

IoT Technician (Smart City)

NSQF LEVEL - 4

1st Year (Volume II of II)

TRADE THEORY

SECTOR: IT & ITES



Directorate General of Training

DIRECTORATE GENERAL OF TRAINING
MINISTRY OF SKILL DEVELOPMENT & ENTREPRENEURSHIP
GOVERNMENT OF INDIA



**NATIONAL INSTRUCTIONAL
MEDIA INSTITUTE, CHENNAI**

Post Box No. 3142, CTI Campus, Guindy, Chennai - 600 032

21	Identify different IoT Applications with IoT architecture.	130. Identify various IoT Applications in smart city viz. smart environment, smart street light and smart water & waste management. (07 hrs) 131. Recognise the functions of various Internets of Things (Smart City) (IoT) applications & their distinctive advantages. (08 hrs) 132. Identify and explore different functional building blocks of IOT enabled system / application. (08 hrs) 133. Test signal flow into IOT enabled system/application as per the IOT architecture. (07 hrs)	Introduction to Internet of Things applications in smart city& their distinctive advantages - smart environment, smart street light and smart water & waste management. What is an IOT? What makes embedded system an IOT? Role and scope of IOT in present and future marketplace. Smart objects, Wired – Cables, hubs etc. Wireless – RFID, WiFi, Bluetooth etc. Different functional building blocks of IOT architecture.
22	Identify, test and interconnect components/parts of IoT system.	134. Connect and test Arduino board to computer and execute sample programs from the example list. (04 hrs) 135. Upload computer code to the physical board	Arduino development board, Pin diagram, Functional diagram, Hardware familiarization and operating instructions. Integrated development

		(Microcontroller) to blink a simple LED. (02 hrs) 136. Write and upload computer code to the physical Arduino board Micro controller to sound buzzer. (02 hrs) 137. Circuit and program to Interface light sensor – LDR with aarduino to switch ON/OFF LED based on light intensity. (03 hrs) 138. Set up & test circuit to interface potentiometer with Arduino board and map to digital values for e.g. 0-1023. (03 hrs) 139. Interface Pushbuttons or switches; connect two points in a circuit while pressing them. This turns on the built-in LED on pin 13 in Arduino, while pressing the button. (03 hrs) 140. Rig up the Circuit and upload a program to Control a relay and switch on/off LED light using Arduino. (02 hrs) 141. Make Circuit and upload a program to Interface of LCD display with a microcontroller to display characters. (03 hrs) 142. Rig up the circuit and upload a program to interface temperature sensor – LM35 with a controller to display temperature on the LCD. (02 hrs) 143. Set up Circuit and upload program to Interface DC motor (actuator) with microcontroller to control on/off/forward/reverse operations. (03 hrs) 144. Rig up Circuit and upload program micro-controller to switch on/off two lights using	Environment, Running Programs on IDE, simple Programming concepts.(10 Hours)

		relay. (03 hrs)	
23-24	Project Work / Industrial Visit (Optional) Broad area: - - Full wave Voltage rectifier with indicator. - Flashing LEDs. - Microcontroller - Interconnection of components of IoT system. - Visit to IoT Lab of different industries.		
25-26	Revision		
27-28	Identify and test various parts of embedded system.	145. Test main heart of embedded system / micro controller and micro controller hardware board / Hardware platform of an embedded system such as Arduino-Uno. (04 hrs) 146. Power up Arduino Uno board, and test its Analog, Digital and I/O pins. (03 hrs) 147. Test and explore sensors and actuators such as LDR, temperature sensors, potentiometers, piezo-element, servo, relay and push buttons, LED, Tri colour LED. (05 hrs) 148. Download and install Arduino Software IDE in computer system. (03 hrs) 149. Test and familiarize with different components of Arduino IDE/Sketch. (04 hrs) 150. Rig up the circuit to test Light dependent resistor to switch ON/OFF based light intensity. (05 hrs) 151. Rig up a test circuit to display 0-9 Numbers on 7 segment display. (05 hrs) 152. Rig up test circuit to control the backlight 16x2 character display. (03 hrs) 153. Rig up the test circuit to control the relay. (04 hrs)	Fundamental idea of embedded system – with architecture and familiarization with different components. Arduino-Uno board - simplest Boards such STM, NXP development board etc. Basics of development boards, Arduino Uno board its needs, hardware familiarization, pin diagram Other available development boards. Concept of integrated development platform (IDE), Its components and Serial Monitor. Concepts behind sensing light, temperature, Motion and other physical parameters. Familiarization with the Sensors/Actuators such as Light dependent resistor, LM35 temperature Sensors, 7 Segment display, 16x2 LCD display, Relays, DC motors (Actuators), Switches, buzzer and motion sensors, Gas Sensors(MQ2 sensor), rain sensor, IR/Obstacle/Proximity sensor, Ultrasonic/Distance measurement Sensors. Pin diagram / connection / Schematic diagrams /Functional

		154. Rig up the test circuit to control DC motor in Forward/Reverse operations using DPDT switch. (03 hrs) 155. Connect the test circuit to sound the Buzzer. (02 hrs) 156. Connect and test the motion sensor along with light /Buzzer /Streetlight and also test for timer and area of motion detection. (04 hrs) 157. Set up a test circuit to test IR sensor. (05 hrs) 158. Set up test circuit to test rain sensor. (05 hrs) 159. Set up test circuit to test ultrasonic sensors. (05 hrs)	Diagram of these Components.
29	Identify and select various types of sensors used in Smart City.	160. Identify various sensors used in Smart city. (03 hrs) 161. Identify Roles and characteristics of various sensors. (04 hrs) 162. Select appropriate sensor as per requirement. (03 hrs) 163. Determine air quality and use noise pollution Sensors. (05 hrs) 164. Measure PM2.5 and PM10 levels using Electrochemical Sensors. (04 hrs) 165. Explore sensors used in weather monitoring system. (03 hrs) 166. Measure air temperature, humidity, atmospheric pressure and solar radiation. (04 hrs) 167. Measure and record Information such as air temperature, wind speed, dew point temperature, wind direction, relative humidity, solar radiation and atmospheric pressure at predetermined intervals by	Principle of operation of various sensors used in Smart city; their roles and characteristics. Selection of appropriate sensor as per requirement. Use of air quality and noise pollution Sensors. Measurement of PM2.5 and PM10 levels using Electrochemical Sensors for pollution control in smart environment. Explore sensors used in weather monitoring system. Measurement and record of Information such as air temperature, wind speed, dew point temperature, atmospheric pressure etc. at predetermined intervals by Weather Stations.

		Weather Stations. (04 hrs)	
30	Position the appropriate sensors and collect the information required in Smart City.	168. Identify sensors node block diagram and its components. (05 hrs) 169. Check connection with sensors and send data wirelessly to a central data logger at program. (05 hrs) 170. Identify interfacing of wireless modules with IoT platform. (02 hrs) 171. Select and Install sensors like NO2, PM2.5, PM10, CO2, O2, air temperature, humidity etc. (05 hrs) 172. Configure sensor node using USB and over the air programming. (05 hrs) 173. Check the battery level and solar panel connection with sensor node. (03 hrs) 174. Control Variable rate controllers manually or automatically using an on-board computer guided by real GPS location. (05 hrs)	Concept of sensor node blocks diagram and its components. Connection with sensors to send data wirelessly to a central data logger at program. Explore interfacing of wireless modules with IoT platform. Selection and Installation of sensors like NO2, PM2.5, PM10, CO2, O2, VOC, air temperature, humidity, etc. Data packet and sensor node configuration tool using USB and Over the air programming. Explore the battery level and solar panel connects with sensor node. Control Variable rate controllers manually or automatically using an on-board computer guided by real GPS location.
31-32	Identify and test Wired & Wireless communication medium such as RS232, RS485, Ethernet, Fiber Optic, Wi-Fi, GSM, GPRS, RF etc. and Communication protocol.	175. Explore the interfacing of Zigbee module to create wireless sensor network. (02 hrs) 176. Check the M2M Wireless Sensor Network (WSN) in IoT Zigbee router, end device and coordinator configuration. (02 hrs) 177. Identify the interfacing of Bluetooth module to create local sensor network. (02 hrs) 178. Explore the interfacing of GSM module to make node as a gateway. (02 hrs) 179. Apply IoT Gateway using WiFi and Ethernet. (02 hrs) 180. Check UART Communication, RS485 Communication, I2C	Introduction to Zigbee, Block diagram of Zigbee based sensor network. Introduction to wireless personal area network system. Introduction to Zigbee networking system. Concept of interfacing of Bluetooth module to local sensor network, interfacing of GSM module to make node as a gateway. IoT Gateway using WiFi and Ethernet. Application of GPS satellites in Location Sensors. Creation of a combine sensor appropriate for local climate monitoring. Concept of Weather Stations.

		Protocol device interfacing SPI Protocol device interfacing, Ethernet configuration, Zigbee interfacing, Wi-Fi AP and Router interfacing. (02 hrs) 181. Identify the Wi-Fi module and lua script for data communication. (02 hrs) 182. Explore the application of GPS satellites in Location Sensors. (02 hrs) 183. Check USB and Ethernet connectivity for data communication. (02 hrs) 184. Create a combine sensor appropriate for local climate monitoring. (02 hrs) 185. Use signals from GPS satellites to determine latitude, longitude and altitude to within feet by Location Sensors for precise positioning. (02 hrs) 186. Operate Global Positioning System (GPS) & Apply satellites broadcasting signals that allow GPS receivers to calculate their position. (02 hrs) 187. Analyze the Vehicle Speed using Tacking system. (01 hr) 188. Use Geographical information system(GIS) consisting of a computer software data base system used to input, store, retrieve, analyze and display in map like form, spatially referenced geographical information for more detailed analysis of city roads, transport, traffic etc. (02 hrs) 189. Analyze the data collected by GPS and supply it to user in usable format such as maps, graphs, charts or reports using	Usage of signals from GPS satellites to determine latitude, longitude and altitude to within feet by Location Sensors for precise positioning. Principle of operation & Application of Global Positioning System (GPS): satellites broadcasting signals that allow GPS receivers to calculate their position. Working principle & Use of Geographical information system(GIS) consisting of a computer software data base system used to input, store, retrieve, analyze and display in map like form, spatially referenced geographical information for more detailed analysis of fields. Working principal of GPS module for vehicle speed measurement. Data Integration Through a Geographical Information System. Use of Computer Hardware and Software to analyze the data collected by GPS and supply it to user in usable format – such as maps, graphs, charts or reports. Tooth technology, operating modes, Pin configuration. Basics of Wi-fi Modules, Pin configurations, Modes of operations. Basics of GSM/GPS modules. Basic blocks of networking, - Specifications, Standards and types of cables, - Concept of wired or wireless communication medium - Different types of networks - Design and establish networks

		suitable Computer Hardware and Software. (02 hrs) 190. Test the android phone and its features, use of sensors & usage. (02 hrs) 191. Check the blue tooth module along and explore the possibility of pairing with Android Smart Phone. (02 hrs) 192. Test Bluetooth module with a micro controller and Program to switch on/off an LED/Buzzer. (02 hrs) 193. Check the GSM Module and its interconnections. (02 hrs) 194. Download mobile app from play store and control (ON/OFF) a simple LED via Bluetooth. (02 hrs) 195. Test GPS module. (02 hrs) 196. Check Wifi module. (02 hrs) 197. Identify Cable and its Pin Mapping (02 hrs) 198. Crimp and Test RJ9 / RJ11 / RJ45 connectors (02 hrs) 199. Understand Frequency Band, Gain, Antenna and Modulation for Wi-Fi. (02 hrs) 200. Understand Frequency Band, Gain, Antenna and Modulation for GPRS. (02 hrs) 201. Understand Frequency Band, Gain, Antenna and Modulation for RF. (02 hrs) 202. Design and Test Local Area Networks over Ethernet & Wi-Fi. (03 hrs) 203. Design and Test Cellular Wide Area Networks over GSM & GPRS. (02 hrs) 204. Design and Test Personal Area Networks over RF. (02 hrs)	
33	Identify Solar Panel Basic Testing, Characteristics, Charge Controller Circuit.	205. Explore and test series combination of Solar PV Modules. (02 hrs)	Basics of solar Electricity, Working principle of PV panel, advantages of solar electricity and components

		206. Test parallel combination of Solar PV Modules. (02 hrs) 207. Check series-parallel combination of Solar PV Modules. (02 hrs) 208. Measure VI Characteristics of Solar PV Module. (02 hrs) 209. Explore and test blocking diode and its working in Solar PV Module. (02 hrs) 210. Observe bypass diode and its working in Solar PV Module. (04 hrs) 211. Measure effect of inclination angle of Solar PV Module. (02 hrs) 212. Explore and test different charging techniques. (02 hrs) 213. Test Buck & Boost converter. (02 hrs) 214. Check effect of change in solar radiation on Solar PV Module. (02 hrs) 215. Explore and test running different applications i.e. LEDs, Dusk to Dawn sensing. (04 hrs) 216. Explore the use of P V Analyzer. (02 hrs) 217. On Grid Smart Energy Management. (02 hrs)	of solar electricity, Various combinations, VI characteristics of solar PV module, effect of inclination angle on PV module, different battery charging techniques.
34-35	Perform installation, configuration and check working of IOT devices, network, database, app and web services. Monitor environmental parameters like Temperature, Humidity, Air Quality, PM2.5, PM10, CO₂ etc.	218. Install Linux Operating System porting. (02 hrs) 219. Configure Local cloud & server. (02 hrs) 220. Configure Over the air (OTA) node. (02 hrs) 221. Explore GUI based parameter configuration, GUI based IoT application. (03 hrs) 222. Check IoT Gateway using Wi-Fi and Ethernet. (02 hrs) 223. Work with the command line and the Shell. (02 hrs) 224. Manage directories and files.	Installation of Linux Operating System porting. Configuration of Local cloud & server. Over the air (OTA) node configuration. GUI based parameter configuration, GUI based IoT application. IoT Gateway using Wi-Fi and Ethernet. User access and data security (Cyber security) by Cryptography. The command line and the Shell, directories and files.

		(02 hrs) 225. Manage user access and data security (Cyber security) by Cryptography. (03 hrs) 226. Set up a Linux file system. (04 hrs) 227. Perform system initialization. (03 hrs) 228. Connect a system to the network. (02 hrs) 229. Install and Configure Linux. (02 hrs) 230. Create Shell Scripts, flow control in the Shell, Advanced Shell features. (02 hrs) 231. Explore Database management system. (02 hrs) 232. Test Cloud and Server Configuration for IoT. (03 hrs) 233. Test Qt based GUI for Sensor output. (02 hrs) 234. Test IoT Web and Application Development Tools for IoT. (02 hrs) 235. Select and Install Carbon dioxide sensors. (03 hrs) 236. Identify and Install Oxygen sensors. (02 hrs) 237. Select and Install Volatile organic compound sensor. (04 hrs) 238. Identify and Install Air temperature, Air humidity atmospheric pressure and UV sensor. (02 hrs) 239. Select and Install Nitric Oxide (NO), Hydrogen Sulphide, Sulphur Dioxide, Carbon Monoxide, Ozone Soil Moisture and Soil Temperature sensor. (02 hrs) 240. Check Magnetic field for smart parking, IR for human presence. (03 hrs) 241. Measure Hall Effect (doors	Linux file system, understanding system initialization. Connection of a system to the network. Installation and Configuration of Linux. Shell Scripts, flow control in the Shell, Advanced Shell features. Database management system. Cloud and Server Configuration for IoT. Qt based GUI, IoT Web and Application Development Tools for IoT. Principle of operation, selection and installation of Carbon dioxide sensors, Oxygen sensors. Volatile organic compound sensor Selection and Installation of Air temperature, Air humidity and atmospheric pressure, UV sensor, Nitric Oxide (NO), Hydrogen Sulphide, Sulphur Dioxide, Carbon Monoxide, Ozone Soil Moisture and Soil Temperature sensor. Study and test of Magnetic field for smart parking, IR for human presence. Study and test of Hall Effect (doors and windows openings), Water presence, Liquid level, Liquid flow, Temperature, Humidity for smart security. Calibration Kits testing for the sensor probes for water quality analysis.

		and windows openings), Water presence, Liquid level, Liquid flow, Temperature, Humidity for smart security. (02 hrs) 242. Test Calibration Kits for the sensor probes for water quality analysis. (02 hrs)	
36-39	Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols, device management and monitoring.	243. Power up the Solar Inverter (similar device) as per the device manual. (04 hrs) 244. Integrate Solar Inverter (similar device) with serial protocol working on Modbus RTU. (05 hrs) 245. Communicate and Verify the parameters on Modbus Master Software. (06 hrs) 246. Power up the Energy Meter (similar device) as per the device manual. (04 hrs) 247. Integrate Energy Meter(similar device) with serial protocol working DLMS protocol. (05 hrs) 248. Communicate and verify the parameters on DLMS server software. (06 hrs) 249. Setup wired Local Area Network and wireless network. (04 hrs) 250. Setup environment for Modbus TCP/IP server client testing. (05 hrs) 251. Communicate and Configure Modbus devices through GSM GPRS network. (06 hrs) 252. Setup Serial to Ethernet protocol converter and verify. (05 hrs) 253. Setup Serial to Wi-Fi protocol converter and verify. (05 hrs) 254. Setup Serial to GPRS protocol converter and verify. (05 hrs) 255. Setup Ethernet IoT Data	- Basics of Industrial protocols Modbus RTU, Modbus TCP, DLMS - Client server communication Basics of Protocol Converters. Basics of IoT Data Acquisition System. Device connectivity over cloud and troubleshooting. GUI based IoT Cloud Configuration utility. IoT device and its parameter configuration Cloud Device Management and troubleshooting.

		Acquisition system, connect to cloud and verify. (08 hrs) 256. Setup WiFiloT Data Acquisition system, connect to cloud and verify (08 hrs) 257. Setup Cellular (GSM / GPRS) IoT Data Acquisition system, connect to cloud and verify (08 hrs) 258. Explore IoT Cloud Configuration utility. (06hrs) 259. Create / modify organization, Connect devices over cloud. (06hrs) 260. Configuration of parameters, alarms, notifications on cloud platform. (06hrs) 261. Explore user management roles and security. (06hrs) 262. Observer Device Diagnostics for troubleshooting. (06hrs) 263. Setup Environment for embedded SCADA testing. (04 hrs)	
40-41	Demonstrate and Deploy responsive Web Application using APIs and generate reports using templates.	264. Explore Web API, required input parameters and output (10 hrs) 265. Map Web API to Widget / Control / Plugin. (20 hrs) 266. Display and configure graphs, charts and other ready to use controls and widgets. (20 hrs) 267. Generate reports using readily available API, templates and to export it to excel, word pdf and other required formats. (10 hrs)	Usage of Web Services / Web API Development of Sample Web Application. Generation and export of Reports User access and rights management.
42-43	Identify and test Smart Lighting system and its components.	268. Rig up circuit to lighting system and measure different parameter such as Voltage, current, Lux using multimeter and Lux Meter. (04 hrs) 269. Test different types of Lighting System such as Outdoor, Indoor, street Light etc. (06	Fundamental science of lighting system. Different types of light Luminaries, Smart Light Drivers. Lumen, Lux, Wattage etc. Sensor integrated, Non-sensor integrated lighting System. Different dimming control methods in lighting system.

		hrs) 270. Check circuits to test and troubleshoot Sensor integrated lighting System. (06 hrs) 271. Apply non-sensor integrated lighting System. (05 hrs) 272. Test different dimming control methods in lighting system. (05 hrs) 273. Rig up the circuit to interface Microcontroller, LDR and Light to vary brightness in accordance with illumination of the light. Upload the code to microcontroller and test for proper operation. (06 hrs) 274. Check Circuit to test and troubleshoot MQ135 pollution sensor module. (06 hrs) 275. Install CCTV Camera for building security and roadside safety. (06 hrs) 276. Rig up the circuit to interface Microcontroller, MQ135 pollution sensors and vary brightness of light in accordance with Fog/Smog environment. Upload the code to microcontroller and test for proper operation. (06 hrs) 277. Test System architecture of smart lighting and Identify • Wired–DALI, GREENBUS2, etc. • Wireless • Hybrid (10 hrs)	Concept of dimming. Basics of interfacing micro controllers. Need of smart lighting. Schematic diagrams, datasheets LDR, Motion sensor, MQ135. Components of System architecture of smart lighting. Principle of CCTV Camera and installation process and recording and recover the data. Concept of Wired – DALI, GREENBUS2, etc. Wireless, Hybrid.
44-45	Identify, select, install and troubleshoot different module / devices used in SMART Street Light based on IoT and Cloud Technology.	278. Install, test and troubleshooting of Smart Light. (04 hrs) 279. Install and test Solar street light. (04 hrs) 280. Execute testing of sensors used in street light like dusk to	Basic concepts of Smart Light-Working Principle of Solar street light, sensors used in street light like dusk to dawn, Temperature sensor. Solar battery management system - Basic concepts battery, types,

		dawn, Temperature sensor. (04 hrs) 281. Check solar battery management system. (06 hrs) 282. Check solar street light components. (06 hrs) 283. Test LED used on solar street light. (04 hrs) 284. Install Security camera on street light. (06 hrs) 285. Apply Smart embedded system that controls the street light based on detection of sunlight. (06 hrs) 286. Configure and Communicate 3 Phase Modbus Energy Meter with IoT based Smart Streetlight Controller. (06 hrs) 287. Observe Over voltage protection and over current protection. (06 hrs) 288. Monitor Smart streetlight management system having with map view based dash board and individual system details. (08 hrs)	preventive maintenance, arrangement of battery and battery management. Solar street light components, LED used on solar street light, Security camera on street light. Smart embedded system that controls the street light based on detection of sunlight. Benefits - ensure safety and to prevent energy wastage.
46	Identify, select, install and troubleshoot different module / devices used in SMART Parking.	289. Install LED display board. (04 hrs) 290. Test Magnetic field for smart parking. (10 hrs) 291. Execute installation of proximity sensor for boom barrier, IR Sensor for presence. (06 hrs) 292. Apply full stack solution to deal with all aspects of parking including high level tools for management and analytics software down to street level occupation sensors and enforcing tools. (10 hrs)	Concept of Smart parking for better management of car park availability and traffic in the city to improve citizen's life - smart parking solution. Connected Parking - LoRa WAN private network for better understanding and better management of car park availability. Use of proximity sensor, IR Sensor in smart parking. Full stack solution to deal with all aspects of parking.
47	Identify, select, install and troubleshoot different module /	293. Use Location Sensors, GPS & GPS integrated circuits. (06 hrs)	Concept of Smart Road & Traffic, Live & Connected roads - Benefits - experience of quicker, safer and

	devices used in SMART Traffic.	294. Apply Solar panel, Antenna & Radio Technology. (06 hrs) 295. Use scanner for real-time traffic and pedestrian estimation. (06 hrs) 296. Carry out Smartphone Detection (Bluetooth, Wifi, 3G/4G-GPRS etc.). (04 hrs) 297. Detect liquid presence over road by Liquid presence sensor for Smart Security. (04 hrs) 298. Apply Linear displacement sensor for Structural Crack monitoring. (04 hrs)	more effective trips. Weather monitoring at risky points: Low cost weather station (Rainfall, Temperature + Humidity, Wind speed & direction), Pluviometer, Vane sensor, Anemometer, Temperature+ Humidity, Liquid presence sensor. Smartphone Detection (Bluetooth, Wifi, 3G/4G-GPRS etc.). Structural Crack monitoring.
48	Apply IoT Application for Water & Waste Management.	299. Select and install pH, Cupric (Cu²⁺), Silver (Ag⁺), Lithium (Li⁺), Conductivity, Temperature for maintenance of water quality. (06 hrs) 300. Install Smart dustbin. (06 hrs) 301. Install GPS based tracking system for smart bin. (06 hrs) 302. Maintain dry waste and wet waste separately. (06 hrs) 303. Install, test & apply different components like Ultrasonic sensors, Wifi module & Thingspeak (IoT Platform) cloud. (06 hrs)	Smart Waste Management system: Definition, Application, working, challenges, constraints, Detection of rubbish levels in containers to optimize the trash collection routes - Concept of Smart Garbage Bin. Maintenance of dry waste and wet waste separately. Different components- Ultrasonic sensors, Wifi module & Thingspeak (IoT Platform) cloud.
49-50	Project Work/Industrial Visit (Optional) Broad Area:- - Cloud based water quality analysis system using different sensors on IoT Explore. - Wireless Building automation system using PIR, camera and Alarm. - Environmental monitoring system using different sensors. - Responsive Web based IoT Smart rooftop management system with Over voltage & current protection using 3 phase MODBUS energy meter with class 1.0 accuracy - Responsive Web application for Smart Energy management system having map view based dash board with Three Phase 415 VAC input, Single MODBUS, Ethernet, SD Card Storage, Remote GSM/GPRS connectivity.		
51-52	Revision & Examination		

Smart City - Identify different IoT Application with IoT Architecture

Identify various IoT application in smart city viz. smart environment, smart street light, smart water and waste management

Objective:

- to identify various IoT application in smart city viz. smart environment, smart street light, smart water and waste management.

Requirement:**PROCEDURE:**

In general, the term IoT (Internet of Things) refers to the rapidly growing number of digital devices – the quantity is now billions – these devices can communicate and interact with others over the network/internet worldwide and they can be remotely monitored and controlled. The IoT includes only smart sensors and other devices. On the operational level of IoT, for example weather data is collected. IoT offers new opportunities for cities to use data to manage traffic, cut pollution, make better use of infrastructure and keep citizens safe and clean.

Advantages of IoT

Companies use IoT for innovative management and for monitoring widely dispersed processes. As a result, they even can control the latter even from distant places as information is continuously fed into applications and data storage. IoT provides an advantage of knowing things in advance. Due to the low cost of IoT, it is now possible to monitor and manage activities that were previously unreachable. The financial aspect is the best advantage, because this new technology could replace humans who are in charge of monitoring and maintaining supplies. Consequently, costs can be significantly reduced and optimized. IoT also makes it possible to gain completely new insights e.g. associating weather influences to industrial productions.



Smart City - Identify different IoT Application with IoT Architecture**Recognise the functions of various Internet of Things (Smart city- IoT) applications and their distinctive advantages****Objective:**

- to recognise the functions of various Internet of Things (Smart city – IoT) applications and their distinctive advantages.

PROCEDURE:

As cities grow and expand, smart and innovative solutions are crucial for improving productivity, increasing operational efficiencies, and reducing management costs. Citizens are gradually equipping their homes with IoT devices such as TV and Internet box. In the real estate sector, connected objects include thermostats, smart alarms, smart door locks, and other systems and appliances. At the United Nations conference on climate change (COP21) held in Paris in 2016, connected objects were extensively addressed and gave to many local communities the opportunity to rethink their environmental objectives in order to reduce their CO2 emissions through the use of IoT. The latter can play a vital role in the context of smart cities. For example, intelligent waste containers can bring real benefits to citizens; they will be able to indicate that they are soon going to be full and must be emptied. Citizens can check through a smart phone application if the waste containers in the street are full or not. Also, after waste containers reports their status, companies can offer route optimization solution to the teams responsible for garbage collection. Places can be equipped with sensors and monitor environmental conditions, cyclists or athletes can find the most "healthy" trips and the city can respond by adjusting the traffic or by planting more trees in some areas. The data will be accessible to all citizens to promote the creation of applications using real-time information for residents. Cities have become hubs for knowledge sharing. The technologies and solutions needed for creating smart cities are just beginning to emerge. Figure 1 describes an example of a smart city.

The investment in IoT will be crucial to build smart cities, services as data using will generate most of the revenues. Safety and security of smart homes will be the second largest market in terms of service revenues. As for services related to health and well-being, they should represent a market of \$ 38 billion in 2020. A practical solution must find the trade-offs between effectiveness and privacy risks. A sophisticated attacker could, for example, take control of various intelligent devices such as lights, cameras, traffic lights, connected cars and many other smart devices in cities. With over 50 billion devices connected by 2020, municipalities will be very

much concerned for the safety of their intelligent cities. However, solutions to address safety, security, and privacy concerns of smart cities relying on diverse intelligent objects fall not only into the technology realm but also in other areas including sociology, legal, and policy management.

Here we will see an overview of IoT in the context of smart cities, and we discuss how IoT can enhance a city's smartness. We also identify the weaknesses and risks associated with IoT deployment and adoption in the smart city environment. Then we will see some background information on IoT and smart city. Only then we will discuss the main architectures used within IoT. Then we will see how IoT can be considered as an enabling technology for smart city and then we describe the weaknesses that need to be addressed when IoT is used for smart cities. Finally, we make some concluding remarks.

Internet of Things and Smart City

In the recent many have provided definitions for the term Internet of Things. IoT may be defined as "Objects having identities and virtual personalities in smart spaces using intelligent interfaces to connect and communicate within social, medical, environmental and users context". Huge investments are currently being made in the IoT area to support the delivery of a wide range services. Various aspects of social and economic life are currently being studied for IoT. Trust in IoT implies that investors do not hesitate to commit to it financially; 100 million euros were invested by large corporations such as Telefónica, SK Telecom, NTT Docomo Ventures, Elliott Management Corporation and industry groups GDF SUEZ, Air Liquide for research and development of IoT.

The deployment of IoT needs communication standards that seamlessly operate among the various objects. Several worldwide organizations are involved in standardizing such communications. These include the International Telecommunication Union (ITU), the Institute of Electrical and Electronics Engineers (IEEE), the Internet Engineering Task Force (IETF), Global Standard1 (GS1), the Organization for the Advancement of Structured Information Standards (OASIS), the Industrial Internet Consortium (IIC), and several others.

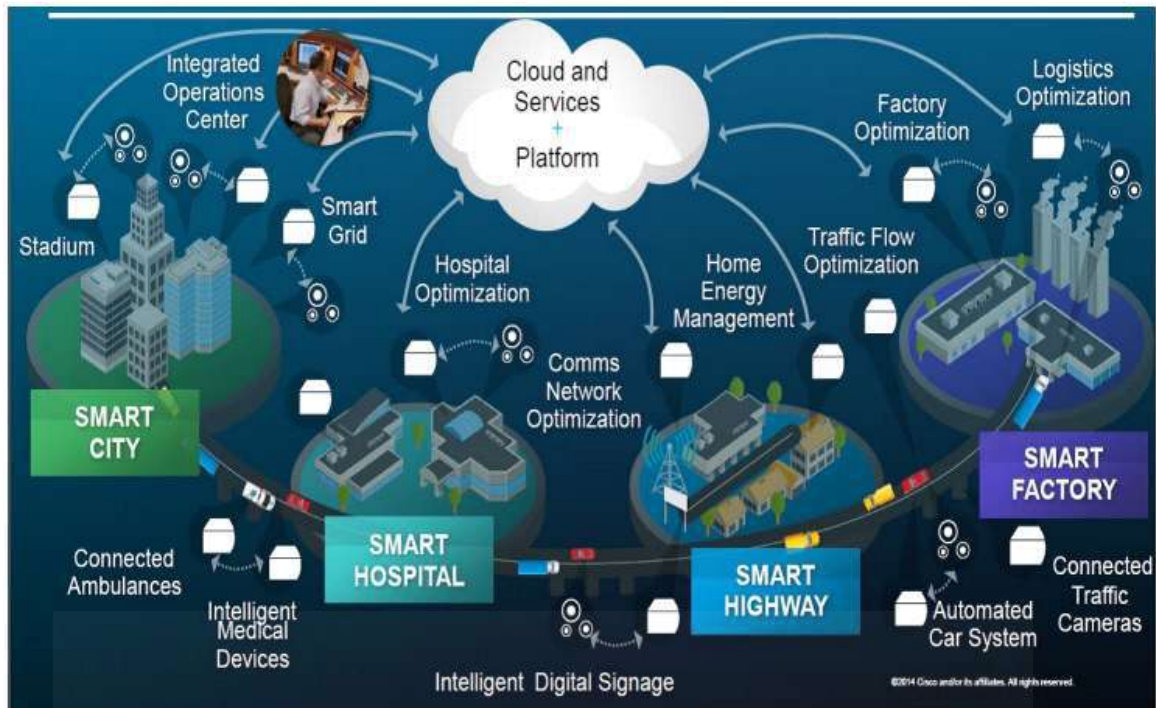


Fig. 1: Development of smart cities

We briefly present some of these IoT standards and initiatives in Table 1. For example, the Internet of Things Standard Global Initiative (IoT-GSI) supported by ITU made two recommendations: the ITU-T Y.2060, which provides an overview of the concept of IoT and ITU-T Y.2061, which describes the conditions for the machine interface oriented towards applications. Various standards were proposed by IEEE and IETF at different levels for sensor networks based on the Internet Protocol (IP). For example, at the link layer, the IEEE 802.15.4 standard is more suitable than Ethernet in industrial environments. At the network level, the IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) standard can adapt the IPv6 protocol for wireless communications. In 2011, the IETF published the IPv6 Routing Protocol (RPL) standard for Low-power Networks.

IETF has also launched a Working Group to standardize an application layer-oriented protocol for connected objects. The reference protocol is called the Constrained Application Protocol (CoAP). CoAP (see RFC 7252 of June 2014) provides methods and commands (such as, HTTP Get) to query an object and change its status. CoAP relies on UDP and can optionally use Datagram Transport Layer Security (DTLS), to provide communication security. Operating systems used in IoT include: TinyOS, Contiki OS, MantisOS, Nano-RK, Android, Brillo (Google), Windows 10 IoT Core, LiteOS (Huawei), Mbed OS (ARM). In addition, several platforms [10] have been developed for IoT: Arrayent, Californium

CoAP Java framework, Erbium, CoAP framework for Contiki, and XMesh networking stack. At the application layer, a large number of applications have been developed: Iobridge Thingspeak, Nimbits, Evrythng, Open.Sen.se, NanoService, exosite One, HP supposed, Isidorey, SensorCloud, Manybots, and so on. Figure 2 compares the 6LoWPAN communication stack with other popular communication stacks.

The Electronic Product Code Global (EPC Global) initiative of the organization Global Standard 1 (GS1) defines a unique individual identifier for identifying an electronic product and the overall EPC network architecture that defines the organization of information systems designed to ensure the exchange of information in an EPC network. One of its main components is the Object Naming Service (ONS) which is based on the Domain Name System (DNS).

In fact, in 1970 the European Article Numbering (EAN) standard emerged for product identification. However, this EAN barcode is actually used to identify a class of products, not individual instances within this class. Furthermore, in IoT, a unique IP address for each connection is required. This is why EPC was proposed by GS1 as a new standard. Meanwhile, OASIS[†] issued various recommendations on network technologies in IoT and messaging technologies such as Message Queue Telemetry Transport (MQTT), Advanced Message Queuing Protocol (AMQP) and the Data Distribution Service for Real-Time Systems (DDS). In 2014, a new

Industrial Internet Consortium was launched in order to coordinate and establish the priorities and enabling technologies of the Industrial Internet. There are thousands of founding and contributing members of ICC and they include: Bosh, Intel, IBM, Schneider, Huawei, Cisco, and several others. There are currently 19 Working Groups and teams working on different areas: Business Strategy and Solution Lifecycle, Legal, Liaison, Security, Technology Testbeds, Marketing and Membership, and so on. Figure 3 summarizes some IoT's protocols and standards and Table 2 details the used acronyms.

A smart city is defined as a city connecting physical infrastructures, ICT infrastructures, social infrastructures and business infrastructures to leverage the collective intelligence of the city. A city can be smart through a large deployment of IoT (especially through machine-to-

machine and human-to-machine communications). Wireless Sensor Networks (WSNs), the sensing-actuation arm of the IoT, seamlessly integrate into urban infrastructure forming a "digital skin" around it. The information generated is shared across diverse platforms and applications to develop a Common Operating Picture (COP) of the city.

IoT Architecture

IoT technologies are expected to be part of large scale networks, with the number of devices in the thousands and areas spanning several kilometers. In the rest of this chapter, we focus primarily on the LoRa Ultra-Narrow Band (UNB) technology which was developed by Semtech and SigFox's.

TABLE

	802.11a	802.11b	802.11g	802.11n	802.11ac	802.11ad	802.15.1	802.15.3	802.15.4	802.15.6	NFC
Network Type	WLAN	WLAN	WLAN	WLAN	WLAN	WLAN	WPAN	WPAN	WPAN	WBAN	Point-to-Point
Date	1999	1999	2003	2009	2014	2012	2002/2005	2003	2007	2011	2011
Network Size	30	30	30	30			7	245	65535	250	-
Bit Rate	54 Mbps	11Mbps	54 Mbps	248 Mbps	3.2 Gbps	≥ 7Gbps	3 Mbps	55 Mbps	250 Kbps	10 Mbps	424 Kbps
Frequency	5 GHz	2.4 GHz	2.4 GHz	2.4/5 GHz	5 GHz	2.4/5/60GHz	2.4 GHz	2.4 GHz	868-915 MHz 2.4 GHz	402-405 MHz	13.56 Mhz
Range	120 m	140 m	140 m	50 m indoor 250 m outdoor	30 m	5 m	100 m	100 m	75 m	2-5 m	0.2 m
Modulation	BPSK, QPSK 16-QAM 64-QAM OFDM	DBPSK DQPSK CCK DSSS	DBPSK DQPSK 16-QAM 64-QAM OFDM	OFDM	OFDM	QAM-256	8DPSK DQPSK PIDQPSK GFSK AFM	QPSK DQPSK 16-QAM 32-QAM 64-QAM	ASK DSSS PSSS		Manschester and Modified Miller
Application	WiFi	WiFi	WiFi	WiFi			Bluetooth		ZigBee		

Table 1 Main communication standards within IoT

Acronym	Description	Acronym	Description
HBase	Hadoop Database	RapidMQ	Rapid Message Queuing
MQTT	Message Queuing Telemetry Transport	DDS	Data Distribution Service
XMPP	Extensible Messaging and Presence Protocol	AMQP	Advanced Message Queuing Protocol
HTTP	HyperText Transfer Protocol	FTP	File Transfer Protocol
Telnet	Telecommunication Network	SSH	Secure SHell
IPv4	Internet Protocol Version 4	IPv6	Internet Protocol version 6
6LowPan	IPv6 over Low power Wireless Personal Area Networks	RPL	IPv6 Routing Protocol for Low-Power and Lossy Networks
BLE	Bluetooth Low Energy	RFID	radio frequency identification
GSM	Global System for Mobile Communications	CDMA	Code division multiple access
OBD2	On-board diagnostics 2	PLC	Power-line communication
RS-232	Recommended Standard 232	Modbus	Modicon Communication Bus
USB	Universal Serial Bus	SPI	Serial Peripheral Interface
AES	Advanced Encryption Standard	SSL	Secure Sockets Layer

Table 2 Acronym table

Simplified OSI	TCP/IP	6LoWPAN	ZigBee
Application	HTTP	HTTP, COAP, MQTT	ZigBee APL
Transport	TCP	TCP, UDP	
Internet	IP	IPv6, RPL	ZigBee NWK
Link		6LoWPAN IEEE 802.15.4 MAC	IEEE 802.15.4 MAC
Physical	WiFi	IEEE 802.15.4 PHY	IEEE 802.15.4 PHY

Fig. 2: Comparison of 6LoWPAN's stack with other stacks

LoRa

LoRa is a wireless technology designed to provide the low-power within wide-area networks (LPWANs) required for Internet of Things services. The technology offers a mix of long range, low power consumption and secure data transmission. The LoRa standard has been developed for IoT-type devices in regional or global networks. This technology provides seamless interoperability among devices without requiring any complex installations. The services targeted include home energy monitoring, alarm systems, remote health monitoring, transportation, environment protection, and so on. This specification defines the communication protocol and system architecture for the underlying network. It supports frequencies in the 433, 868 or 915 MHz ISM bands, depending on the area where it is deployed. In Europe, it uses either Gaussian Frequency Shift Keying (GFSK) or the proprietary LoRa modulation system, which works with a version of Chirp Spread Spectrum using 125 KHz channel bandwidth. LoRa architecture is describes in Figure 4.

The hierarchical star-based topology is used by LoRa networks. IoT devices in such networks can be servers, end-points, or gateways. Data rates can range, in Europe, from 0.3 Kbps up to 50 Kbps when channel aggregation is employed. In North America, the minimum data rate is 0.9 Kbps because of Federal Communications Commission (FCC) requirements. The payload for this technology can range from 2 to 255 bytes. This standard is optimized for low cost and battery operated sensors. The devices are asynchronous and communicate only when they have data ready to send whether event-driven or scheduled. Power consumption is proportional to the time devices spent while in the listening mode. LoRa is gaining significant attention in IoT networks that are being deployed by wireless network operators. It can be deployed with minimum upfront infrastructure investments and operating costs. When increased network capacity is required, further Gateways can be added. It has been estimated that the deployment cost of this technology in unlicensed bands needs much less capital than even a 3G software upgrade. Major Telecom

operators (e.g., Swisscom, NKE Electronics, and others) are deploying this technology for nationwide networks because of its benefits over competing technologies. These benefits include bi-directional communications, mobility for asset tracking, security, and accurate localization.

SigFox

SigFox created an ultra-narrowband IoT communications system designed to support IoT deployments over long ranges, e.g. in excess of 20 km between a client device and a base station. SigFox uses license-exempt spectrum for its product, namely the 868 MHz band in Europe and 915 MHz band in the US, to transmit data over a narrow spectrum to and from connected objects. The ultra-narrow band operation is achieved using bandwidth channels lower than 1 KHz transmitting data payloads of 12 bytes uplink and 8 bytes downlink with a protocol overhead of 26 bytes. One of the advantages of SigFox devices is their resource efficiency. The power demand is negligible because devices are only "on" when they are transmitting; this means that the power demand is a fraction of that for a device operating on cellular networks.

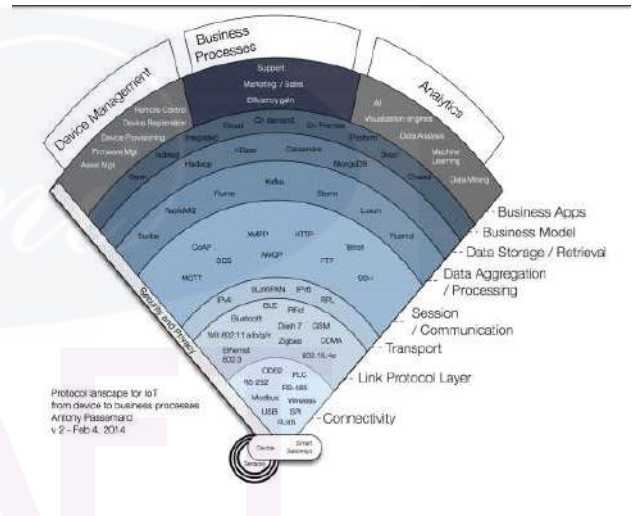


Fig. 3: Internet of Things protocol stack

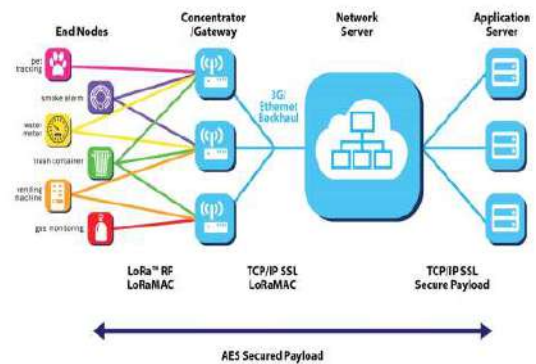


Fig. 4: LoRa architecture [20]

Standard	SIGFOX	LoRaWAN	LTE-M	IEEE 802.11ah (low power WiFi)	Dash7 Alliance Protocol 1.0	Ingenu RPMA	nWave
Frequency Band	868 MHz/902 MHz ISM	868 MHz/902 MHz ISM	Cellular	License- exempt bands below 1 GHz; excluding the TV White Spaces	433, 868, 915 MHz	2.4 GHz ISM	Sub- GHz ISM
Range	30-50km (rural), 3-10km (urban), 1000km LoS	2-5k (urban), 15k (rural)	2.5- 5km	Up to 1Km (outdoor)	0.5 km	>500 km LoS	10km (urban), 20- 30km (rural)

Table 3 Comparison of low power WAN technologies [21]

Sig Fox technology allows deploying very efficient, low throughput communications by limiting the number of antennas (base stations). For the same level of coverage, SigFox requires around 1,000 times less antennas and base stations, compared with some cellular networks. This technology offers access to a service management

interface, which can enable the control of main communication parameters such as battery and temperature settings, signal quality, volume of exchanged data, and others. Networks based on SigFox technology have already connected thousands of devices in several international cities. They are currently operational in 14 countries, covering an area of more than 1.2 million km² and reach 223 million people.

In general, IoT can be divided into three layers: the perception layer, the network layer and the application layer. The perception layer is mainly used to capture and gather, distinguish and identify the information of objects in the physical world. This layer includes RFID tags, cameras, GPS, sensors, laser scanners, and so on. The network layer is used to forward packets over a reliable communication medium. The application layer processes the data, aggregates various sources and displays it. Table 3 compares different low power WAN technologies used in IoT use case scenarios.



Smart City - Identify different IoT Application with IoT Architecture**Identify and explore different functional building blocks of IoT enabled system/ applications****Objective:**

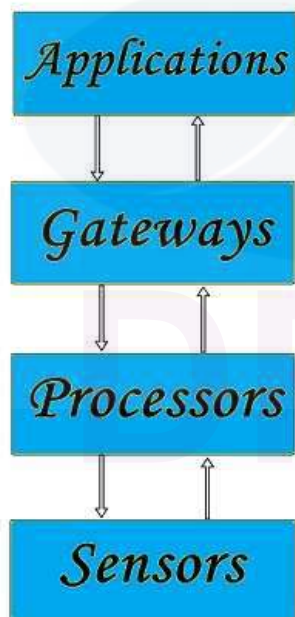
- to Identify and explore different functional building blocks of IoT enabled system/ applications.

Requirements:

- Cloud
- Server
- Internet connectivity
- Sensors
- IoT devices

PROCEDURE:**Elements of IoT**

Following are the four building blocks of IoT system and they have their own characteristics.

**Sensors**

Sensors are the front end of the IoT devices. They really mean “things” in IoT. Their main task is to get necessary data from surroundings and pass it further to database or processing systems. They must be uniquely findable from their IP address because they are basic front end interface in the large network of other

devices. Sensors collect real time data and can either work autonomously or can be user controlled.

Examples of sensors are: gas sensor, water quality sensor, moisture sensor, etc.

Processors

As computer and other electrical systems, processors are the brain of the IoT system. The main job of processors is to process raw data collected by the sensors and transform them to some meaningful information and knowledge. In short, we can say that its job is to give intelligence to the data. Processors are easily controllable by applications and their one more important job is to secure data. They perform encryption and decryption of data. Microcontroller, embedded hardware devices, etc. can process the data using processors attached within the devices.

Gateways

Main task of gateways is to route the processed data and transfer it to proper databases or network storage for proper utilization. In other words, gateway helps in communication of the data. Communication and network connectivity are essentials for IoT systems.

Examples of gateways are LAN, WAN, PAN, etc.

Applications

Applications are another end of an IoT system. Applications do proper utilization of all the data collected and provide interface to users to interact with that data. These applications could be cloud based applications which are responsible for rendering data collected. Applications are user controllable and are delivery points of particular services.

Examples of applications are: smart home apps, security system control applications, industrial control hub applications, etc.

If we summarize the elements of IoT then we can say that the raw data gathered by the sensors are transferred to embedded processors. Processors transform raw data to meaningful information and then again transfer it to remote cloud based applications or database systems via gateway device connectivity. It then transfers to the applications for the proper applied usage and also for data analysis via big data.

Smart City - Identify different IoT Application with IoT Architecture

Test signal flow into IoT enabled system/ applications as per the IoT architecture

Objective:

- to test signal flow into IoT enabled system/ applications as per the IoT architecture.

Requirements:

- Cloud
- Server
- Computer system
- Internet Connectivity
- Sensors
- IoT devices

PROCEDURE:

INTRODUCTION

The IOT provides potentialities (capabilities) which is necessary for the development of a huge number of applications but only a very small part of these applications is currently available to our society. The new applications would likely improve the quality of our livelihood in many domains and environments such as at home, while travelling, when sick, at work, when jogging and at the gym. Such environments furnish objects with prehistoric brainpower and the communication capabilities are neglected most of the times. Such objects are enabled to communicate and elaborate the information perceived from the surrounding and it implies that different environments are having a wide range of applications which can be deployed.

These can be grouped into the following domains:

Smart Interactive Office Automation Smart Museum System

Smart Air and Sound Pollution Monitoring System Smart Car Parking System

Smart Gym System

Smart Irrigation System

Smart Garbage Monitoring System Smart Healthcare System

Smart Wild Animal Intrusion Detection System

Smart Traffic Congestion Monitoring System In the above-listed applications domains, we distinguish between those applications that are directly relevant or near enough to our present-day livelihood and the future domains, which can only elaborate in a short time.

Since our present civilization is not organized for their deployment. In the following subsections, we dispense complete details of each application. This paper is divided into 5 sections: Section I gives a brief concept of IOT and its application domains. Section II provides the working principle, implementation and architecture of the above listed application domains. Section III throws light on the comparative study of each application discussed in Section II like the Hardware components used, programming language and communication type. Section IV concludes the paper and Section V shows the experimental results and scope of future works for each application discussed.

APPLICATION WITH ARCHITECTURE

Smart Interactive Home and Office Automation System

The word automation means working by itself with little or no direct human control. A smart office automation system makes employees or customers life easy and comfort, which induce it and grow their ability to stay connected. This can achieve by using various advanced technology and different tools to improve the efficiency of users. A smart office is one that assures the effective and optimal utilization of IoT devices. The Internet can be used as a medium to connect and control office appliances under the Internet of Things(IoT)

- System Architecture: There are different techniques and devices can be used to implement this system. We can use Arduino or Raspberry Pi web-based, email-based, mobile based, SMS based, Zigbee based, Cloud-based, etc. In this system the IoT devices that are used are listed in Table 1: Figure 1 shows a block diagram of a system where the main controller is Raspberry Pi which is connected with electrical appliances(fan, light, etc) via a relay module.
- The system can use Wi-Fi or Ethernet port for connection of Internet. To operate this system we have a mobile application which is basically an interface between system and users. The mobile application provides facility to ON/OFF the electrical appliance, sensing camera information, etc. As we can see in figure 1, visitor counter. This module actually counts the number of visitors entered into office.

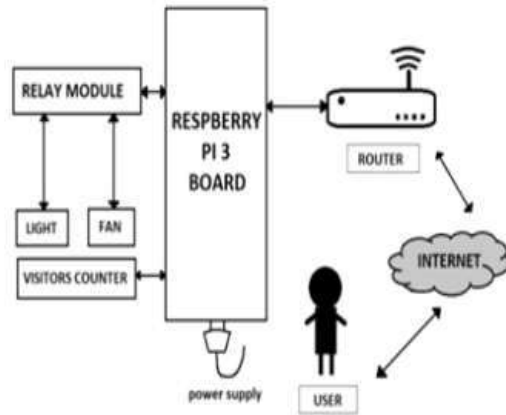


Fig. 1 System Architecture

Table. 1 Components used for Smart Office Automation System

Component	Working
Raspberry Pi-3	Used as a microcontroller and interface between all other devices.
Relay circuit	It allows low power circuit to switch a relatively high voltage or current ON/OFF.
PIR sensor	It detects movements of human being at the time, when he/she comes in contact with it.
Mobile	It is used as an interface between a user and system.

To implement this we used two PIR sensor placed at the entrance door with some distance between them. The main purpose of using the two PIR sensor is for detecting the direction of visitors. In other words, we can say to

check whether visitor is entering into office or exiting from office. The logic behind this is supposed one PIR sensor is PIR1(outside) and another one is PIR2(inside). When PIR1(outside) is cut followed by PIR2(inside) that means a visitor to enter and when PIR2(inside) is cut followed by PIR1(outside) then the visitor is exited from office as shown in figure 2.

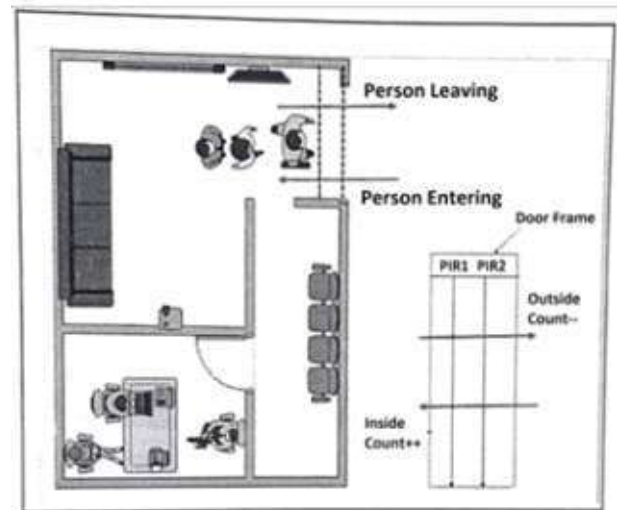


Fig. 2 Visitor Counter

The logic of cutting sensor is:
 If (PIR1) f
 i f (PIR2) f
 counter;
 g
 g else if (PIR2) f
 if (PIR1) f
 counter ++ ;
 g

Connect and test Arduino board to computer and execute sample programs from the example list

Objective:

- to connect and test Arduino board to computer and execute sample programs from the example list.

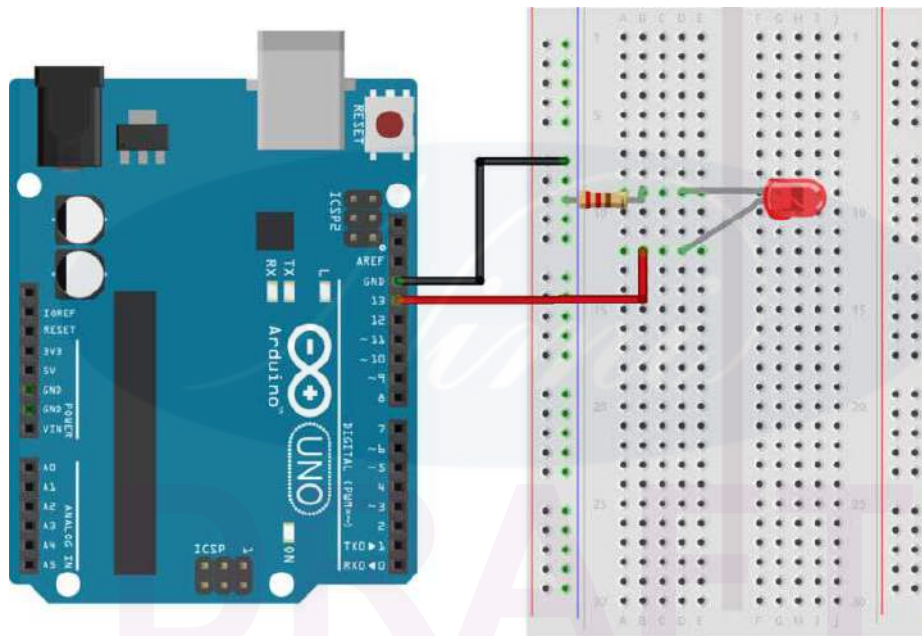
Requirement:

- Arduino Uno Board
- Breadboard – half size
- Jumper Wires
- USB Cable
- LED (5mm)

- Push button switch
- 10k Ohm Resistor
- 220 Ohm Resistor

PROCEDURE:

Simple Arduino Projects For Beginners



Arduino Projects

In this we're going to help you create a few simple arduino projects that are perfect for beginners. These basic projects will help you understand how to set up the Arduino software and then connect the components to perform a specific action.

If you're completely brand new to Arduino, make sure you download our free ebook below. This guide was created for the absolute beginner and will help you to understand the Arduino board along with its parts and components.

Tools and Parts Needed

In order to complete the projects in this tutorial, you'll need to make sure you have the following items.

Need the parts? Get the [Arduino Starter Kit w/ Project Ebook](#)

Download The Software

At this point, we're ready to download the free software known as the IDE. The Arduino IDE is the interface where you will write the sketches that tell the board what to do

To install the software, you will need to click on the link that corresponds with your computer's operating system.

Write and upload computer code to the physical Arduino board micro controller to sound buzzer

Objective:

- to write and upload computer code to the physical Arduino board micro controller to sound buzzer.

Requirement:

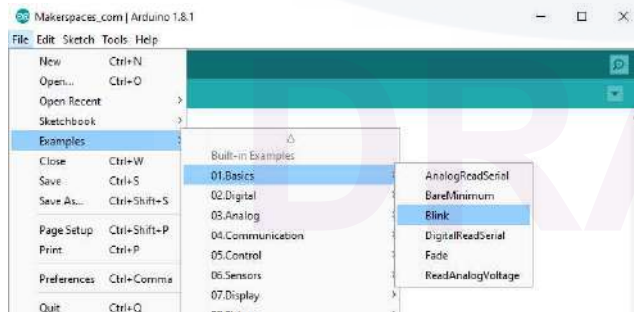
- Arduino Uno Board
- Breadboard – half size
- Jumper Wires
- USB Cable
- LED (5mm)
- Push button switch
- 10k Ohm Resistor
- 220 Ohm Resistor

PROCEDURE:

Upload the Blink Sketch

Now it's time to upload the sketch (program) to the Arduino and tell it what to do. In the IDE, there are built-in example sketches that you can use which make it easy for beginners.

To open the blink sketch, you will need to go to **File > Examples > Basics > Blink**



Now you should have a fully coded blink sketch that looks like the image below.



Next, you need to click on the verify button (check mark) that's located in the top left of the IDE box. This will compile the sketch and look for errors. Once it says "Done Compiling" you are ready to upload it. Click the upload button (forward arrow) to send the program to the Arduino board.



The built-in LEDs on the Arduino board will flash rapidly for a few seconds and then the program will execute. If everything went correctly, the LED on the breadboard should turn on for a second and then off for a second and continue in a loop.



Circuit and program to interface light sensor - LDR with Arduino to switch ON/OFF LED based on light intensity

Objective:

- to circuit and program to interface light sensor- LDR with Arduino to switch On/Off LED based on light intensity.
-

Requirements:

- LED
- LDR
- Arduino
- USB
- Connecting wires
- Computer system

PROCEDURE:**Instructions I**

- Connect the 3.3v output of the Arduino to the positive rail of the breadboard
- Connect the ground to the negative rail of the breadboard
- Place the LDR on the breadboard
- Attach the 10K resistor to one of the legs of the LDR
- Connect the A0 pin of the Arduino to the same column where the LDR and resistor is connected (Since the LDR gives out an analog voltage, it is connected to the analog input pin on the Arduino. The Arduino, with its built-in ADC (Analog to Digital Converter), then

converts the analog voltage from 0-5V into a digital value in the range of 0-1023). - Now connect the other end of the 10K resistor to the negative rail

- And the the second (free) leg of the LDR to the positive rail

Pretty much this is what we need for the light sensing. Basic circuits like this can be done without an Arduino aswell. However, if you want to log the values and use it to create charts, run other logics etc. I will recomend an Arduino or ESP8266 or may be a ESP32 for this.

Now, as we want our circuit to do something in the real world other than just displaying the values on the computer screen we will be attaching a LED to the circuit. The LED will turn on when its dark and will go off when its bright. To achieve this we will:

- Place the LED on the breadboard
- Connect the 220ohm resistor to the long leg (+ve) of the LED
- Then we will connect the other leg of the resistor to pin number 13 (digital pin) of the Arduino
- and the shorter leg of the LED to the negative rail of the breadboard

Set up & test circuit to interface potentiometer with Arduino board and map to digital values for eg- 0-1023

Objective:

- to set up and test circuit to interface potentiometer with Arduino board and map to digital values.

Requirement:

- Arduino or Genuino Board
- Potentiometer *or*
- 10K ohm photoresistor and 10K ohm resistor
- built-in LED on pin 13 *or*
- 220 ohm resistor and red LED

PROCEDURE:

A potentiometer is a simple knob that provides a variable resistance, which we can read into the Arduino board as an analog value. In this example, that value controls the rate at which an LED blinks.

We connect three wires to the Arduino board. The first goes to ground from one of the outer pins of the potentiometer. The second goes from 5 volts to the other outer pin of the potentiometer. The third goes from analog input 2 to the middle pin of the potentiometer.

By turning the shaft of the potentiometer, we change the amount of resistance on either side of the wiper which is connected to the center pin of the potentiometer. This changes the relative “closeness” of that pin to 5 volts and ground, giving us a different analog input. When the shaft is turned all the way in one direction, there are 0 volts going to the pin, and we read 0. When the shaft is turned all the way in the other direction, there are 5 volts going to the pin and we read 1023. In between, `analogRead()` returns a number between 0 and 1023 that is proportional to the amount of voltage being applied to the pin.

In this example we use a variable resistor (a potentiometer or a photoresistor), we read its value using one analog input of an Arduino or Genuino board and we change the blink rate of the built-in LED accordingly.

The resistor's analog value is read as a voltage because this is how the analog inputs work.

Circuit

With a potentiometer click the image to enlarge

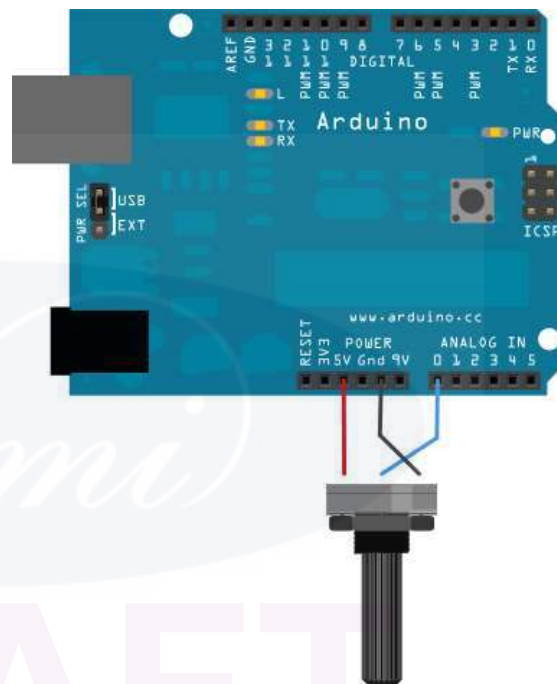
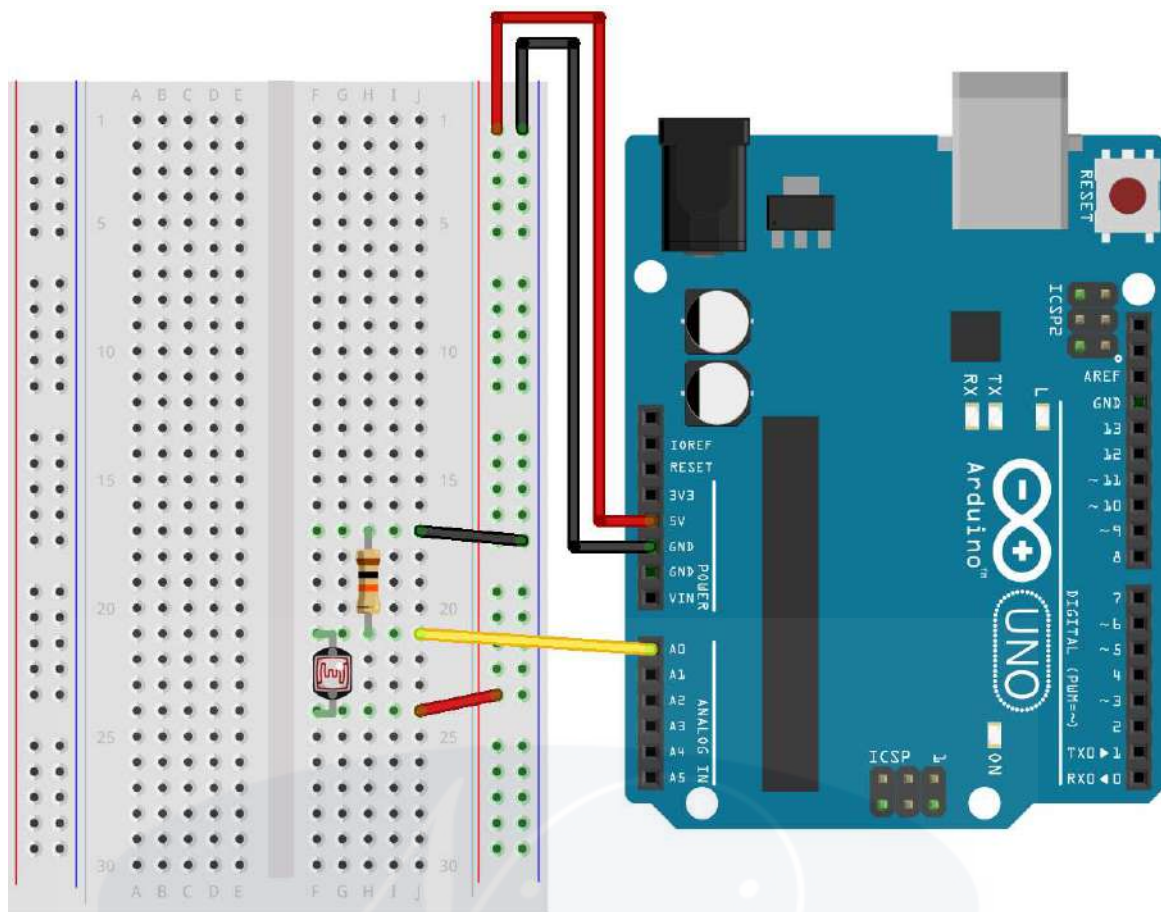


image developed using [Fritzing](#). For more circuit examples, see the [Fritzing project page](#)

With a photoresistor click the image to enlarge



Made with  Fritzing.org

image developed using [Fritzing](#). For more circuit examples, see the [Fritzing project page](#)

Connect three wires to the Arduino or Genuino board. The first goes to ground from one of the outer pins of the potentiometer. The second goes from 5 volts to the other outer pin of the potentiometer. The third goes from analog input 0 to the middle pin of the potentiometer.

For this example, it is possible to use the board's built in LED attached to pin 13. To use an additional LED, attach its longer leg (the positive leg, or anode), to digital pin 13 in series with the 220 ohm resistor, and its shorter leg (the negative leg, or cathode) to the ground (GND) pin next to pin 13.

The circuit based on a photoresistor uses a resistor divider to allow the high impedance Analog input to

measure the voltage. These inputs do not draw almost any current, therefore by Ohm's law the voltage measured on the other end of a resistor connected to 5V is always 5V, regardless the resistor's value. To get a voltage proportional to the photoresistor value, a resistor divider is necessary. This circuit uses a variable resistor, a fixed resistor and the measurement point is in the middle of the resistors. The voltage measured (V_{out}) follows this formula:

$$V_{out} = V_{in} * (R_2 / (R_1 + R_2))$$

where V_{in} is 5V, R_2 is 10k ohm and R_1 is the photoresistor value that ranges from 1M ohm in darkness to 10k ohm in daylight (10 lumen) and less than 1k ohm in bright light or sunlight (>100 lumen).

Rig up the circuit and upload a program to control a relay switch on/off LED light using Arduino

Objective:

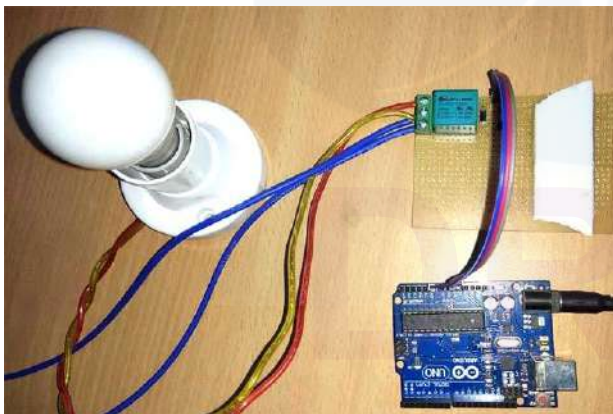
- to rig up the circuit and upload a program to control a relay switch on/off LED light using Arduino.

Requirements:

- Arduino
- 5v or 6v relay
- AC appliance or Bulb
- BC547 transistor
- 1k resistor
- Breadboard or PCB
- Connecting jumper wire
- Power supply
- 1n4007 diode
- Screw terminal or terminal block

PROCEDURE:

Arduino Relay Control



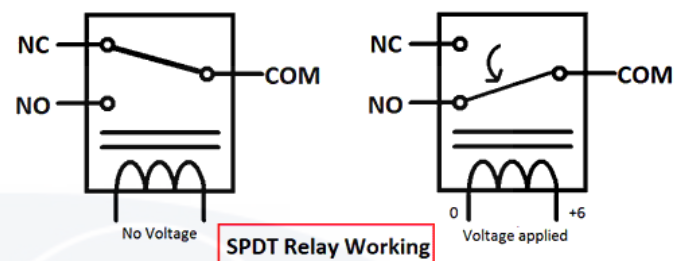
LED Blinking is a very common and almost first program for every embedded learner or beginner. In which we blink an LED with having some delay. So today we are here with the same project but here we will use an AC bulb instead of normal LED and will blink an AC bulb.

Whenever we need to connect any AC Appliance in our embedded circuits, we use a Relay. So in this arduino relay control tutorial we will simply learn **How to interface a Relay with Arduino**. Here we are not using any Relay Driver IC like ULN2003 and will only use an NPN transistor to control relay.

Relay:

Relay is an electromagnetic switch, which is controlled

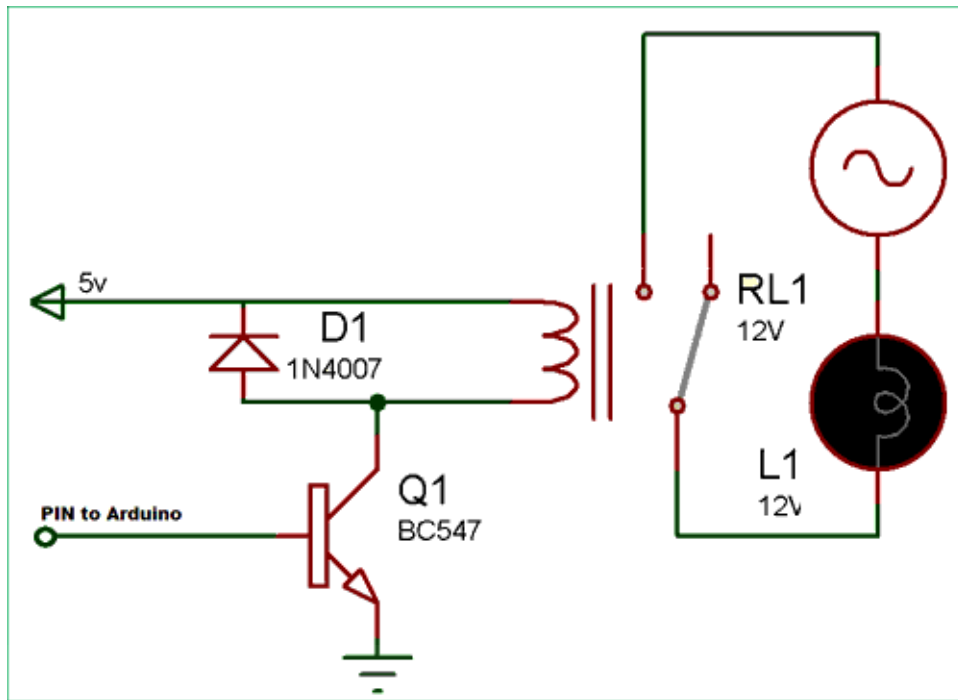
by small current, and used to switch ON and OFF relatively much larger current. Means by applying small current we can switch ON the relay which allows much larger current to flow. A relay is a good example of controlling the AC (alternate current) devices, using a much smaller DC current. Commonly used Relay is **Single Pole Double Throw (SPDT) Relay**, it has five terminals as below:



When there is no voltage applied to the coil, COM (common) is connected to NC (normally closed contact). When there is some voltage applied to the coil, the electromagnetic field produced, which attracts the Armature (lever connected to spring), and COM and NO (normally open contact) gets connected, which allow a larger current to flow. Relays are available in many ratings, here we used 6V operating voltage relay, which allows 7A-250VAC current to flow.

The relay is always configured by using a small **Driver circuit** which consists a Transistor, Diode and a resistor. **Transistor** is used to amplify the current so that full current (from the DC source – 9v battery) can flow through a coil to fully energize it. The **resistor** is used to provide biasing to the transistor. And **Diode** is used to prevent reverse current flow, when the transistor is switched OFF. Every Inductor coil produces equal and opposite EMF when switched OFF suddenly, this may cause permanent damage to components, so Diode must be used to prevent reverse current. A **Relay module** is easily available in the market with all its Driver circuit on the board or you can create it on perf board or PCB like below. Here we have used 6V Relay module.

Here to **turn on the Relay with Arduino** we just need to make that Arduino Pin High (A0 in our case) where Relay module is connected. Below given is **Relay Driver Circuit** to build your own Relay module:



Relay Driver Circuit



DRAFT

Smart City - Identify, test and interconnect components/ parts of IoT system

Make circuit and upload a program to interface of LCD display with a microcontroller to display characteristics

Objective:

- to make circuit and upload a program to interface of LCD display with a microcontroller to display characteristics.

Requirements:

- Arduino Uno
- Microcontroller
- LCD
- Jumper wires
- Computer system

PROCEDURE:

Interfacing LCD to Arduino – Display Text and Characters on LCD Screen using Arduino

In this guide, we're learning how to interface LCD to Arduino and display text characters on LCD screen. We're interfacing 16×2 LCD to Arduino as a demonstration with circuit and code. Let's begin.

A **Liquid Crystal Display** commonly abbreviated as **LCD** is basically a display unit built using Liquid Crystal technology. When we build real life/real world electronics based projects, we need a medium/device to display output values and messages. The most basic form of electronic display available is **7 Segment display** – which has its own limitations. The next best available option is **Liquid Crystal Displays** which comes in different size specifications. Out of all available LCD modules in market, the most commonly used one is **16×2 LCD Module** which can display 32 ASCII characters in 2 lines (16 characters in 1 line). Other commonly used LCD displays are 20×4 Character LCD, Nokia 5110 LCD module, 128×64 Graphical LCD Display and 2.4 inch TFT Touch screen LCD display.

In this article, we are going to learn how to interface lcd to arduino with 2 examples – one being interfacing a **16×2 LCD module to Arduino** and the other being interfacing a **20×4 LCD module to Arduino**.

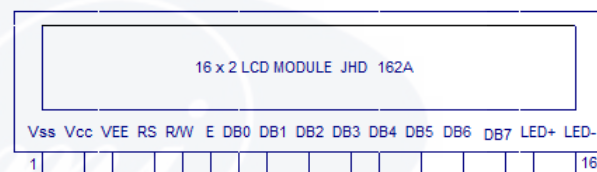
Interfacing 16×2 LCD to Arduino uno

LCD modules form a very important part in many arduino based embedded system designs. So the knowledge on interfacing LCD module to arduino is very essential in designing embedded systems. This section of the article is about interfacing an Arduino to 16×2 LCD. JHD162A is the LCD module used here. JHD162A is a 16×2 LCD module based on the **HD44780 driver from Hitachi**. The JHD162A has 16 pins and can be operated in 4-bit mode (using only 4 data lines) or 8-bit mode (using all 8 data

lines). Here we are using the LCD module in 4-bit mode. First, I will show you how to display a plain text messages on the LCD module using arduino and then I have designed a useful project using LCD and arduino – a **digital thermometer**. Before going in to the details of the project, let's have a look at the JHD162A LCD module.

16×2 LCD Module Pin Out Diagram

The JHD162A lcd module has 16 pins and can be operated in 4-bit mode or 8-bit mode. Here we are using the LCD module in 4-bit mode. Before going in to the details of the project, let's have a look at the JHD162A LCD module. The schematic of a JHD162A LCD pin diagram is given below.



The name and functions of each pin of the 16×2 LCD module is given below.

Pin1(Vss): Ground pin of the LCD module.

Pin2(Vcc): Power to LCD module (+5V supply is given to this pin)

Pin3(VEE): Contrast adjustment pin. This is done by connecting the ends of a 10K potentiometer to +5V and ground and then connecting the slider pin to the VEE pin. The voltage at the VEE pin defines the contrast. The normal setting is between 0.4 and 0.9V.

Pin4(RS): Register select pin. The JHD162A has two registers namely **command register** and **data register**. Logic HIGH at RS pin selects data register and logic LOW at RS pin selects command register. If we make the RS pin HIGH and feed an input to the data lines (DB0 to DB7), this input will be treated as data to display on LCD screen. If we make the RS pin LOW and feed an input to the data lines, then this will be treated as a command (a command to be written to LCD controller – like positioning cursor or clear screen or scroll).

Pin5(R/W): Read/Write modes. This pin is used for selecting between read and write modes. Logic HIGH at this pin activates read mode and logic LOW at this pin activates write mode.

Pin6(E): This pin is meant for enabling the LCD module. A HIGH to LOW signal at this pin will enable the module.

Pin7(DB0) to Pin14(DB7): These are data pins. The commands and data are fed to the LCD module through these pins.

Pin15(LED+): Anode of the back light LED. When operated on 5V, a 560 ohm resistor should be connected in series to this pin. In arduino based projects the back

light LED can be powered from the 3.3V source on the arduino board.

Pin16(LED-): Cathode of the back light LED.

For knowing more about LCD module JHD162A and its pin functions, read this article: [Interfacing 16×2 LCD and 8051 microcontroller](#). The circuit diagram of interfacing LCD to arduino for displaying a text message is shown below.



Rig up the circuit and upload a program to interface temperature sensor - LM 35 with a controller to display temperature on the LCD

Objective:

- to rig up the circuit and upload a program to interface temperature sensor – LM 35 with a controller to display temperature on the LCD.
-

Requirements:

- Arduino UNO
- LCD module
- Temperature sensor
- Computer system

PROCEDURE:

- In this article, we are presenting a tutorial on how to interface LM35 and Arduino along with its program. Once we successfully interface arduino and lm35, we will go on to build a temperature display using arduino and a 16×2 LCD module which constantly monitors temperature around the measurement field/range of LM35 and displays the same on LCD module. So lets get to building the project!
- LM35 is an analog, linear temperature sensor whose output voltage varies linearly with change in temperature. LM35 is three terminal linear temperature sensor from National semiconductors. It can measure temperature from -55 degree celsius to +150 degree celsius. The voltage output of the LM35 increases 10mV per degree Celsius rise in temperature. LM35 can be operated from a 5V supply and the stand by current is less than 60uA. The pin out of LM35 is shown in the figure below.

- So that's all info you need about LM35 for this particular temperature display project using arduino uno. So lets get to LM35 temperature sensor interfacing with arduino!
- We are using Arduino Uno as our board and LM35 can be connected to arduino as shown in circuit diagram.
- **Note:-** LM35 is an analog temperature sensor. This means the output of LM35 is an analog signal. Microcontrollers dont accept analog signals as their input directly. We need to convert this analog output signal to digital before we can feed it to a microcontroller's input. For this purpose, we can use an ADC(Analog to Digital Converter).If we are using a basic microcontroller like 8051, we need to use an external ADC to convert analog output from LM35 to digital. We then feed the output of ADC (converted digital value) to input of 8051. But modern day boards like Arduino and most modern day micro controllers come with inbuilt ADC. Our arduino uno has an in built 10 bit ADC (6 channel). We can make use of this in built ADC of arduino to convert the analog output of LM35 to digital output. Since Arduino uno has a 6 channel inbuilt ADC, there are 6 analog input pins numbered from A0 to A5. Connect analog out of LM35 to any of these analog input pins of arduino.

Setup circuit and upload a program to interface DC motor (actuator) with microcontroller to control on/off/forward/reverse operations

Objective:

- to setup circuit and upload a program to interface DC motor (actuator) with microcontroller to control on/off/forward/reverse operations.

Requirement:

- Arduino UNO
- USB Cable
- Resistors (R1, R2, R3, R4) – 1K Ω
- Diodes (D1, D2, D3, D4) – 1N4007
- Transistors (Q1, Q2, Q3, Q4) – 2N2222
- DC Motor
- Push Button
- Potentiometer – 10K Ω
- Connecting Wires
- Breadboard
- 9V Battery
- Battery Connector

- Brushless DC motor,
- Geared DC motor,
- Servo motor,
- Stepper motor and
- DC Linear Actuator.

Different types of motors are used in different applications like Robotics, precision positioning, industrial automation etc.

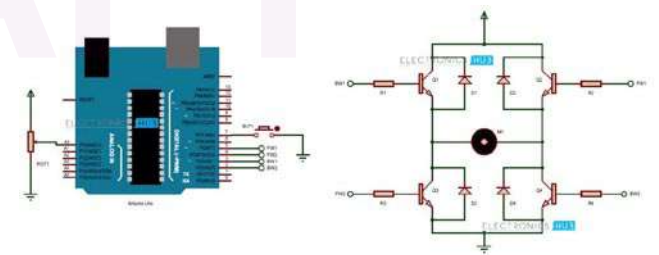
Generally, when a DC motor is associated with any microcontroller based system, it is often connected using a Motor Driver IC. A Motor Driver IC provides the necessary current for the motor to run. It can also control the direction of the rotation.

In this project, an Arduino based speed and direction control of DC motor without using Motor Driver IC is designed. A DC Motor can't be connected to a Microcontroller as the output current of the Microcontroller is very small and it cannot drive the motor.

Hence, we use transistors to form an H-bridge to drive the motor.

The circuit diagram, description and its working are mentioned below.

Circuit Diagram



Component Description

Arduino Uno

It is a Microcontroller Based prototyping board. The microcontroller used on the Arduino Uno board is ATmega328p. Arduino is responsible for controlling the speed and direction of the motor with the help of other components.

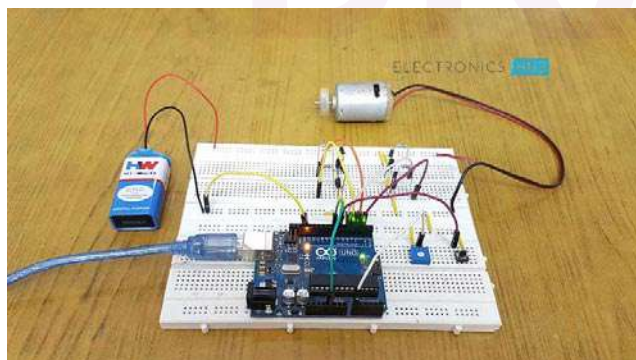
2N2222

It is an NPN transistor with an output current of 800mA.

PROCEDURE:

Speed and Direction Control of DC Motor using Arduino

In this project, I will show you how to achieve Speed and Direction Control of DC Motor using Arduino UNO. It is a simple project using Arduino UNO and a few easily available components to control the speed of rotation of a DC Motor and also its direction of rotation.



Introduction

DC Motors are found everywhere: electronics, toys, fans, tools, discs, pumps etc. DC Motor is an actuator that converts the DC supply to rotation or movement.

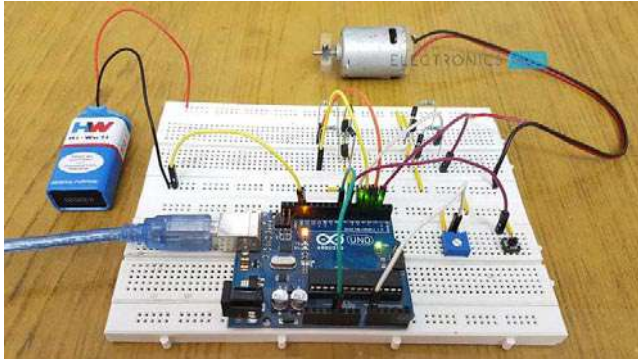
There are different types of DC motors:

- Brushed DC motor,

The maximum output current that is available from Arduino's I/O pins is 50mA, which is not sufficient to drive a motor. Hence, four transistors with high current capability are used.

Circuit Design

Arduino is the main processing unit of the project. The wiper terminal of the POT is connected to the Analog Pin (A0) of the Arduino. The other terminals are connected to Vcc and GND. Four transistors are connected as shown in the circuit diagram.



With the load i.e. a DC Motor in the center, they form an H – bridge. Transistors Q1 and Q4 form the backward direction path while transistors Q2 and Q3 form forward rotation path.

The inputs to the transistors are given from the Arduino.

The pins 3 and 2 of the Arduino are connected to the base of Q1 and Q4 respectively. Pins 5 and 4 are connected to base of Q2 and Q3 respectively. All these connections are made through four 1 K Ω resistors.

A DC Motor is an inductive load and can produce back EMF when we are changing the direction. In order to eliminate the effect of any back EMF, four diodes are connected across the collector and emitter of each transistor.

Working

The aim of this project is to control the speed and direction of a DC Motor without using a Motor Driver IC. Hence, we need to form an H-bridge using transistors in order to drive the motor. The working of the project is explained here assuming all the connections are made as per the circuit diagram.

The POT is connected to the analog pin A0 of the Arduino. This is used to adjust the speed of the motor. The normal operation of the motor is to rotate in forward direction.

When a button, which is connected to the Pin 7 of the Arduino, is activated or pressed, the direction of the rotation is reversed and continue to rotate in that direction until the button is pressed once again.

For forward rotation of the motor, transistors Q2 and Q3 must be turned on. Hence, the outputs 5 and 4 of the Arduino are high.

Rig up circuit and upload program microcontroller to switch on/ off two lights using Relay

Objective:

- to Rig up circuit and upload program microcontroller to switch on/off two lights using Relay.

Requirements:

- 1 Arduino UNO (or any Arduino compatible)
- 1 Relay module (any 5v relay will work but you'll need other components)
- 1 noise sensor FC-04 or similar
- 1 12v AC/DC adapter to power the Arduino
- A couple of female/male cables to connect the components
- 1 AC male plug
- 1 AC female plug
- AC power wire
- 1 Box to contain it all
- Software: Arduino IDE

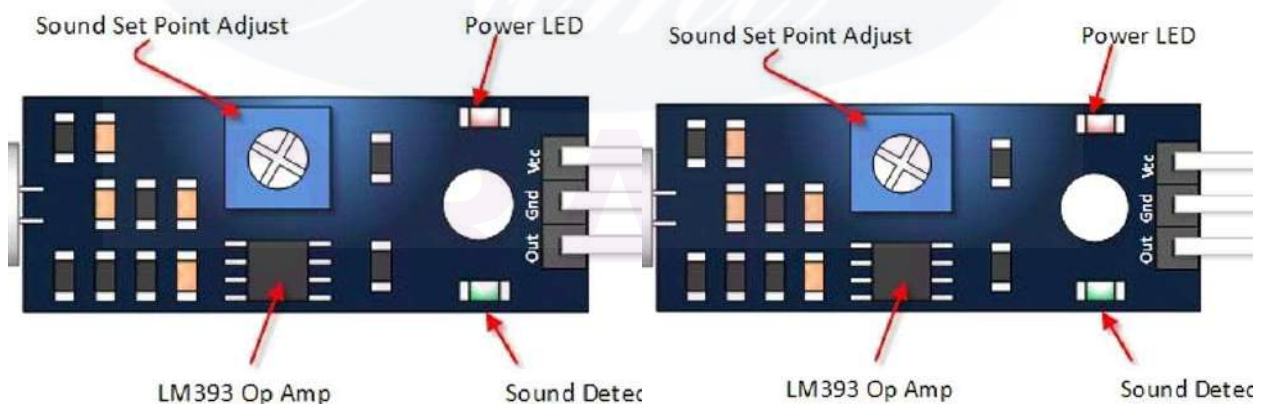
PROCEDURE:

Rig up Circuit and upload program micro-controller to switch on/off two lights using.

An simple, fun and useful device to impress your friends, feel like a bon vivant, or simply turn the lights off in the laziest way possible.

I set it to recognise two claps. Sometimes it can take other repeating noises in the room (like coughing) by mistake, but can be improved to any kind of clapping pattern.

Please be careful while you're working with AC power. Don't touch the connectors in the relay when is connected. If you're young, ask an adult for help.



The FC-04 sensor has three pins. Connect VCC to the 3.3v output pin and GND to GND on the Arduino board. For this tutorial we will connect the OUT data to pin 2 on the Arduino board, but you can use any digital pin.

I've found out that some sound sensors return an analog value with the level of noise detected. The one I'm using only returns a digital signal when it gets to the level set by the potentiometer in the sensor.

Smart City - Identify and test various parts of embedded system

Test main heart of embedded system / micro controller and micro controller hardware board /Hardware platform of an embedded system such as Arduino-Uno

Objectives:

- test the main heart of Arduino – UNO
 - micro controller hardware board
 - hardware platform of an embedded system.
-

Requirements:

- Arduino UNO – 1No
- USB cable – 1No
- DC Power Barrel Jack – 1 No

PROCEDURE:**Basics of Arduino UNO**

Let's have a look at the basic details of Arduino UNO:

- Arduino UNO is a Microcontroller board designed by Arduino.cc in Italy.
- It used Atmega328 Microcontroller which acts as a brain of this board.
- Arduino Bootloader is installed on Atmega328 which makes it capable to work with Arduino Programming.
- Arduino is an open-source platform so it has a lot of support from third-party developers.
- Anyone can design its Libraries for different sensors and modules.

Arduino UNO Pinout

The digital and analogue pins are capable of multiple functions and it totally depends on your projects' requirement. If you want to use SPI modules then you have to stick to SPI Pins and if you want to interface Serial module like GSM or GPS then you need to use Serial Pins. We can also design software serial as well.

Arduino UNO Memory Features

Memories are of main concern while selecting a microcontroller for your project. If you have bigger data or code etc to save then you shouldn't select this one, I would recommend Arduino Mega. Although the SD card option is always there, isn't it? So, let's have a look at its memory features:

- It has a flash memory of 32Kb.
- Arduino UNO has SRAM of 2KB.
- EEPROM memory of UNO is 1Kb.
- Bootloader of 2Kb is installed so we are left with 30kb Flash memory.

Installing Arduino Software Objective:

To download and install the Arduino software for programming your Arduino Uno microcontroller. You will run a simple "sketch" or program to blink an LED.

- A.** You will be connecting your computer to the Arduino via a USB 2.0 cable. This cable will provide power to the Arduino, and will allow you to transfer programs from your computer to the Arduino. In addition, if your program is using the Arduino serial interface, then you can display the data on your computer.

Our discussion of C programming was limited primarily to its application with the Arduino IDE (Integrated Development Environment). For this reason, we did not discuss input/out of data (excepting the above), files or advanced C features. However, you will be able to program and allow your Arduino to operate independently of the computer, and you may power it with a separate wall-mount power supply or even a battery.

- B.** Arduino home page is shown below (<https://www.arduino.cc/en/Guide/HomePage>)

Smart City - Identify and test various parts of embedded system

Power up Arduino Uno board, and test its Analog, Digital and I/O pins.

Objective:

- to test its Analog, Digital and I/O pins.

Requirements:

- Arduino Uno
- USB A-to-B Cable
- LED 5mm
- 220 Ω Resistor
- USB A-to-B Cable
- Breadboard – Half Size
- Jumper wires
- (1) 10K Ω Resistor
- (1) Push Button Switch

Task 1: Test its Analog, Digital and I/O pins

Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

How to use Arduino Board

The 14 digital input/output pins can be used as input or output pins by using `pinMode()`, `digitalRead()` and `digitalWrite()` functions in arduino programming. Each pin operate at 5V and can provide or receive a maximum of 40mA current, and has an internal pull-up resistor of 20-50 KOhms which are disconnected by default. Out of these 14 pins, some pins have specific functions as listed below:

- **Serial Pins 0 (Rx) and 1 (Tx):** Rx and Tx pins are used to receive and transmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.
- **External Interrupt Pins 2 and 3:** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **PWM Pins 3, 5, 6, 9 and 11:** These pins provide an 8-bit PWM output by using `analogWrite()` function.
- **SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK):** These pins are used for SPI communication.

- **In-built LED Pin 13:** This pin is connected with an built-in LED, when pin 13 is HIGH – LED is on and when pin 13 is LOW, its off.

Along with 14 Digital pins, there are 6 analog input pins, each of which provide 10 bits of resolution, i.e. 1024 different values. They measure from 0 to 5 volts but this limit can be increased by using AREF pin with `analogReference()` function.

- Analog pin 4 (SDA) and pin 5 (SCA) also used for TWI communication using Wire library.

Arduino Uno has a couple of other pins as explained below:

- **AREF:** Used to provide reference voltage for analog inputs with `analogReference()` function.
- **Reset Pin:** Making this pin LOW, resets the microcontroller.

Communication:

Arduino can be used to communicate with a computer, another Arduino board or other microcontrollers. The ATmega328P microcontroller provides UART TTL (5V) serial communication which can be done using digital pin 0 (Rx) and digital pin 1 (Tx). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The ATmega16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. There are two RX and TX LEDs on the arduino board which will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (not for serial communication on pins 0 and 1). A SoftwareSerial library allows for serial communication on any of the Uno's digital pins. The ATmega328P also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus.

Arduino Uno to ATmega328 Pin Mapping

When ATmega328 chip is used in place of Arduino Uno, or vice versa, the image below shows the pin mapping between the two.

Arduino function		Arduino function
reset	(PCINT14/RESET) PC8	PC5 (ADC5/SCL/PCINT13) analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	PC4 (ADC4/SDA/PCINT12) analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	PC3 (ADC3/PCINT11) analog input 3
digital pin 2	(PCINT18/INT0) PD2	PC2 (ADC2/PCINT10) analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	PC1 (ADC1/PCINT9) analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	PC0 (ADC0/PCINT8) analog input 0
VCC	VCC	GND
GND	GND	analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	AVCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	VCC
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	PB5 (SCK/PCINT5) digital pin 13
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	PB4 (MISO/PCINT4) digital pin 12
digital pin 7	(PCINT23/AIN1) PD7	PB3 (MOSI/OC2A/PCINT3) digital pin 11 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0	PB2 (SS/OC1B/PCINT2) digital pin 10 (PWM)
		PB1 (OC1A/PCINT1) digital pin 9 (PWM)

Digital Pins 11, 12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

Software

Arduino IDE (Integrated Development Environment) is required to program the Arduino Uno board. [Download it here.](#)

Programming Arduino

Once arduino IDE is installed on the computer, connect the board with computer using USB cable. Now open the arduino IDE and choose the correct board by selecting Tools>Boards>Arduino/Genuino Uno, and choose the correct Port by selecting Tools>Port. Arduino Uno is programmed using Arduino programming language based on Wiring. To get it started with Arduino Uno board and blink the built-in LED, load the example code by selecting Files>Examples>Basics>Blink. Once the example code (also shown below) is loaded into your IDE, click on the 'upload' button given on the top bar. Once the upload is finished, you should see the Arduino's built-in LED blinking. Below is the example code for blinking:

```
// the setup function runs once when you press reset or
power the board

void setup()
{ // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);}
```

// the loop function runs over and over again forever

void loop()

```
{ digitalWrite(LED_BUILTIN, HIGH); // turn the LED on
(HIGH is the voltage level)
```

```
delay(1000); // wait for a second
```

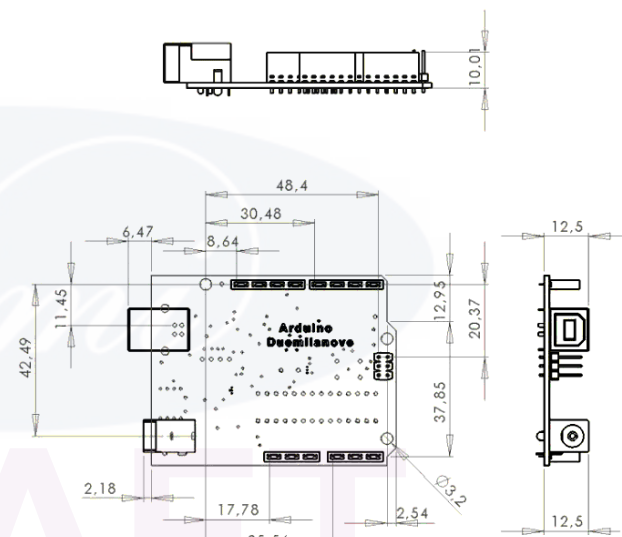
```
digitalWrite(LED_BUILTIN, LOW); // turn the LED off
by making the voltage LOW
```

```
delay(1000); // wait for a second}
```

Applications

- Prototyping of Electronics Products and Systems
- Multiple DIY Projects.
- Easy to use for beginner level DIYers and makers.
- Projects requiring Multiple I/O interfaces and communications.

Arduino Uno 2D Model



Smart City - Identify and test various parts of embedded system

Test and explore sensors and actuators such as LDR, temperature sensors, potentiometers, piezo- element, servo, relay and push buttons, LED, Tri color LED

Objective:

- to explore the sensors and actuators such as LDR, temperature sensors, potentiometer, and piezo – elements, servo, relay, push buttons, LED and Tri color LED.
-

Requirements:

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard – Half Size
- (1) Potentiometer – Rotary
- (10) LED 5mm
- (10) 220 Ω Resistor
- (11) Jumper Wires

Task1: To explore the sensors and actuators**LDR – Light Dependent Resistor**

A **Light Dependent Resistor** (also known as a photo resistor or LDR) is a device whose resistivity is a function of the incident electromagnetic radiation. Hence, they are light-sensitive devices. They are also called as photoconductors, photoconductive cells or simply photocells.

They are made up of semiconductor materials that have high resistance. There are many different symbols used to indicate a photo resistor or LDR, one of the most commonly used symbol is shown in the figure below. The arrow indicates light falling on it.

Working Principle of Photo resistor (LDR)

Photo resistors work based off of the principle of photoconductivity. Photoconductivity is an optical phenomenon in which the material's conductivity is increased when light is absorbed by the material.

When light falls i.e. when the photons fall on the device, the electrons in the valence band of the semiconductor material are excited to the conduction band. These photons in the incident light should have energy greater than the bandgap of the semiconductor material to make the electrons jump from the valence band to the conduction band.

Temperature sensors

A temperature sensor plays an important role in many applications. For example, maintaining a specific temperature is essential for equipment used to fabricate medical drugs, heat liquids, or clean other equipment. For applications like these, the responsiveness and accuracy of the detection circuit can be critical for quality control.

More frequently, however, temperature detection is part of preventative reliability. For example, while an appliance might not actually perform any high temperature activities, the system itself could be at risk to overheating. This risk arises from specific external factors such as a harsh operating environment or internal factors like self-heating of electronics. By detecting when overheating occurs, the system can take preventative action. In these cases, the temperature detection circuit must be reliable over the expected operating temperature range for the application.

Potentiometer

A potentiometer is a manually adjustable variable resistor with 3 terminals. Two terminals are connected to both ends of a resistive element, and the third terminal connects to a sliding contact, called a wiper, moving over the resistive element. The position of the wiper determines the output voltage of the potentiometer. The potentiometer essentially functions as a variable voltage divider. The resistive element can be seen as two resistors in series (potentiometer resistance), where the wiper position determines the resistance ratio of the first resistor to the second resistor.

A potentiometer is also commonly known as a **pot meter or pot**. The most common form of pot meter is the single turn rotary pot meter. This type of pot is often used in audio volume control (logarithmic taper) as well as many other applications. Different materials are used to construct potentiometers, including carbon composition, cermet, wire wound, and conductive plastic or metal film.

**Piezo Element**

A piezoelectric sensor is a device that uses the piezoelectric effect to detect changes in pressure, acceleration, temperature, strain, or force by converting

them to an electrical charge. A piezoelectric disc generates voltage when deformed. The converse is also true; when voltage is applied, the piezoelectric material distorts. If attached to a metal diaphragm, it becomes a kind of speaker that can be used to create a high-pitched tone. All Electronics carries both non-encased and encased piezo elements in various sizes and ratings.

LED

When exposed to light photodiodes produce a current that is directly proportional to the intensity of the light. This light generated current flows in the opposite direction to current in a normal diode or LED. As more photons hit the photodiode the current increases causing a voltage across the diode. As the voltage across the diode increases the linearity decreases.

In addition to emitting light, an LED can be used as a photodiode light sensor / detector. This capability may be used in a variety of applications including ambient light level sensor and bidirectional communications. As a photodiode, an LED is sensitive to wavelengths equal to or shorter than the predominant wavelength it emits. A green LED would be sensitive to blue light and to some green light, but not to yellow or red light. For example, a red LED will detect light emitted by a yellow LED and a yellow LED will detect light emitted by a green LED but a

green LED will not detect light emitted by a red or yellow LED. All three LEDs will detect “white” light or light from a blue LED. White light contains a blue light component which can be detected by the green LED. Recall that visible light wavelengths can be listed from longest wavelength to shortest wavelength as Red, Orange, Yellow, Green, Blue, Indigo, Violet (remember the mnemonic “Roy G. Biv”). Violet is the shortest wavelength light with the most energetic photons and red has the longest wavelength light with the least energetic photons of all of the visible colors of light. LED’s with clear plastic encapsulation will be more sensitive to broad-spectrum illumination (like general room lighting) than LED’s with colored encapsulation (such as those included in the ADALP2000 Analog Parts Kit).

To use the LED as an optical detector, do not forward bias the LED into quadrant #1 of the current-voltage (I-V) curve. (Quadrant 1 is when the operating voltage and current are both positive.) Allow the LED to operate in the solar cell mode, quadrant #4 (operating voltage is positive, current is negative), or in the photodiode mode quadrant #3 (operating voltage is negative, current is negative). In the solar cell mode, no applied bias voltage is used. The solar cell (or LED in this case) generates its own current and voltage.



Download and install Arduino Software IDE in computer system

Objective:

- to install Arduino Software IDE in computer system.

Requirement:

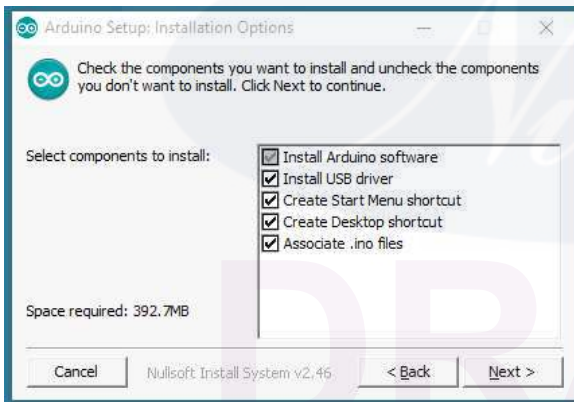
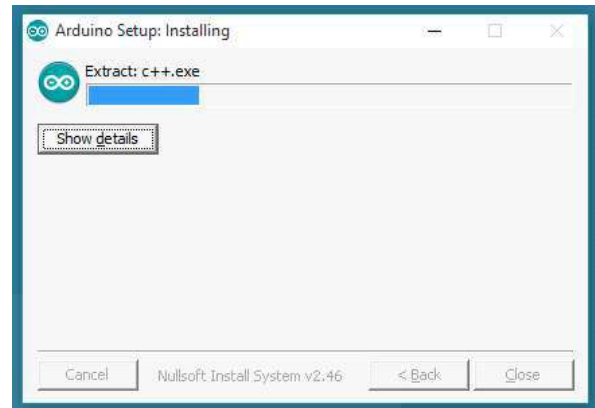
- Arduino IDE (.exe) files
- System 64 bit
- Windows / Linux operating system

PROCEDURE:**Task 1: Installing Arduino Software IDE in computer system.**

Download the Arduino Software (IDE)

Get the latest version from the [download page](#). You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually. The Zip file is also useful if you want to create a [portable installation](#).

Choose the installation directory (we suggest to keep the default one)



The process will extract and install all the required files to execute properly the Arduino Software (IDE)

Proceed with board specific instructions

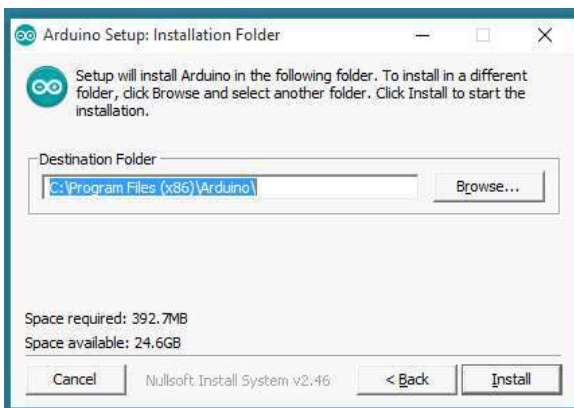
When the Arduino Software (IDE) is properly installed you can go back to the [Getting Started Home](#) and choose your board from the list on the right of the page.

Last revision 2016/08/09 by SM

The text of the Arduino getting started guide is licensed under a [Creative Commons Attribution-HYPERLINK](http://creativecommons.org/licenses/by-sa/3.0/) "http://creativecommons.org/licenses/by-sa/3.0/" [Share AlikeHYPERLINK](http://creativecommons.org/licenses/by-sa/3.0/) "http://creativecommons.org/licenses/by-sa/3.0/" [3.0 License](http://creativecommons.org/licenses/by-sa/3.0/). Code samples in the guide are released into the public domain.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

Choose the components to install



Test and familiarize with different components of Arduino IDE/Sketch

Objective:

- to test and familiarize with different components of Arduino IDE/Sketch.
-

Requirements:

- Arduino IDE
- Computer system
- Arduino UNO
- USB
- Sensors required

PROCEDURE:**Task 1: To test and familiarize with different components of Arduino IDE/ Sketch**

When we start developing an Arduino project usually we tend to use the **Arduino IDE**. There are, anyway, several Arduino IDE alternative we should consider if we are not satisfied with the standard IDE. As we know, Arduino is one of the most popular prototyping board (with several alternatives) when it comes the time to develop an IoT project or when we are building a DIY project.

Usually, the Arduino IDE is the first IDE we use when approaching the first time to Arduino. This happens for several reasons: it is easy to use, it supports all the Arduino boards, it has a built-in library manager easy to use. Moreover, the Arduino IDE is very user-friendly without too many options, menus and so on that could scare an inexperienced user. It is so easy that we do not have to worry about how it works, we can focus only on the development process. We write the Arduino code, and the Arduino IDE compiles it and upload the compiled code into the Arduino board. That's is.

Moreover, it is open source and it runs on several OS such as Windows, Mac OS X, and Linux. Anyway, if you do not feel comfortable with Arduino IDE there are several alternatives you can use when developing your Arduino project. There are some missing features in Arduino IDE that could bother you, especially if you are used to IDE like Eclipse, Visual Studio or IntelliJ, just to name a few. One missing feature in Arduino IDE, that really bothers me, is the autocomplete. That is a way to predict the rest of the bothers me, is the autocomplete. That is a way to predict the rest of the commands so that you do not have to write it all. There are some other features I

like in an IDE such as showing function parameters, hints related to the syntax error and so on.

Arduino IDE alternative list

This article wants to provide a list of Arduino IDE alternative you can use when creating your Arduino project if you do not like the standard Arduino IDE. The order is random.



This is a plugin for Eclipse. We all know Eclipse and the IDE. If you programmed in Java for sure you had the chance to test and appreciate the Eclipse IDE features. It is almost a standard in the IDE. Eclipse is a very versatile environment that supports several programming languages ranging from Java to C/C++ and so on. This platform uses the plugins to extend its features and add new platform support.

This is a plugin for Eclipse IDE that helps to develop Arduino applications. You can download the plugin directly from the Eclipse Marketplace. Of course, it is required you install first the Eclipse C++ IDE. Once, this plugin is installed in Eclipse IDE, you have to select the platform or in other words the SDK that includes all the libraries necessary to build the application. Once the plugin is configured, you are ready to develop your Arduino sketch. The IDE is essentially the Eclipse IDE so if you are familiar with Eclipse you do not have problems to use it. This IDE has all the features you have in the standard Arduino IDE such as Serial monitor to debug the Arduino sketch too.

Rig up the circuit to test Light Dependent Resistor to switch ON/OFF based light intensity

Objective:

- to test the Light Dependent Resistor to switch ON/OFF based light intensity.

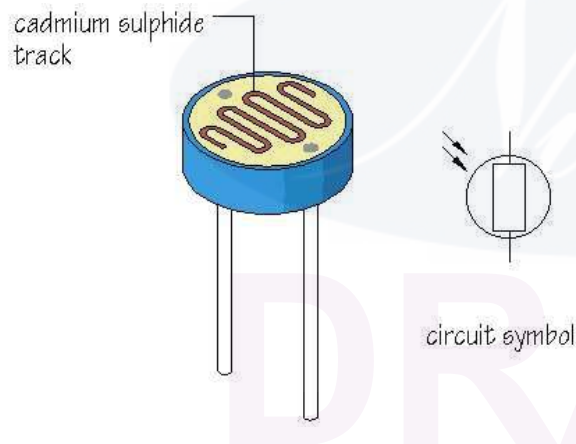
Requirements:

- Arduino Board
- Bread Board
- Jumper wires
- Indicator LED
- 100 OHM resistance
- 10k resistance

LDR

PROCEDURE:

TASK 1: Test Light Dependent Resistor to switch ON/OFF based light intensity



Majority of street lights, outdoor lights, and a number of indoor home appliances are typically operated and maintained manually in many occasions. This is not only risky, however additionally it leads to wastage of power with the negligence of personnel or uncommon circumstances in controlling these electrical appliances ON and OFF. Hence, we can utilize the light sensor circuit for automatic switch OFF the loads based on daylight's intensity by employing a light sensor. This article discusses in brief about what is a light dependent resistor, how to make a light dependent resistor circuit and its applications.

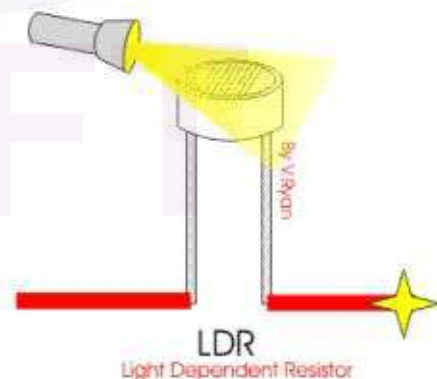
What is a Light Dependent Resistor?

An LDR or light dependent resistor is also known as

photo resistor, photocell, photoconductor. It is a one type of resistor whose resistance varies depending on the amount of light falling on its surface. When the light falls on the resistor, then the resistance changes. These resistors are often used in many circuits where it is required to sense the presence of light. These resistors have a variety of functions and resistance. For instance, when the LDR is in darkness, then it can be used to turn ON a light or to turn OFF a light when it is in the light. A typical light dependent resistor has a resistance in the darkness of 1M Ω , and in the brightness a resistance of a couple of K Ω .

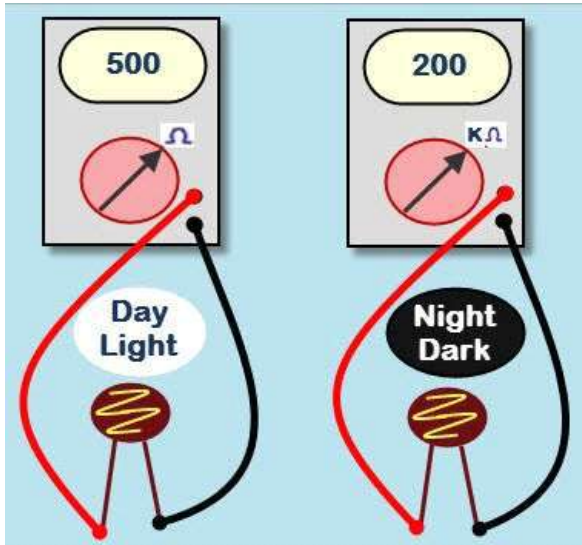
Working Principle of LDR

This resistor works on the principle of photo conductivity. It is nothing but, when the light falls on its surface, then the material conductivity reduces and also the electrons in the valence band of the device are excited to the conduction band. These photons in the incident light must have energy greater than the band gap of the semiconductor material. This makes the electrons to jump from the valence band to conduction.



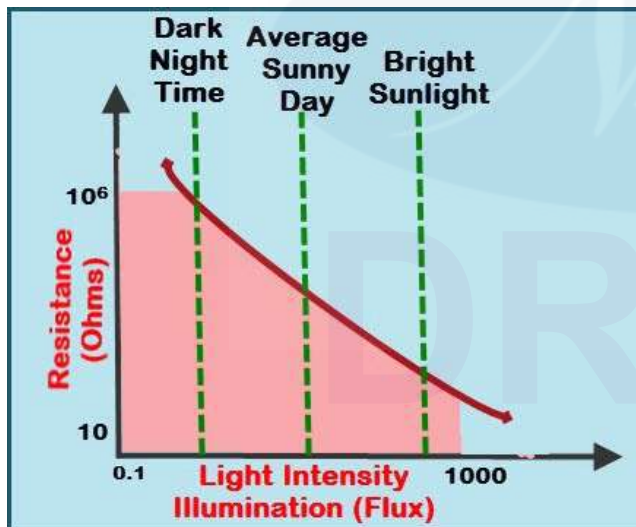
Working Principle of LDR

These devices depend on the light, when light falls on the LDR then the resistance decreases, and increases in the dark. When a LDR is kept in the dark place, its resistance is high and, when the LDR is kept in the light its resistance will decrease.



Variation of LDR Resistance with Variation in Light Intensity

If a constant “V” is applied to the LDR, the intensity of the light increased and current increases. The figure below shows the curve between resistance Vs illumination curve for a particular light dependent resistor.



Light Intensity vs LDR Resistance

Types of light Dependent Resistors

Light dependent resistors are classified based on the materials used.

Intrinsic Photo Resistors

These resistors are pure semiconductor devices like silicon or germanium. When the light falls on the LDR,

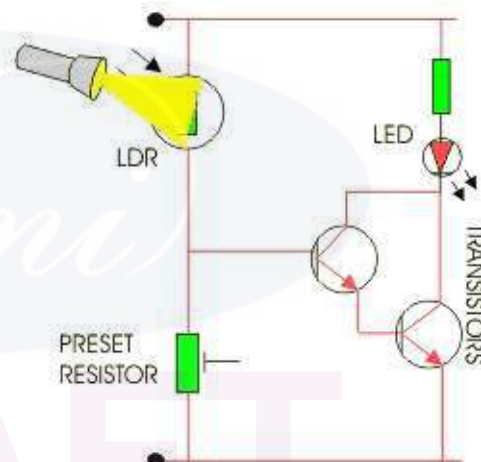
then the electrons get excited from the valence band to the conduction band and number of charge carriers increases.

Extrinsic Photo Resistors

These devices are doped with impurities and these impurities creates a new energy bands above the valence band. These bands are filled with electrons. Hence this decrease the band gap and small amount of energy is required in moving them. These resistors are mainly used for long wavelengths.

Circuit Diagram of a Light Dependent Resistor

The circuit diagram of a LDR is shown below. When the light intensity is low, then the resistance of the LDR is high. This stops the current flow to the base terminal of the transistor. So, the LED does not light. However, when the light intensity onto the LDR is high, then the resistance of the LDR is low. So current flows onto the base of the first transistor and then the second transistor. Consequently the LED lights. Here, a preset resistor is used to turn up or down to increase or decrease the resistance.



Light Dependent Resistor Circuit

Light Dependent Resistor Applications

Light dependent resistors have a low cost and simple structure. These resistors are frequently used as light sensors. These resistors are mainly used when there is a need to sense the absence and presence of the light such as burglar alarm circuits, alarm clock, light intensity meters, etc. LDR resistors mainly involves in various electrical and electronic projects. For better understanding of this concept, here we are explaining some real time projects where the LDR resistors are used.

- to Rig up a test circuit to display 0-9 Numbers on 7 segment display.

The seven segment display consists of 7 LEDs labelled 'a' through 'g'. By forward biasing different LEDs, we can display the digits 0 through 9. Seven segment displays are of two types, common cathode and common anode. In common anode type anodes of all the seven LEDs are tied together, while in common cathode type all cathodes are tied together. The seven segment display used here is a common anode type. Resistor R1 to R7 are current limiting resistors. IC 7446 is a decoder/driver IC used to drive the seven segment display.

D1
7 segment display

+5V DC

IC1
7446

IC2
7490

Clock

R1 to R7 = 220 ohm

All ICs must be mounted on holders.

Smart City - Identify and test various parts of embedded system

Rig up test circuit to control the backlight 16x2 character display

Objective:

- to rig up test circuit to control the backlight 16*2 character display.

Requirements:

- Arduino or Genuino Board
- LCD Screen (compatible with Hitachi HD44780 driver)
- pin headers to solder to the LCD display pins
- 10k ohm potentiometer
- 220 ohm resistor
- hook-up wires
- breadboard

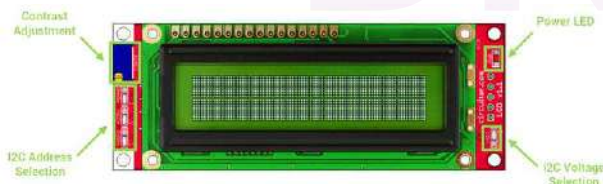
PROCEDURE:

The LCD Nanoshield uses an I2C communication interface, allowing greater flexibility when connecting it together with other Nanoshields:

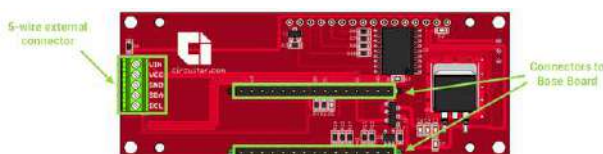
- You can connect it together with other Nanoshields that use I2C communications without any conflicts.
- Only **two** Arduino pins are used for all communication, freeing more pins for other functions.
- You can connect many independent LCDs on your project at the same time (up to **eight!**).

The internal LCD controller is compatible with the HD44780 chip from Hitachi, a de facto standard in the market for this kind of LCD. This is the same standard used in the LCD library that comes with the Arduino IDE.

To use the LCD Nanoshield, we provide an Arduino library that allows you to easily control the LCD using only two I/O pins of the I2C interface.



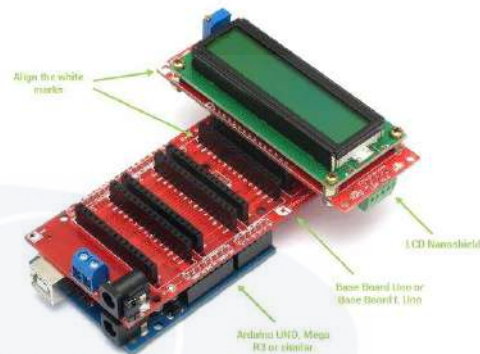
Top view (click to enlarge)



Bottom view (click to enlarge)

Arduino + Base Board Uno connection

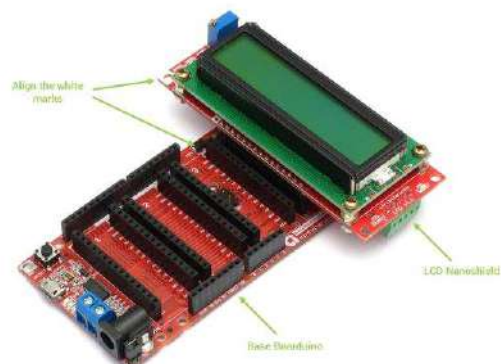
The easiest way to use the LCD Nanoshield with an Arduino is to use the Base Board Uno or the Base Board L Uno. You just need to snap the boards together and upload our sample code to verify it's working (see the code samples section below). This type of connection can be used with Arduino UNO, Mega R3, Duemilanove, and similar boards (contact us if you have questions about compatibility with other versions). The picture below shows how the final assembly looks like.



Connecting to an Arduino using the Base Board Uno (click to enlarge)

Base Boarduno connection

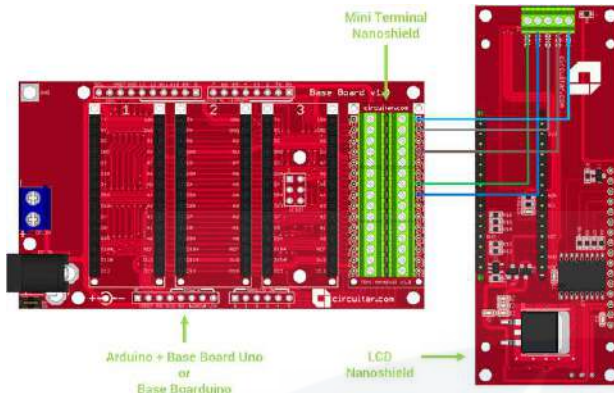
It is also possible to connect the LCD Nanoshield to our Arduino-compatible microcontroller board, the Base Boarduno. The connection is done in the same way as with the Base Board, as shown in the picture below. You just need to snap the boards together and upload our sample code to verify it's working (see the code samples section below).



Connecting to a Base Boarduino (click to enlarge)

Direct connection

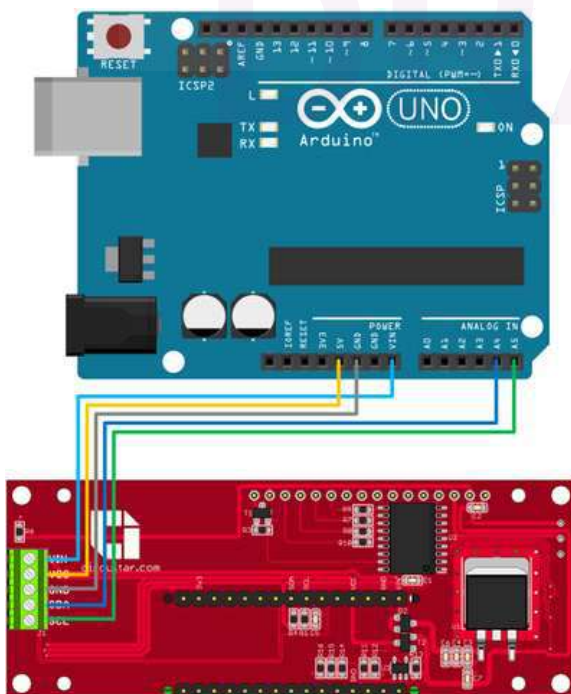
By using the [Mini Terminal HYPERLINK “https://www.circuitar.com/nanoshields/modules/mini-terminal/index.html”](https://www.circuitar.com/nanoshields/modules/mini-terminal/index.html) Nanoshield, it is possible to securely connect the LCD Nanoshield to an Arduino equipped with a [Base Board](https://www.circuitar.com/nanoshields/modules/base-boarduino/index.html) or to a [Base Boarduino](https://www.circuitar.com/nanoshields/modules/base-boarduino/index.html). This connection uses only five wires, and is useful when the LCD needs to be mounted away from the Base Board – for instance when it must be mounted on a panel or case. The diagram below shows how to make that connection.



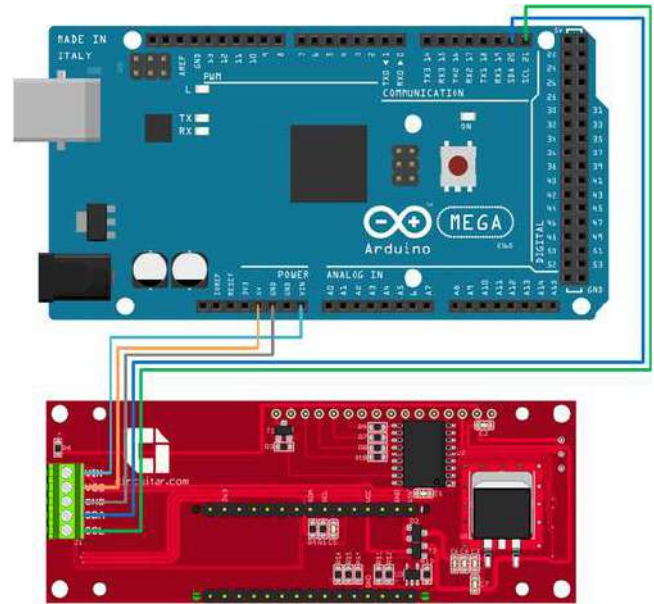
Connection using a Mini Terminal (click to enlarge)

Arduino wiring

The following diagram show how to connect the LCD Nanoshield directly to an Arduino UNO or an Arduino Mega.



Connecting to an Arduino UNO (click to enlarge)

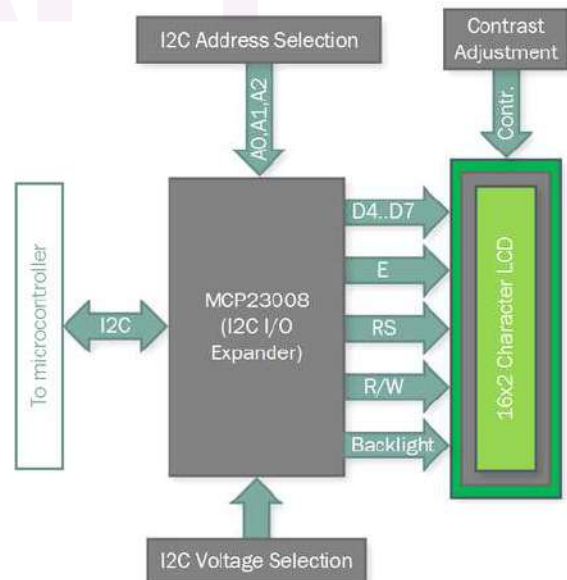


Connecting to an Arduino Mega (click to enlarge)

Pinout

LCD	Arduino	Função
SDA	A4	I2C data line
SCL	A5	I2C clock line
VIN	VIN	External power supply
VCC	VCC	5V external power supply
3V3	3V3	3.3V external power supply (optional)
GND	GND	Ground

Pinout table



Rig up the test circuit to control the relay

Objective:

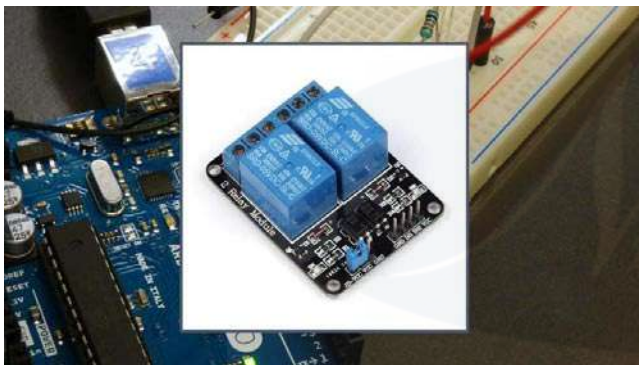
- to rig up the test circuit to control the relay.

Requirements:

- Arduino UNO
- PIR Motion Sensor
- ESP8266 Microcontroller
- Jumper Wires

PROCEDURE:

Here we will see how to control mains voltage with the Arduino using a relay module. We make a brief introduction to the relay module and build a simple project example with the Arduino. The example we'll build shows how to control a relay module with an Arduino and a PIR motion sensor.



A relay is an electrically operated switch that can be turned on or off, letting the current go through or not, and can be controlled with low voltages, like the 5V provided by the Arduino pins.

Controlling a relay module with the Arduino is as simple as controlling any other output as we'll see later on.



This relay module has two channels (those blue cubes). There are other models with one, four and eight channels.

This module should be powered with 5V, which is appropriate to use with an Arduino. There are other relay modules that are powered using 3.3V, which is ideal for ESP32, ESP8266, and other microcontrollers.

Get a relay module:

- 5V 2-channel relay module
- 5V 1-channel relay module
- 5V 8-channel relay module
- 3.3V 1-channel relay module
- Relay Pinout

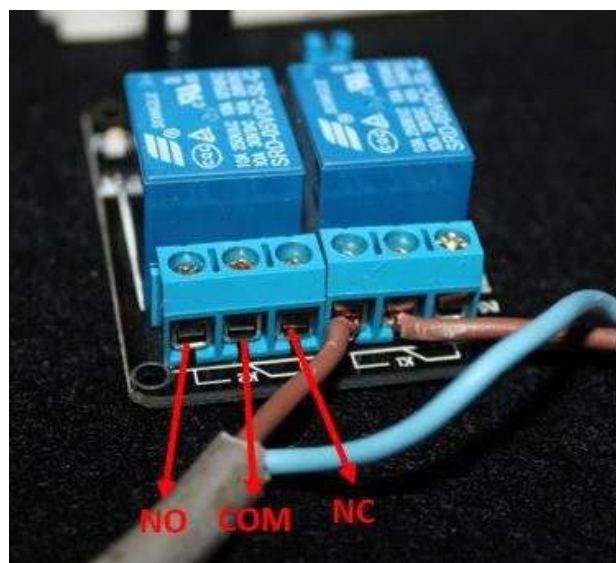
The following figure shows the relay module pinout.



The six pins on the left side of the relay module connect high voltage, and the pins on the right side connect the component that requires low voltage—the Arduino pins.

Mains voltage connections

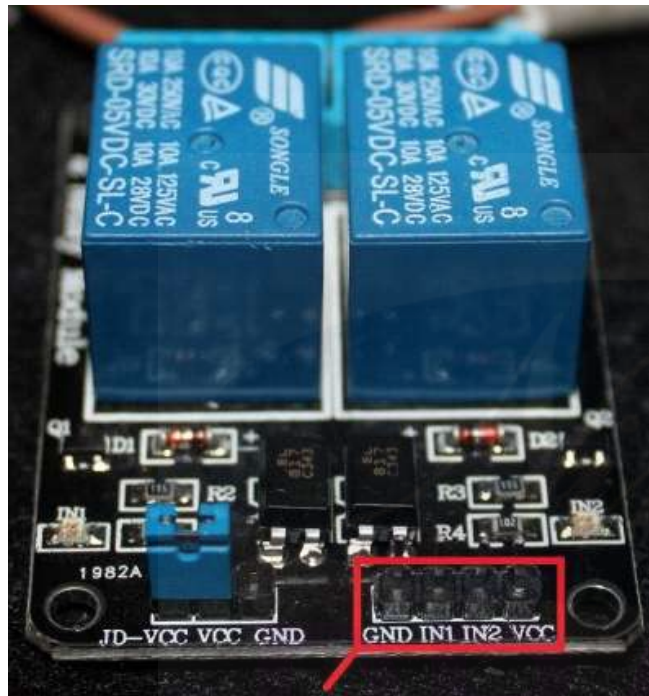
The high-voltage side has two connectors, each with three sockets: common (COM), normally closed (NC), and normally open (NO).



- **COM: common pin**
- **NC (Normally Closed):** the normally closed configuration is used when you want the relay to be closed by default, meaning the current is flowing unless you send a signal from the Arduino to the relay module to open the circuit and stop the current.
- **NO (Normally Open):** the normally open configuration works the other way around: the relay is always open, so the circuit is broken unless you send a signal from the Arduino to close the circuit.

Pin wiring

The low-voltage side has a set of four pins and a set of three pins.



The set at the right consists of VCC and GND to power up the module, and input 1 (IN1) and input 2 (IN2) to control the bottom and top relays, respectively.

The second set of pins consists of GND, VCC, and JD-VCC pins. The JD-VCC pin powers the electromagnet of the relay.

The connections between the relay module and the Arduino are really simple:

- GND: goes to ground
- IN1: controls the first relay (it will be connected to an Arduino digital pin)
- IN2: controls the second relay (it should be connected to an Arduino digital pin if you are using this second relay. Otherwise, you don't need to connect it)
- VCC: goes to 5V

Rig up the test circuit to control DC motor in Forward/ Reverse operations using DPDT switches

Objective:

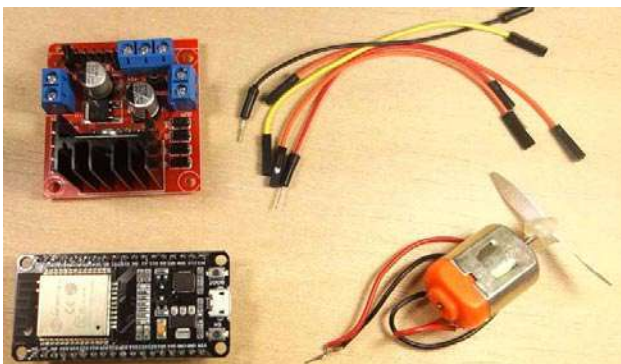
- to rig up the test circuit to control DC motor in Forward/ Reverse operation using DPDT Switches.

Requirements:

- ESP32 DOIT DEVKIT V1 Board – read ESP32 Development Boards Review and Comparison
- DC motor
- L298N motor driver
- Power source: 4x 1.5 AA batteries or Bench power supply
- 2x 100nF ceramic capacitors (optional)
- 1x SPDT slide switch (optional)
- Jumper wires

PROCEDURE:

Take a quick look on how the L298N motor driver works. Then, we'll show you an example on how to control the speed and direction of a DC motor using the ESP32 with Arduino IDE and the L298N motor driver.

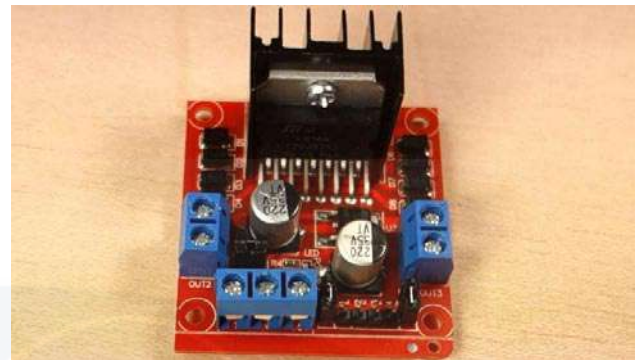


L298N Motor Driver

There are many ways to control a DC motor. The method we'll use here is suitable for most hobbyist motors, that require 6V or 12V to operate.

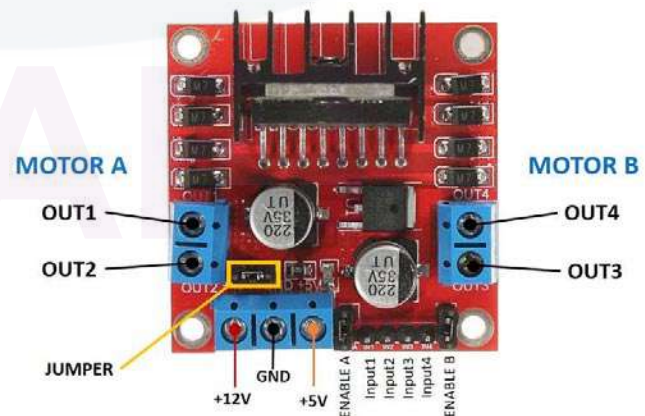
We're going to use the L298N motor driver that can handle up to 3A at 35V. Additionally, it allows us to drive two DC motors simultaneously, which is perfect to build a robot.

The L298N motor driver is shown in the following figure:



L298N Motor Driver pinout

Let's take a look at the L298N motor driver pinout and see how it works.



The motor driver has a two terminal block in each side for each motor. OUT1 and OUT2 at the left and OUT3 and OUT4 at the right.

- OUT1: DC motor A + terminal
- OUT2: DC motor A – terminal
- OUT3: DC motor B + terminal
- OUT4: DC motor B – terminal

At the bottom you have a three terminal block with **+12V**, **GND**, and **+5V**. The **+12V** terminal block is used to power up the motors. The **+5V** terminal is used to power up the L298N chip. However, if the jumper is in place, the chip is powered using the motor's power supply and you don't need to supply 5V through the **+5V** terminal.

It's important to note that despite the +12V terminal name, with the setup we'll use here (with the jumper in place) you can supply any voltage between 6V and 12V. In this tutorial will be using 4 AA 1.5V batteries that combined output approximately 6V, but you can use any other suitable power supply. For example, you can use a bench power supply to test this tutorial.

In summary

- **+12V:** The +12V terminal is where you should connect your power supply
- **GND:** power supply GND
- **+5V:** provide 5V if jumper is removed. Acts as a 5V output if jumper is in place
- **Jumper:** jumper in place – uses the motors power supply to power up the chip.

Jumper removed: you need to provide 5V to the +5V terminal. If you supply more than 12V, you should remove the jumper

At the bottom right you have four input pins and two enable terminals. The input pins are used to control the direction of your DC motors, and the enable pins are used to control the speed of each motor.

- **IN1:** Input 1 for Motor A
- **IN2:** Input 2 for Motor A
- **IN3:** Input 1 for Motor B
- **IN4:** Input 2 for Motor B
- **EN1:** Enable pin for Motor A
- **EN2:** Enable pin for Motor B

There are jumper caps on the enable pins by default. You need to remove those jumper caps to control the speed of your motors.

Control DC motors with the L298N

Now that you're familiar with the L298N Motor Driver, let's see how to use it to control your DC motors.

Enable pins

The enable pins are like an ON and OFF switch for your motors. For example:

- If you send a HIGH signal to the enable 1 pin, motor A is ready to be controlled and at the maximum speed;
- If you send a LOW signal to the enable 1 pin, motor A turns off;

- If you send a PWM signal, you can control the speed of the motor. The motor speed is proportional to the duty cycle. However, note that for small duty cycles, the motors might not spin, and make a continuous buzz sound.

SIGNAL ON THE ENABLE PIN	MOTOR STATE
HIGH	Motor enabled
LOW	Motor not enabled
PWM	Motor enabled: speed proportional to duty cycle

Input pins

The input pins control the direction the motors are spinning. Input 1 and input 2 control motor A, and input 3 and 4 control motor B.

- If you apply LOW to input1 and HIGH to input 2, the motor will spin forward;
- If you apply power the other way around: HIGH to input 1 and LOW to input 2, the motor will rotate backwards. Motor B can be controlled using the same method but applying HIGH or LOW to input 3 and input 4.

Controlling 2 DC Motors – ideal to build a robot

If you want to build a robot car using 2 DC motors, these should be rotating in specific directions to make the robot go left, right, forward or backwards.

For example, if you want your robot to move forward, both motors should be rotating forward. To make it go backwards, both should be rotating backwards.

To turn the robot in one direction, you need to spin the opposite motor faster. For example, to make the robot turn right, enable the motor at the left, and disable the motor at the right. The following table shows the input pins' state combinations for the robot directions.

DIRECTION	INPUT 1	INPUT 2	INPUT 3	INPUT 4
Forward	0	1	0	1
Backward	1	0	1	0
Right	0	1	0	0
Left	0	0	0	1
Stop	0	0	0	0

Connect the test circuit to sound the Buzzer

Objective:

- to connect the test circuit to sound the Buzzer.
-

Requirement:

- Breadboard (optional)
- Relay
- Connecting wire
- Battery (according to the rating of relay)
- Battery caps

PROCEDURE:**Task 1: Connecting a Buzzer to test circuit**

If a buzzer operates from a low enough voltage and draws low enough current, it can be interfaced directly to an Arduino Uno pin.

The buzzer used in this example can operate from a voltage between 3 to 28V and draws only 4mA of current at 12V. When the current drawn by the buzzer was measured at 5V, it was found that it only drew about 1.1mA which is well within the drive capability of an Arduino Uno pin.

See the [Arduino buzzer circuit](#) if you want to connect a buzzer to the Arduino that operates from a different voltage to the Arduino or if you are using a buzzer that draws more current than an Arduino pin can deliver.

Circuit Diagram

In this circuit, the positive lead of the buzzer is connected to pin 13 of the Arduino, the negative lead of the buzzer is connected to GND.

Arduino Sketch

The Blink Arduino example program can be used to test the circuit. This is a program that is built into the Arduino IDE and can be found under File 'Examples' '1.Basics' 'Blink'

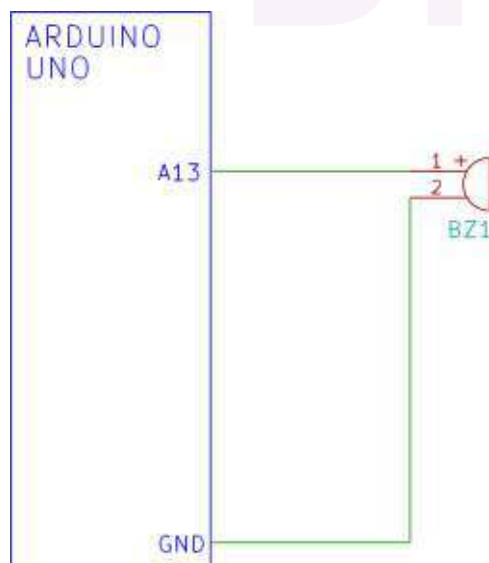
The Blink sketch will switch the buzzer on and off continuously.

Can't see the video? [View on YouTube](#) '!

Buzzer Details

The buzzer used in this article has the part number KPEG260 and the RS Components stock number: 535-8275, the brand is Kingstate.

This is a piezo buzzer that has built in circuitry that produces the audible buzzer tone. A plain piezo disk will not work in this circuit as it does not have any circuitry to drive it, although an Arduino sketch could be written to drive a piezo disk that is connected to an Arduino pin.



Smart City - Identify and test various parts of embedded system**Connect and test the motion sensor along with light/ Buzzer/ Streetlight and also timer and area of motion detection****Objective:**

- to connect and test the motion sensor along with light/ Buzzer/ Streetlight and also timer and area of motion detection.

Requirements:

- Arduino UNO
- PIR Sensor
- 5V Buzzer
- Breadboard
- Connecting Wires
- Power Supply
- Circuit Design

PROCEDURE:**Introduction:**

we will learn about PIR Sensor and how can it be used as a Motion Sensor through the Arduino PIR Sensor Tutorial. By going through this project, you can understand how PIR Sensor works and how to hook up a PIR Sensor to Arduino.

We have made a project using Arduino, PIR Sensor and GSM Module called GSM based Home Security System using Arduino. If you understand how a PIR Sensor works, then you can many such interesting projects and even more complex ones.

A PIR Sensor or a Passive Infrared Sensor is an electronic device that measures the infrared (IR) light emitted by the objects in its observable area. The term 'Passive' in the PIR Sensor indicates that the sensor actually doesn't emit any infrared light but rather passively detects it that is emitted by its surrounding objects.

Every object, with its surface temperature greater than absolute zero i.e. -2730°C emits heat in the form of infrared radiation. Humans cannot see this radiation as the radiations are in infrared wavelength.

But PIR Sensors detect these radiations and change them into appropriate electrical signals.

PIR Sensor

A typical PIR Sensor looks like the one shown in the image below. To connect with external devices, it has only three pins namely VCC, Digital OUT (Data) and GND.



On the top of sensor board, there is a special type of lens called Fresnel Lens that is covering up the actual Pyroelectric Sensor. The job of the Fresnel Lens is to focus all the infrared radiation onto the pyroelectric sensor.



If you look at the back of the PIR Sensor board, the whole circuitry is housed there. The brain of the PIR Sensor Module is the BISS0001 PIR Motion Detector IC. Near this IC, we have two potentiometers, one for adjusting the Sensitivity and the other is for adjusting the delay time.

Using Sensitivity Adjust, you can control the range of field of view and in our sensor, it is up to 7 meters. Using the Delay Time Adjust, you can control the duration for which the Digital Out will stay HIGH when a moving object is detected.

How PIR Sensor works?

PIR Sensors are complicated than most other sensors. PIR Motion Sensor may seem simple when implemented as all you need to do is check for a HIGH signal on the Digital Out Pin of the Sensor whenever motion is detected.

But, internally, there is a lot going on and the input and output of the sensor are dependent on several variables.

The actual PIR Sensor i.e. the one which is covered with a lens, consists of two slots and both these slots are made up of IR Sensitive materials. Under normal condition where there is no movement in front of the sensor, both the slots in the Sensor detect same amount of infrared radiation.

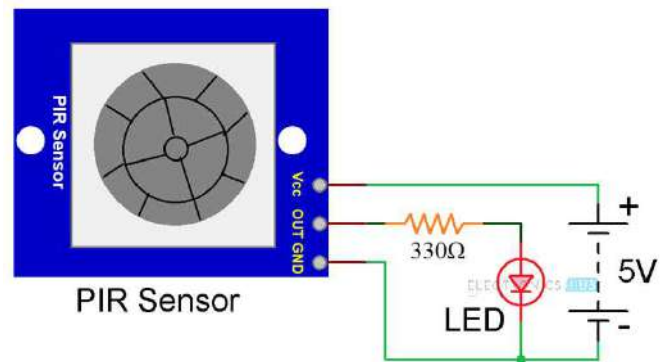
When there is movement in front of the sensor, like a human or a cat, their radiation is interpreted by one of the slots first and the differential output between the two slots becomes positive.

As the person moves away, the second slot detects the radiation and the differential output will become negative. Based these output pulses, a motion is detected.

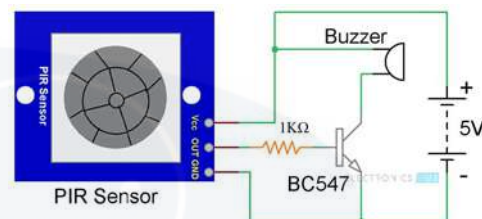
Testing the PIR Sensor

Since the Digital Out Pin of the PIR Sensor is either HIGH or LOW based on the movement detected, you can build a simple circuit to test the PIR Sensor.

The first circuit consists of a PIR Sensor and an LED. When the PIR Sensor detects motions, the LED turns ON. The duration for which the LED is ON can be adjusted with the help of Delay Adjust POT.



A similar PIR Sensor testing circuit is shown below but it consists of a buzzer. In order to drive the buzzer, an NPN Transistor like BC547 or 2N2222 can be used. The buzzer will be activated when the sensor detects any movement.



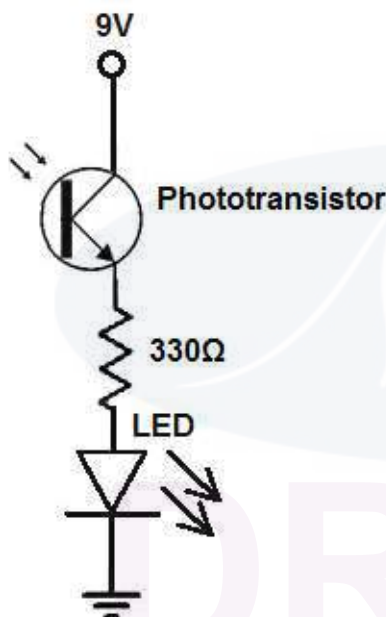
Set up a test circuit to test IR sensor

Objective:

- to set up a test circuit to test IR sensor.

Requirements:

- Arduino UNO
- IR LED
- Breadboard
- Jumper Wires

PROCEDURE:**IR Detector Circuit**

This circuit is very simple.

When the IR phototransistor isn't exposed to any infrared light, there can be no current flow through the transistor, because infrared light is what produces base current in the transistor. The base current triggers a larger current from emitter to collector. Without infrared, there is no base current. Therefore, no amplified current can be produced.

When the transistor is exposed to infrared light, a base current is produced. This base current triggers a larger current from emitter to collector. If you expose the IR phototransistor to an infrared light source such as an infrared LED, a TV remote control, a flame, or sunlight, the LED will light up. This is because the infrared induces a base current in the transistor, which causes sufficient current amplification to drive and light the LED. Without the infrared light, no base current will be produced and, thus, no amplification will occur. The infrared light is what triggers the whole transistor to allow current flow.

And this is how the IR detector circuit works.

Testing the Circuit

Again, when testing this circuit, the sunlight will make this circuit work, since the sun emits infrared light, as do flames as well. Probably the best way to produce the infrared light, though, which you will need to test this circuit is an infrared LED.

An infrared LED is an LED which emits infrared light. We cannot see infrared light, because it's outside of the visible spectrum, which is why we build circuits to be able to detect it. Infrared LEDs can be found at many online retailers. The LED is usually a dark purple or blue color. Like other LEDs, the cathode lead is shorter than the anode LED, so it connects the same way in a circuit as you would a regular LED.

To produce infrared LED from an IR LED, you just give the LED sufficient voltage and it will emit infrared light.

Just place the DC power which the LED needs for power. Check the datasheet of the IR LED in use to find out how much voltage it needs. It will be called the forward voltage on the datasheet. Also check the maximum forward current listed on the datasheet and use a resistor to make sure the LED does not receive more than this specification. Once the LED is supplied with the sufficient current, it will emit infrared light. With it emitting IR light, now just point the IR LED to our original Infrared (IR) detector circuit. This will trigger the LED to light in the circuit. And, thus, our test will be complete.

Smart City - Identify and test various parts of embedded system

Set up test circuit to test rain sensor

Objective:

- to set up test circuit to test rain sensor.
-

Requirements:

- Arduino UNO
- Bread board
- Jumper wires
- Potentiometer
- Control LED

PROCEDURE:

This is a simple rain sensor circuit which can be built by a school grade student very easily and can be used for displaying its relatively useful feature, probably among his friends or in a science fair exhibition.

Using IC 555 as the Comparator

The circuit is basically rigged using IC 555 as a comparator and is typically configured to sense the low resistance through water across its relevant inputs.

When water (due to rain fall) comes across the above inputs, a low resistance is developed here. The preset P1 is suitably adjusted such that any type of water across the sensing inputs triggers the IC appropriately.

The sudden low resistance at pin #2 of the IC acts like a pulse which exceeds the potential at pin #2 more than 1/3 of the supply voltage.

This activation instantly makes the output of the IC go low, ringing the connected buzzer. The buzzer circuit is comprehensively explained here, if you wanted to build one.

As long as the sensing input stays immersed under water, the output continues with the above situation.

However the moment, water is removed from the specified input terminals, the potential at pin #2 reverts to less than 1/3 of the supply voltage, making the output go high, back to its original position, switching off the buzzer.

The above operation effectively indicates the commencement of a rain fall when the sensor is appropriately placed for the detection.

The charge inside the capacitor C1 keeps the buzzer ringing for some period of time even after the water from the sensing inputs is completely removed.

Therefore the value of C1 must be appropriately chosen, or may be completely eliminated if the feature is not required.

Making the Sensor Unit

The explained rain sensor circuit obviously needs to be placed indoors, therefore only the sensor terminals are required to be positioned outdoors through long connecting flexible wires.

The figure shows a simple way of making the sensor unit.

A small plastic of around 2 by 2 inches is used and a couple metal screws are fixed over the plate. The distance between the screws should be such that no residual water is able to stick or clog between them and water formation across it is detected only as long as the rain fall persists.

The wires from the screws should be carefully terminated to the relevant points on the circuit. The circuit must be housed inside a suitable plastic enclosure along with the buzzer and the battery.

Parts List

R1 = 1M, R2 = 100K,

P1 = 1M pre-set, can be replaced with a 1M fixed resistor

IC = 555, C1 = 10uF/25V,

Simple Rain Sensor Circuit using a Single Transistor

If you think the above circuit a bit over complex, then perhaps you could implement the design using a single transistor and a resistor, as shown in the following image:

The working of the above circuit is rather simple. When water droplets or rain droplets fall on the sensor device, made using screw heads, the water bridges across the screw heads allowing small electrical current to pass across the metal, triggering the base of the transistor. As soon as this happens, the transistor begins conducting and amplifies the conduction across its collector/emitter terminals.

This results in the switching ON of the connected buzzer which now begins buzzing or beeping indicating the commencement of rain outside, and warning the user regarding the same.

Set up test circuit to test ultrasonic sensors

Objective:

- to set up test circuit to test ultrasonic sensors.

Requirement:

Hardware:

- 1 x Arduino (I used UNO R3 but any model works)
- 1 x Ultrasonic sensor.
- A few female jumper cables.
- A few male jumper cables.
- 1 x A/B type USB cable (Usually included with Arduino)

Software:

- Arduino IDE

PROCEDURE:

How Does an Ultrasonic Sensor Work?

An ultrasonic sensor is a device that uses ultrasonic waves to measure an object's distance. Ultrasonic transducers — which are the microphone and speaker tandems — send and receive ultra-high frequency sound waves to obtain an object's distance or proximity. The ultra-high frequency sound waves are reflected from an object's surface creating a unique echo pattern.

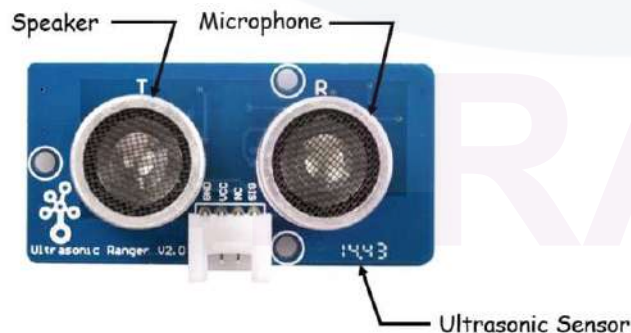


Figure 1. An HC-SR04 ultrasonic sensor.

Figure 2 illustrates the ultrasonic sensor's ultra-high frequency sound waves being reflected from an object's surface.

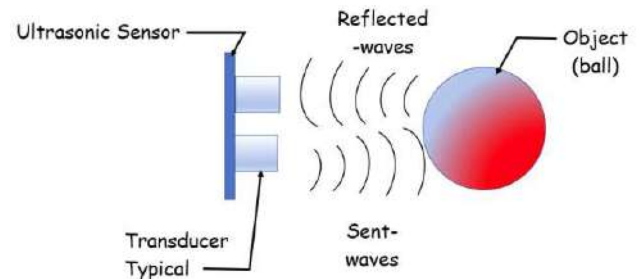


Figure 2. Sent and reflected waves: ultrasonic sensor basic operation.

Wiring an Ultrasonic Sensor to an Arduino Using TinkerCad Circuits

With your basic understanding of how an ultrasonic sensor works, you are now ready to wire the device to an Arduino. To explore the operation of the ultrasonic sensor, you can build a virtual functional circuit using TinkerCad Circuits.

TinkerCad Circuits is a free online circuit simulator that allows a variety of electrical and electronic circuits to be simulated prior to wiring them on a real breadboard. You can even test Arduino projects (including the code) with TinkerCad Circuits. You can gain valuable electronics knowledge through experimentation prior to committing to building your physical circuit.

Figure 3 shows a functional ultrasonic sensor Arduino project built with TinkerCad Circuits.

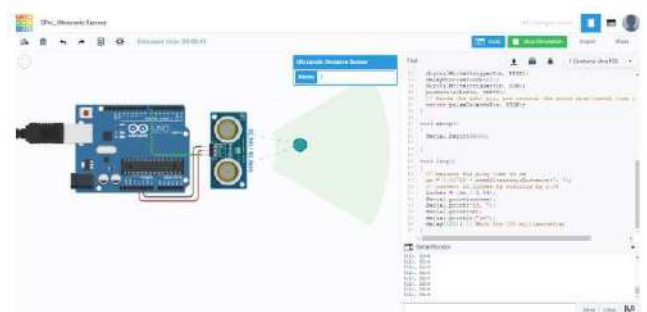


Figure 3. An online ultrasonic sensor and Arduino TinkerCard Circuit.

Use Figure 4 as a reference if you have a breadboard to experiment with the ultrasonic sensor.

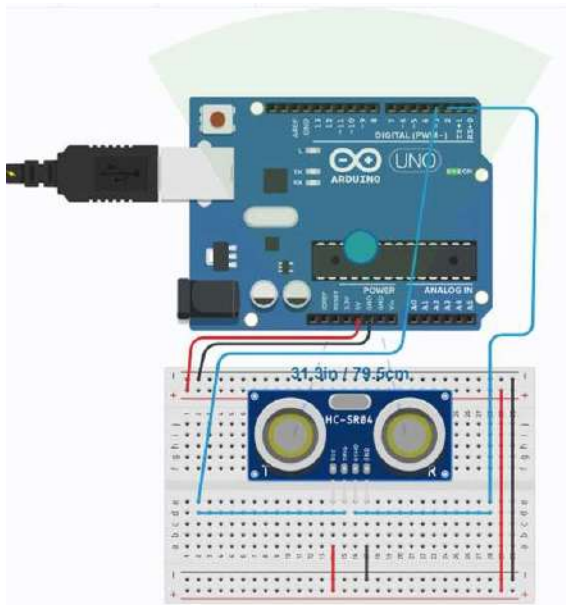


Figure 4. A breadboard wiring version built in TinkerCad Circuits.

Wiring an Ultrasonic Sensor With an Arduino on a Breadboard

You can use the ultrasonic sensor Arduino circuit built in TinkerCad Circuits or the electrical wiring diagram shown in Figure 5 to build your sensing device.

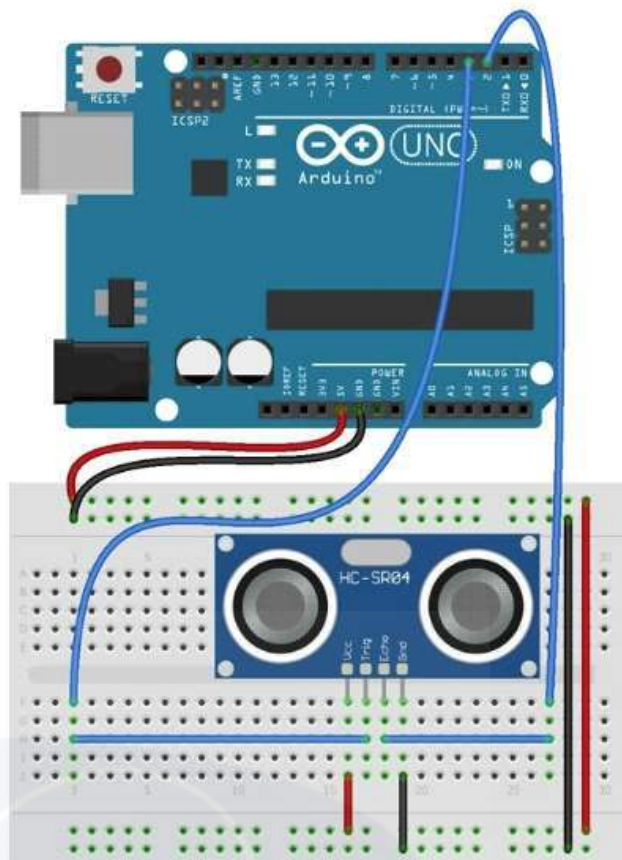


Figure 5. The electrical wiring diagram for an actual ultrasonic sensor to an Arduino.

If you are using a 4-pin ultrasonic sensor, the normally closed pin (NC) is wired to ground. You can place the ultrasonic sensor as shown on the breadboard and with jumper wires complete the wiring to the Arduino.

Here's the circuit I built using a 4-wire jumper harness to wire the ultrasonic sensor to the Arduino.

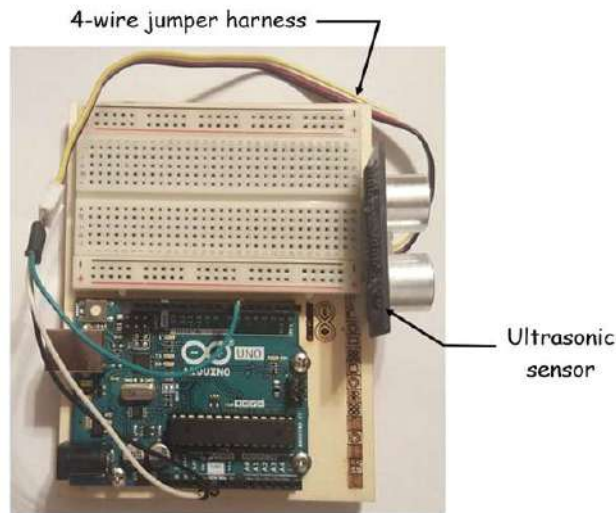


Figure 7. The 4-wire color-coded jumper wiring connections.

You have now successfully wired an ultrasonic sensor to an Arduino! You are now ready to install (upload) the ultrasonic sensor code to the Arduino.

The Ultrasonic Sensor Code

The last part of the project is uploading the ultrasonic sensor code to the Arduino. Connect the Arduino to your desktop PC or laptop computer using a USB cable. In Arduino IDE, enter the code shown below. You can also download the code to your desktop PC or laptop computer's hard-drive. In the IDE, you can upload the code by clicking the horizontal arrow.

This sketch reads a PING))) ultrasonic rangefinder and returns the distance to the closest object in range. To do this, it sends a pulse to the sensor to initiate a reading, then listens for a pulse to return. The length of the returning pulse is proportional to the distance of the object from the sensor.

Figure 6. The author's actual circuit.

The 4-wire jumper harness is color coded. Figure 7 shows the wiring connections between the Arduino and the ultrasonic sensor.

Arduino	Ultrasonic Ranger
5V	Red
GND	Black
Not Conencted	White
D7	Yellow

Smart City - Identify and select various types of sensors used in smartcity

Identify various sensors used in Smart city

Objective:

- to identify various sensors used in Smart city.

Requirements:

- Arduino UNO
- Temperature Sensor
- Proximity Sensor
- Accelerometer
- IR Sensor (Infrared Sensor)
- Pressure Sensor
- Light Sensor
- Ultrasonic Sensor
- Smoke, Gas and Alcohol Sensor
- Touch Sensor
- Color Sensor
- Humidity Sensor
- Tilt Sensor
- Flow and Level Sensor

PROCEDURE:

Sensors are everywhere. They're in our homes and workplaces, our shopping centers and hospitals. They're embedded in smart phones and an integral part of the Internet of Things (IoT). Sensors have been around for a long time. The first thermostat was introduced in the late 1880s and infrared sensors have been around since the late 1940s. The IoT and its counterpart, the Industrial Internet of Things (HYPERLINK "https://en.wikipedia.org/wiki/Industrial_Internet_of_Things"IIoTHYPERLINK "https://en.wikipedia.org/wiki/Industrial_Internet_of_Things"), are bringing sensor usage to a new level.

Broadly speaking, sensors are devices that detect and respond to changes in an environment. Inputs can come from a variety of sources such as light, temperature, motion and pressure. Sensors output valuable information and if they are connected to a network, they can share data with other connected devices and management systems.

Sensors are crucial to the operation of many of today's businesses. They can warn you of potential problems before they become big problems, allowing businesses to perform predictive maintenance and avoid costly downtime. The data from sensors can also be analyzed for trends allowing business owners to gain insight into crucial trends and make informed evidence-based decisions.

Sensors come in many shapes and sizes. Some are purpose-built containing many built-in individual sensors, allowing you to monitor and measure many sources of data. In brownfield environments, it's key for sensors to include digital and analog inputs so that they can read data from legacy sensors.

There are many types of IoT sensors and an even greater number of applications and use cases. Here are 10 of the more popular types of IoT sensors and some of their use cases.

1. Temperature Sensors

Temperature sensors measure the amount of heat energy in a source, allowing them to detect temperature changes and convert these changes to data. Machinery used in manufacturing often requires environmental and device temperatures to be at specific levels. Similarly, within agriculture, soil temperature is a key factor for crop growth.

2. Humidity Sensors

These types of sensors measure the amount of water vapor in the atmosphere of air or other gases. Humidity sensors are commonly found in heating, vents and air conditioning (HVAC) systems in both industrial and residential domains. They can be found in many other areas including hospitals, and meteorology stations to report and predict weather.

3. Pressure Sensors

A pressure sensor senses changes in gases and liquids. When the pressure changes, the sensor detects these changes, and communicates them to connected systems. Common use cases include leak testing which can be a result of decay. Pressure sensors are also useful in the manufacturing of water systems as it is easy to detect fluctuations or drops in pressure.

4. Proximity Sensors

Proximity sensors are used for non-contact detection of objects near the sensor. These types of sensors often emit electromagnetic fields or beams of radiation such as infrared. Proximity sensors have some interesting use cases. In retail, a proximity sensor can detect the motion between a customer and a product in which he or she is interested. The user can be notified of any discounts or special offers of products located near the sensor. Proximity sensors are also used in the parking lots of malls, stadiums and airports to indicate parking availability. They can also be used on the assembly lines of chemical, food and many other types of industries.

5. Level Sensors

Level sensors are used to detect the level of substances including liquids, powders and granular materials. Many industries including oil manufacturing, water treatment and beverage and food manufacturing factories use level sensors. Waste management systems provide a common use case as level sensors can detect the level of waste in a garbage can or dumpster.

6. Accelerometers

Accelerometers detect an object's acceleration i.e. the rate of change of the object's velocity with respect to time. Accelerometers can also detect changes to gravity. Use cases for accelerometers include smart pedometers and monitoring driving fleets. They can also be used as anti-theft protection alerting the system if an object that should be stationary is moved.

7. Gyroscope

Gyroscope sensors measure the angular rate or velocity, often defined as a measurement of speed and rotation around an axis. Use cases include automotive, such as car navigation and electronic stability control (anti-skid) systems. Additional use cases include motion sensing for video games, and camera-shake detection systems.

8. Gas Sensors

These types of sensors monitor and detect changes in air quality, including the presence of toxic, combustible or hazardous gasses. Industries using gas sensors

include mining, oil and gas, chemical research and manufacturing. A common consumer use case is the familiar carbon dioxide detectors used in many homes.

9. Infrared Sensors

These types of sensors sense characteristics in their surroundings by either emitting or detecting infrared radiation. They can also measure the heat emitted by objects. Infrared sensors are used in a variety of different IoT projects including healthcare as they simplify the monitoring of blood flow and blood pressure. Televisions use infrared sensors to interpret the signals sent from a remote control. Another interesting application is that of art historians using infrared sensors to see hidden layers in paintings to help determine whether a work of art is original or fake or has been altered by a restoration process.

10. Optical Sensors

Optical sensors convert rays of light into electrical signals. There are many applications and use cases for optical sensors. In the auto industry, vehicles use optical sensors to recognize signs, obstacles, and other things that a driver would notice when driving or parking. Optical sensors play a big role in the development of driverless cars. Optical sensors are very common in smart phones. For example, ambient light sensors can extend battery life. Optical sensors are also used in the biomedical field including breath analysis and heart-rate monitors.

Vimmi
DRAFT

Identify Roles and characteristics of various sensors

Objective:

- to identify roles and characteristics of various sensors.

Requirements:

- Temperature sensor
- Humidity sensor
- Gas sensor
- Motion sensor
- Smoke sensor
- Pressure sensor
- Image sensor
- IR sensor
- Accelerometer sensor

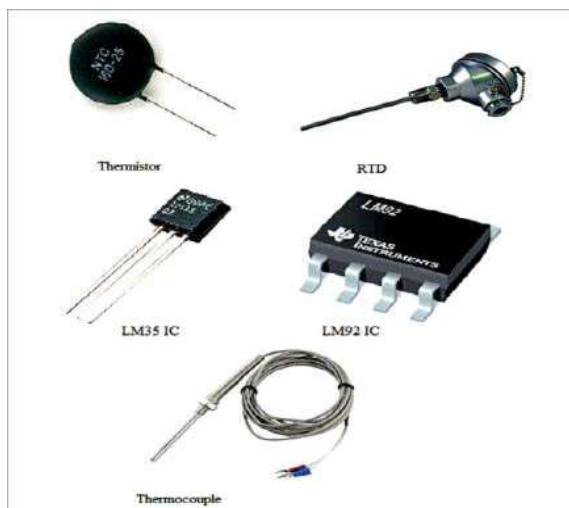
PROCEDURE:

Temperature sensors

These devices measure the amount of heat energy generated from an object or surrounding area. They find application in air-conditioners, refrigerators and similar devices used for environmental control. They are also used in manufacturing processes, agriculture and health industry.

Temperature sensors can be used almost in every IoT environment, from manufacturing to agriculture. In manufacturing, sensors are used to monitor the temperature of machines. In agriculture, these can be used to monitor the temperature of soil, water and plants.

Temperature sensors include thermocouples, thermistors, resistor temperature detectors (RTDs) and integrated circuits (ICs).



Humidity sensors

The amount of water vapour in air, or humidity, can affect human comfort as well as many manufacturing processes in industries.

So monitoring humidity level is important.

Most commonly used units for humidity measurement are relative humidity (RH), dew/frost point (D/F PT) and parts per million (PPM).



Motion sensors

Motion sensors are not only used for security purposes but also in automatic door controls, automatic parking systems, automated sinks, automated toilet flushers, hand dryers, energy management systems, etc.

You use these sensors in the IoT and monitor them from your smartphone or computer.

HC-SR501 passive infrared (PIR) sensor is a popular motion sensor for hobby projects.



Gas sensors

These sensors are used to detect toxic gases. The sensing technologies most commonly used are electrochemical, photo-ionisation and semiconductor. With technical advancements and new specifications, there are a multitude of gas sensors available to help extend the wired and wireless connectivity deployed in IoT applications.



Smoke sensors

Smoke detectors have been in use in homes and industries for quite a long time. With the advent of the IoT, their application has become more convenient and user-friendly. Furthermore, adding a wireless connection to smoke detectors enables additional features that increase safety and convenience.



Pressure sensors

These sensors are used in IoT systems to monitor systems and devices that are driven by pressure signals. When the pressure range is beyond the threshold level, the device alerts the user about the problems that should be fixed. For example, BMP180 is a popular digital pressure sensor for use in mobile phones, PDAs, GPS navigation devices and outdoor equipment. Pressure sensors are also used in smart vehicles and aircrafts to

determine force and altitude, respectively. In vehicle, tyre pressure monitoring system (TPMS) is used to alert the driver when tyre pressure is too low and could create unsafe driving conditions.

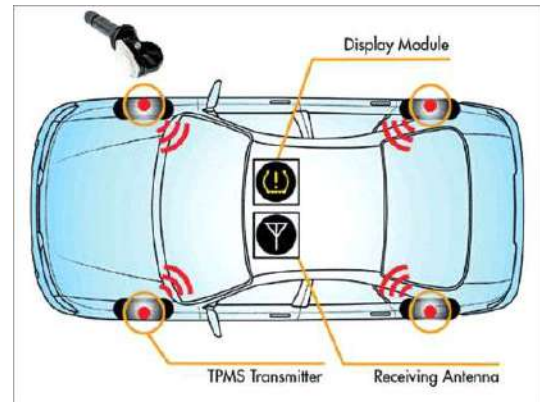
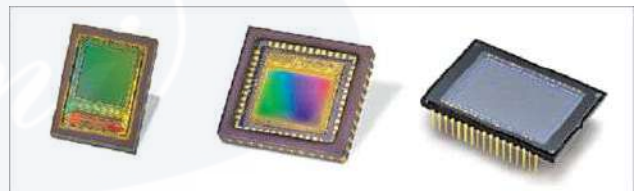


Image sensors

These sensors are found in digital cameras, medical imaging systems, night-vision equipment, thermal imaging devices, radars, sonars, media house and biometric systems. In the retail industry, these sensors are used to monitor customers visiting the store through IoT network. In offices and corporate buildings, they are used to monitor employees and various activities through IoT networks.



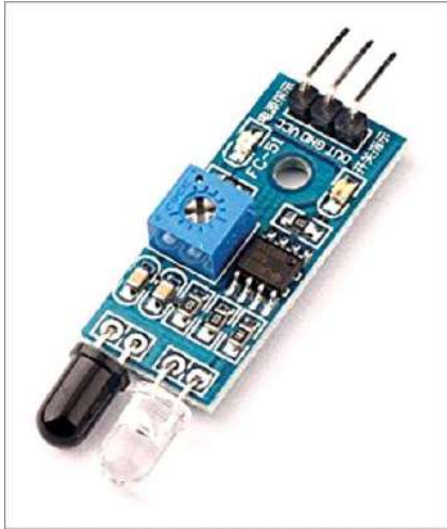
Accelerometer sensors

These sensors are used in smartphones, vehicles, aircrafts and other applications to detect orientation of an object, shake, tap, tilt, motion, positioning, shock or vibration. Different types of accelerometers include Hall-effect accelerometers, capacitive accelerometers and piezoelectric accelerometers.



IR sensors

These sensors can measure the heat emitted by objects. They are used in various IoT projects including healthcare to monitor blood flow and blood pressure, smartphones to use as remote control and other functions, wearable devices to detect amount of light, thermometers to monitor temperature and blind-spot detection in vehicles.



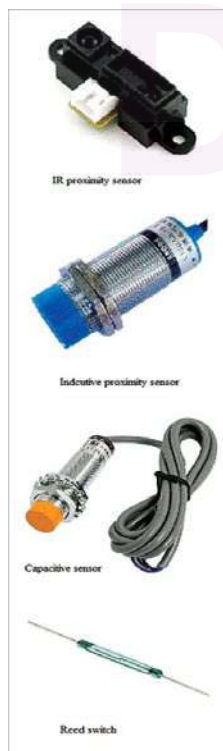
Ultrasonic Sensor

A transducer that works on the principle similar to the sonar or radar and estimate attributes of the target by interpreting is called as ultrasonic sensors or transceivers. There are different types of sensors that are classified as active and passive ultrasonic sensors that can be differentiated based on the working of sensors.



Proximity sensors

These sensors detect the presence or absence of a nearby object without any physical contact. Different types of proximity sensors are inductive, capacitive, photoelectric, ultrasonic and magnetic. These are mostly used in object counters, process monitoring and control.



Smart City - Identify and select various types of sensors used in smartcity

Select appropriate sensor as per requirement

Objective:

- to select appropriate sensor as per requirement.

Requirement:

- IOT sensors
- Arduino UNO

PROCEDURE:

IoT platforms function and deliver various kinds of intelligence and data using a variety of sensors. They serve to collect data, pushing it and sharing it with a whole network of connected devices. All this collected data makes it possible for devices to autonomously function, and the whole ecosystem is becoming “smarter” every day. By combining a set of sensors and a communication network, devices share information with one another and are improving their effectiveness and functionality.

Take Tesla vehicles as an example. All of the sensors on a car record their perception of the surroundings, uploading the information into a massive database. The data is then processed and all the important new pieces of information are sent to all other vehicles. This is an ongoing process, through which a whole fleet of Tesla vehicles is becoming smarter every day.

Let's take a look at some of the key sensors, extensively being used in the IoT world.

Temperature sensors

By definition, “A device, used to measure amount of heat energy that allows to detect a physical change in temperature from a particular source and converts the data for a device or user, is known as a Temperature Sensor.”

These sensors have been deployed for a long time in a variety of devices. However, with the emergence of IoT, they have found more room to be present in an even greater number of devices.

Only a couple of years ago, their uses mostly included A/C control, refrigerators and similar devices used for environmental control. However, with the advent of the IoT world, they have found their role in manufacturing processes, agriculture and health industry.

In the manufacturing process, many machines require specific environment temperature, as well as device temperature. With this kind of measurement, the manufacturing process can always remain optimal.

On the other hand, in agriculture, the temperature of soil is crucial for crop growth. This helps with the production of plants, maximizing the output.

Followed are some sub-categories of Temp Sensors:

- **Thermocouples:** These are voltage devices that

indicate temperature measuring with a change in voltage. As temperature goes up, the output voltage of the thermocouple rises.

- **Resistor temperature detectors (RTD):** The resistance of the device is directly proportional to the temperature, increase in a positive direction when the temperature rises resistance going up.
- **Thermistors:** It is a temperature sensitive resistor that changes its physical resistance with the change in temperature.
- **IC (Semiconductor):** They are linear devices where the conductivity of the semiconductor increases linearly and it takes advantage of the variable resistance properties of semiconductor materials. It can provide a direct temperature reading in digital form, especially at low temperatures.
- **Infrared sensors:** It detects temperature by intercepting a portion of emitted infrared energy of the object or substance, and sensing its intensity, can be used to measure temperature of solids and liquids only, not possible to use it on gases because of their transparent nature.

Proximity sensor

A device that detects the presence or absence of a nearby object, or properties of that object, and converts it into signal which can be easily read by user or a simple electronic instrument without getting in contact with them.

Proximity sensors are largely used in the retail industry, as they can detect motion and the correlation between the customer and product they might be interested in. A user is immediately notified of discounts and special offers of nearby products.

Another big and quite an old use-case is vehicles. You are reversing your car and are alarmed about an obstacle while taking reverse, that's the work of proximity sensor.

They are also used for parking availability in places such as malls, stadiums or airports.

Following are some of the Proximity Sensors sub-categorised:

- **Inductive Sensors:** Inductive proximity sensors are used for non-contact detection to find out the presence of metallic objects using electromagnetic field or a beam of electromagnetic radiation. It can operate at higher speeds than mechanical switches and also seems more reliable because of its robustness.
- **Capacitive Sensors:** Capacitive proximity sensors

can detect both metallic as well as non-metallic targets. Nearly all other materials are dielectric different from air. It can be used to sense very small objects through a large portion of target. So, generally used in difficult and complicated applications.

- **Photoelectric Sensors:** Photoelectric sensor is made up of light-sensitive parts and uses a beam of light to detect the presence or absence of an object. It is an ideal alternative of inductive sensors. And used for long distance sensing or to sense non-metal object.
- **Ultrasonic Sensors:** Ultrasonic sensors are also used to detect the presence or to measure the distance of targets similar to radar or sonar. This makes a reliable solution for harsh and demanding conditions.

Pressure sensor

A pressure sensor is a device that senses pressure and converts it into an electric signal. Here, the amount depends upon the level of pressure applied.

There are plenty of devices that rely on liquid or other forms of pressure. These sensors make it possible to create IoT systems that monitor systems and devices that are pressure propelled. With any deviation from standard pressure range, the device notifies the system administrator about any problems that should be fixed.

Deployment of these sensors is not only very useful in manufacturing, but also in the maintenance of whole water systems and heating systems, as it is easy to detect any fluctuation or drops in pressure.

Water quality sensor

Water quality sensors are used to detect the water quality and Ion monitoring primarily in water distribution systems.

Water is practically used everywhere. These sensors play an important role as they monitor the quality of water for different purposes. They are used in a variety of industries.

Following is a list of the most common kind of water sensors in use:

- **Chlorine Residual Sensor:** It measures chlorine residual (i.e. free chlorine, monochloramine & total chlorine) in water and most widely used as disinfectant because of its efficiency.
- **Total Organic Carbon Sensor:** TOC sensor is used to measure organic element in water.
- **Turbidity Sensor:** Turbidity sensors measure suspended solids in water, typically it is used in river and stream gaging, wastewater and effluent measurement.
- **Conductivity Sensor:** Conductivity measurements are carried out in industrial processes primarily to obtain information on total ionic concentrations (i.e. dissolved compounds) in water solutions.

- **pH Sensor:** It is used to measure the pH level in the dissolved water, which indicates how acidic or basic (alkaline) it is.
- **Oxygen-Reduction Potential Sensor:** The ORP measurement provides insights into the level of oxidation/reduction reactions occurring in the solution.

Chemical sensor

Chemical sensors are applied in a number of different industries. Their goal is to indicate changes in liquid or to find out air chemical changes. They play an important role in bigger cities, where it is necessary to track changes and protect the population.

Main use cases of chemical sensors can be found in Industrial environmental monitoring and process control, intentionally or accidentally released harmful chemical detection, explosive and radioactive detection, recycling processes on Space Station, pharma industries and laboratory etc.

Following are most common kind of chemical sensors in use:

- Chemical field-effect transistor
- Chemiresistor
- Electrochemical gas sensor
- Fluorescent chloride sensor
- Hydrogen sulfide sensor
- Nondispersive infrared sensor
- pH glass electrode
- Potentiometric sensor
- Zinc oxide nanorod sensor

Gas sensor

Gas sensors are similar to the chemical ones, but are specifically used to monitor changes of the air quality and detect the presence of various gases. Like chemical sensors, they are used in numerous industries such as manufacturing, agriculture and health and used for air quality monitoring, detection of toxic or combustible gas, hazardous gas monitoring in coal mines, oil & gas industries, chemical laboratory research, manufacturing – paints, plastics, rubber, pharmaceutical & petrochemical etc.

Following are some common Gas sensors:

- Carbon dioxide sensor
- Breathalyzer
- Carbon monoxide detector
- Catalytic bead sensor
- Hydrogen sensor
- Air pollution sensor
- Nitrogen oxide sensor

- Oxygen sensor
- Ozone monitor
- Electrochemical gas sensor
- Gas detector
- Hygrometer

Smoke sensor

A smoke sensor is a device that senses smoke (airborne particulates & gases), and its level.

They have been in use for a long period of time. However, with the development of IoT, they are now even more effective, as they are plugged into a system that immediately notifies the user about any problem that occurs in different industries.

Smoke sensors are extensively used by manufacturing industry, HVAC, buildings and accommodation infra to detect fire and gas incidences. This serves to protect people working in dangerous environments, as the whole system is much more effective in comparison to the older ones.

Common Type of Smoke Sensors

Smoke sensors detect the presence of Smoke, Gases and Flame surrounding their field. It can be detected either optically or by the physical process or by the use of both the methods.

- Optical smoke sensor (Photoelectric): Optical smoke sensor used the light scatter principle trigger to occupants.
- Ionization smoke sensor: Ionization smoke sensor works on the principle of ionization, kind of chemistry to detect molecules causing a trigger alarm.

IR sensors

An infrared sensor is a sensor which is used to sense certain characteristics of its surroundings by either emitting or detecting infrared radiation. It is also capable of measuring the heat being emitted by the objects.

They are now used in a variety of IoT projects, especially in Healthcare as they make monitoring of blood flow and blood pressure simple. They are even used in a wide array of regular smart devices such as smartwatches and smartphones as well.

Other common use includes home appliances & remote control, breath analysis, Infrared vision (i.e. visualize heat leaks in electronics, monitor blood flow, art historians to see under layers of paint), wearable electronics, optical communication, non-contact based temperature measurements, automotive blind-angle detection.

Their usage does not end there, they are also a great tool for ensuring high-level security in your home. Also, their application includes environment checks, as they can detect a variety of chemicals and heat leaks. They are going to play an important role in the smart home industry, as they have a wide-range of applications.

Level sensors

A sensor which is used to determine the level or amount of fluids, liquids or other substances that flow in an open or closed system is called Level sensor.

Like IR sensors, level sensors are present in a wide array of industries. They are primarily known for measuring fuel levels, but they are also used in businesses that work with liquid materials. For example, the recycling industry, as well as the juice and alcohol industry rely on these sensors to measure the number of liquid assets in their possession.

Best use cases of level sensor is, fuel gauging & liquid levels in open or closed containers, sea level monitoring & Tsunami warning, water reservoirs, medical equipment, compressors, hydraulic reservoirs, machine tools, beverage and pharmaceutical processing, high or low-level detection etc.

This helps better streamline their businesses, as sensors collect all the important data at all times. With the use of these sensors, any product manager can precisely see how much liquid is ready to be distributed and whether the manufacturing should be stepped up.

There are two basic level measurement types:

Point level sensors: Point level sensors usually detect the particular specific level and respond to the user if the sensing object is above or below that level. It is integrated into single device to get an alarm or trigger

Continuous level Sensor: Continuous level sensors measure liquid or dry material levels within a specified range and provide outputs which continuously indicate the level. The best example of it is fuel level display in the vehicle.

Image sensors

Image sensors are instruments which are used to convert optical images into electronic signals for displaying or storing files electronically.

The major use of image sensor is found in digital camera & modules, medical imaging and night vision equipment, thermal imaging devices, radar, sonar, media house, Biometric & IRIS devices.

Two main types of sensors are used in:

- CCD (charge-coupled device), and
- CMOS (complementary metal-oxide semiconductor) imagers.

Although each type of sensor uses different technology to capture images, both CCD and CMOS imagers use metal-oxide semiconductors, having the same degree of sensitivity to light, and no inherent quality difference

An average consumer would think that this is a regular camera, but even though this is not far from the truth, image sensors are connected with a wide range of different devices, making their functionality much better.

One of the best-known uses includes the car industry, in which imagery plays a very important role. With these sensors, the system can recognize signs, obstacles and many other things that a driver would generally notice on the road. They play a very important role in IoT industry, as they directly affect the progress of driverless cars.

They are also implemented in improved security systems, where images help capture details about the perpetrator.

In the retail industry, these sensors serve to collect data about customers, helping businesses get a better insight into who is actually visiting their store, race, gender, age are only some of the useful parameters that retail owners get by using these IoT sensors.

Motion detection sensors

A motion detector is an electronic device which is used to detect the physical movement (motion) in a given area and it transforms motion into an electric signal; motion of any object or motion of human beings

Motion detection plays an important role in the security industry. Businesses utilize these sensors in areas where no movement should be detected at all times, and it is easy to notice anybody's presence with these sensors installed.

These are primarily used for intrusion detection systems, automatics door control, boom barrier, smart camera (i.e motion based capture/video recording), toll plaza, automatic parking systems, automated sinks/toilet flusher, hand dryers, energy management systems (i.e. Automated Lighting, AC, Fan, Appliances Control) etc.

On the other hand, these sensors can also decipher different types of movements, making them useful in some industries where a customer can communicate with the system by waving a hand or by performing a similar action. For example, someone can wave to a sensor in the retail store to request assistance with making the right purchase decision.

Even though their primary use is correlated with the security industry, as the technology advances, the number of possible applications of these sensors is only going to grow.

Following are key motion sensor types widely used:

- **Passive Infrared (PIR):** It Detects body heat (infrared energy) and the most widely used motion sensor in home security systems.
- **Ultrasonic:** Sends out pulses of ultrasonic waves and measures the reflection off a moving object by tracking the speed of sound waves.
- **Microwave:** Sends out radio wave pulses and measures the reflection off a moving object. They cover a larger area than infrared & ultrasonic sensors, but they are vulnerable to electrical interference and more expensive.

Accelerometer sensors

Accelerometer is a transducer that is used to measure the physical or measurable acceleration experienced by an object due to inertial forces and converts the mechanical motion into an electrical output. It is defined as rate of change of velocity with respect to time

These sensors are now present in millions of devices, such as smartphones. Their uses involve detection of vibrations, tilting and acceleration in general. This is great for monitoring your driving fleet, or using a smart pedometer.

In some instances, it is used as a form of anti-theft protection, as the sensor can send an alert through the system if an object that should remain stationary is moved.

They are widely used in cellular & media devices, vibration measurement, automotive control and detection, free fall detection, aircraft and aviation industries, movement detection, sports academy/athletes behavior monitoring, consumer electronics, industrial & construction sites etc.

There are various kinds of accelerometers and following are few mainly used in IoT projects:

- **Hall-effect accelerometers:** Hall-effect accelerometers are using Hall principle to measure the acceleration, it measures the voltage variations caused by changes in a magnetic field around them.
- **Capacitive accelerometers:** Capacitive accelerometers sensing output voltage depends on the distance between two planar surfaces. Capacitive accelerometers are also less prone to noise and variation with temperature.
- **Piezoelectric accelerometers:** Piezoelectric sensing principle is working on the piezoelectric effect. Piezo-film based accelerometers are best used to measure vibration, shock, and pressure.

Each accelerometer sensing technology has its own advantages and compromises. Before selecting, it's important to understand the basic differences of the various types and the test requirements.

Gyroscope sensors

A sensor or device which is used to measure the angular rate or angular velocity is known as Gyro sensors, Angular velocity is simply defined as a measurement of speed of rotation around an axis. It is a device used primarily for navigation and measurement of angular and rotational velocity in 3-axis directions. The most important application is monitoring the orientation of an object.

Their main applications are in car navigation systems, game controllers, cellular & camera devices, consumer electronics, robotics control, drone & RC control helicopter or UAV control, vehicle control/ADAS and many more.

There are several different kinds of gyro sensors which are selected by their working mechanism, output type, power, sensing range and environmental conditions.

- Rotary (classical) gyroscopes
- Vibrating Structure Gyroscope
- Optical Gyroscopes
- MEMS(micro-electro-mechanical systems) Gyroscopes

These sensors are always combined with accelerometers. The use of these two sensors simply provides more feedback to the system. With gyroscopic sensors installed, many devices can help athletes improve the efficiency of their movements, as they gain access to the athletes movement during sports activities.

This is only one example of its application, however, as the role of this sensor is to detect rotation or twist, its application is crucial for the automation of some manufacturing processes.

Humidity sensors

Humidity is defined as the amount of water vapour in an atmosphere of air or other gases. The most commonly used terms are "Relative Humidity (RH)

These sensors usually follow the use of temperature sensors, as many manufacturing processes require perfect working conditions. Through measuring humidity, you can ensure that the whole process runs smoothly, and when there is any sudden change, action can be taken immediately, as sensors detect the change almost instantaneously.

Their applications and use can be found in Industrial & residential domain for heating, ventilating, and air conditioning systems control. They can also be found in Automotive, museums, industrial spaces and greenhouses, meteorology stations, Paint and coatings industries, hospitals & pharma industries to protect medicines

Optical sensors

A sensor which measures the physical quantity of light rays and convert it into electrical signal which can be easily readable by user or an electronic instrument/device is called optical sensor.

Optical sensors are loved by IoT experts, as they are

practical for measuring different things simultaneously. The technology behind this sensor allows it to monitor electromagnetic energy, which includes, electricity, light and so on.

Due to this fact, these sensors have found use in healthcare, environment monitoring, energy, aerospace and many more industries. With their presence oil companies, pharmaceutical companies and mining companies are in a much better position to track environmental changes while keeping their employees safe.

Their main use can be found in ambient light detection, digital optical switches, optical fibres communications, due to electrical isolation best suited for oil and gas applications, civil and transportation fields, high speed network systems, elevator door control, assembly line part counters and safety systems.

Following are key type of optical sensors:

- **Photodetector:** It uses light sensitive semiconductor materials like photocells, photodiodes or phototransistors to work as photodetector
- **Fiber Optics:** Fibers optics carry no current, So its immune to electrical & electromagnetics interference and even in damaged condition no sparking or shock hazard happens.
- **Pyrometer:** It estimates the temperature of an object by sensing the color of the light and Objects radiate light according to their temperature and produce same colors at same temperature.
- **Proximity & Infrared:** Proximity use light to sense objects nearby and Infrared are used where visible light would be inconvenient.

It is clear that IoT has become incredibly popular, and current trends show that it is the future. It simply helps with automation of various processes, making these systems quite useful for both regular consumers and businesses.

We are yet to see the full potential this technology carries, as the whole platform is becoming smarter through the fusion of all the abovementioned sensors. When you consider the fact that all of the measured data is collected and can be analyzed, it is obvious that IoT is going to become even smarter in the future.

Determine Air quality and use noise pollution sensor

Objective:

to determine air quality and use noise pollution sensor.

Requirement

- Arduino UNO
- LCD
- Jumper Wires
- Breadboard

PROCEDURE:

The device is intended to provide the user with a cost-efficient means of determining air quality. Our sensor focuses on the five components of the Environmental Protection Agency's Air Quality Index: ozone, particulate matter, carbon monoxide, sulphur dioxide, and nitrous oxide. This device detects all of these pollutants except sulphur dioxide. The device also includes a town gas sensor to alert the user to gas leaks or the presence of flammable gases. Furthermore, a temperature and humidity sensor is included as these conditions can impact the performance of the gas sensors.

We have yet to fully calibrate our device, but we have extracted data from sensor data sheets to make some preliminary estimations. The sensors used are relatively inexpensive and vary greatly from component to component so they need to be calibrated with known concentrations of the target gasses. We have not yet had the opportunity do so.

Step 1: Materials

Control and Power

- Arduino Uno
- 5V power supply
- RGB 16x2 LCD shield

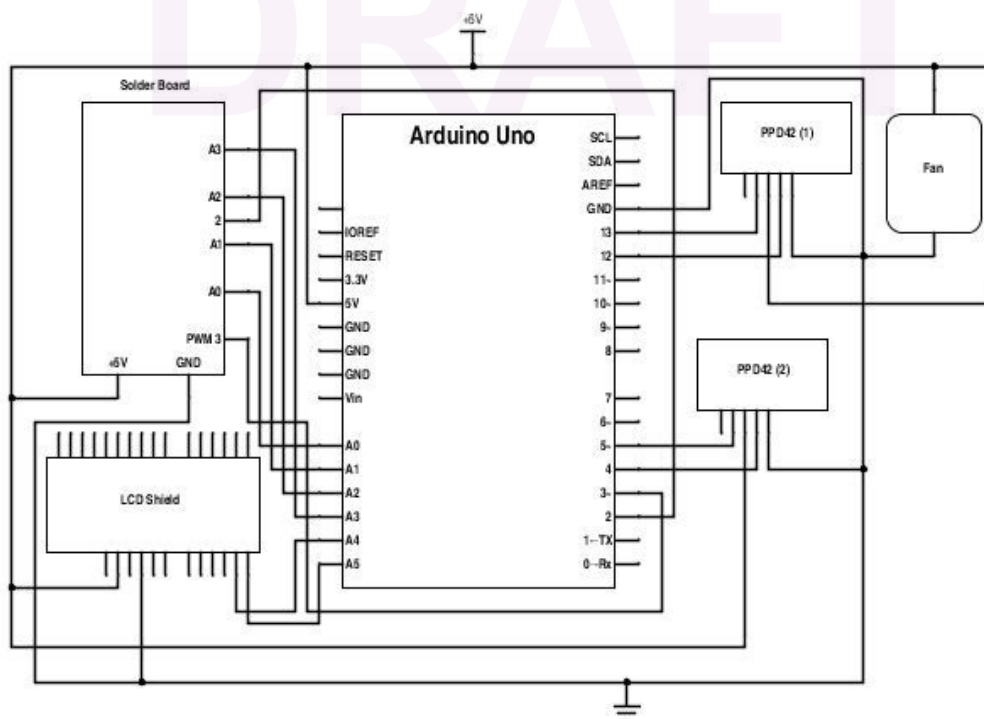
Sensors

- Shinyei PPD42 Particulate Matter Detector
- MQ-2 Gas Sensor
- MQ-9 Gas Sensor
- MiCS-2714 Gas Sensor (NO2)
- MiSC-2614 Gas Sensor (Ozone)
- Keyes DHT11 Temperature and Humidity Sensor

Box and Assembly

- Access to 3D printer
- Solder Board
- 5V fan
- 10 to 15 wires of gauge 24

Step 2: Overall Circuit Diagram



The circuit diagram above is the entirety of this pollution detector works. A detailed circuit diagram for the solder board follows later. Note that you can change most of the digital ports and analog ports that sensors enter should you need to do so (for any reason); this will only require that you edit the code we have provided to account for these changes.

Step 3: Particulate Matter Sensor

Each Shinyei Detector has two signal outputs: one for small particulate matter (left yellow wire in the picture above) and one for larger particulate matter. These output signals are connected the digital inputs of the Arduino. The detector needs to be powered by supplying +5V and ground to the ports on the detector. See the overall circuit diagram for details.

Each detector uses an infrared LED and a photodetector to measure scattering off of small airborne particulates. Internal circuitry turns the photodetector output into digital output signals. Generally the sensor outputs a +5V signal, when it detects particles is sends out a low-voltage pulse. The fraction of the time that the output signal is low or the "low-pulse occupancy percentage" is proportional to the concentration of particulate matter in the air.

Step 4: Gas Sensor Circuit Board

Above is the circuit diagram for the circuit board hosting the gas sensors and temperature/humidity sensor? Details about mounting each of the separate devices are in the following steps. Note that your circuit board can look different from ours physically. In fact, we recommend you print a circuit board for the surface mount devices instead of using a solder board. It should work just as well as long as you follow the circuit diagram.

Step 5: Ozone and NO2 Sensors

We use surface-mount sensors MiCS-2614 and MiCS-2714 to detect Ozone and Nitrogen Dioxide respectively.

Both of these devices use an internal resistor as their sensing element. The sensing resistor is connected between the pins (G) and (K) in the diagram above. Use an ohmmeter to check that you have found the right pins. The resistance should be on the order of 10-20 k Ω .

Both devices also have a heating element between pins (A) and (H). This heating element keeps the sensing element at the appropriate temperature. The resistance of the heating element is 50-60 Ω .

Ideally these devices should be surface mounted onto a circuit board. However, in the absence of a circuit board printer it is still possible to carefully solder to the back of these devices using very low temperature solder and much care.

As shown in the solder board circuit diagram, we place the 82 Ω resistor and the 131 Ω resistor in series with the heating elements of the MiCS-2614 and MiCS-2714 units respectively. This ensure that the heating elements receive the proper level of power. If you don't have access

to a 131 Ω resistor (it is not a standard value) use a 120 Ω resistor and a 12 Ω resistor in series.

We place the sensing resistors in both devices in series with 22k Ω resistors to create a voltage divider. From the voltage at the output of the voltage divider we can calculate the sensing resistance.

$$R_{\text{senor}} = 22\text{k}\Omega * (5V / V_{\text{out}} - 1)$$

Step 6: MQ Toxic Gas Sensors

We used MQ-2 and MQ-9 gas sensors to measure toxic gasses including Propane, Butane, LPG and Carbon Monoxide.

The MQ-2 and MQ-9 are very similar to the MiCS detectors. They use a gas-sensitive resistor (SnO₂) to detect concentrations of toxic gases and have an internal heating element to keep the sensor at the right temperature. The circuits we use for these devices are practically the same as the circuits for the MiCS sensors, except that we use a transistor rather than a resistor to regulate heater power in the MQ-9.

Refer to the solder board circuit diagram for mounting details. For the MQ-2 sensor, connect the pins marked A to the 5V power, connect the pin marked G to ground, and connect the pin marked S is connected to ground in series with a 47 k Ω resistor. For the MQ-9 gas sensor, connect the pin marked A to the transistor, the pin marked B to the 5 V power, the pin marked G to ground, and the pin marked S to ground in series with a 10 k Ω resistor.

Step 7: Temperature and Humidity Sensors

This sensor is provided because temperature and humidity play a role in the gas concentrations that our sensors detect. High humidity and temperature as well as dramatic changes in either would have detrimental effects on the accuracy of readings It is therefore helpful to be able to monitor these variables. Both temperature and humidity can be read from this single sensor. Oriented as it is in the photo above, the left pin is to be attached to power, the middle pin is the output signal, and the right pin is grounded. The output signal for this component goes to a digital port on the Arduino. Our code is set up such expecting the temperature signal in digital port 2. This can be changed to another digital port should you need to; simply alter the code in accordance to what port you have chosen. Refer to the solder board diagram to use this component.

Step 8: Power and Fan

If you look at the circuit diagram for the entire project, you will see that we need only one input voltage of 5 V. A common adaptor like the one shown above can be used to power the project. Furthermore, you will need a fan to ensure airflow through the box and prevent overheating. We used the fan above but any fan that uses 5 V and is of the appropriate size can be used.

Step 9: Container

While there are many ways to make an effective box, we chose to use an UP 3D printer for our box. We have attached the STL that we used for the final printing.

Attachments

Step 10: Coding

The code for extracting raw data from the device is attached above. This code will print the sensor resistance values, Shinyei PPD42 low-pulse occupancy percentages, and temperature and humidity readings to the computer via the serial monitor. It will cycle through the raw data on LCD screen as well.

In order to make the code work you will first need to download the libraries for the LCD shield and temperature and humidity sensors. You will find the libraries at the following websites

LCD shield code: <https://learn.adafruit.com/rgb-lcd-shield/using-th...>

Temperature and humidity sensor code: <https://github.com/adafruit/DHT-sensor-library>

Step 11: Interpreting the Data

We are in the process of determining how to transform raw sensor values into meaningful outputs. Calibration against known pollution sources will eventually be necessary to ensure accuracy. In meantime we have used sensor data sheets and prior research to make approximations.

To estimate particulate matter concentrations we use information from a research paper by David Holstius. The paper correlates the Shinyei PPD42 dust sensor outputs with EPA measurements. The graphs in the appendix show best fit lines for the data. We used the graphs to approximate PM2.5 concentration in micrograms per cubic meter as:

$PM2.5 = 5 + 5 * (\text{small PM low-pulse occupancy percentage})$



Measure PM2.5 and PM10 levels using Electrochemical Sensors

Objective:

- to measure PM2.5 and PM10 level using Electrochemical Sensors.

Requirements:

- Arduino Uno
- Breadboard & Wires
- Basic Speaker
- MQ-2 Smoke Detecting Sensor
- MQ-7 Carbon Monoxide Co Gas Sensor
- Air Quality Sensor Hazardous Gas Detection Module
- DHT11 Temperature and Humidity Sensor

PROCEDURE:**Introduction**

The sensor will extract air quality information with multiple sensors, graph that data in real time, and even set off a speaker alarm if the air quality reaches a harmful critical point.

Hardware**Arduino Uno**

Arduino Uno is one of the cheapest and most popular microcontrollers available today. Since the Uno has an onboard analog-to-digital converter, the analog signals produced from our sensors can be properly read. Other popular microcontrollers do not have this feature, so that's why we chose the Uno for this project.

Bread Board

A breadboard is a tool used to prototype and test circuits. Breadboards have two long rails (red rail for power and blue rail for ground) on each of its sides that are each connected vertically down the board. This is where engineers typically plug in a power supply and ground so that the whole rail can be easily used as a power and ground connection. At the center of the board, there are groups of 5-holed horizontal lines. Each of these lines is horizontally connected (5-hole connection) which are used for the circuit itself. You can even connect one grouping to another to make a 10-hole connection and so on and so forth.

Basic Speaker

You know what a speaker is. But here's some fun cocktail party knowledge: a speaker is essentially a piece of vibrating metal that moves up by a magnet when fed power and falls down when unpowered. When the magnet is turned on and off at a specific frequency, a tone can be played. You won't need to know much about how to code one of these, but it's always nice to know!

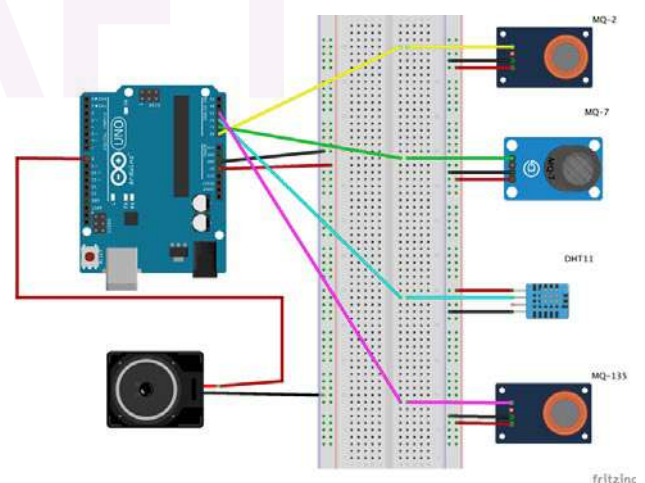
Sensors

Each of the sensors we're going to use operate on the same fundamental principle. A specific type of material is placed inside a sensor and an electric current is fed through it. When certain environmental conditions occur that react with the sensor's specific material, the material's resistance to the current increases or decreases. Thus the current will be affected and the sensor can then translate that change into proper sensor units.

Wiring

The wiring of each of these sensors follows the same pattern: connect ground pin of the sensor to the ground of the Arduino, connect the power pin of the sensor to the 5.5V pin of the Arduino, and connect the data pin of the sensor to a GPIO pin on the Arduino.

You can connect the sensors exactly as we did by following the diagrams exactly.



Smart City - Identify and select various types of sensors used in smartcity

Explore sensors used in weather monitoring system

Objective:

- to explore sensors used in weather monitoring system.

Requirement:

- Thermometer
- Hygrometer
- Barometer

PROCEDURE:

In IOT enabled weather monitoring system project, Arduino Uno measures 4 weather parameters using respective 4 sensors. These sensors are temperature sensor, humidity sensor, light sensor and rain level sensor. These 4 sensors are directly connected to Arduino Uno since it has an inbuilt Analog to digital converter. Arduino calculates and displays these weather parameters on an LCD display. Then it sends these parameters to the Internet using IOT techniques. The process of sending data to the internet using Wi-Fi is repeated after constant time intervals. Then the user needs to visit a particular website to view this data. The project connects and stores the data on a web server. Thus the user gets Live reporting of weather conditions. Internet connectivity or Internet connection with Wi-Fi is compulsory in this IOT weather monitoring project.

Description in Detail:

Imagine a situation where scientists/nature analysts want to monitor changes in a particular environment say

volcano or a rain-forest. And these people are from different places in the world. In this case, SMS based weather monitoring has its own limitations. Since it sends SMS to few numbers. And time for sending SMS increases as the number of mobile numbers increases. In order to send this data to everyone, a person who receives this SMS can upload/add data to some place where everyone can see it. And what else apart from the Internet connects everyone in this world? However, a person doing it manually is time consuming and tedious job. And then there arises a need of an automated solution for this. So in such scenarios, IOT – Internet of Things proves really effective. Using Internet of Things, we can upload these weather parameters data to the cloud using internet connectivity over a WiFi module through wireless communication. Thus this project is also categorized under Wireless communication projects.

Two things are necessary to view this weather reporting over the Internet. One is the Internet and another is a device to access a URL / website. This device can be laptop or desktop or a tablet or even a smartphone. NOTE that Internet connectivity is required at both places. One where is project is placed and another from where user monitors this data.

DRAFT

Smart City - Identify and select various types of sensors used in smartcity

Measure air temperature, humidity, atmospheric pressure and solar radiation

Objective:

- to measure air temperature, Humidity, Atmospheric pressure and solar radiation.

Requirement:

Hardware

- 1x Breadboard
- 1x Arduino Uno
- 4x Jumper Wires
- 1x HDC100x sensor

Software

- Arduino IDE.

Link: <https://www.arduino.cc/en/main/software>

- Arduino library for HDC100x sensor.

Link: https://github.com/RFgermany/HDC100X_Arduino_Library

PROCEDURE:

Here we will see how to set up a whole weather station system, which separated into multi parts:

- Working with Air Temperature and Humidity sensor.
- Working with Air Pressure sensor.
- Working with Solar radiation / Luminosity sensor.

The sensor implemented to record ambient temperature and humidity in this HDC100x from Watterot.

It is a four – pin digital sensor. One of the important criteria of this sensor is its sensitivity with direct sun light exposure and rain.

Therefore, the material preventing the entering of sun light and positioning of the weather station should be considered.

A plastic box was implemented to store this sensor and to avoid the effect of solar radiation, a brown thick paper was applied to cover the top of the weather station box.

According EPA (1987), the standard measurement height for temperature and relative humidity sensors is at least 2.0 meters above the ground surface.

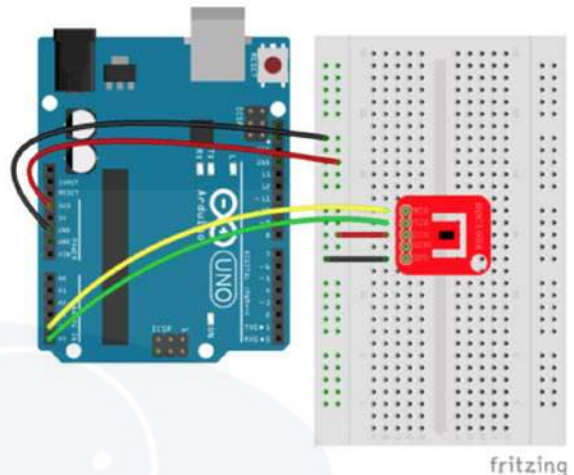
Working with Air Temperature and Humidity sensor

PROCEDURE

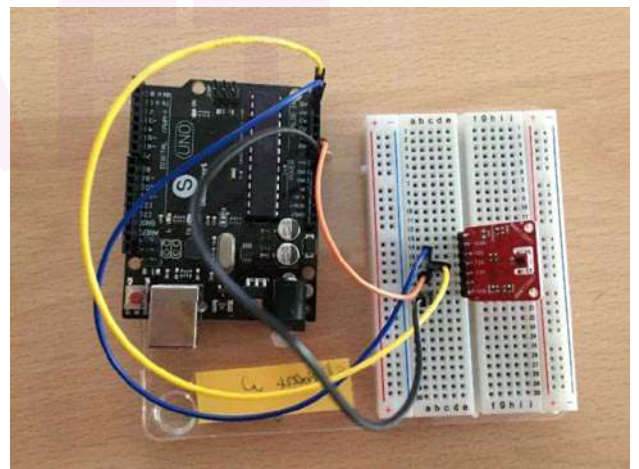
Hardware hookup

- HDC100x is a four-pin sensor, which has the most four important pins needed to be connect:
- **GND:** ground pin, needed to be connect with GND pin on Arduino Uno.

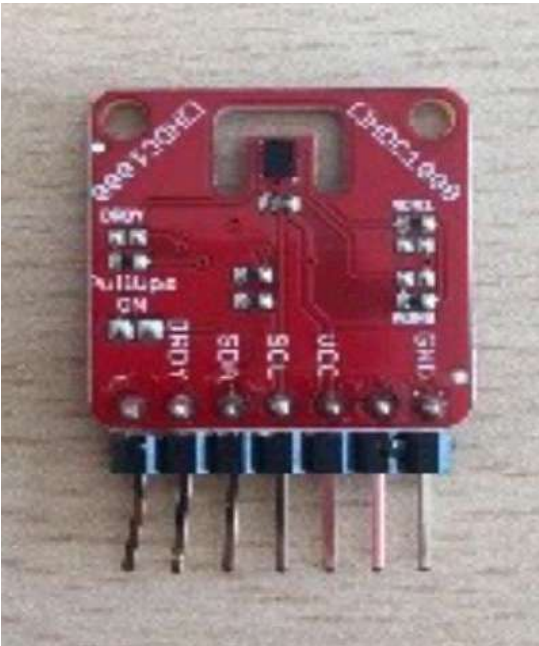
- **VCC:** power pin, needed to be connect with 3.3V pin on Arduino Uno to supply the power for the sensor.
- **SCL:** needed to be connect with A5 on Arduino Uno.
- **SDA:** needed to be connect with A4 on Arduino Uno.



WIRING DIAGRAM OF HDC100X SENSOR TO ARDUINO



REAL WORK APPLICATION



HDC100X SENSOR

Nimi
DRAFT

Smart City - Identify and select various types of sensors used in smartcity

Measure and record Information such as air temperature, wind speed, dew point temperature, wind direction, relative humidity, solar radiation and atmospheric pressure at predetermined intervals by Weather Stations

Objective:

- to measure and record information such as air temperature, wind speed, dew point temperature, wind direction, relative humidity, solar radiation and atmospheric pressure at predetermined intervals by weather stations.

Requirement:

- Arduino Uno
- ESP8266 Wi-Fi Shield
- DHT11 Sensor
- BMP180 Sensor
- Breadboard
- Jumper Wires

PROCEDURE:

Weather stations normally comprises of few sensor to measure environmental parameters and a monitoring or logging system to analyze these parameters.

we will learn how to build such a wireless IoT based weather station which can measure critical environmental parameters like Temperature, Humidity and Pressure. Since our weather station is IoT enabled, we can send these parameters to a ThingSpeak channel (IoT cloud) where we can store, analyze and access the data remotely.

We will be using the Arduino board along with DHT11 sensor, BMP180 sensor and ESP8266 wifi module. The DHT11 sensor senses the temperature and humidity, while BMP180 sensor calculates the pressure and ESP8266 is used for internet connectivity.

ThingSpeak enables live monitoring from anywhere in the world and we can also view the logged data which will be stored on their website and even graph it over time to analyze it.

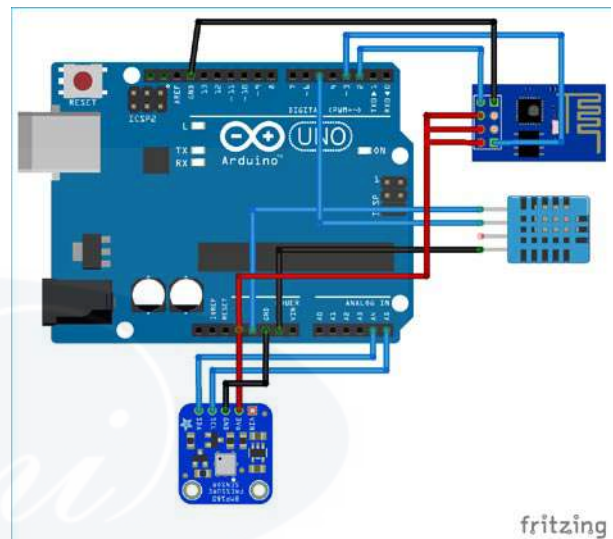
Circuit Diagram

The complete circuit for Arduino based IoT Weather Station is shown below.

The DHT11 sensor is powered by the 5V pin of the Arduino and its data pin is connected to pin 5 for one wire communication. The BMP180 sensor is powered by the 3.3V pin of Arduino and its data pins SCL (Serial Clock) and SDA (Serial Data) are connected to A4 and A5 pin of Arduino for I2C communication.

The ESP8266 module is also powered by the 3.3V pin of the Arduino and its Tx and Rx pins are connected to Digital pins 2 and 3 of Arduino for serial communication.

You can use the below table as reference for making your connections.



S.NO.	Pin Name	Arduino Pin
1	ESP8266 VCC	3.3V
2	ESP8266 RST	3.3V
3	ESP8266 CH_PD	3.3V
4	ESP8266 RX	TX
5	ESP8266 TX	RX
6	ESP8266 GND	GND
7	BMP180 VCC	3V
8	BMP180 GND	GND
9	BMP180 SDA	A4
10	BMP180 SCL	A5
11	DHT-11 VCC	5V
12	DHT-11 Data	5
13	DHT-11 GND	GND

Setting up your ThingSpeak Channel

ThingSpeak is an open data platform that allows you to aggregate, visualize, and analyze live data in the cloud. You can control your devices using ThingSpeak, you can send data to ThingSpeak from your devices, and even you can create instant visualizations of live data, and send alerts using web services like Twitter and Twilio. ThingSpeak has integrated support from the numerical

computing software MATLAB. MATLAB allows ThingSpeak users to write and execute MATLAB code to perform preprocessing, visualizations, and analyses. ThingSpeak takes minimum of 15 seconds to update your readings. We have also done other interesting projects with ThingSpeak like NodeMCU Temperature and Humidity Monitoring



Smart City - Position the appropriate sensors and collect the information required in Smart City

Identify sensors node block diagram and its components

Objective:

- to identify sensors node block diagram and its components.

Requirement:

- Arduino UNO
- A/D converter
- Smart sensors
- RFID
- NFC

PROCEDURE:

Internet of Things:

The Internet of Things (IoT) is about interconnecting embedded systems, bringing together two evolving technologies: wireless connectivity and sensors. These connected embedded systems are independent microcontroller-based computers that use sensors to collect data. These IoT systems are networked together usually by a wireless protocol such as WiFi, Bluetooth, 802.11.4, or a custom communication system. The networking protocol is selected based on the distribution of nodes and the amount of data to be collected.

This data is sent over the network to the main hub or computer. This main computer collects and analyzes the data, storing it in memory and even making system decisions based on the results of the analysis.

Crucial to modern IoT nodes is the need for security with some form of data encryption, the most common being AES256. This security is critical for helping the microcontroller perform a secure boot, insuring that the core is running the code is meant to run. Encryption security is also used to encrypt the data transmitted over the network, insuring that it is viewed only by those systems authorized to receive the data.

Smart Sensors for the Internet of Things

Smart Sensors are sensors that have some data processing on board. These can be anywhere from environmental sensors that perform basic processing of the data, to high accuracy MEMS sensors with a gyroscope and accelerometer with integrated digital processing for sensor fusion calculations.

A/D Converters for the Internet of Things

Some of the systems being monitored in an IoT network already supply an analog signal. This is common in automotive and industrial systems. This analog signal can be fed to an analog to digital converter (ADC). An ADC measures an analog signal and converts the magnitude to a binary number. While many

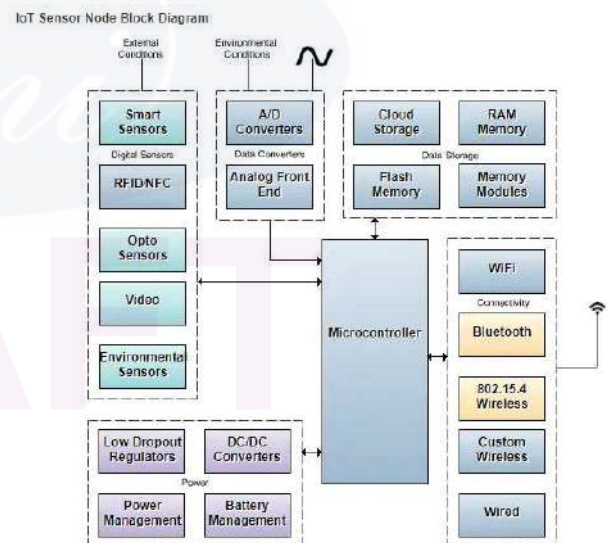
microcontrollers have an ADC onboard, some high speed and/or high precision applications may require a more sophisticated ADC and so an external chip is used.

RFID and NFC for the Internet of Things

RFID and NFC sensors are useful for detecting the presence of items that have been identified with an RFID or NFC tag.

Analog Front Ends

An Analog Front End (AFE) is used for more complex waveforms when just an analog to digital converter (ADC) is not sufficient. An AFE has a higher level of integration and includes an ADC as well as signal conditioning blocks that can include a programmable gain amplifier (PGA) and filtering circuits. It does the work of an ADC and several opamps.



Optical Sensors for the Internet of Things

Optical sensors detect the presence of light at visible or non-visible wavelengths.

Environmental Sensors

Environmental sensors detect the status of the present environment, including pressure, temperature, position, acceleration, motion, or orientation. These can be simple temperature sensors or complex MEMS devices. These sensors provide a digital output that can easily be read by the main microcontroller.

Low Dropout Regulators

Low Dropout (LDO) Voltage Regulators are low loss voltage regulators that insure the system gets the proper voltage with the best possible efficiency.

DC/DC Converters

DC-to-DC converters take a battery voltage either step the voltage up or down. Stepping the voltage up insures that the system is operating at the correct voltage. DC/DC converters can also step the voltage down with an efficiency much higher than a linear regulator or resistor divider network.

Power Management ICs for the Internet of Things

Sophisticated IoT nodes with high performance microcontrollers can have a power management chip to insure the proper distribution of power to the microcontroller as well as the rest of the system.

Battery Management ICs

For battery powered systems, especially those that use rechargeable batteries, a battery management IC can

be essential for extending the service life of the system. These ICs monitor the battery voltage and insure that the appropriate charging current and voltage is applied.

Microcontrollers for the Internet of Things

Although an 8-bit microcontroller may be useful for simple IoT nodes, most IoT applications use either a 16-bit or a 32-bit microcontroller. 16-bits are used for low to medium complexity nodes and provide a good compromise between low power and data throughput. However, most IoT nodes use 32-bits for performance and a high level of integration. Some microcontrollers have sensor interfaces, voltage regulators, and on-chip RF radios, perfect for IoT nodes.

802.15.4 Wireless

Some IoT network nodes are dispersed in such a way that a peer-to-peer, or mesh network, is more appropriate for the exchange of data. 802.15.4 Networking supports peer-to-peer communications. Standards include ZigBee, Thread, Wireless HART, and 6LoWPAN.



Smart City - Position the appropriate sensors and collect the information required in Smart City

Check connection with sensors and send data wirelessly to a central data logger at program

Objective:

- to check connection with sensors and send data wirelessly to a central data logger at program.

Requirements:

- Arduino Uno
- Adafruit Data Logging Shield
- DHT11 module
- CR1220 3V battery
- SD card

PROCEDURE:

Step 1: Materials

This is a minimal project with only few external component, for this the BOM list will be really short. However, let's see what material is requested:

- NodeMcu V3 (or any compatible ESP8266 μ processor);
- RGB led (common anode);
- Resistors for led ($1 \times 10\Omega$, $1 \times 22\Omega$, $1 \times 100\Omega$, $1 \times 10k\Omega$)
- DS18B20 (Maxim Integrated thermometer);
- LM35 (Texas Instrument thermometer);
- External battery (optional);
- Cable;
- Connector (to make it more "advanced");
- Box (optional, again to make it more "advanced");
- Led holder (optional);

Note: As I said you need to chose one of the two methods. If you choose LM35 thermometer, you'll need a few others component.

- Attiny 45/85;
- AVR programmer (or Arduino as ISP);
- Resistor ($1 \times 1k\Omega$, $1 \times 2k\Omega$, $1 \times 10k\Omega$, $1 \times 18k\Omega$)
- 2.54mm strip connector (optional)
- Diode ($2 \times 1N914$)
- Perfboard or PCB;

Step 2: Choosing the Sensor

Choosing the sensor can be a difficult step: today there are tons of transducers (TI offers 144 different element) both analog and digital with different temperature range, accuracy and case.

Analog Sensors (46 parts available from TI):

Pros:

- The data logger can easily be changed from temperature to another quantity (voltage, current, ...);
- May be a little cheaper;
- Easy to use since it doesn't require any special library;
- Cons:
 - Require ADC (that can influence the measurement's accuracy) and other external components. Since esp8266 has only one ADC (and not really accurate) I would suggest to use an external one.
 - Needs dedicated cable with noise rejection since any inducted voltage can change the result.

After a bit of thinking I decided to use LM35, a linear sensor with $+10\text{mV}/^\circ\text{C}$ scale factor with 0.5°C accuracy and a very low current (about $60\mu\text{A}$) with a operating voltage from 4V to 30V. For more detail I suggest to see the datasheet: LM35.

Digital Sensors (highly recommended)

Pros:

- Almost any external components needed;
- Integrated ADC

Cons:

- Request library or software can decode the digital signal (I2C, SPI, Serial, One Wire, ...);
- More expensive;

I've chosen DS18B20 because I found a set of 5 waterproof sensors on Amazon and because it's widely documented on internet. Main feature is 9-12bit measurement, 1-Wire bus, 3.0 to 5.5 supply voltage, 0.5°C accuracy. Again, for more detail here is the datasheet: DS18B20.

Step 3: LM35

Let's analyse how I've implemented the external ADC and other feature for the LM35 thermometer. I found a cable with three wires, one with shielding and two without. I decided to add a decoupling capacitor to stabilize the supply voltage near the sensor.

To convert analog temperature to digital, I've used Attiny85 microprocessor in a dip8 package (again for

more information see the datasheet: attiny85). Most important thing for us is the 10 bit ADC (not really the best one but enough precise for me). To communicate with Esp8266 I decided to use Serial communication keeping in mind that esp8266 works with 3.3V and attiny85 at 5V (as it needs to power the sensor). To achieve that, i used a simple voltage divider (see schematic). To read negative temperature we need to add some external components (2x1N914 and 1x18k resistor), since I don't want to use negative power supply.

SESP8266 Code

- OTA update: you don't need to plug esp8266 to your computer every time you need to upload the code (you have to do it only the first time);
- Wireless manager, if wireless network change, you don't need to reupload the sketch. You can simply configure again the network parameters connecting to esp8266 access point;
- Thingspeak data trasmission;
- Both LM35 and DS18B20 supported;
- Simple User Interface (RGB led indicates some useful information);

Common LM35 and DS18B20 configuration

You need to change pin definition, token, channel number, user and password for OTA update. Line from 15 to 23.

```
#define red YOURPINHERE #define green YOURPINHERE
```

```
#define blue YOURPINHERE
```

```
const char* host = "select host address"; //not really needed you can leave esp8266-webupdate
```

```
const char* update_path = "/firmware"; //to change the address for updating ex: 192.168.1.5/firmware
```

```
const char* update_username = "YOURUSERHERE";
```

```
const char* update_password = "YOURPASSWORDHERE";
```

```
unsigned long myChannelNumber = CHANNELNUMBERHERE;
```

```
const char * myWriteAPIKey = "WRITEAPIHERE";
```

Step 6: ESP8266 Code: LM35 User

You need to connect the attiny board to esp8266, to power the ADC unit use the VU pin and G pin. You need to choose which pin you want to use for serial communication (to keep hardware serial free for debug purpose). Tx pin must be selected but isn't really used. (Line 27).

```
SoftwareSerial mySerial (RXPIN, TXPIN);
```

On the top you need to add:

```
#define LM35USER
```

Step 7: ESP8266 Code: DS18B20 User

As first operation you need to identify the device Address for each sensor. Compile and program this code to the esp and look in serial for the results. Code can be found here (search for this title in the page: «Read individual DS18B20 Internal Addresses»). Connect only one sensor to obtain the address, results should be something like this (random number here! Just as example):

```
0x11, 0x22, 0x33, 0xD9, 0xB1, 0x17, 0x45, 0x12
```

Then you need to change my code in the section "Configuration for DS18B20" (line 31 to 36)":

```
#define ONE_WIRE_BUS ONEWIREPINHERE
```

```
#define TEMPERATURE_PRECISION TEMPBITPRECISION //(from 9 to 12)
```

```
#define delayDallas READINTERVAL //(In Milliseconds, minimum is 15s or 15000mS)
```

```
DeviceAddress blueSensor = { 0x11, 0x22, 0x33, 0xD9, 0xB1, 0x17, 0x45, 0x12}; //CHANGE WITH YOUR ADDRESS
```

```
DeviceAddress redSensor = { 0x11, 0x22, 0x33, 0xD9, 0xB1, 0x17, 0x45, 0x12}; //CHANGE WITH YOUR ADDRESS
```

```
DeviceAddress greenSensor = {0x11, 0x22, 0x33, 0xD9, 0xB1, 0x17, 0x45, 0x12 }; //CHANGE WITH YOUR ADDRESS
```

On the top you need to add:

```
#define DSUSER
```

ESP8266 Little Trick

After a bit of testing I found that if you plug esp8266 without programming, it will not run the code until you press reset one time. To solve this issue, after a bit of research, I discovered that you have to add a pull-up resistor from 3.3V to D3. This will tell the processor to load the code from flash memory.

With this method, D3 can directly be used to data input for DS18B20 sensors.

First Time Operation

If you've uploaded the code correctly but never use Wifi manager library it's time to configure your wifi connection. Wait until you see the RGB led flashing faster than before, then search with your mobile or PC the wifi network called "AutoConnectAp" and connect.

After connection, open a web browser and enter 192.168.4.1, you'll find the GUI interface of wifi manager (see photos) and press "Configure Wifi". Wait for esp8266 to search wifi networks, and select the desired one. Insert password and press "save".

Esp8266 will restart (don't care RGB led this time

because it'll output some random information) and connect to the network.

Conclusion

In the end, here is a graph taken from the data logger in action while logging my freezer temperature. In orange is the DS18B20 and in blue the LM35 and it's circuit. You

can see the biggest difference in accuracy from digital to analog sensor (with my poor "ADC circuit") that give some non-physical data.

Summing up, if you want to build this logger I suggest using the DS18B20 digital temperature sensor since it's easier to read and almost "plug and play", it's more stable and accurate, it runs at 3.3V and require only one pin for lots of sensors.



Smart City - Position the appropriate sensors and collect the information required in Smart City

Identify interfacing of wireless modules with IoT platfor

Objective:

- to identify interfacing of wireless modules with IoT Platform.

Requirements:

- UART Interface
- Arduino UNO
- Arduino IDE

PROCEDURE:

There is much that goes into IoT application creation and development. In addition to understanding system design, you'll also need to understand the following:

- How your device will connect to the internet.
- How you're going to provide your device with credentials.
- How your device will be set up and managed.
- How to select an IoT module for your device.

And while picking an Internet of Things connectivity module isn't the first step you'll take, it's an important one. We've outlined some of the most popular IoT module types in this article, so you can understand what will work best for you.

ZIGBEE

ZigBee is used for building low power, short-range mesh networks. If the mesh is well designed and well distributed—meaning it has a relatively dense number of nodes—it has the ability to cover a large area. If there are holes in the mesh, the true range performance is pretty limited.

We've seen some application creators struggle to sell products that use ZigBee's hardware because their customers don't design the mesh properly (and then have numerous issues to deal with). It's worth noting that there are other proprietary versions of ZigBee that can work really well, like MeshConnect from CEL. This product is especially popular for smart home and energy solutions.

6LOWPAN & THREAD

6LoWPAN and Thread are rapidly growing standards. They are low power versions of IP-based communications that allow end nodes to use 6LoWPAN instead of WiFi to communicate back to access points. Google and Nest are big proponents of these technologies. We at Link Labs can't use 6LoWPAN because the header size is too large for our technology, but this is certainly a space to watch. If 6LoWPAN

becomes a standard type of wireless (especially in homes), it'll be an attractive module option.

The issue here is that you need a bridge from 6LoWPAN to ethernet somewhere in the system. The Nest thermostat, for example, acts as that bridge—the thermostat has a WiFi chip and a Thread chip in it. But, in order for 6LoWPAN to work with an application, you have to sell some kind of access point that can connect to the internet.

Z-WAVE

Z-Wave is a closed ecosystem of sub-GHz RF devices that work very well together. People tend to like Z-Wave because its range is very good, it's very easy to use, and it has good interoperability. For significant home automation, it's definitely more popular than ZigBee.

Like Thread and 6LoWPAN, Z-Wave needs its own proprietary access point or gateway. So, in addition to selling a product, you also have to sell an access point. This makes your go-to marketing strategy more complex, which is a downside.

WIFI

WiFi modules give users an easy gateway to the internet. If you're selling an IoT solution like a smart thermostat, which is made for homes and businesses that are already WiFi-connected, WiFi is a go-to option.

One big issue with WiFi is provisioning: How can you get the password for the customer's network onto the device? Someone has to type it in, and that's a major headache for many customers. Things like WiFi protected setup (WPS) were created in response to this issue. WPS was supposed to allow customers to push a device on their routers and allow devices to join for a short period of time. But, they were never standardized and are still not used regularly.

It's also important to consider WiFi from a security perspective. If you use a connected product that has a WiFi module in it, you must understand that you are now operating a device inside someone's network. Thus, you need to be a good steward and make sure you are not creating vulnerabilities for that network.

Power budget is another issue. We don't think you'll find very many battery-powered WiFi IoT applications out there, because staying connected to WiFi is relatively power intensive.

BLUETOOTH

This is a really interesting use case, because customers actually carry the Bluetooth-to-internet bridge around in their pocket. The Apple iBeacon ecosystem is one type of software development kit (SDK) that provides a persistent Bluetooth-to-internet connection that runs in the background.

For example, with the use of a product like Gimbal's Proximity Beacon, Bluetooth-enabled devices can feed data from a person's phone to the internet without them knowing it. If someone has the ESPN app running in the background of her phone, and she walks into a store with a Proximity Beacon in use, store owners can tell she's in their store without her ever knowing it. These

kinds of real-life use cases make Bluetooth and Bluetooth Low Energy (BLE) an interesting module option.

CELLULAR

The advantages of using a cellular module are that it offers ubiquitous coverage and is an out-of-the-box solution. The issues with cellular lie in provisioning and subscription management. You don't want to be stuck paying for data for a bunch of products that are still sitting on the shelf.

There are a lot of companies in the mobile virtual network operator (MVNO) space—like KORE Wireless—or strictly in the device management space—like Jasper—that make that process easier. Aeris is another really good company that makes getting a data plan for your IoT devices easier, and we've seen it in use a lot more lately.



Smart City - Position the appropriate sensors and collect the information required in Smart City

Select and Install sensors like NO₂, PM_{2.5}, PM₁₀, CO₂, O₂, air temperature, humidity etc

Objective:

- to select and Install sensors like NO₂, PM_{2.5}, PM₁₀, CO₂, O₂, air temperature, Humidity.

Requirement:

- Arduino Uno
- 5V power supply
- RGB 16x2 LCD shield
- Shinyei PPD42 Particulate Matter Detector
- MQ-2 Gas Sensor
- MQ-9 Gas Sensor
- MiCS-2714 Gas Sensor (NO₂)
- MiSC-2614 Gas Sensor (Ozone)
- Keyes DHT11 Temperature and Humidity Sensor
- Access to 3D printer
- Solder Board
- 5V fan
- 10 to 15 wires of gauge 24

PROCEDURE:

New Calibrated Air Quality Sensors for Smart Cities

Libelium released a **new Gases PRO Sensor Board** in May 2015. This platform features precision sensors to enable the implementation of urban monitoring, air quality, industrial, environmental or farming projects with high requirements in terms of accuracy, reliability, and measurement range. The new gas sensors line is available for both Waspnote OEM and Plug & Sense!.

Two different Sensor Boards are available to monitor and measure gas levels: Gases Sensor Board (Gases) and Gases PRO Sensor Board (Gases PRO).

Comparative Gases and Gases PRO Sensor Boards:

Number of sensors: Gases PRO integrates 16 different gas sensors. The original Gases board integrates 12. See the complete list of parameters in the tables below.

Increased accuracy: Gases PRO measures exact values of gas concentration in ppms. The Gases board shows a range of concentration low / medium / high. Details of the accuracy ranges figure in the tables below.

Precision sensors: A sensitive new particulate matter sensor—commonly known as dust sensor—is included in the Gases PRO board. This sensor allows the measurement of PM₁, PM_{2.5}, and PM₁₀ particles and comes calibrated from the factory.

Should I choose Gases or Gases PRO?

Libelium has created different systems in terms of accuracy and pricing. The customer should consider Gases PRO if she needs maximum performance sensors for metering that require accurate ppm or percentage. If the project just needs to detect gas presence or gas levels, the standard Gases Sensor Board is sufficient.

The new Gases PRO Sensor Board can read up to 16 gas sensors; it also has a high-end sensor for three parameters: **temperature, humidity and pressure.**

Gases PRO [New Version]

Parameter	Range	Accuracy *
Temperature	-40 to +85 °C	Calibrated ±1 °C (±0.5 °C at 25 °C)
Humidity	0 to 100% HR	Calibrated ±3% RH (at 25 °C, range 20 ~ 80% RH)
Pressure	30 to 110 kPa	Calibrated ±0.1 kPa (range 0 ~ 65 °C)
Carbon Monoxide for	0 to 500 ppm	Calibrated ±1 ppm

high concentrations CO

Carbon Monoxide for low concentrations CO	0 to 25 ppm	Calibrated ± 0.1 ppm
Carbon Dioxide CO₂	0 to 5000 ppm	Calibrated ± 50 ppm (range 0~2500 ppm) ± 200 ppm (range 2500~5000 ppm)
Molecular Oxygen O₂	0 to 30%	Calibrated $\pm 0.1\%$
Ozone O₃	0 to 18 ppm	Calibrated ± 0.2 ppm
Nitric Oxide for low concentrations NO	0 to 20 ppm	Calibrated ± 0.2 ppm
Nitric Dioxide high accuracy NO₂	0 to 20 ppm	Calibrated ± 0.1 ppm
Sulfur Dioxide high accuracy SO₂	0 to 20 ppm	Calibrated ± 0.1 ppm
Ammonia for low concentrations NH₃	0 to 100 ppm	Calibrated ± 0.5 ppm
Ammonia for high concentrations NH₃	0 to 500 ppm	Calibrated ± 3 ppm
Methane and other combustible gases CH₄	0 to 100 % / LEL	Calibrated ± 0.15 % LEL
Molecular Hydrogen H₂	0 to 1000	Calibrated ± 10 ppm
Hydrogen Sulfide H₂S	0 to 100 ppm	Calibrated ± 0.1 ppm
Hydrogen Chloride HCl	0 to 50 ppm	Calibrated ± 1 ppm
Hydrogen Cyanide HCN	0 to 50 ppm	Calibrated ± 0.2 ppm
Phosphine PH₃	0 to 20 ppm	Calibrated ± 0.1 ppm
Ethylene Oxide ETO	0 to 100 ppm	Calibrated ± 1 ppm
Chlorine Cl₂	0 to 50 ppm	Calibrated ± 0.1 ppm
Particle Matter – Dust	0.5 to 16 μ m (16 steps) (includes PM1, PM2.5 and PM10)	Calibrated

* Accuracy values have been calculated at average conditions: 20°C / 101300 Pa. Accuracy levels may differ with different temperature and pressure levels, aging and in presence of third type of gases which cause cross sensitivity. See each Gas Sensor documentation for more information.

Gases v2.0 [Old Version]

Parameter	Range	Accuracy *
Temperature	40 to +125 °C	Not calibrated ± 2 °C (range 0 ~ 70 °C), ± 4 °C (range -40 ~ 125 °C)
Humidity	0 to 100% HR	Not calibrated < $\pm 4\%$ RH (at 25 °C, range 30 ~ 80% RH), < $\pm 6\%$ RH (range 0 ~ 100% RH)
Pressure	15 to 115 kPa	Not calibrated < $\pm 1.5\%$ V (range 0 ~ 85 °C)
Carbon Monoxide CO	30 to 1000 ppm	Not calibrated

Carbon Dioxide CO²	350 to 10000 ppm	Not calibrated
Molecular Oxygen O²	0 to 30%	Not calibrated
Ozone O³	0.010 to 1 ppm	Not calibrated
Nitric Dioxide NO²	0.05 to 5 ppm	Not calibrated
Ammonia NH³	10 to 100 ppm	Not calibrated
Methane and other combustible gases CH₄	500 to 10000 ppm	Not calibrated
Molecular Hydrogen H²	500 to 10000 ppm	Not calibrated
Hydrogen Sulfide H₂S	0.1 to 3 ppm	Not calibrated
Isobutane C₄H₁₀	50 to 5000 ppm	Not calibrated
Ethanol CH₃CH₂OH	1 to 30 ppm	Not calibrated
	50 to 5000 ppm	Not calibrated

Toluene C₆H₅CH₃ ppm calibrated	1 to 30 Not
Volatile Organic Compounds ppm calibrated	30 to 400 Not
Hydrocarbons Not calibrated	-

How are the Gases PRO sensors calibrated?

All the Gases PRO sensors are linear, so they behave according to a simple line like $y = f(x) = m \cdot x + c$. On the other hand, the Gases Sensor Board sensors show a logarithmic response, which is harder to handle.

All the gas sensors for Gases PRO are calibrated by the sensor manufacturers, in laboratories. To obtain maximum accuracy, a two-point calibration process is

performed on controlled concentrations of gas in vacuum chambers. Due to the linear nature of the sensors, these two points are enough to correct the two possible drifts in m (slope) and c (constant offset). Our sensor manufacturers provide a calibration certificate for each individual sensor, with two calibration parameters. Libelium stores these parameters in an EEPROM, a non-volatile memory chip inside each gas sensor. When the user performs a sensor reading, the software system designed by Libelium reads the sensor signal and the two parameters. These three values are processed by the system, and a temperature compensation calculation is applied. From this data we obtain the concentration value in ppms – the value we want to measure.

The new Temperature, Humidity and Pressure sensor for Gases PRO is calibrated in the factory. It comes with three calibration points for temperature, six points for humidity and nine points for pressure. These three parameters are automatically inter-compensated to obtain extreme accuracy.

Gases PRO Sensor Board is included in Libelium's new Smart Cities sensing solution, Plug & Sense! Smart Environment PRO. This board also allows to obtain information about particle size and density ranging from 1um to 10um (PM1 / PM2.5 / PM10) in its surroundings, using the new particulate matter sensor—commonly known as dust sensor.

Smart City - Position the appropriate sensors and collect the information required in Smart City

Configure sensor node using USB and over the air programming

Objective:

- to configure sensor node using USB and over the air programming.

Requirement:

- Arduino IDE
- Arduino UNO
- Windows 7, 10, Vista

PROCEDURE:

ESP32 Over-the-air (OTA) Programming – Web Updater Arduino IDE

Quick guide that shows how to do over-the-air (OTA) programming with the ESP32 using the OTA Web Updater in Arduino IDE. The OTA Web Updater allows you to update/upload new code to your ESP32 using a browser, without the need to make a serial connection between the ESP32 and your computer.



OTA programming is useful when you need to update code to ESP32 boards that are not easily accessible. The example we'll show here works when the ESP32 and your browser are on your local network.

The only disadvantage of the OTA Web Updater is that you have to add the code for OTA in every sketch you upload, so that you're able to use OTA in the future.

How does OTA Web Updater Works?

- The first sketch should be uploaded via serial port. This sketch should contain the code to create the OTA Web Updater, so that you are able to upload code later using your browser.
- The OTA Web Updater sketch creates a web server you can access to upload a new sketch via web browser.
- Then, you need to implement OTA routines in every sketch you upload, so that you're able to do the next updates/uploads over-the-air.

- If you upload a code without a OTA routine you'll no longer be able to access the web server and upload a new sketch over-the-air.

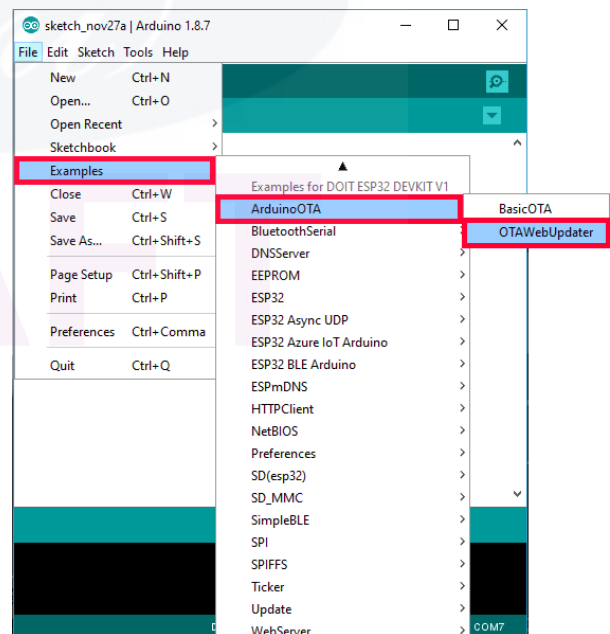
Prerequisites

Before proceeding with this tutorial you should have the ESP32 add-on installed in your Arduino IDE. Follow one of the next tutorials to install the ESP32 on the Arduino IDE, if you haven't already.

- [Installing the ESP32 Board in Arduino IDE \(Windows instructions\)](#)
- [Installing the ESP32 Board in Arduino IDE \(Mac and Linux instructions\)](#)

ESP32 OTA Web Updater

When you install the ESP32 add-on for the Arduino IDE, it will automatically install the ArduinoOTA library. Go to **File > Examples > ArduinoOTA > OTAWebUpdater**.



The following code should load.

```
/*
```

* OTAWebUpdater.ino Example from ArduinoOTALibrary

* Rui Santos

* Complete Project Details <https://randomnerdtutorials.com>

```

*/
#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include <ESPmDNS.h>
#include <Update.h>
const char* host = "esp32";
const char* ssid = "REPLACE_WITH_YOUR_SSID";
const      char*      password      =
"REPLACE_WITH_YOUR_PASSWORD";
WebServer server(80);
/*
* Login page
*/
const char* loginIndex =
"<form name='loginForm'>"
"<table width='20%' bgcolor='A09F9F' align='center'>"
"  <tr>"
"    <td colspan=2>"
"      <center><font size=4><b>ESP32 Login Page</b></font></center>"
"    <br>"
"    </td>"
"  <br>"
"  <br>"
"  </tr>"
"  <td>Username:</td>"
"  <td><input type='text' size=25 name='userid'><br></td>"
"  </tr>"
"  <br>"
"  <br>"
"  <tr>"
"    <td>Password:</td>"
"    <td><input      type='Password'      size=25"
"    name='pwd'><br></td>"
"  <br>"
"  <br>"
"  </tr>"
"  <tr>"
"    <td><input type='submit' onclick='check(this.form)' "
"    value='Login'></td>"
"  </tr>"

```

```

"</table>"
"</form>"
"<script>"
"function check(form)"
"{ " "if(form.userid.value=='admin' && "
"form.pwd.value=='admin')"
"{" "window.open('/serverIndex') "
"}" "else"
"{"
"  alert('Error Password or Username')/*displays error "
"message*/"
"  }"
"  }"
"</script>";
/*
* Server Index Page
*/
const char* serverIndex =
"<script src='https://ajax.googleapis.com/ajax/libs/jquery/"
"3.2.1/jquery.min.js'></script>"
"<form method='POST' action='#' enctype='multipart/"
"form-data' id='upload_form'>"
"<input type='file' name='update'>"
"<input type='submit' value='Update'>"
"</form>"
"<div id='prg'>progress: 0%</div>"
"<script>"
"$('form').submit(function(e){"
"  e.preventDefault();"
"  var form = $('#upload_form')[0];"
"  var data = new FormData(form);"
"  $.ajax({"
"    url: '/update',"
"    type: 'POST',"
"    data: data,"
"    contentType: false,"
"    processData:false,"
"    xhr: function() {"
"      var xhr = new window.XMLHttpRequest();"
"      xhr.upload.addEventListener('progress', function(evt) "
"      {"
"        if (evt.lengthComputable) {"
"          var per = evt.loaded / evt.total;"

```

```

    "$('#prg').html('progress: ' + Math.round(per*100) +
    '%');"
    "}"
    "}, false);"
    "return xhr;"
    "},"
    "success:function(d, s) {"
    "console.log('success!')"
    "},"
    "error: function (a, b, c) {"
    "}"
    "});"
    "});"
    "</script>";
/*
* setup function
*/
void setup(void) {
Serial.begin(115200);
// Connect to WiFi network
WiFi.begin(ssid, password);
Serial.println("");
// Wait for connection
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.print("Connected to ");
Serial.println(ssid);
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
/*use mdns for host name resolution*/
if (!MDNS.begin(host)) { //http://esp32.local
Serial.println("Error setting up MDNS responder!");
while (1) {
delay(1000);
}
}
Serial.println("mDNS responder started");
/*return index page which is stored in serverIndex */
server.on("/", HTTP_GET, []() {

```

```

server.sendHeader("Connection", "close");
server.send(200, "text/html", loginIndex);
});
server.on("/serverIndex", HTTP_GET, []() {
server.sendHeader("Connection", "close");
server.send(200, "text/html", serverIndex);
});
/*handling uploading firmware file */
server.on("/update", HTTP_POST, []() {
server.sendHeader("Connection", "close");
server.send(200, "text/plain", (Update.hasError()) ?
"FAIL" : "OK");
ESP.restart();
}, []() {
HTTPUpload& upload = server.upload();
if (upload.status == UPLOAD_FILE_START) {
Serial.printf("Update: %s\n", upload.filename.c_str());
if (!Update.begin(UPDATE_SIZE_UNKNOWN)) { //
start with max available size
Update.printError(Serial);
}
} else if (upload.status == UPLOAD_FILE_WRITE) {
/* flashing firmware to ESP*/
if (Update.write(upload.buf, upload.currentSize) !=
upload.currentSize) {
Update.printError(Serial);
}
} else if (upload.status == UPLOAD_FILE_END) {
if (Update.end(true)) { //true to set the size to the current
progress
Serial.printf("Update Success: %u\nRebooting...\n",
upload.totalSize);
} else {
Update.printError(Serial);
}
}
});
server.begin();
}
void loop(void) {
server.handleClient();
delay(1);
}

```


View raw code

You should change the following lines on the code to include your own network credentials:

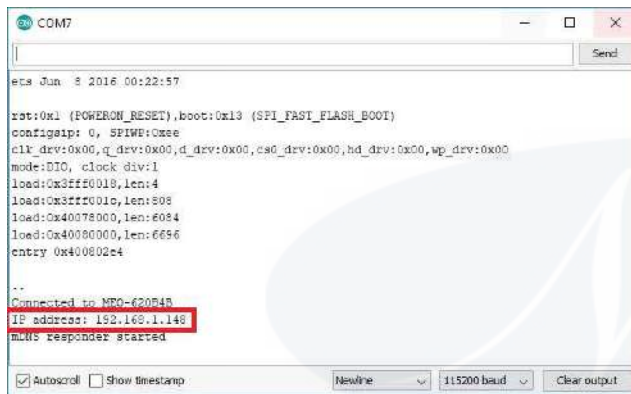
```
const char* ssid = "";  
const char* password = "";
```

The OTAWebUpdater example for the ESP32 creates an asynchronous web server where you can upload new code to your board without the need for a serial connection.

Upload the previous code to your ESP32 board. Don't forget to enter your network credentials and select the right board and serial port.



After uploading the code, open the Serial Monitor at a baud rate of 115200, press the ESP32 enable button, and you should get the ESP32 IP address:

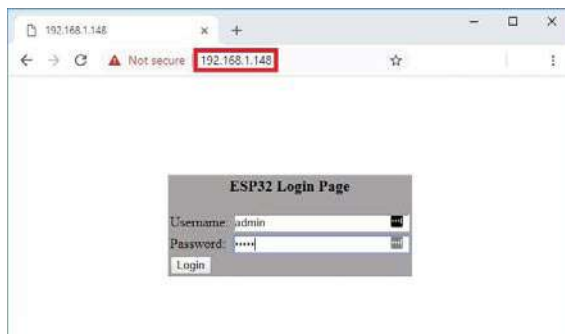


Now, you can upload code to your ESP32 over-the-air using a browser on your local network.

To test the OTA Web Updater you can disconnect the ESP32 from your computer and power it using a power bank, for example (this is optional, we're suggesting this to mimic a situation in which the ESP32 is not connected to your computer).

Update New Code using OTA Web Updater

Open a browser in your network and enter the ESP32 IP address. You should get the following:



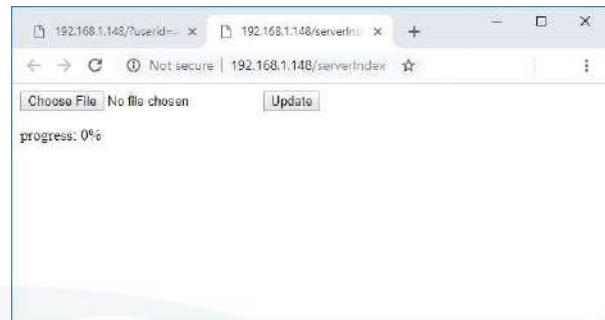
Enter the username and the password:

- **Username:** admin
- **Password:** admin

You can change the username and password on the code.

Note: After you enter the username and password, you are redirected to the /serverIndex URL. You don't need to enter the username and password to access the /serverIndex URL. So, if someone knows the URL to upload new code, the username and password don't protect the web page from being accessible from others.

A new tab should open on the /serverIndex URL. This page allows you to upload a new code to your ESP32. You should upload .bin files (we'll see how to do that in a moment).



Preparing the New Sketch

When uploading a new sketch over-the-air, you need to keep in mind that you need to add code for OTA in your new sketch, so that you can always overwrite any sketch with a new one in the future. So, we recommend that you modify the OTAWebUpdater sketch to include your own code.

For learning purposes let's upload a new code that blinks an LED (without delay). Copy the following code to your Arduino IDE.

```
/*
```

```
* Rui Santos
```

```
* Complete Project Details https://  
randomnerdtutorials.com
```

```
*/
```

```
#include <WiFi.h>
```

```
#include <WiFiClient.h>
```

```
#include <WebServer.h>
```

```
#include <ESPmDNS.h>
```

```
#include <Update.h>
```

```
const char* host = "esp32";
```

```
const char* ssid = "REPLACE_WITH_YOUR_SSID";
```

```
const char* password = "REPLACE_WITH_YOUR_PASSWORD";
```

```
//variables to blink without delay:
```

```

const int led = 2;
unsigned long previousMillis = 0;    // will store last time
LED was updated
const long interval = 1000;          // interval at which to
blink (milliseconds)
int ledState = LOW;                  // ledState used to set the LED
WebServer server(80);
/*
 * Login page
 */
const char* loginIndex =
"<form name='loginForm'>"
  "<table width='20%' bgcolor='A09F9F' align='center'>"
"<tr>"
  "<td colspan=2>"
"<center><font size=4><b>ESP32 Login Page</b></font></center>"
  "<br>"
"</td>"
  "<br>"
  "<br>"
  "</tr>"
  "<td>Username:</td>"
  "<td><input type='text' size=25 name='userid'><br></td>"
"</tr>"
  "<br>"
  "<br>"
  "<tr>"
  "<td>Password:</td>"
  "<td><input type='Password' size=25 name='pwd'><br></td>"
  "<br>"
  "<br>"
  "</tr>"
  "<td><input type='submit' onclick='check(this.form)' value='Login'></td>"
"</tr>"
"</table>"
"</form>"
"<script>"
"function check(form)"

```

```

{"
  "if(form.userid.value=='admin' && form.pwd.value=='admin')
{"
  "window.open('/serverIndex')
}"
  "else"
  {"
    "alert('Error Password or Username')/*displays error message*/"
  }
}
"</script>";
/*
 * Server Index Page
 */
const char* serverIndex =
"<script src='https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js'></script>"
"<form method='POST' action='#' enctype='multipart/form-data' id='upload_form'>"
  "<input type='file' name='update'>"
  "<input type='submit' value='Update'>"
  "</form>"
"<div id='prg'>progress: 0%</div>"
"<script>"
  "$('form').submit(function(e){
    "e.preventDefault();"
    "var form = $('#upload_form')[0];"
    "var data = new FormData(form);"
    "$.ajax({
      "url: '/update',"
      "type: 'POST',"
      "data: data,"
      "contentType: false,"
      "processData:false,"
      "xhr: function() {"
        "var xhr = new window.XMLHttpRequest();"
        "xhr.upload.addEventListener('progress', function(evt)
        {"
          "if (evt.lengthComputable) {"
            "var per = evt.loaded / evt.total;"

```

```

    "$(#prg').html('progress: ' + Math.round(per*100) +
    '%');"
    }
    }, false);"
    "return xhr;"
    },
    "success:function(d, s) {"
    "console.log('success!')
    },
    "error: function (a, b, c) {"
    }
    },"
    },"
    "</script>";
/*
 * setup function
 */
void setup(void) {
pinMode(led, OUTPUT);
Serial.begin(115200);
// Connect to WiFi network
WiFi.begin(ssid, password);
Serial.println("");
// Wait for connection
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.print("Connected to ");
Serial.println(ssid);
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
/*use mdns for host name resolution*/
if (!MDNS.begin(host)) { //http://esp32.local
    Serial.println("Error setting up MDNS responder!");
    while (1) {
        delay(1000);
    }
}
Serial.println("mDNS responder started");

```

```

/*return index page which is stored in serverIndex */
server.on("/", HTTP_GET, []() {
    server.sendHeader("Connection", "close");
    server.send(200, "text/html", loginIndex);
});
server.on("/serverIndex", HTTP_GET, []() {
    server.sendHeader("Connection", "close");
    server.send(200, "text/html", serverIndex);
});
/*handling uploading firmware file */
server.on("/update", HTTP_POST, []() {
    server.sendHeader("Connection", "close");
    server.send(200, "text/plain", (Update.hasError()) ?
    "FAIL" : "OK");
    ESP.restart();
}, []() {
    HTTPUpload& upload = server.upload();
    if (upload.status == UPLOAD_FILE_START) {
        Serial.printf("Update: %s\n", upload.filename.c_str());
        if (!Update.begin(UPDATE_SIZE_UNKNOWN)) { //
start with max available size
            Update.printError(Serial);
        }
    } else if (upload.status == UPLOAD_FILE_WRITE) {
        /* flashing firmware to ESP*/
        if (Update.write(upload.buf, upload.currentSize) !=
upload.currentSize) {
            Update.printError(Serial);
        }
    } else if (upload.status == UPLOAD_FILE_END) {
        if (Update.end(true)) { //true to set the size to the
current progress
            Serial.printf("Update Success: %u\nRebooting...\n",
upload.totalSize);
        } else {
            Update.printError(Serial);
        }
    }
});
server.begin();
}
void loop(void) {
server.handleClient();

```

```

delay(1);

//loop to blink without delay
unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= interval) {
  // save the last time you blinked the LED
  previousMillis = currentMillis;
  // if the LED is off turn it on and vice-versa:
  ledState = not(ledState);
  // set the LED with the ledState of the variable:
  digitalWrite(led, ledState);
}
}

```

[View raw code](#)

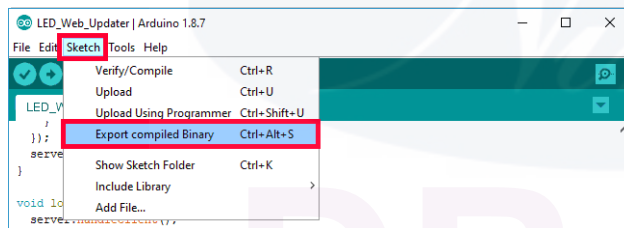
As you can see, we've added the "blink without delay" code to the OTAWebUpdater code, so that we're able to make updates later on.

After copying the code to your Arduino IDE, you should generate a .bin file.

Generate a .bin file in Arduino IDE

Save your sketch as LED_Web_Updater.

To generate a .bin file from your sketch, go to **Sketch > Export compiled Binary**



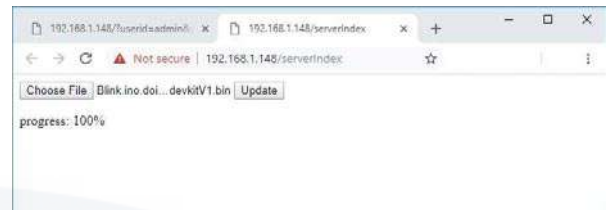
A new file on the folder sketch should be created. Go to **Sketch > Show Sketch Folder**. You should have two files in your Sketch folder: the .ino and the .bin file. You should upload the .bin file using the OTA Web Updater.



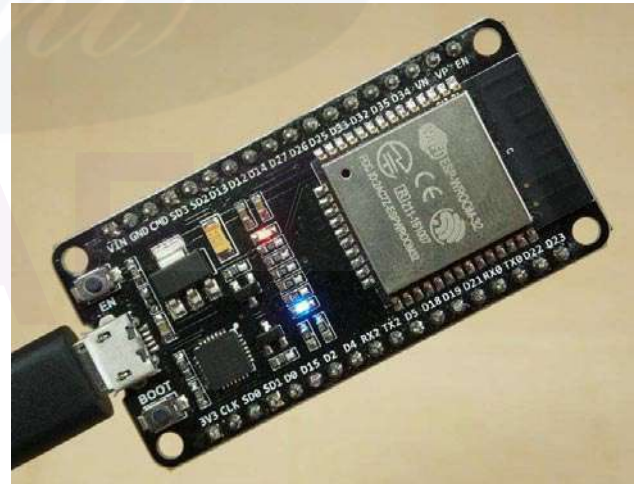
Upload a new sketch over-the-air to the ESP32

In your browser, on the ESP32 OTA Web Updater page, click the **Choose File** button. Select the .bin file generated previously, and then click **Update**.

After a few seconds, the code should be successfully uploaded.



The ESP32 built-in LED should be blinking.



Congratulations! You've uploaded a new code to your ESP32 over-the-air.

Smart City - Position the appropriate sensors and collect the information required in Smart City

Check the battery level and solar panel connection with sensor node

Objective:

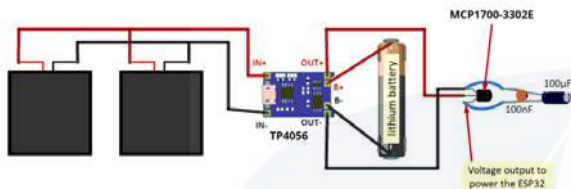
- to check the battery level and solar panel connection with sensor node.

Requirements:

- Arduino IDE
- Arduino
- Node mcu WiFi
- 12 volt battery

PROCEDURE:**ESP32 Solar Powered – Circuit Overview**

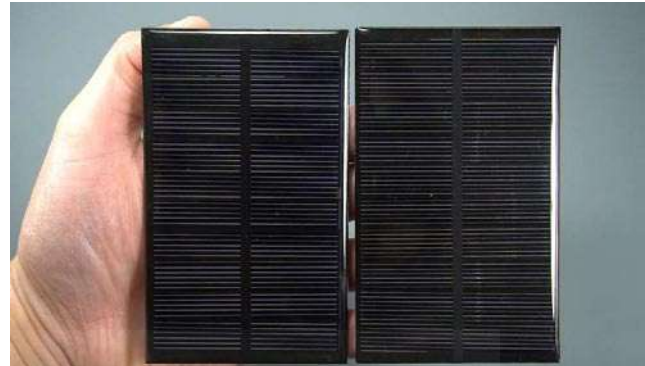
The following diagram shows how the circuit to power the ESP32 with solar panels works.



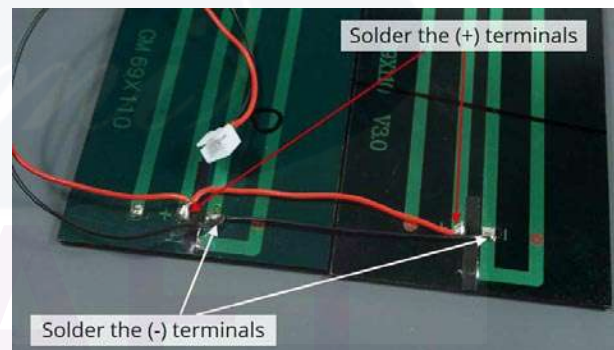
- The solar panels output between 5V to 6V with direct sun.
- The solar panels charge the lithium battery through the TP4056 battery charger module. This module is responsible for charging the battery and prevent overcharging.
- The lithium battery outputs 4.2V when fully charged.
- You need to use a low dropout voltage regulator circuit (MCP1700-3302E) to get 3.3V from the battery output.
- The output from the voltage regulator will power the ESP32 through the 3.3V pin.

Solar Panels

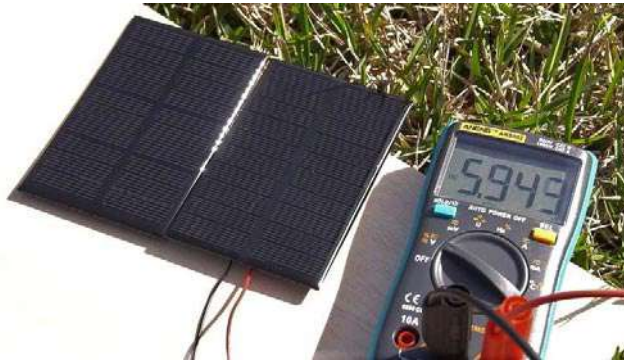
The solar panels we're using have an output voltage up to between 5V to 6V. If you want your battery to charge faster, you can use several solar panels in parallel. In this example we're using two mini solar panels as shown in the following figure.



To wire solar panels in parallel solder the (+) terminal of one solar panel to the (+) terminal of the other solar panel. Do the same for the (-) terminals. It may help taking a look at the following figure.



When wiring solar panels in parallel you'll get the same output voltage, and double the current (for identical solar panels). As you can see in the following figure, the solar panels output approximately 6V.



In the picture above, we're using the ANENG AN8002 multimeter, read our review here: [ANENG AN8002 Multimeter Review – Best Low Cost Multimeter?](#)

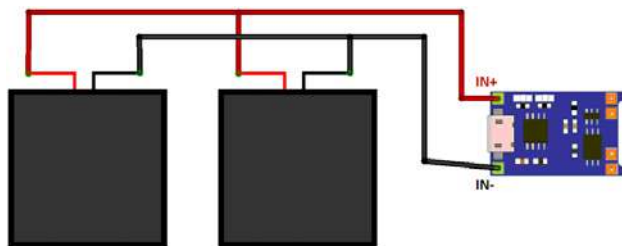
TP4056 Charger Module

The TP4056 lithium battery charger module comes with circuit protection and prevents battery over-voltage and reverse polarity connection.

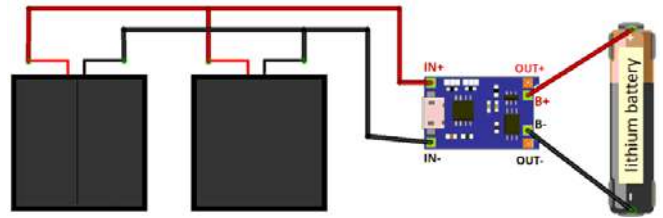
The TP4056 module lights up a red LED when it's charging the battery and lights up a blue LED when the battery is fully charged.



Wire the solar panels to the TP4056 lithium battery charger module as shown in the schematic diagram below. Connect the positive terminals to the pad marked with IN+ and the negative terminals to the pad marked with IN-.



Then, connect the battery holder positive terminal to the B+ pad, and the battery holder negative terminal to the B- pad.



The OUT+ and OUT- are the battery outputs. These lithium batteries output up to 4.2V when fully charged (although they have 3.7V marked in the label).

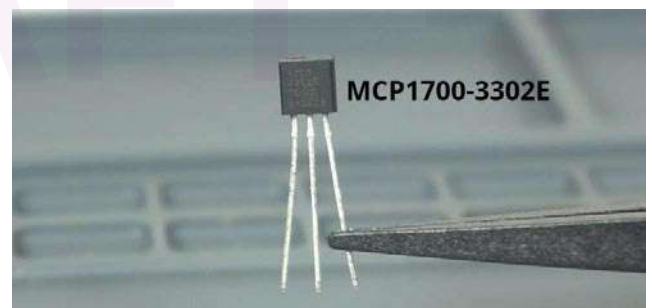


To power the ESP32 through its 3.3V pin, we need a voltage regulator circuit to get 3.3V from the battery output.

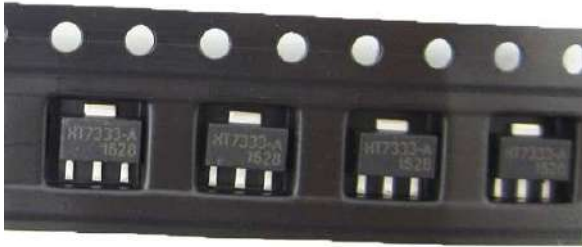
Voltage Regulator

Using a typical linear voltage regulator to drop the voltage from 4.2V to 3.3V isn't a good idea, because as the battery discharges to, for example 3.7V, your voltage regulator would stop working, because it has a high cutoff voltage.

To drop the voltage efficiently with batteries, you need to use a low-dropout regulator, or LDO for short, that can regulate the output voltage.



After researching LDOs, the [MCP1700-3302E](#) is the best for what we want to do. There is also a good alternative like the HT7333-A.



Any LDO that has similar specifications to these two are also good alternatives. Your LDO should have similar specs when it comes to output voltage, quiescent current, output current and a low dropout voltage. Take a look at the datasheet below.



MCP1700

Low Quiescent Current LDO

Features:

- 1.6 μ A Typical Quiescent Current
- Input Operating Voltage Range: 2.3V to 6.0V
- Output Voltage Range: 1.2V to 5.0V
- 250 mA Output Current for Output Voltages > 2.5V
- 200 mA Output Current for Output Voltages < 2.5V
- Low Dropout (LDO) Voltage
 - 178 mV Typical @ 250 mA for $V_{OUT} = 2.8V$
- 0.4% Typical Output Voltage Tolerance
- Standard Output Voltage Options:
 - 1.2V, 1.8V, 2.5V, 2.8V, 3.0V, 3.3V, 5.0V
- Stable with 1.0 μ F Ceramic Output Capacitor
- Short Circuit Protection
- Overtemperature Protection

General Description:

The MCP1700 is a family of CMOS low dropout (LDO) voltage regulators that can deliver up to 250 mA of current while consuming only 1.6 μ A of quiescent current (typical). The input operating range is specified from 2.3V to 6.0V, making it an ideal choice for two and three primary cell battery-powered applications, as well as single cell Li-Ion-powered applications.

The MCP1700 is capable of delivering 250 mA with only 178 mV of input to output voltage differential ($V_{OUT} = 2.8V$). The output voltage tolerance of the MCP1700 is typically $\pm 0.4\%$ at $+25^{\circ}C$ and $\pm 3\%$ maximum over the operating junction temperature range of $-40^{\circ}C$ to $+125^{\circ}C$.

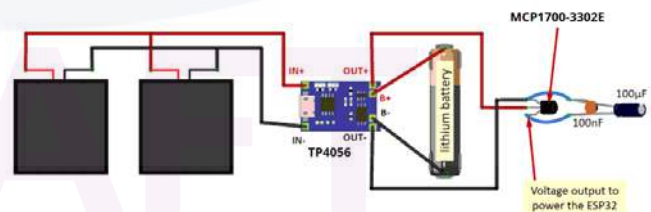
Output voltages available for the MCP1700 range from 1.2V to 5.0V. The LDO output is stable when using only 1 μ F output capacitance. Ceramic, tantalum or aluminum electrolytic capacitors can all be used for

Here's the MCP1700-3302E pinout: GND, VIN and VOUT pins.



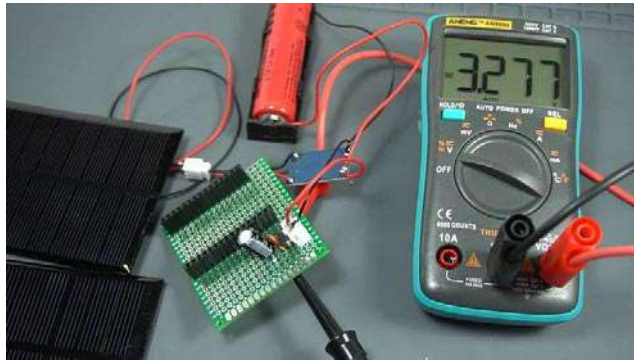
The LDOs should have a ceramic capacitor and an electrolytic capacitor connected in parallel to GND and Vout to smooth the voltage peaks. Here we're using a 100uF electrolytic capacitor, and a 100nF ceramic capacitor.

Follow the next schematic diagram to add the voltage regulator circuit to the previous setup.

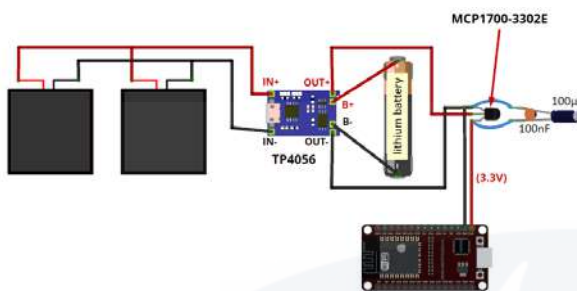


Warning: electrolytic capacitors have polarity! The lead with the white/gray strip should be connected to GND.

The Vout pin of the voltage regulator should output 3.3V. That is the pin that will power the ESP32 or ESP8266.



Finally, after making sure that you're getting the right voltage on the Vout pin of the voltage regulator, you can power the ESP32. Connect the Vout pin to the 3.3V pin of the ESP32 and GND to GND.

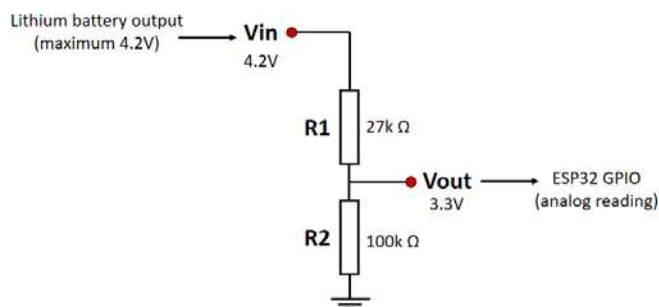


If you're using an ESP8266 instead, you can follow the same circuit. Wire the output of the MCP1700-3302E to the ESP8266 3.3V pin and GND to GND.

Battery Voltage Level Monitoring Circuit

When you have your ESP32 powered with batteries or solar powered as in this case, it can be very useful to monitor the battery level. One way to do that is reading the output voltage of the battery using an analog pin of the ESP32.

However, the battery we're using here outputs a maximum of 4.2V when fully charged, but the ESP32 GPIOs work at 3.3V. So, we need to add a voltage divider so that we're able to read the voltage from the battery.



The voltage divider formula is as follows:

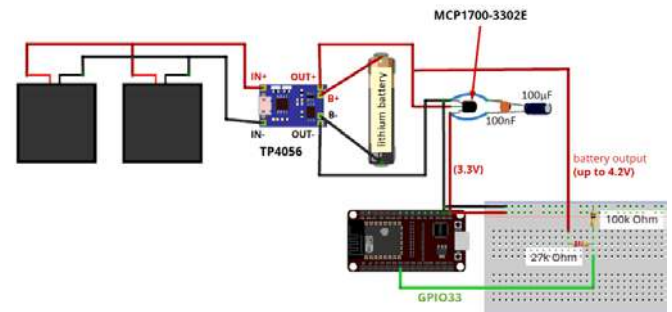
$$V_{out} = (V_{in} \cdot R_2) / (R_1 + R_2)$$

So, if we use $R_1 = 27k \text{ Ohm}$, and $R_2 = 100k \text{ Ohm}$, we get:

$$V_{out} = (4.2 \cdot 100k) / (27k + 100k) = 3.3V$$

So, when the battery is fully charged, the Vout outputs 3.3V that we can read with an ESP32 GPIO.

Add two resistors to your circuit as shown in the following schematic diagram.



In this case, we're monitoring the battery level through GPIO33, but you can use any other suitable GPIO. Read our [ESP32 GPIO guide](#) to learn which GPIOs are the best to use.

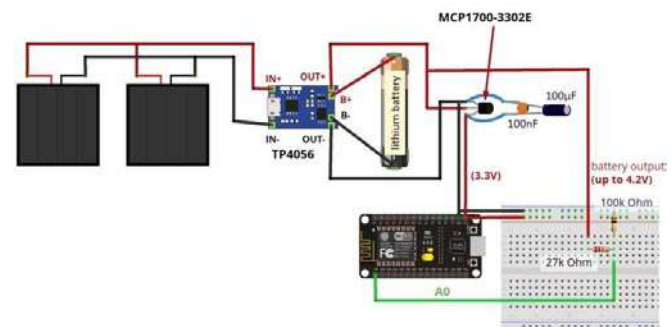
Finally, to get the battery level, you can simply read the voltage on GPIO33 using the `analogRead()` function in your code (if you're using Arduino IDE).

```
analogRead(33);
```

You can also use the `map()` function, to convert the analog values to a percentage:

```
float batteryLevel = map(analogRead(33), 0.0f, 4095.0f, 0, 100);
```

If you're using ESP8266, it just supports analog reading on the A0 pin. So, you need to wire the circuit as follows:



With the ESP8266 to read the analog value use:

```
analogRead(0);
```


Smart City - Position the appropriate sensors and collect the information required in Smart City

Control Variable rate controllers manually or automatically using an on- board computer guided by real GPS location

Objective:

- we will see how to interface Arduino Mega with NEO-6M GPS module.

Components Needed:**Hardware**

- Arduino Mega
- Neo-6M GPS module

Software

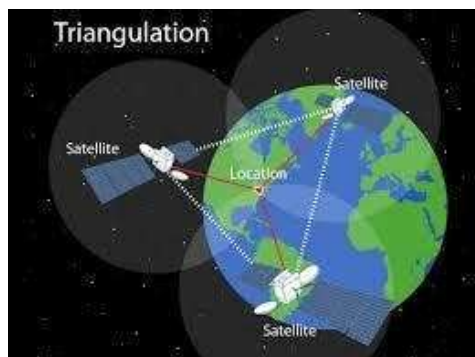
- Arduino IDE

PROCEDURE:**Information About GPS****What is GPS?**

The Global Positioning System (GPS) is a satellite-based navigation system made up of at least 24 satellites. GPS works in any weather conditions, anywhere in the world, 24 hours a day, with no subscription fees or setup charges.

How GPS works

GPS satellites circle the Earth twice a day in a precise orbit. Each satellite transmits a unique signal and orbital parameters that allow GPS devices to decode and compute the precise location of the satellite. GPS receivers use this information and trilateration to calculate a user's exact location. Essentially, the GPS receiver measures the distance to each satellite by the amount of time it takes to receive a transmitted signal. With distance measurements from a few more satellites, the receiver can determine a user's position and display it.



To calculate your 2-D position (latitude and longitude) and track movement, a GPS receiver must be locked on to the signal of at least 3 satellites. With 4 or more satellites, the receiver can determine your 3-D position

(latitude, longitude and altitude). Generally, a GPS receiver will track 8 or more satellites, but that depends on the time of day and where you are on the earth.

Once your position has been determined, the GPS unit can calculate other information, such as

- Speed
- Bearing
- Track
- Trip dist
- Distance to destination

What is signal

GPS satellites transmit at least 2 low-power radio signals. The signals travel by line of sight, meaning they will pass through clouds, glass and plastic but will not go through most solid objects, such as buildings and mountains. However, modern receivers are more sensitive and can usually track through houses. A GPS signal contains 3 different types of information

Pseudorandom code

It is an I.D. code that identifies which satellite is transmitting information. You can see which satellites you are getting signals from on your device's satellite page.

Ephemeris data

Ephemeris data is needed to determine a satellite's position and gives important information about the health of a satellite, current date and time.

Almanac data

Almanac data tells the GPS receiver where each GPS satellite should be at any time throughout the day and shows the orbital information for that satellite and every other satellite in the system.

Neo-6M GPS Module

The NEO-6M GPS module is shown in the figure below. It comes with an external antenna and does not come with header pins. So you will need to solder it.



Overview of NEO-6M GPS Module

NEO-6M GPS Chip

The heart of the module is a NEO-6M GPS chip from u-blox. It can track up to 22 satellites on 50 channels and achieves the industry's highest level of sensitivity i.e. -161 dB tracking, while consuming only 45mA supply current. The u-blox 6 positioning engine also boasts a Time-To-First-Fix (TTFF) of under 1 second.

One of the best features the chip provides is Power Save Mode (PSM). It allows a reduction in system power consumption by selectively switching parts of the receiver ON and OFF. This dramatically reduces power consumption of the module to just 11mA making it suitable for power sensitive applications like GPS wristwatch.

The necessary data pins of NEO-6M GPS chip are broken out to a "0.13 pitch headers. This includes pins required for communication with a microcontroller over UART.

Note:- The module supports baud rate from 4800bps to 230400bps with default baud of 9600.

NEO-6M GPS chip



Position Fix LED Indicator

There is an LED on the NEO-6M GPS Module which indicates the status of Position Fix. It'll blink at various rates depending on what state it's in

- No Blinking ==> means It is searching for satellites
- Blink every 1s – means Position Fix is found

LED Indicator



3.3V LDO Regulator

The operating voltage of the NEO-6M chip is from 2.7 to 3.6V. But, the module comes with MIC5205 ultra-low dropout 3V3 regulator from MICREL. The logic pins are also 5-volt tolerant, so we can easily connect it to an Arduino or any 5V logic microcontroller without using any logic level converter.



3.3 LDO Regulator

Battery & EEPROM

The module is equipped with an HK24C32 two wire serial EEPROM. It is 4KB in size and connected to the NEO-6M chip via I2C. The module also contains a rechargeable button battery which acts as a super-capacitor.

An EEPROM together with battery helps retain the battery backed RAM (BBR). The BBR contains clock data, latest position data (GNSS or bit data) and module configuration. But it is not meant for permanent data storage.

As the battery retains clock and last position, time to first fix (TTFF) significantly reduces to 1s. This allows much faster position locks.

Without the battery the GPS always cold-start so the initial GPS lock takes more time. The battery is automatically charged when power is applied and maintains data for up to two weeks without power.

Rechargeable button battery



HK24C32 serial EEPROM



Pinout

- GND is the Ground Pin and needs to be connected to GND pin on the Arduino.
- TxD (Transmitter) pin is used for serial communication.
- RxD (Receiver) pin is used for serial communication.
- VCC supplies power for the module. You can directly connect it to the 5V pin on the Arduino.



Arduino Mega

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.



Nimi
DRAFT

Smart City - Position the appropriate sensors and collect the information required in Smart City

Explore the interfacing of Zigbee module to create wireless sensor network

Objective:

- to explore the interfacing of ZigBee module to create wireless sensor network.

Requirements:

- ZigBee
- Arduino UNO
- Arduino IDE

PROCEDURE:

Introduction:

The unprecedented growth of industries and vehicular traffic have seriously affected the purity of clean air and environment. The world health organization announced that nearly 2.4 million people die every year due to this air pollution. So, the air pollution has become one of the greatest challenges for human health in the world[1]. The major components of air pollutants are particulate matter, SO₂, CO₂, CO and NO_x (i.e NO₂ and NO) etc.

Certain atmospheric pollutants react with each other and produce other pollutants called secondary pollutants. Though carbon dioxide (CO₂) is not considered as an air pollutant, it is considered here due to its importance in green house effect.

With the advancements in network technologies, the wireless sensor network (WSN) is found to be an efficient method of measuring and monitoring air pollutants.

WSN is one of the most significant technologies in 21st century. A WSN is a system comprised of radio frequency (RF) transceivers, sensors, microcontrollers and power sources. Recent advances in wireless sensor network technology have led to the development of low cost, low power multifunctional sensor nodes. There are various technologies available for WSN, viz., Wi-Fi, Bluetooth, and ZigBee etc. It is an undisputed fact that ZigBee has many advantages over other methods. To name a few, it operates in industrial scientific and medical (ISM) band of 2.4 GHz, and it is low cost, high reliability, very long battery life, high security, self-healing properties, large number of nodes supported, ease of deployment, guaranteed delivery, route optimization.

Multi-hop communication over the ISM band is also possible in WSN since it consumes less power than traditional single hop communication. A communication network is composed of many nodes each of which can transmit and receive data over communication links. The ZigBee network layer supports star, tree and mesh topologies. The ZigBee coordinator is responsible for initiating and maintaining the devices on the network and

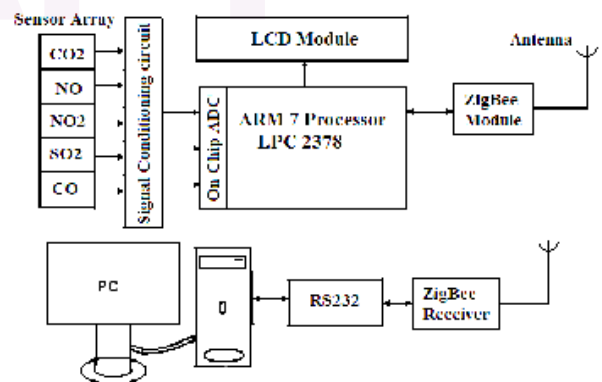
all other devices known as end devices, directly communicate with the ZigBee coordinator.

In mesh and tree topologies the ZigBee coordinator is responsible for starting the network and for choosing certain key network parameters but the network may be extended through the use of ZigBee routers. In tree networks routers move data and control messages through the network using a hierarchical routing strategy. The IEEE 802.15.4 standard defines three frequency bands of operation: 868 MHz, 916 MHz and the 2.4 GHz bands for ZigBee. 2.4 GHz bands are used in most of the commonly available wireless communication devices globally because of ISM band.

In addition, this band offers the highest data rate of 250Kbps and 16 channels between 2.4 GHz and 2.4835 GHz at the physical layer. Typical transmission distances are within the range from 30 meters in an indoor non line of sight environment to over 100 meters in a line of sight environment. Although defined as 25mW, transmit output power of ZigBee is within 10mW. Hence, the ZigBee modules employ dipole type antenna to increase gain of the antenna.

ZigBee standard is built on top of the IEEE 802.15.4 standard. The IEEE 802.15.4 standard defines the physical and MAC (Medium Access Control) layers for low rate wireless personal area network.

Wireless Network Module



ZigBee technology is known by its very low power consumption, which in combination with low power sensor circuits comprises a system that can operate for a long period of time. By using the ultra low power MSP430 microcontroller and the CC2480 ZigBee

coprocessor, it is possible to achieve several years of operation using common off-the-shelf batteries. The ZigBee module ez430-RF2480 by Texas instruments is used.

This is a complete wireless development tool that includes the ultra low power microcontroller MSP430 and the low power transceiver CC2480. It operates in the 2-4 GHz industrial scientific medical (ISM) free radio frequency band with 16 channels. It provides extensive hardware support for packet handling, data buffering, burst transmissions, clear channel assessment, link quality indication and wake on radio. There are five power down modes by which the power consumption can be minimized.

Sensor Array:

A simple multimode network is developed using ZigBee technology. The sensor array consists of sensors like CO sensor, CO2 sensor, VOC sensor and NOX sensors along with temperature and humidity sensors. The temperature is measured using DS 1820 sensor and

resistive type humidity sensor is used. Carbon monoxide is measured using TGS 2442- CO sensor, carbon dioxide is measured using TGS 4161-CO2 sensor, the presence of volatile organic compounds such as Benzene is measured using TGS 823- VOC sensor.

Nitrogen dioxide is measured using TGS 2106 sensor. The electrical response of the gas sensors is converted into voltage signals by using suitable signal conditioning circuits.

Central Server

The central server here is the personal computer (PC) with accessibility to the internet. The server (PC) is connected to the ZigBee modem using the RS232 communication standard. The data received from the ZigBee receiver is stored and displayed in the PC. A graphical user interface is developed, for the continuous analysis of the data.

The data displayed on the internet will help to check the real-time air pollutants level. For this a normal browser on any PC can be used.



Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Check the M2M Wireless Sensor Network (WSN) in IoT ZigBee router, end device and coordinator configuration

Objective:

- to check the M2M Wireless Sensor Network (WSN) in IoT ZigBee router, end device and coordinator configuration.

Requirements:

- ZigBee
- Arduino UNO
- Arduino IDE

PROCEDURE:

The ZigBee or ZigBee/IEEE 802.15.4 protocol is a specification created for wireless networking. It includes hardware and software standard design for WSN (Wireless sensor network) requiring high reliability, low cost, low power, scalability and low data rate. ZigBee-style self-organizing ad-hoc digital radio networks were conceived in the 1990s, but the IEEE 802.15.4-2003/ ZigBee specification was ratified on December 14, 2004. And only half year later the ZigBee Alliance announces availability of Specification 1.0 (on June 13, 2005).

A WSN consist in inexpensive wireless sensor which are capable of collecting, storing, processing environmental information, and communicating to neighbors nodes. For example, a WSN, at home can be used for light control, heating ventilation air conditioning (HVAC), security monitoring, and emergency event detection. And for industrial control can be used to improve the current manufacturing control system, detect unstable situations, control production pipelines, and so on.

ZigBee provides ultra low consumption and efficiency (thanks to the adaptable duty cycle, the low rate rates and the low coverage radio), and enable large scale networks for the WPN, making it one of the most convenient standards for this purpose.

The ZigBee Alliance

The ZigBee alliance objective is to work on the interoperability issues of IEEE 802.15.4/ZigBee protocol stacks. There are three levels of membership: Adopter, Participant, and Promoter.

- **Adopter:** Offers access to final, approved specifications, use of the Zigbee Member logo, participation in interoperability events, workshops and developers conferences.
- **Participant:** Offers full participation in all Alliance committees, work/task groups and member meetings. Participants earn voting rights in work groups and

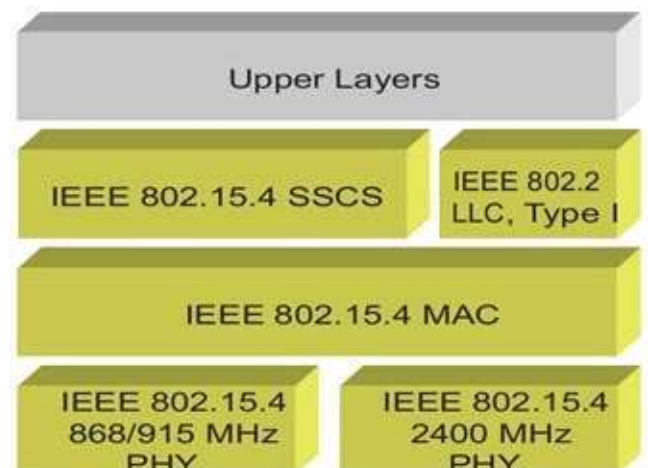
have early access to all Zigbee Alliance standards and specifications in development.

- **Promoter:** Offers automatic voting rights in all work groups, final approval rights on all standards and a seat on the Alliance Board of Directors.



The IEEE 802.15.4

Before start being more technical about ZigBee, let's introduce the IEEE 802.15.4 standard. Basically what it does is to specify the physical and data link layer protocols for low-rate wireless personal area network (LR-WPAN) and it was defined in 2003.

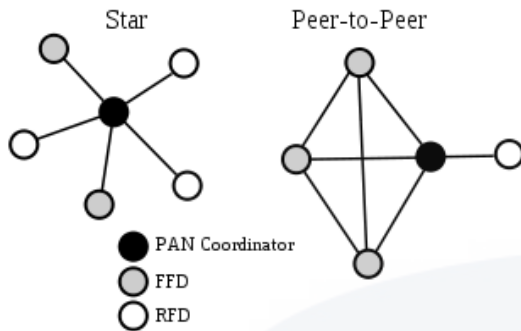


Devices in a LR-WPAN can be classified as full function device (FFDs) and reduce function devices (RFDs). One device is designated as the PAN coordinator which is the responsible for maintaining the network and managing other devices

It also defines two types of network topologies: star topology and peer-to-peer topology.

The **star** pattern is structured, where the coordinator of the network will necessarily be the central node. **Peer-to-Peer** networks can form arbitrary patterns of connection, and their extension is limited by the distance between each pair of nodes.

It can operate on channel 868 MHz in Europe, 916 MHz in America and Australia and 2400 MHz worldwide, also some other were added.



The Physical layer (PHY)

Is the layer that provides the data transmission capabilities. There are three different states defined for the devices: Transmitting, receiving and sleeping. This allows to the device to save energy when duty cycles are defined and the device sleeps. It also characterize the link quality/strength of a received signal of a link - Link Quality Indication (LQI).

The Data Link Layer

Is the layer in charge of data transference between the nodes of the network. It introduces - on the MAC sublayer - the superframe structure. The superframe is defined by the time between two beacons send by the coordinator of the network. The superframe can be divided between active period and inactive period, this - together with the sleeping state- allows to save energy during the inactive periods.

The beacons are used also to synchronizing with other devices, announcing the existence of a PAN, and informing pending data in coordinators.

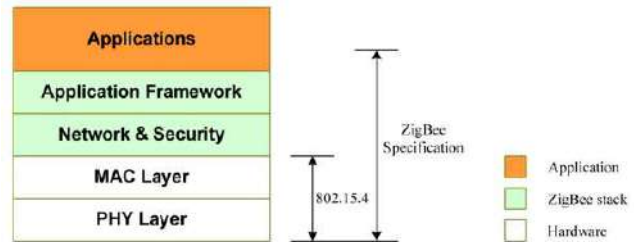
ZigBee - First release

The ZigBee protocol uses the IEEE 802.15.4 standard for the first two layers and adds the network and security, and framework protocol on top of it.

The network layer provides reliable and secure transmission among devices.

It defines three types of devices:

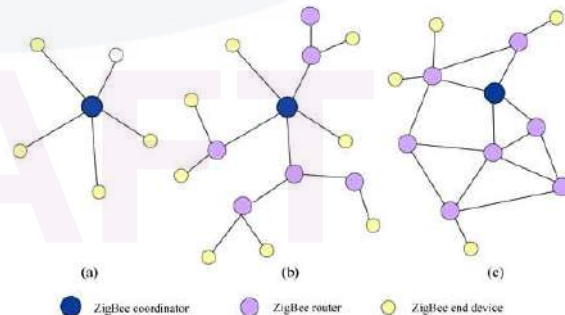
- **ZigBee Coordinator:** responsible for initializing, maintaining and controlling the network. There is one and only one per network
- **ZigBee Router:** connected to the coordinator or other routers. It can have zero or more children nodes. Participate in multi hop routing
- **ZigBee End Devices:** does not participate in routing



Also it defines three types of network topology: Star, tree and mesh.

In the star topology there is no presence of ZigBee Router. There is only a ZigBee Coordinator and zero or more ZigBee End Devices.

In the case of the tree, or cluster tree, topology the ZigBee Coordinator and also the ZigBee Router can announce beacons. The ZigBee Router can define dynamics duty cycle per cluster allowing devices to go to sleep not depending on the ZigBee Coordinator but on a more close ZigBee Router. The device would choose a slot and transmit its beacon to avoid collision. It also enables guaranty bandwidth (GTS).



Finally in the mesh topology there is no synchronization, therefore no beacon enable mode, meaning that the ZigBee Coordinator and the ZigBee Routers must be always on. And because of the dynamics of the network there is no bandwidth guaranty.

	Pros	Cons
Star	1. Easy to synchronize 2. Support low power operation 3. Low latency	1. Small scale
Tree	1. Low routing cost 2. Can form superframes to support sleep mode 3. Allow multihop communication	1. Route reconstruction is costly 2. Latency may be quite long
Mesh	1. Robust multihop communication 2. Network is more flexible 3. Lower latency	1. Cannot form superframes (and thus cannot support sleep mode) 2. Route discovery is costly 3. Needs storage for routing table

Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Identify the interfacing of Bluetooth module to create local sensor network

Objective:

- to identify the interfacing of Bluetooth module to create local sensor network.

Requirements:

- Mobile with Bluetooth
- Arduino IDE
- Arduino UNO

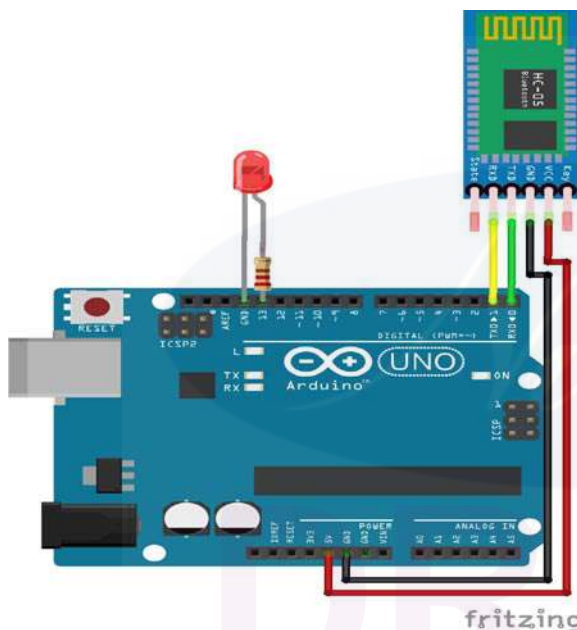
PROCEDURE:

Note: Don't Connect RX to RX and TX to TX of Bluetooth to Arduino you will receive no data, Here TX means Transmit and RX means Receive

How Does it Work?

HC 05/06 works on serial communication. Here the android app is designed sending serial data to the Bluetooth module when certain button is pressed. The Bluetooth module at other end receives the data and send to arduino through the TX pin of Bluetooth module(RX pin of Arduino). The Code fed to Arduino checks the received data and compares. If received data is 1 the LED turns on turns OFF when received data is 0.

Open the serial monitor and watch the received data



Schematic

Let's Start Building

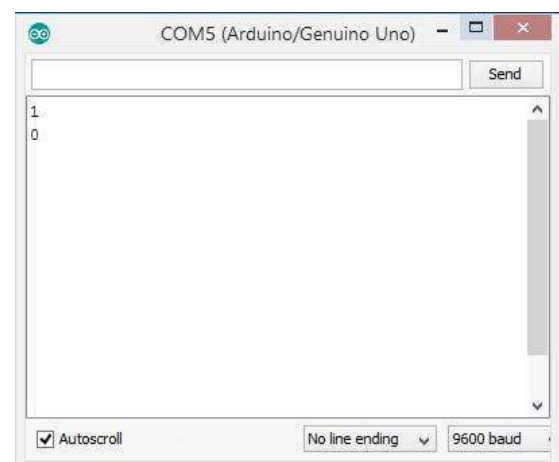
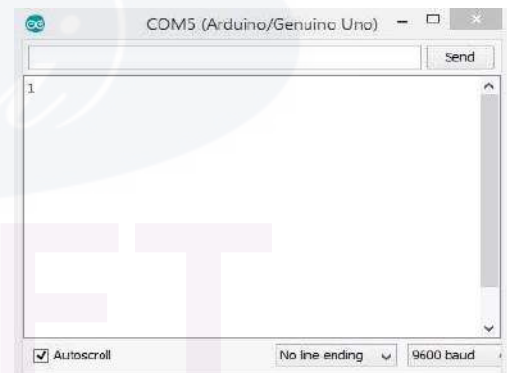
The circuit is so simple and small, there are only a few connections to be made

Arduino Pins		Bluetooth Pins
RX (Pin 0)	————>	TX
TX (Pin 1)	————>	RX
5V	————>	VCC
GND	————>	GND

Connect a LED negative to GND of Arduino and positive to pin 13 with a resistance valued between 220 Ω – 1K Ω .

And you're done with the circuit

LED turns ON



LED turns OFF

- You can download the android app from the app link given below:
- <https://github.com/Mayoogh/Arduino-Bluetooth-Basic>
- Pair your device with HC 05/06 Bluetooth module1) Turn ON HC 05/06 Bluetooth module2) Scan for available device3) Pair to HC 05/06 by entering default password 1234 OR 0000
- Install LED application on your android device
- Open the Application

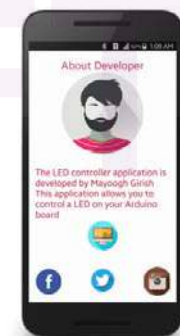


- After connecting successfully
- Press ON button to turn ON LED and OFF button to turn OFF the LED

- Press paired devices
- Select your Bluetooth module from the List (HC 05)



- Pairing screen



- Disconnect button to disconnect from Bluetooth module.

This is just basic tutorial on interfacing Bluetooth module with Arduino This project can be improved to a higher level like Home automation using a smartphone, Smartphone controlled robot and much more.

Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Explore the interfacing of GSM module to make node as a gateway

Objective:

- to explore the interfacing of GSM module to make node as a gateway.

Requirement:

- Raspberry Pi 3
- rhydoLABZ GSM/GPRS Modem
- Connecting Wires
- 5v 1A power source (For GSM modem)

PROCEDURE:

The GSM modem, phones provides a low cost, long range, wireless communication channel for embedded, microcontroller based applications that need connectivity rather than high data rates. Machinery such as industrial refrigerators, freezers, HVAC, monitoring and vending machines, vehicle service, warning system, security system, remote control etc. could benefit from being connected to a GSM modem.

This tutorial describes how to use an AVR to control a GSM modem using AT commands. The interface between modem and host is a textual protocol called AT-Commands. These commands enable GSM modem setup, dialing, text messaging etc. This particular application connects an AVR ATmega8 and SIM300, SIM900 module using a RS232 connection. Most GSM modems have same AT commands.

Contents

- Types of GSM Modem
- Points to consider while choosing GSM modem
- Interface to GSM modems.
- Implementation of AT-Command set.
- PDU string compression and decompression.
- SMS transmission, how to send and receive.

Types of GSM Modems

There are many GSM modems are available in market Simcom SIM300, SIM900 and Quectel M10. lets review on its features

Points to consider while choosing GSM Modem

There are may GSM modems available in market these

are SIM300, SIM900 and SIM900A, Quectel M10 is much reliable and a little bit costly. SIM300 is older modem now it is becoming absolute.

- Power supply requirement
- Available interface UART(TTL) or RS232
- Size
- Price
- Audio interface (MIC/SPK)

Power Supply Requirement

Most GSM modem require 4.2V to operate and they are compatible with 5V and 3.3V some modems are also available in 3.3V. depending on your microcontroller requirement choose correct modem. GSM modem requires more current while sending SMS, making call and during registering on network. **Remember always use 2 Amp Power supply.** Most new people make mistakes in selecting correct power supply this will cause problem or modem reset while sending SMS.

Available interface UART(TTL) or RS232

As we are using AVR ATmega8 we prefer TTL UART Interface. Most GSM Modem have RS232 and UART (TTL) Interface.

Few GSM Modem comes with USB Interface also, when you are interfacing GSM Modem with PC use USB Interface. It eliminates need of USB to serial converter and few modems are USB powered so no need of external power supply.

Size, Price

Size and price of each modem varies based on its manufacturer use of external components.

Audio Interface (MIC/SPK)

Audio interface is required if you are using GSM Modem to make calls or to interface with DTMF decoder for making IVRS system.

Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Apply IoT Gateway using WiFi and Ethernet

Objective:

- to apply IoT Gateway using WiFi and Ethernet.

Requirements:

- Arduino UNO
- Arduino Ethernet

PROCEDURE:

Arduino Ethernet Shield

Stand alone arduino board/microcontroller can not communicate with devices present on a network. Arduino uno does not have a proper hardware and software to communicate with devices present on network. In order to make arduino to communicate with devices presents on a network we need to attach an external module with arduino which could work as an intermediate between arduino and network. External module/peripheral must contain all the network configuration protocols(MAC, IP etc) and translators which can translate the network commands interpret-able by arduino.

Arduino Ethernet shield is an external board which can be mounted on arduino female headers and this Ethernet shield can now bring arduino over network. Arduino with Ethernet shield can talk/communicate with the devices/peripherals present not only on a home or office network but can go further over internet.



On Arduino Ethernet shield resides an intelligent unit which contains all the network configuration protocols(TCP, UDP etc) and IP stack. Arduino Ethernet shield has a RJ45 port to communicate with the switches and routers connected to a network. It also has a SD card slot. Data can be read and write to SD card. So arduino Ethernet shield not only provides internet access to arduino it also ads up an external portable EEPROM to arduino in the form of SD card.

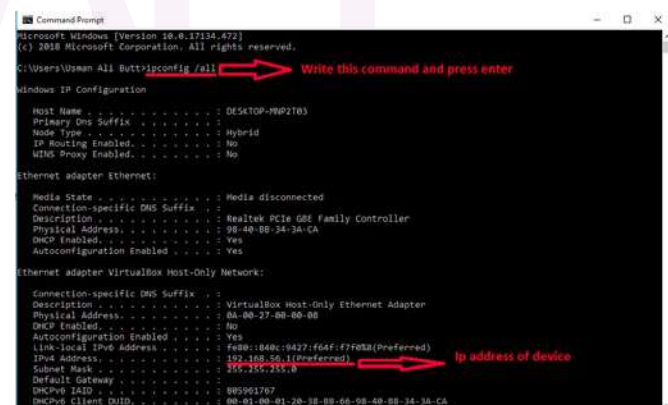
Arduino communicates with the Ethernet shield on SPI protocol. Arduino SPI pins 10, 11, 12 and 13 make connection and talks with the arduino Ethernet shield. Many arduino Ethernet libraries are present on internet which can be utilized to work with arduino and Ethernet shield. Arduino Ethernet Libraries are a great path to start with. Libraries contain arduino Ethernet board initialization and startup functions. We only need to write our main code rest of initialization and controlling part is done by the library(Thanks to arduino open source community).

Arduino Ethernet shield Web controlled Fan

Now lets put all the above in practical form and switch On/Off a fan using arduino uno and arduino Ethernet shield. We will connect our arduino with our home network and try to toggle arduino gpio(general purpose input output pin) using our laptop, cell phone or computer.

Every device on a network is identified by an IP and mac address. So if we want our arduino to connect with our home network we must first identify what IP our router is assigning to clients. The best and easiest way is to open from your computer and laptop Command Prompt. You can find cmd by pressing windows button on keyboard and then in search type cmd. Open cmd and enter ipconfig /all command. This command will display the IP that router assigned to your system.

See the below picture.



Arduino Ethernet shield IP assignment

Your router is assigning the IP to clients in series 192.168.56.x. X represents the client. So now you know which IP address series your router is working on. How to use this IP? In the project code i will explain it.

Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Check UART Communication, RS485 communication, I2C Protocol device Interfacing SPI Protocol device interfacing, Ethernet configuration, ZigBee interfacing, Wi-Fi – AP and router Interfacing

Objective:

- to Check UART Communication, RS485 communication, I2C Protocol device Interfacing SPI protocol device interfacing, Ethernet configuration, ZigBee interfacing, Wi-Fi AP and Router Interfacing.

Requirements:

- Arduino
- GSM Module
- GPS Module
- 16x2 LCD
- Power Supply
- Connecting Wires
- 10 K POT

PROCEDURE:

Arduino based Vehicle Tracker using GPS and GSM



Vehicle Tracking System using GPS and Arduino

In our previous article, we have learned about "[How to interface GPS module with Computer](#)" and How to make a [GPS updated Clock](#)". In this project we are going one step ahead with GPS and going to **track a vehicle using GPS and GSM**. This **Vehicle Tracking System** can also be used for Accident Detection Alert System, Soldier Tracking System and many more, by just making few changes in hardware and software.

Tracking of vehicle is a process in which we track the vehicle location in form of **Latitude and Longitude** (GPS coordinates). GPS Coordinates are the value of a location. This system is very efficient for outdoor application purpose.

This kind of **Vehicle Tracking System Project** is widely in tracking Cabs/Taxis, stolen vehicles, school/colleges buses etc.

GPS Module and Its Working:

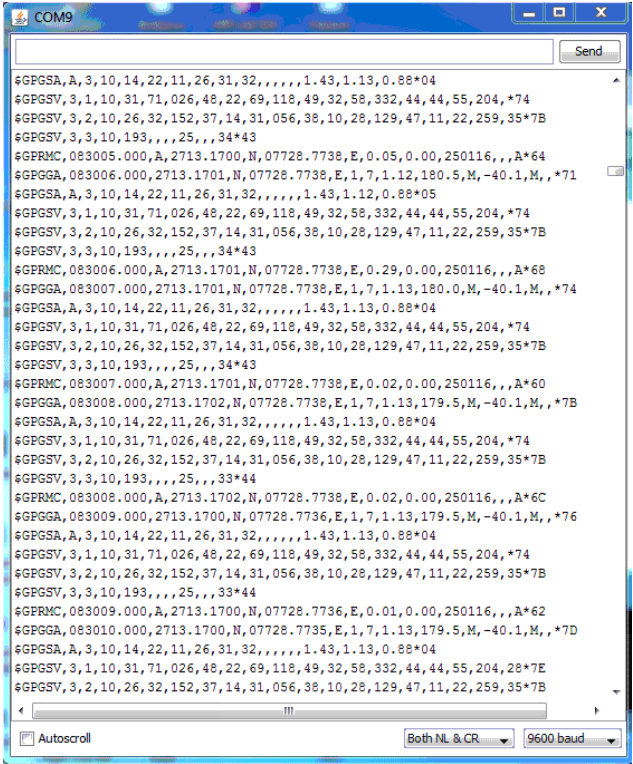
GPS stands for Global Positioning System and used to detect the Latitude and Longitude of any location on the Earth, with exact UTC time (Universal Time Coordinated). GPS module is the main component in our vehicle tracking system project. This device receives the coordinates from the satellite for each and every second, with time and date.



GPS module sends the data related to tracking position in real time, and it sends so many data in NMEA format (see the screenshot below). NMEA format consist several sentences, in which we only need one sentence. This sentence starts from **\$GPGGA** and contains the coordinates, time and other useful information. This **GPGGA** is referred to **Global Positioning System Fix Data**. Know more about [Reading GPS data and its strings here](#).

We can extract coordinate from \$GPGGA string by counting the commas in the string. Suppose you find \$GPGGA string and stores it in an array, then Latitude can be found after two commas and Longitude can be found after four commas. Now these latitude and longitude can be put in other arrays.

\$GPGGA,HHMMSS.SSS, latitude, N,longitude, E,FQ,NOS, HDP,altitude, M,height, M,,checksum data



Identifier	Description
\$GPGGA	Global Positioning system fix data
HHMMSS.SSS	Time in hour minute seconds and milliseconds format.
Latitude	Latitude (Coordinate)
N	Direction N=North, S=South
Longitude	Longitude(Coordinate)
E	Direction E= East, W=West
FQ	Fix Quality Data
NOS	No. of Satellites being Used
HPD	Horizontal Dilution of Precision
Altitude	Altitude from sea level
M	Meter
Height	Height
Checksum	Checksum Data

Below is the \$GPGGA String, along with its description:

\$GPGGA, 104534.000, 7791.0381, N,06727.4434, E,1,08,0.9,510.4, M,43.9,M,,*47

Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Identify the Wi-Fi module and lua script for data communication

Objective:

- to identify the Wi-Fi module and lua script for data communication.

Requirements:

- 32-bit RISC CPU
- 64 KiB of instruction RAM
- External QSPI flash – 512 KiB to 4 MiB
- IEEE 802.11 b/g/n Wi-Fi
- Integrated TR switch
- WEP or WPA/WPA2 authentication
- 16 GPIO pins
- SPI, I²C,
- I²S interfaces
- UART on dedicated pins

PROCEDURE:

NodeMcu ESP8266 V3 Lua CH340 Wifi Dev. Board is a fast leading edge low-cost WiFi technology. Modern high-level mature LUA based technology. It is an integrated unit with all available resources on board. It is super simple to complement your existing Arduino projects or any development board that has I/O pins available.

Modern Internet development tools such as Node.js can take advantage of the NodeMCU with the built-in API to put your idea on the fast track immediately. NodeMCU is built based on the mature ESP8266 technology to take advantage of the abundant resources available on the web.

NodeMCU has ESP-12 based serial WiFi integrated onboard to provide GPIO, PWM, ADC, I2C and 1-WIRE resources at your fingertips, built-in USB-TTL serial with super reliable industrial strength CH340 for superior stability on all supported platforms. This module is one of the cheapest available wifi-modules in the market. V3 or Version3 is the latest version of this module. This tutorial, however, will facilitate you to connect all the versions of ESP8266 NodeMcu, i.e V1, V2 or V3.

- **Arduino-like hardware IO** – Advanced API for hardware IO, which can dramatically reduce the redundant work for configuring and manipulating hardware. Code like Arduino, but interactively in Lua script.
- **Nodejs style network API** Event-driven API for network applications, which facilitates developers writing code running on a 5mm*5mm sized MCU in Nodejs style. Greatly speed up your IOT application developing process.

- **Development Kit** The Development Kit based on ESP8266, integrates GPIO, PWM, IIC, 1-Wire and ADC all on one board. Power your development in the fastest way combination with NodeMCU Firmware!

The NodeMcu is an open-source firmware and development kit that helps you to prototype your IoT product with few Lua script lines. The Development Kit based on ESP8266, integrated GPIO, PWM, IIC, 1-Wire and ADC all in one board.

Features:

- Uses CH340G instead of CP2102.
- Wireless 802.11 b/g/n standard
- WiFi at 2.4GHz, support WPA / WPA2 security mode
- Support STA/AP/STA + AP three operating modes
- Built-in TCP/IP protocol stack to support multiple TCP Client connections (5 MAX)
- Support UART / GPIO data communication interface
- Remote firmware upgrade (OTA)
- Support Smart Link Smart Networking
- ESP8266 has IO Pin
- Don't need to download resetting
- A great set of tools to develop ESP8266
- Lowest cost WI-FI
- FOR Arduino like hardware IO
- Greatly speed up your IOT application developing process
- Open-source, Interactive, Programmable, Low cost, Simple, Smart, WI-FI enabled
- WI-FI MCU ESP8266 integrated and easy to prototyping development kit.
- We provide the best platform for IoT application development at the lowest cost.
- NodeMCU has built-in USB-TTL serial with super reliable industrial strength CH340G for superior stability on all supported platforms.
- Advanced API for hardware IO, which can dramatically reduce the redundant work for configuring and manipulating hardware.
- Event-driven API for network applications, which facilitates developers writing code running on a 5mm*5mm sized MCU in Nodes style.

Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Explore the application of GPS satellites in Location Sensors

Objective:

- to explore the application of GPS satellites in Location sensors.

PROCEDURE:**Applications of GPS**

Up until now, we've looked at how you can use GPS receivers to tell you where you are, to navigate between points and to make digital maps of various features. But GPS isn't just used by civilians; it's also used by pilots, boat captains, farmers, surveyors, scientists and the military (just to name a few!).

While typical civilian handheld GPS receivers are usually accurate to about 5 meters, there are also very expensive, highly advanced GPS receivers that are capable of providing positions accurate to within a centimeter! These receivers have revolutions lots of industries, where highly accurate positioning is used for so many different tasks. The following sections provide a quick summary of how GPS is used in some industries.

Aviation

Almost all modern aircraft are fitted with multiple GPS receivers. This provides pilots (and sometimes passengers) with a real-time aircraft position and map of each flight's progress. GPS also allows airline operators to pre-select the safest, fastest and most fuel-efficient routes to each destination, and ensure that each route is followed as closely as possible when the flight is underway.

Marine

When high accuracy GPS is fitted to boats and ships, it allows captains to navigate through unfamiliar harbors, shipping channels and waterways without running aground or hitting known obstacles. GPS is also used to position and map dredging operations in rivers, wharfs and sandbars, so other boats know precisely where it is deep enough for them to operate.

Farming

Farmers rely on repeat planting season after season to maximize their crop productions. By putting GPS receivers on tractors and other agricultural equipment, farmers can map their plantations and ensure that they return to exactly the same areas when sowing their seeds in future. This strategy also allows farmers to continue working in low-visibility conditions such as fog and darkness, as each piece of machinery is guided by its

GPS position instead of visual references. High accuracy GPS is also used to map soil sample locations, allowing farmers to see where the soil is most fertile across individual fields or even entire farms!

Science

Scientists use GPS technology to conduct a wide range of experiments and research, ranging from biology to physics to earth sciences. Traditionally, when scientists wanted to understand where and how far animals roam, they had to tag animals with metal or plastic bands and then follow them to various locations to monitor their movement. Today, scientists can fit animals with GPS collars or tags that automatically log the animal's movement and transmit the information via satellite back to the researchers. This provides them with more detailed information about the animal's movements without having to relocate specific animals.

Earth scientists also use GPS technology to conduct a wide range of research. By installing high accuracy GPS receivers on physical features such as glaciers or landslips, scientists can observe and study both the speed and direction of movement, helping them to understand how landscapes change over time. Similarly, GPS receivers can be installed on solid bedrock to help understand very small and very slow changes in tectonic plate motion across the world.

Surveying

Surveyors are responsible for mapping and measuring features on the earth's surface and under water with high accuracy. This includes things like determining land boundaries, monitoring changes in the shape of structures or mapping the sea floor. Surveyors have historically required line-of-sight between their instruments in order to undertake such work, but the availability of high accuracy GPS receivers has reduced the need for this. GPS can either be setup over a single point to establish a reference marker, or it can be used in a moving configuration to map out the boundaries of various features. This data can then be transferred into mapping software to create very quick and detailed maps for customers.

Military

The GPS system was originally developed by the United

States Department of Defense for use by the US military, but was later made available for public use. Since then, GPS navigation has been adopted by many different military forces around the world, including the Australian Defense Force. Some countries have even decided to develop their own satellite navigation networks for use during wartimes.

Today, GPS is used to map the location of vehicles and other assets on various battlefields in real time, which helps to manage resources and protect soldiers on the ground. GPS technology is also fitted to military vehicles and other hardware such as missiles, providing them with tracking and guidance to various targets at all times of the day and in all weather conditions.



Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Check USB and Ethernet connectivity for data communication

Objective:

- to check USB and Ethernet connectivity for data communication.

Required Hardware:

- 1 x Arduino or 1 x Starter Kit for Raspberry Pi + Raspberry Pi
- 1 x GPRS+GPS Quad band Module for Arduino (SIM908)
- 1 x GPRS antenna
- 1 x GPS antenna
- 1 x SIM card
- 1 x PC

PROCEDURE:

Setting up a Direct Ethernet Connection using a USB to Ethernet Adapter

KB0166: Setting up a Direct Ethernet Connection using a USB to Ethernet Adapter

In some cases a network connection to your main router is not close or convenient, and the direct USB cable connection will not work properly.

A reliable and easy solution is to install a USB to Ethernet adapter. This creates a one port

network connection between the clock and a USB port on your PC or MAC.

NOTE: Some models may come with a driver CD. Make sure to follow the manufacturer's instructions

to install the drivers first.

Solution:

1. Connect the Ethernet cable to the port on the bottom of the clock and the port on the adapter.
2. Plug the USB connection of the adapter into your computer. Windows should recognize the connection and the Ethernet port connection lights should activate underneath the clock.
3. If you are using Windows, go to the Command Prompt to verify the IP address of the new connection.

NOTE: Apple MAC users can select the Apple Icon\System Preferences\Network and the "USB Ethernet" option to

view the new network settings described below.

Using Windows 7, Vista, or Windows 10:

1. Click on the Windows Start icon on the task bar.
2. In the Search programs and files.. window, type: cmd
3. Press Enter on the keyboard to go to the Command Prompt.

Using Windows 8:

1. Click on the Search (magnifying glass) icon on the task bar located on the right side of the desktop.
2. In the Search Apps window, type in the letters: cmd
3. Press Enter on the keyboard to go to the Command Prompt.

At the Command Prompt type >ipconfig –all

Press Enter to review the Ethernet adapter Local Area Connection Settings:

In this example, the IP address of the computer is 169.254.241.227 and the subnet mask is 255.255.0.0

4. Assign the IP Address and Subnet in your Clock.
 - a. Choose an available IP address.
 - b. The first 9 digits of the clock's IP address must match your computer's IP address, exactly.
 - c. The last 3 digits of your clock's IP address must be different.

Using the example above, we will assign the clock's IP address to: 169.254.241.230.

- a. Enter into Supervisor Mode at your employee time clock terminal.
- b. Press the padlock key on the time clock.
- c. Enter in your security code – the default security code is 00 00 00.
- d. Using the Arrow keys, scroll through the options until you find the option View IP Address – press Enter.
- e. The current IP address of the clock will appear. Press Enter.
- f. Enter the desired IP address including leading zeroes if needed. (For example .076 etc.)

The IP address in this Example is: 169.254.241.230

- g. The clock will display IP Address Set Successfully.
- h. Enter the Subnet Mask as 255.255.000.000
- i. The clock will display Subnet Mask Set Successfully.

- j. Enter the Gateway as 000.000.000.000.
- k. The clock will display Gateway Address Set Successfully.
- l. Enter the DNS as 000.000.000.000
- m. The clock will display DNS Address Set successfully.
- n. Press Clear to exit out of Supervisor Mode.
- o. To log into the time clock open a web browser on your computer, and in the address bar type in the IP address of the clock and press Enter.



Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Create a combine sensor appropriate for local climate monitoring

Objective:

- to create a combine sensor appropriate for local climate monitoring.
-

Requirements:

- Temperature sensor
- Humidity sensor
- GPS
- Arduino UNO

PROCEDURE:**Introduction:**

Internet of Things (IoT) is adding value to products and applications in the recent years. The connectivity of the IoT devices over the network has widely reduced the power consumption, robustness and connectivity to access data over the network. IoT is powering many frontiers of industries and is seen as a promising technology to take Big Data Analytics to a level higher. Weather monitoring system as a module is an issue among IoT research community and it has been widely addressed. A new weather monitoring system is developed using various sensors connecting to Raspberry Pi. The implementation and data visualization on the data collected are discussed in this paper in detail. Weather parameters like temperature, humidity, PM 2.5 and PM 10 concentrations and Air Quality Index (AQI) are monitored and visualized in graphical means using the Raspberry Pi as server and data accessed over the intranet or internet in a specified subnet or world wide web. The data visualization is provided as result and proves to be a robust framework for analyzing weather parameters in any geographical location studying the effect of smog and PM 2.5 concentration.

an implementation of weather monitoring system using Internet of Things (IoT). A raspberry pi based implementation is proposed to monitor PM2.5, PM 10, temperature, humidity and the air quality index (AQI) of the Particulate Matter pollutants available. Internet of Things is playing a leading role in providing solutions to many applications with the support of software, internet and embedded systems. There are various IoT devices available in the market ranging from micro controllers to microprocessors. The microprocessors in IoT are normally ARM processors like Raspberry Pi and Intel Edison. There are many technologies developed for weather monitoring using IoT devices.

In countries like India , China and Thailand smog is a natural phenomenon affecting human lifestyle and daily routine of those cities. The implementation done is based on the health issues faced by the people in Thailand. Studies have been done to analyze the PM 2.5 concentrations in Thailand for more than a decade from all over Thailand. The data analyzed by states that the predictive analytics of PM 2.5 in Bangkok, Thailand using linear regression says that the parameters of various parameters of timestamp, temperature and humidity are responsible for the concentration of PM 2.5. The data of PM 2.5 was taken from Berkeley Earth database and temperature and other weather parameters are populated using API connectivity to a local weather station in Weather Channel. The data is not populated from the same geographic location. To address these issues, an IoT based weather monitoring system is designed and developed at Thai-Nichi Institute of Technology.

Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Use signals from GPS satellites to determine latitude, longitude and altitude to within feet by Location Sensors for precise positioning

Objective:

- to use signals from GPS satellites to determine latitude, longitude and altitude to within feet by Location sensors for precise Positioning.

Requirements:

- Arduino UNO
- GPS receiver

PROCEDURE:

Location Sensors use signals from GPS satellites to determine latitude, longitude, and altitude to within feet. Three satellites minimum are required to triangulate a position. Precise positioning is the cornerstone of precision agriculture. GPS integrated circuits like the NJR NJG1157PCD-TE1 are a good example of location sensors.

Optical Sensors use light to measure soil properties. The sensors measure different frequencies of light reflectance in near-infrared, mid-infrared, and polarized light spectrums. Sensors can be placed on vehicles or aerial platforms such as drones or even satellites. Soil reflectance and plant color data are just two variables from optical sensors that can be aggregated and processed. Optical sensors have been developed to determine clay, organic matter, and moisture content of the soil. Vishay, for example, offers hundreds of photodetectors and photodiodes, a basic building block for optical sensors.

Electrochemical Sensors provide key information required in precision agriculture: pH and soil nutrient levels. Sensor electrodes work by detecting specific ions in the soil. Currently, sensors mounted to specially designed “sleds” help gather, process, and map soil chemical data.

Mechanical Sensors measure soil compaction or

“mechanical resistance.” The sensors use a probe that penetrates the soil and records resistive forces through use of load cells or strain gauges. A similar form of this technology is used on large tractors to predict pulling requirements for ground engaging equipment. Tensiometers, like Honeywell FSG15N1A, detect the force used by the roots in water absorption and are very useful for irrigation interventions.

Dielectric Soil Moisture Sensors assess moisture levels by measuring the dielectric constant (an electrical property that changes depending on the amount of moisture present) in the soil. Airflow Sensors measure soil air permeability. Measurements can be made at singular locations or dynamically while in motion. The desired output is the pressure required to push a predetermined amount of air into the ground at a prescribed depth. Various types of soil properties, including compaction, structure, soil type, and moisture level, produce unique identifying signatures.

Agricultural Weather Stations are self-contained units that are placed at various locations throughout growing fields. These stations have a combination of sensors appropriate for the local crops and climate. Information such as air temperature, soil temperature at various depths, rainfall, leaf wetness, chlorophyll, wind speed, dew point temperature, wind direction, relative humidity, solar radiation, and atmospheric pressure are measured and recorded at predetermined intervals. This data is compiled and sent wirelessly to a central data logger at programmed intervals. Their portability and decreasing prices make weather stations attractive for farms of all sizes.

Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Operate Global Positioning System (GPS) & Apply satellites broadcasting signals that allow GPS receivers to calculate their position

Objective:

- to operate Global Positioning System (GPS) and apply satellites broadcasting signals that allow GPS receivers to calculate their position.

Requirements:

- System
- Global Positioning System

PROCEDURE:

Global Positioning System or “GPS” was originally developed by the US Air Force to assist them with the military operations. GPS has made a significant impact on all positioning, navigation and monitoring applications in the world and over the course of the decades, it has evolved to a point where it is used in almost every aspect of our lives. The journey of GPS started with Sputnik in 1957 when Soviet scientists launched the very first satellite into space. The scientists at MIT discovered that the frequency of the radio signals emanated from the Russian satellite changed based on its location. The event also known as the Doppler Effect gave the scientists the idea that the location of satellites in orbit can be tracked from the ground by measuring the changes in the frequency of the radio signals. What seemed like a monumental loss in the technology race turned into a massive advantage in the hands of the accomplished scientists at MIT. They immediately started working on a new satellite system that would allow the US army to keep track of strategical locations and targets on the ground. TRANSIT was the first satellite navigation system built by the US Navy to locate US submarines carrying nuclear missiles in 1959. With the help of private tech companies like Aerospace Corporation, the military started working on another system; a series of satellites in orbit that send signals to receivers on the ground continuously. After 11 years of research and development, the US Air Force launched the first satellite of the 24-satellite navigation system called NAVSTAR.

An unfortunate incident that happened in 1983 changed the fate of the GPS technology forever. Korean Air Flight 007 wandered off course into Soviet airspace and was shot down shortly after. Following the aftermath of the event, President Ronald Reagan signed an executive order to make GPS available for civilian use to improve navigation and air safety once it is fully operational. Another significant event in the history of GPS is the removal of selective availability that was in effect since the first Gulf War by Bill Clinton in 2000. Since then, the US military expanded the size of the constellation to 31

satellites – 27 active and four spares in case of a malfunction – orbiting Earth and increased the accuracy of the GPS signals significantly.

Private tech companies and the DoD kept working on the development of GPS-based applications, and the result is the wide-range of devices we use in our daily lives.

So how does a GPS tracking device work? GPS units use a method called trilateration to determine the position, speed, and elevation of an object. GPS satellites broadcast their location and time constantly, and every satellite has a sophisticated atomic clock inside to be used for timing calculations. Tracking devices calculate the distance and time that it takes the GPS signals to travel to the surface from the satellites. The device requires signals from at least four satellites to provide an accurate location. With this in mind, the GPS satellite system was designed to have a minimum of four satellites in range for users with a clear view of the sky. Data from additional satellites further enhances the accuracy of the location. Each satellite is powered by solar energy and orbits the Earth twice a day, transmitting signals regularly.

As the GPS technology improved, various industries have started adopting the advanced tools it introduced to the market. Especially for the vehicle industry, GPS tracking systems have been a groundbreaking development. Today, tracking devices are being used for many purposes; mainly to monitor the movements of people, vehicles, and assets. Features such as real-time tracking, route history, geofence zones and event alerts allow users to provide personal safety measures for family members and enhance vehicle security. Fleet companies use the data provided by GPS and telematics devices to reduce operational costs and enforce better driving habits. On the other hand, concerned parents and family members employ the services of personal locators to keep an eye on their loved ones. The ability to track the location of a person in real-time and receive instant notifications of important events make personal tracking devices one of the best safety systems on the market.

When it comes to selecting the right GPS tracker, there are mainly two options you can choose from; battery-powered also known as portable trackers and hardwired tracking devices. Each type comes with its own set of advantages and disadvantages, and they can be used

for different purposes. Portable trackers are the go-to device for keeping track of a person or an object. Innovations in the field introduced devices with longer battery lives and access to a wealth of information. Portable devices can also be moved from one object to another easily and with optional magnetic cases; the placement options are limitless. Hardwired tracking devices are more suitable for tracking vehicles, and they need to be connected to a power source. The go-to place for a hardwired tracker is under the dashboard where it can be attached to the car battery discreetly. While they are not mobile, hardwired trackers are less susceptible to tampering and provide better accuracy when locating a target. Fleet companies prefer wired tracking devices; with additional sensors and access to the vehicle engine computer, these trackers provide the necessary information to eliminate the causes of fuel waste, get ahead of driver related problems and lower maintenance expenses.

GPS is an abbreviation for Global Positioning System. This term basically means a navigation structure which is used for the accurate location of an individual or a vehicle (Zap, 2008). The device which is usually used for this purpose is a GPS tracking system unit.

This unit can be used to tell the precise location of a person or any vehicle, whether it is stationary or moving. The GPS tracking unit records the data within the unit and then transmits it to a central server with the help of a satellite modem which directly fits inside the unit. The unit can be used to view the position of any person or vehicle in a real-time scenario (or later) whenever it is necessary to check the situation of any vehicle (Smith, 2006). The software used for this purpose is called GPS tracking software which helps to view the location of any target intended. Nowadays, mobiles having the GPS capability are being used to track the precise position of any person with the help of GPS software which is specifically designed for mobile systems.

There are basically three types of GPS tracking devices which are as follows:

- Data pushers
- Data pullers
- Data loggers

Data pushers

The data pushers are those types of GPS tracking units which are usually used for any personal tracking, asset tracking or the tracking of any type of vehicle. In case of personal as well as asset tracking, the location of any specific person or vehicle is sent to the server over short time intervals where the data is stored as well as examined closely (Zap, 2008). As far as the vehicle tracking is concerned, the GPS tracking unit not only sends the position of the vehicle to the server but is also able to analyze the speed at which the vehicle is in motion or any other data required. These units send all the important information to the server where all the required coordinates are analyzed clearly and then are stored in

the device. These GPS trackers could be used by anyone, including security agencies in case they want to track a potential suspect or can be used by parents to track the position of their kids and thus remain informed of their children's activities. Therefore, these types of GPS tracking unit devices are very important and useful.

Data pullers

The data pullers are employed for tracking the location same as data pushers. The main difference in both of these units is that the data pushers do not send any kind of information to the server, rather it is requested to send any required data. Also, the data pushers send the data at regular intervals, however, the data pullers are always on and could be used to retrieve data at any time.

Data loggers

The data loggers are used to store the position of any vehicle or person in its internal memory and do not send any data to a central server.

Therefore, all of the three types of GPS tracking units are very beneficial and helpful in tracking the exact location of any specified person, vehicle or fleet tracking.

Understanding GPS Technology

When many people begin researching and learning about the various products offered throughout the GPS tracking industry they are often left feeling overwhelmed. Tracking System understands that there are many terms and words associated with vehicle management and GPS monitoring technology that parents, businesses, and police agencies will likely hear for the first time as they begin their journey into the GPS and telematics industries. This glossary was designed to help bring simplicity and understanding to many of the terms, words, and phrases frequently used throughout the GPS monitoring industry, providing a resource for anyone interested in a complete and comprehensive overview of GPS-related definitions.

A-GPS

A-GPS is an acronym associated with the term "Assisted-Global Positioning System". Assisted GPS basically refers to the combination of both GPS satellite and cellular/wireless technologies. A-GPS is a term often associated with and used to describe how location-based data recorded onto a GPS tracker is transmitted.

Active

Active tracking is synonymous with the more commonly used term for real-time tracking. Active simply refers to having instant remote access to GPS tracking data.

Alkaline Battery

A form of battery often used in both the GPS Tracking Key (AAA alkaline battery) and GPS Tracking Key Pro (AA alkaline battery) models, alkaline batteries are the most commonly used form of battery in consumer electronics products. Known for having a long shelf life,

alkaline batteries are also viewed as being high-quality sources of power.

Atomic Clock

Although the name sounds pretty intimidating, an atomic clock is simply a way of calculating time through the periodic movements of atoms. Atomic clockwork is utilized by Global Positioning System units to provide a timing mechanism that allows GPS accuracy to be calculated with precision.

Auto Theft Recovery

The process of safely recovering a vehicle or mobile asset if it is stolen. Real-time GPS trackers are often used by people looking for an auto-theft recovery/prevention system because the devices can alert users if a vehicle is moved, and the technology provides 24/7 access to an automobile's location.

Auxiliary Power

Auxiliary power refers to the supplemental or secondary source of power that is used by a tracking unit or electronic device. This could refer to a GPS unit using an alternative source of power such as a cigarette lighter adapter to power the system rather than the device's primary source of power, an internal battery system such as a lithium ion or alkaline.

Car Tracker

GPS vehicle management system that is used to monitor the movements of an automobile. Commonly used by parents to observe teen driving behaviours, businesses to monitor employee driving activity and police agencies to conduct surveillance.

Covert Surveillance

The behaviour or act of acquiring, observing or recording data or information related to the activities of another in a secret fashion. Private investigators and police departments have a fundamental need to gather information in a covert way, making vehicle tracking systems very popular among security professionals and law enforcement.

Drivers

A form of the computer program that has controlling power over a device or GPS unit. When GPS tracking devices are connected to a personal computer it may be necessary or required to load new drivers to ensure proper system operation.

Elderly Tracking

The process of monitoring and observing the driving habits and behaviours of a senior citizen in an effort to improve personal safety. Many families choose to hard-wire real-time tracking systems to the vehicles being operated by an elderly family member to ensure that the senior is driving safely or not becoming lost.

External GPS Antenna

An essential piece of equipment or hardware that is used

to acquire radio signals from GPS satellites, external GPS antennas are designed separately from the monitoring device. Although the external antenna is independent of the GPS unit, it allows the GPS hardware to capture GPS signals. External antennas are popular among businesses looking for some versatility and flexibility in regards to the placement of the tracking device.

Fleet Management

The process of overseeing and managing numerous mobile assets and vehicles of a single company in an effort to improve routing, driver safety, customer service and overall company efficiency. Having remote access to all vehicles operating in the field provides companies with a way to safeguard equipment, control employee routes more effectively and reduces the costs associated with fuel consumption.

Geo-Fence

Geo-fence, or geo-fencing, is an alert feature that is common among many real-time tracking systems. Essentially how the feature works is by allowing the user to set a virtual boundary or "fence" around a particular area on a map. The alert is programmed into the GPS unit, and when the object/asset being monitored enters or exits through that virtual boundary an alert is sent to the GPS user via email or cellular text message.

Geotagging/Geocoding

The process of linking, connecting or tagging forms of digital media such as photographs, video data, RSS feeds or even web pages with geographical data that includes longitude and latitude positions acquired through GPS tracking. This feature is popular among outdoor adventurers who enjoy documenting their journeys and uploading the information for friends and family.

Global Positioning System (GPS)

The heart and soul of both the GPS vehicle tracking and GPS navigational industries. The most simplistic way to describe the Global Positioning System would be as an organized network of satellites orbiting the planet that were designed and are controlled by the U.S. Department of Defense.

Google Earth

Satellite image program designed and updated by Google that many GPS systems use as a primary source of showcasing recorded or live tracking data. The computer information program offers users access to historical imagery, street views, 3D buildings and more.

GPRS

GPRS are the acronyms for "General Packet Radio Services", which basically is in reference to a form of radio technology used by mobile communication devices and networks that allow the transfer of data.

GPS Navigation Device

The most commonly used form of GPS technology, navigation systems are electronic devices that are often

onboard vehicles and installed to an automobile's 12-volt system and placed on the front windshield. The GPS receives satellite signals and uses that data to accurately determine position. The devices then work with mapping programs to help drivers with routing or directions. This form of GPS technology is perfect for those travelling to new places, or who commonly become lost.

GPS Tracking System

An electronic device that is used to capture location-based data and store the information to internal memory (GPS receiver/passive), or transmit the information where it can be accessed live remotely via the personal computer (Active/real-time). Using GPS satellites to accurately determine the position of a vehicle, asset or person, a GPS tracking system will record an assortment of location-based data ranging from speeds travelled, addresses arrived/departed, maximum speed travelled, stops made, time en route and more. GPS tracking systems are often described as the ultimate fleet management solutions.

GPX Format

The data format that allows GPS users to exchange files, regardless if the systems are supported by conflicting operating systems or programming software. GPX format provides users with a unique way to write, read and download routes.

GSM

A term used to define "Global System Mobile Communications", which is currently the only form of cellular service offered throughout Europe. An international standard for digital mobile telecommunications systems, this form of technology was created to provide an alternative solution to analogue. Mobile communication devices and trackers that utilize GSM use SIM cards.

Hard-Wire

The connection between electrical wires/cables and a vehicle's or piece of equipment's circuit (Often 12-volt system in automobiles). This form of connection offers an alternative source of power that can keep a device powered by means other than an internal battery.

Interface

Reference to a point of interaction among a particular set of components. Applicable to both levels of hardware or software, an interface provides independent objects with the ability to communicate with each other via a connection, device or another form of a system. Past-Track monitoring software interfaces with Google Earth software, allowing a GPS user to view recorded GPS data over the satellite image program.

Internal Antenna

A piece of hardware that is essential in acquiring GPS signals so a system can calculate data. Unlike an external antenna, internal antennas are designed into the GPS

unit itself. This engineering design allows tracking devices increased portability.

Lithium Battery

One of the newest forms of battery technology, lithium-ion batteries provide long-lasting power and are much lighter than alkaline batteries, making them much more efficient for high-tech systems. Lithium batteries have a long shelf-life, and most importantly are not susceptible to leaking like alkaline batteries.

NMEA

Acronyms for the "National Marine Electronics Association", a unifying force behind the electronics industries that provides a file extension format that is an industry standard. Initially developed to aid in marine electronic equipment in the 1908s, the technology is now common among GPS technological systems.

Operating System (OS)

Software consisting of multiple data and programs that basically operates on computers and manages hardware. Some of the most common operating systems people utilize every day on their personal computers include Microsoft Windows programs and Mac OS X. However, Google has been in the process of creating an OS that many people believe could be one of the efficient.

Passive Tracking System

Commonly labelled as data loggers or receivers, passive trackers record GPS data and store that information in internal memory for review at a later time. The data from a passive tracker is manually downloaded via USB port or download cable, resulting in the system being much more cost-effective and not having monthly service fees. The GPS Tracking Key and GPS Tracking Key Pro are the two most popular passive tracking systems among consumers and businesses.

RFID

Radio Frequency Identification: A form of technology that utilizes radio frequency tags that are connected to objects that will transmit information or data to a receiver.

Real-Time Tracking

A form of GPS tracking that allows users to remotely access GPS data live as it happens. The real-time tracking unit receives GPS signals and then transmits that information via wireless networks, allowing users to have instant access to the location of a person, vehicle, asset or object. This form of GPS tracking is commonly used in fleet management and auto-theft recovery applications.

Sampling Rate

The frequency or rate in which position updates are acquired and stored in the internal memory of a GPS receiver. Think of the sampling rate like "breadcrumbs" strong across a path, independent pieces of data that are used in conjunction with one another to provide GPS users with a way to monitor travels.

Satellite

Both a natural (When planets revolve around a star such as a sun) and unnatural phenomenon (Object created by man such as a space probe) where a particular object revolves around another object in regular intervals. Satellites are used for a variety of communication purposes.

Short Message Service (SMS)

A form of technology that is similar to that used by paging systems, SMS provides an avenue to send short text messages electronically to mobile communication devices such as cellular phones. Although the amount of data is often limited by a set number of characters, it is one of the most efficient ways to transmit data and alerts from a real-time tracking device to a user's personal cell phone.

Teen Tracking

The process of monitoring the driving habits of teenagers to improve teen safety and overall road safety for everyone sharing the highways and roadways. With many teens having a tendency to drive at dangerous speeds, and motor vehicle accidents leading the list of causes of premature death among teenagers, parents are utilizing both real-time tracking and passive monitoring to observe teen driving behaviours.

Telematics

Telematics basically refers to the technology of transmitting, receiving and storing data or information between telecommunication systems. Also referred to as telemetry, telematics plays a significant role in the GPS tracking industry because the technology is often used to describe the use of GPS vehicle management or navigational systems since the devices integrate

computer and wireless technologies along with mobile communication systems for data transfer and reception.

Triangulation

The sophisticated process of accurately determining the exact position of a GPS receiver operating on Earth by using a mathematical system that measures distance via relative space between three GPS satellites orbiting in space. When a GPS system is placed in an optimum location where it has a direct line of sight to the sky, the process of triangulation results in GPS tracking data is accurate within 2.5 meters.

USB Port

Universal Serial Bus Ports are referred to as a standardized electrical point of connection on a personal computer that provides an avenue interface to numerous other electronic equipment such as printers, scanners, keyboards and more. Passive vehicle tracking devices such as the GPS Tracking Key have a USB drive engineered into the hardware, allowing users to connect the GPS system directly to a personal computer USB port to manually download GPS tracking data.

WAAS

Established in the mid-1990s by the Department of Transportation, Wide Area Augmentation Services (WAAS) refers to a system that was created to boost accuracy and the availability of both GPS navigation and tracking data. Ground stations were developed and strategically placed throughout the country to monitor and observe GPS satellite signals, accounting for any gaps in service or errors, resulting in WAAS-enabled GPS receivers having a superior accuracy of location-based data and position.

Water Resistant

One common misconception is that water resistant is the same as waterproof. IT IS NOT! Water resistant refers to a limited ability to keep moisture off hardware, circuitry and GPS components. People looking to use a car tracker that is water resistant on the outside of their vehicle should take additional precautions to safeguard their GPS tracking unit from the elements.

Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Analyze the vehicle speed using tracking system

Objective:

- to analyze the vehicle speed using tracking system.

Requirement:

- RFID
- Bluetooth low energy
- Low energy wireless IP networks
- ZigBee
- Z-wave
- LTE advance
- Wi-Fi direct

PROCEDURE:

Internet of things in vehicle tracking system (VTS) currently works on 3 main technologies:

- Radio-frequency identification, or RFID,
- Global Positioning System, or GPS and
- On-board diagnostics, or OBD II
- RFID helps control and track products
- GPS tracking solutions obtain real-time information on routes
- OBD II obtains information on vehicle maintenance and driving conditions.

GPS or the Global Positioning Satellite system has been around long before IoT devices came out.

While GPS data transmitted to a host over a mobile data connection is enough to track the moment of a vehicle in real time, IoT essentially makes Vehicle Tracking solutions a lot more efficient, economical, automated, and reliable.



This how an IoT based typical fleet tracking system works

- Location data collected by on-vehicle sensors, including GPS, OBD II sensors.

- Data from all sensors is periodically sent to the local gateway over a mobile data connection as the vehicle moves.
- Gateway sends information to network server where speed, location, direction and other variables are collected and send to a secure application server in the cloud
- Application server sends reports and alerts on movement and safety of the vehicle to the driver's or supervisor's mobile.

IoT Devices, also known as Connected Devices, are regular household and commercial appliances and gadgets, except you can connect and control them over a network: a personal network or the internet.

Yes, you can switch on your Air Conditioner with your smartphone from anywhere if both are connected to the internet. But that's only a smaller part of what Connected Devices can achieve.

Imagine how easy it would be to track things if everything could connect to the network. We can only anticipate the way it will revolutionize different industries.

Consumers are already the biggest beneficiary of this era of connected devices while businesses are still traversing how this technology breakthrough will improve their bottom line.

Growth is already significant in the IoT Sector. In 2015, [DHL HYPERLINK "https://www.dpdhl.com/content/dam/dpdhl/presse/pdf/2015/DHLTrendReport_Internet_of_things.pdf"](https://www.dpdhl.com/content/dam/dpdhl/presse/pdf/2015/DHLTrendReport_Internet_of_things.pdf) & [HYPERLINK "https://www.dpdhl.com/content/dam/dpdhl/presse/pdf/2015/DHLTrendReport_Internet_of_things.pdf"](https://www.dpdhl.com/content/dam/dpdhl/presse/pdf/2015/DHLTrendReport_Internet_of_things.pdf) Cisco conducted a study to understand the implications of Internet of Thing (IoT) Devices on logistics sector.

As per the study, there were 15 billion connected devices in 2015 and they will rise to 50 billion by 2020—a staggering rise of 333% since 2012 in the number of IoT devices.

The report further indicates that Logistics Industry would be the biggest beneficiary of IoT.

Benefits of IoT in fleet management

- IoT offers many new tools, like in-vehicle streaming camera, driver mobile apps etc. making communication easier and rapid.

- Scheduling, freight management, chauffeur and truck tracking, and effective routing are some advantages of Internet of Things.
- IoT supports third party APIs for traffic reporting, park-assist, average speed, driving etiquettes monitoring etc.
- Logistic companies can move to the cloud for their daily operations and track fleets anywhere, anytime and on any device.
- Collective data from IoT devices assist fleet management systems to get valuable insights into chauffeur behaviour, acts of law breaking by the driver, over-speeding etc.
- With the help of IoT and GPS tracking system, the fleet managers are able to access vital information such as the road conditions, weather, and the traffic. This access helps them in deploying their vehicles and drivers in an optimized way.
- Businesses can ensure that all their vehicles are working properly by setting alerts for maintenance, wear & tear of tire, deviations in temperature, and many more.

Moreover, you can run preventive maintenance with your vehicle management system that keeps your fleet in the best condition and safe.

It also prevents any sudden expenses or incurring delays due to breakdowns.

- A fleet will generate a huge amount of information such as the tracking speed, fuel consumption, distance traveled, utilization of vehicles, and many more.

To monitor all these data manually would be a tough task. Moreover, it's prone to human errors.

However, with the help of IoT and the real-time GPS tracking software, you can keep an eye on all these data and subsequently analyze it to enhance the overall efficiency of your fleet system.

- IoT automates various processes in fleet management. IoT enables companies to set up an automatic flow of logistics processes as the interconnected devices and work as part of an integrated process.
- Vehicle Idling pumps tons of gases into the air; it also results in usage of billions of gallons of fuel. We can stop this with IoT; we can send a real-time notification to drivers and monitor the idling time closely.
- IoT helps in tracking the performances of your vehicle components. It compares the present performance of that part with its standard performance. It sends notification if it finds any irregularity or drops in performance.
- Early adopters of IoT enjoys an absolute advantage over others due to two reasons.

First, it reduces the overall cost by optimum utilization of resources, and second, it helps in acquiring, sharing and acting on real-time data which results in quick and precise decisions.

- Proper application of IoT shrinks total costs by optimal use of resources and enhanced performance.
- Gathering, sharing, and acting on real time info helps a firm make rapid decisions and prompt improvements.

DRAFT

Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Use Geographical Information System consisting of computer software data base system used to input, store, retrieve, analyze and display in map like form, spatially referenced geographical information for more detailed analysis of city roads, transport and traffic etc

Objective:

- to use Geographical Information System consisting of computer software data base system used to input, store, retrieve, analyze and display in map like form, spatially referenced geographical information for more detailed analysis of city roads, transport and traffic etc.

Requirements:

- System having GPS
- GPS hardware drives
- GUI Graphical User Interface
- Database Management System DBMS

PROCEDURE:

DATA TYPES IN GEOGRAPHIC INFORMATION SYSTEMS (MAP DATA REPRESENTATION)

A Geographic information system stores two types of data that are found on a map- the geographic definitions of earth surface features and the attributes or qualities that those features possess. Not all systems use the same logic for achieving this. Nearly all, however, use one or a combination of both of the fundamental map representation techniques:

- Raster Based
- Vector Based

Vector Systems

Some Common Vector Data Formats

Name	Description
GBF/DIME (Geographic Base File/Dual Independent Map Encoding)	The data format developed by the US Bureau of the Census for digitally storing street maps to assist in the gathering and tabulation of data for the 1970 census, which was used again for the 1980 census
TIGER (Topologically Integrated Geographic Encoding and Referencing system)	Created by the US Bureau of the census as an improvement to the GBF/DIME and was used in the 1990 census. This format is known as TIGER/SDTS for the 2000 census.
DLG (Digital Line Graphs)	The format used for the USGS topographic maps
AutoCAD DXF (Data Exchange Format)	One of the file formats of AutoCAD and AutoCAD Map that has become de facto industry standard and is widely used as an export format in many GIS
IGDS (Intergraph Design System) DGN File	A proprietary file format for Intergraph CAD systems, which is also widely used in the mapping industry.

ArcInfo Coverage	The proprietary data format of ArcInfo; stores vector graphical data using a topological structure explicitly defining spatial relationships.
ArcInfoE00	The export format of ArcInfo.
Shapefiles	The data format for ArcView GIS that defines the geometry and attribute of geographically referenced objects by three specific files (i.e., a main file, an index file, and a database table.)
CGM (Computer Graphics Metafile)	An ISO standard for vector data format that is widely used in PC-based computer graphic applications.

Source: C. P. Lo, Albert. W. Yeung, *Concepts and Techniques of Geographic Information Systems*; PHI (2002), P. 87.

With vector representation, the boundaries or the course of the features are defined by a series of points that when joined with straight lines, form the graphic representation of that feature. The points themselves are encoded with a pair of numbers giving the x and the y coordinates. That is the information about points, lines and polygons is encoded and stored as a collection of x and y coordinates. The location of a point feature, such as a bore well/bore hole can be described by a single x, y coordinate. For example, the location of a point, which is 300 meters east, and 200 meters north of an origin (0, 0) would be stored as the x, y co-ordinate pair (300,200). Linear features, such as roads and rivers, can be stored as a collection of point coordinates. Polygonal features such as sales territories and river catchments can be stored as a closed loop of coordinates.

Raster Systems

With raster systems, the graphic representation of features and the attributes they possess are merged into unified data file. In fact, we typically do not define features at all. Rather, the study area is divided into a fine mesh of grid cells in which we record the condition or attribute of the earth's surface at that point. That is Raster data are stored as a matrix of pixels (Contraction of the term picture element) with each pixel location assigned a row and column designation.

Some Commonly Used Raster File Formats

Format Name	Description
BMP	The format used by bit map graphics in Microsoft windows application.
PCX	A proprietary bit map format of Zsoft that is supported by many image scanners.
TIFF (Tagged Image File Format)	A nonproprietary system-independent format designed as a non proprietary format for the storage of scanned images and the exchange of data between graphics packages. TIFF/JPEG, commonly referred to as TIFF 6.0, is an extension of the TIFF format.
GeoTIFF	An extension of the TIFF format that contains georeferencing information in its file header. GeoTIFF is developed and maintained as the result of the concerned effort of GIS software developers, commercial data suppliers, and data users to provide a publicly available and platform – interoperable standard for the support of raster geographic data in TIFF.

GIF (Graphic Interchange Format)	A cross platform format, proprietary to CompuServe that is widely used for the transmission of images on the world wide web.
JPEG (Joint Photographic Experts Group)	Another cross-platform format developed primarily for storage of photographic images and is also widely used for graphics on the world wide web
PNG (Portable Network Graphics)	A patent free raster format that is intended to replace the proprietary GIF format
GRID	The proprietary raster format used by ESRI (Environmental Systems Research Institute Inc.) in its software products such as ArcInfo and ArcView GIS

Source: C.P.Lo. Albert. W. Yeung. *Concepts and Techniques of Geographic Information Systems*; PHI (2002), P. 87.

Querying the GIS

GIS stores both spatial and non-spatial data in a database system, which links the two types of data to provide flexible and powerful ways of querying or asking about the data. In some cases we query by location For example “what land use is at this location?” In other cases we query by attribute -“What areas have high levels of carbon-di-oxide?” Some time we undertake simple queries such as the above and at other time we ask about complex compounds of conditions like- “Show all wetlands that are larger than 1 hectare and that are adjacent to industrial lands”

APPLICATIONS OF GEOGRAPHIC INFORMATION SYSTEMS

At a broad range of scales, map have become increasingly important as legal document that convey land ownership and jurisdictional boundaries, as tools to support decision making (For example, in urban planning).

GIS provides the ability to completely model utility networks, such as those supplying water, power and telecommunications to a large number of consumers. Such a system may operate at a variety of scales,

modeling service connections to consumers, service districts as well as detailed facilities inventories and layouts, such as transformers, valves conduits, and schematic diagrams.

What's in a GIS map?

Today we look for it to guide our decision-making through fuller control of our natural resources, a giant step in itself and more than ambitious enough in many respects. GIS has become a key element at many levels:

1. In customer support (to respond to service failure),
2. In maintenance and daily operations (To identify work requirements and assess inventories) and
3. In planning (to respond to projected needs).

Business Application of GIS

Business organizations operate in an ever changing and challenging environment, in which competitive forces require that information technology be exploited to the full. On widely cited model of business, the Value Chain Model by Porter identifies five primary business activities:

1. Inbound Logistics (Inputs)
2. Operations
3. Outbound Logistics (Outputs)
4. Marketing and sales and
5. Services.

GIS TOOLS

There are an extraordinary range of software packages that are presented as tools for developing IS applications. There are about 63 GIS software systems currently available. Most of these are sold as application software only or as turnkey systems that include hardware, software and user training. A few are offered as consulting services. Some of the products such as base networks or data files, for which the user must develop data input and retrieval routing, are limited in the form but broad in scope. There are a number of GIS software packages available today. Some of them are listed below:

1. AGIS	12.GIMMS	23. PMAP
2. ARC/INFO	13.GISIN	24. SICAD
3. ARCVIEW GIS	14. GRAMS	25. SPANS
4. ATLAS GRAPHICS	15. GRASS	26. STRINGS
5. CARIS-GIS	16. IDRISI	27. TIGRIS
6. EPPL 7	17. IGDS/DMRS	28.TOPOLOGIC
7. ERPAS	18.IMAGE	29. TRANSCAD
8. FMS/AS	19.MAPINFO	30. UFOSNET
9. GEO-Graphics	20.MIPS	31. USEMAP
10. GEOMEDIA	21.MGE (INTERGRAPH)	32. VANGO
11. GFIS	22. PC ARC/INFO	

Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Analyze data collected by GPS and supply it to user in usable format such as maps, graphs, charts or reports using suitable Computer Hardware and Software

Objective:

- To analyze data collected by GPS and supply it to user in usable format such as maps, graphs, charts or reports using suitable Computer Hardware and Software.

Requirements:

Hardware components

- Neo 6M GPS Module × 1
- ESP12E NodeMCU 1.0 × 1
- mini breadboard × 2

Software apps and online services

Thing Speak API

PROCEDURE:

WHAT DO WE MEAN BY COLLECTING DATA?

Essentially, collecting data means putting your design for collecting information into operation. You've decided how you're going to get information – whether by direct observation, interviews, surveys, experiments and testing, or other methods – and now you and/or other observers have to implement your plan. There's a bit more to collecting data, however. If you are conducting observations, for example, you'll have to define what you're observing and arrange to make observations at the right times, so you actually observe what you need to. You'll have to record the observations in appropriate ways and organize them so they're optimally useful.

Recording and organizing data may take different forms, depending on the kind of information you're collecting. The way you collect your data should relate to how you're planning to analyze and use it. Regardless of what method you decide to use, recording should be done concurrent with data collection if possible, or soon afterwards, so that nothing gets lost and memory doesn't fade.

Some of the things you might do with the information you collect include:

- Gathering together information from all sources and observations
- Making photocopies of all recording forms, records, audio or video recordings, and any other collected materials, to guard against loss, accidental erasure, or other problems
- Entering narratives, numbers, and other information into a computer program, where they can be arranged and/or worked on in various ways

- Performing any mathematical or similar operations needed to get quantitative information ready for analysis. These might, for instance, include entering numerical observations into a chart, table, or spreadsheet, or figuring the mean (average), median (midpoint), and/or mode (most frequently occurring) of a set of numbers.
- Transcribing (making an exact, word-for-word text version of) the contents of audio or video recordings
- Coding data (translating data, particularly qualitative data that isn't expressed in numbers, into a form that allows it to be processed by a specific software program or subjected to statistical analysis)
- Organizing data in ways that make them easier to work with. How you do this will depend on your research design and your evaluation questions. You might group observations by the dependent variable (indicator of success) they relate to, by individuals or groups of participants, by time, by activity, etc. You might also want to group observations in several different ways, so that you can study interactions among different variables.

There are two kinds of variables in research. An **independent variable** (the intervention) is a condition implemented by the researcher or community to see if it will create change and improvement. This could be a program, method, system, or other action. A dependent variable is what may change as a result of the independent variable or intervention. A **dependent variable** could be a behavior, outcome, or other condition. A smoking cessation program, for example, is an independent variable that may change group members' smoking behavior, the primary dependent variable.

WHAT DO WE MEAN BY ANALYZING DATA?

Analyzing information involves examining it in ways that reveal the relationships, patterns, trends, etc. that can be found within it. That may mean subjecting it to statistical operations that can tell you not only what kinds of relationships seem to exist among variables, but also to what level you can trust the answers you're getting. It may mean comparing your information to that from other groups (a control or comparison group, statewide figures, etc.), to help draw some conclusions from the data. The

point, in terms of your evaluation, is to get an accurate assessment in order to better understand your work and its effects on those you're concerned with, or in order to better understand the overall situation.

There are two kinds of data you're apt to be working with, although not all evaluations will necessarily include both. **Quantitative data** refer to the information that is collected as, or can be translated into, numbers, which can then be displayed and analyzed mathematically. **Qualitative data** are collected as descriptions, anecdotes, opinions, quotes, interpretations, etc., and are generally either not able to be reduced to numbers, or are considered more valuable or informative if left as narratives. As you might expect, quantitative and qualitative information needs to be analyzed differently.

QUANTITATIVE DATA

Quantitative data are typically collected directly as numbers. Some examples include:

- The frequency (rate, duration) of specific behaviors or conditions
- Test scores (e.g., scores/levels of knowledge, skill, etc.)
- Survey results (e.g., reported behavior, or outcomes to environmental conditions; ratings of satisfaction, stress, etc.)
- Numbers or percentages of people with certain characteristics in a population (diagnosed with diabetes, unemployed, Spanish-speaking, under age 14, grade of school completed, etc.)

Data can also be collected in forms other than numbers, and turned into quantitative data for analysis. Researchers can count the number of times an event is documented in interviews or records, for instance, or assign numbers to the levels of intensity of an observed event or behavior. For instance, community initiatives often want to document the amount and intensity of environmental changes they bring about – the new programs and policies that result from their efforts. Whether or not this kind of translation is necessary or useful depends on the nature of what you're observing and on the kinds of questions your evaluation is meant to answer.

Quantitative data is usually subjected to statistical procedures such as calculating the mean or average number of times an event or behavior occurs (per day, month, year). These operations, because numbers are "hard" data and not interpretation, can give definitive, or nearly definitive, answers to different questions. Various kinds of quantitative analysis can indicate changes in a dependent variable related to – frequency, duration, timing (when particular things happen), intensity, level, etc. They can allow you to compare those changes to one another, to changes in another variable, or to changes in another population. They might be able to tell you, at a particular degree of reliability, whether those changes are likely to have been caused by your

intervention or program, or by another factor, known or unknown. And they can identify relationships among different variables, which may or may not mean that one causes another.

QUALITATIVE DATA

Unlike numbers or "hard data," qualitative information tends to be "soft," meaning it can't always be reduced to something definite. That is in some ways a weakness, but it's also a strength. A number may tell you how well a student did on a test; the look on her face after seeing her grade, however, may tell you even more about the effect of that result on her. That look can't be translated to a number, nor can a teacher's knowledge of that student's history, progress, and experience, all of which go into the teacher's interpretation of that look. And that interpretation may be far more valuable in helping that student succeed than knowing her grade or numerical score on the test.

Qualitative data can sometimes be changed into numbers, usually by counting the number of times specific things occur in the course of observations or interviews, or by assigning numbers or ratings to dimensions (e.g., importance, satisfaction, ease of use).

The challenges of translating qualitative into quantitative data have to do with the human factor. Even if most people agree on what 1 (lowest) or 5 (highest) means in regard to rating "satisfaction" with a program, ratings of 2, 3, and 4 may be very different for different people. Furthermore, the numbers say nothing about why people reported the way they did. One may dislike the program because of the content, the facilitator, the time of day, etc. The same may be true when you're counting instances of the mention of an event, such as the onset of a new policy or program in a community based on interviews or archival records. Where one person might see a change in program he considers important another may omit it due to perceived unimportance.

Qualitative data can sometimes tell you things that quantitative data can't. It may reveal why certain methods are working or not working, whether part of what you're doing conflicts with participants' culture, what participants see as important, etc. It may also show you patterns – in behavior, physical or social environment, or other factors – that the numbers in your quantitative data don't, and occasionally even identify variables that researchers weren't aware of.

It is often helpful to collect both quantitative and qualitative information.

Quantitative analysis is considered to be objective – without any human bias attached to it – because it depends on the comparison of numbers according to mathematical computations. Analysis of qualitative data is generally accomplished by methods more subjective – dependent on people's opinions, knowledge, assumptions, and inferences (and therefore biases) – than that of quantitative data. The identification of patterns, the interpretation of people's statements or other communication, the spotting of trends – all of these

can be influenced by the way the researcher sees the world. Be aware, however, that quantitative analysis is influenced by a number of subjective factors as well. What the researcher chooses to measure, the accuracy of the observations, and the way the research is structured to ask only particular questions can all influence the results, as can the researcher's understanding and interpretation of the subsequent analyses.

WHY SHOULD YOU COLLECT AND ANALYZE DATA FOR YOUR EVALUATION?

Part of the answer here is that not every organization – particularly small community-based or non-governmental ones – will necessarily have extensive resources to conduct a formal evaluation. They may have to be content with less formal evaluations, which can still be extremely helpful in providing direction for a program or intervention. An informal evaluation will involve some data gathering and analysis. This data collection and sensemaking is critical to an initiative and its future success, and has a number of advantages.

- **The data can show whether there was any significant change in the dependent variable(s) you hoped to influence.** Collecting and analyzing data helps you see whether your intervention brought about the desired results

The term “significance” has a specific meaning when you’re discussing statistics. The level of significance of a statistical result is the level of confidence you can have in the answer you get. Generally, researchers don’t consider a result significant unless it shows at least a 95% certainty that it’s correct (called the .05 level of significance, since there’s a 5% chance that it’s wrong). The level of significance is built into the statistical formulas: once you get a mathematical result, a table (or the software you’re using) will tell you the level of significance.

Thus, if data analysis finds that the independent variable (the intervention) influenced the dependent variable at the .05 level of significance, it means there’s a 95% probability or likelihood that your program or intervention had the desired effect. The .05 level is generally considered a reasonable result, and the .01 level (99% probability) is considered about as close to certainty as you are likely to get. A 95% level of certainty doesn’t mean that the program works on 95% of participants, or that it will work 95% of the time. It means that there’s only a 5% possibility that it isn’t actually what’s influencing the dependent variable(s) and causing the changes that it seems to be associated with.

- **They can uncover factors that may be associated with changes in the dependent variable(s).** Data analyses may help discover unexpected influences; for instance, that the effort was twice as large for those participants who also were a part of a support group. This can be used to identify key aspects of implementation.

- **They can show connections between or among various factors that may have an effect on the results of your evaluation.** Some types of statistical procedures look for connections (“correlations” is the research term) among variables. Certain dependent variables may change when others do. These changes may be similar – i.e., both variables increase or decrease (e.g., as children’s proficiency at reading increases, the amount of reading they do also increases). Or the opposite may be observed – i.e. the two variables change in opposite directions (as the amount of exercise they engage in increases, peoples’ weight decreases). Correlations don’t mean that one variable causes another, or that they both have the same cause, but they can provide valuable information about associations to expect in an evaluation.

- **They can help shed light on the reasons that your work was effective or, perhaps, less effective than you’d hoped.** By combining quantitative and qualitative analysis, you can often determine not only what worked or didn’t, but why. The effect of cultural issues, how well methods are used, the appropriateness of your approach for the population – these as well as other factors that influence success can be highlighted by careful data collection and analysis. This knowledge gives you a basis for adapting and changing what you do to make it more likely you’ll achieve the desired outcomes in the future.

- **They can provide you with credible evidence to show stakeholders that your program is successful, or that you’ve uncovered, and are addressing limitations.** Stakeholders, such as funders and community boards, want to know their investments are well spent. Showing evidence of intermediate outcomes (e.g. new programs and policies) and longer-term outcomes (e.g., improvements in education or health indicators) is becoming increasingly important to receiving – and retaining – funding.

- **Their use shows that you’re serious about evaluation and about improving your work.** Being a good trustee or steward of community investment includes regular review of data regarding progress and improvement.

- **They can show the field what you’re learning, and thus pave the way for others to implement successful methods and approaches.** In that way, you’ll be helping to improve community efforts and, ultimately, quality of life for people who benefit.

WHEN AND BY WHOM SHOULD DATA BE COLLECTED AND ANALYZED?

As far as data collection goes, the “when” part of this question is relatively simple: data collection should start no later than when you begin your work – or before you begin in order to establish a baseline or starting point – and continue throughout. Ideally, you should collect data for a period of time before you start your program or

intervention in order to determine if there are any trends in the data before the onset of the intervention. Additionally, in order to gauge your program's longer-term effects, you should collect follow-up data for a period of time following the conclusion of the program.

The timing of analysis can be looked at in at least two ways: One is that it's best to analyze your information when you've collected all of it, so you can look at it as a whole. The other is that if you analyze it as you go along, you'll be able to adjust your thinking about what information you actually need, and to adjust your program to respond to the information you're getting. Which of these approaches you take depends on your research purposes. If you're more concerned with a summative evaluation – finding out whether your approach was effective, you might be more inclined toward the first. If you're oriented toward improvement – a formative evaluation – we recommend gathering information along the way. Both approaches are legitimate, but ongoing data collection and review can particularly lead to improvements in your work.

The “who” question can be more complex. If you're reasonably familiar with statistics and statistical procedures, and you have the resources in time, money, and personnel, it's likely that you'll do a somewhat formal study, using standard statistical tests. (There's a great deal of software – both for sale and free or open-source – available to help you.)

If that's not the case, you have some choices:

- You can hire or find a volunteer outside evaluator, such as from a nearby college or university, to take care of data collection and/or analysis for you.
- You can conduct a less formal evaluation. Your results may not be as sophisticated as if you subjected them to rigorous statistical procedures, but they can still tell you a lot about your program. Just the numbers – the number of dropouts (and when most dropped out), for instance, or the characteristics of the people you serve – can give you important and usable information.
- You can try to learn enough about statistics and statistical software to conduct a formal evaluation yourself. (Take a course, for example.)
- You can collect the data and then send it off to someone – a university program, a friendly statistician or researcher, or someone you hire – to process it for you.
- You can collect and rely largely on qualitative data. Whether this is an option depends to a large extent on what your program is about. You wouldn't want to conduct a formal evaluation of effectiveness of a new medication using only qualitative data, but you might be able to draw some reasonable conclusions about use or compliance patterns from qualitative information.
- If possible, use a randomized or closely matched

control group for comparison. If your control is properly structured, you can draw some fairly reliable conclusions simply by comparing its results to those of your intervention group. Again, these results won't be as reliable as if the comparison were made using statistical procedures, but they can point you in the right direction. It's fairly easy to tell whether or not there's a major difference between the numbers for the two or more groups. If 95% of the students in your class passed the test, and only 60% of those in a similar but uninstructed control group did, you can be pretty sure that your class made a difference in some way, although you may not be able to tell exactly what it was that mattered. By the same token, if 72% of your students passed and 70% of the control group did as well, it seems pretty clear that your instruction had essentially no effect, if the groups were starting from approximately the same place.

Who should actually collect and analyze data also depends on the form of your evaluation. If you're doing a participatory evaluation, much of the data collection - and analyzing - will be done by community members or program participants themselves. If you're conducting an evaluation in which the observation is specialized, the data collectors may be staff members, professionals, highly trained volunteers, or others with specific skills or training (graduate students, for example). Analysis also could be accomplished by a participatory process. Even where complicated statistical procedures are necessary, participants and/or community members might be involved in sorting out what those results actually mean once the math is done and the results are in. Another way analysis can be accomplished is by professionals or other trained individuals, depending upon the nature of the data to be analyzed, the methods of analysis, and the level of sophistication aimed at in the conclusions.

HOW DO YOU COLLECT AND ANALYZE DATA?

Whether your evaluation includes formal or informal research procedures, you'll still have to collect and analyze data, and there are some basic steps you can take to do so.

IMPLEMENT YOUR MEASUREMENT SYSTEM

We've previously discussed designing an observational system to gather information. Now it's time to put that system in place.

- Clearly define and describe what measurements or observations are needed. The definition and description should be clear enough to enable observers to agree on what they're observing and reliably record data in the same way.
- Select and train observers. Particularly if this is part of a participatory process, observers need training to know what to record; to recognize key behaviors, events, and conditions; and to reach an acceptable level of inter-rater reliability (agreement among observers).
- Conduct observations at the appropriate times for the

appropriate period of time. This may include reviewing archival material; conducting interviews, surveys, or focus groups; engaging in direct observation; etc.

- Record data in the agreed-upon ways. These may include pencil and paper, computer (using a laptop or handheld device in the field, entering numbers into a program, etc.), audio or video, journals, etc.

ORGANIZE THE DATA YOU'VE COLLECTED

How you do this depends on what you're planning to do with it, and on what you're interested in.

- Enter any necessary data into the computer. This may mean simply typing comments, descriptions, etc., into a word processing program, or entering various kinds of information (possibly including audio and video) into a database, spreadsheet, a GIS (Geographic Information Systems) program, or some other type of software or file.
- Transcribe any audio- or videotapes. This makes them easier to work with and copy, and allows the opportunity to clarify any hard-to-understand passages of speech.
- Score any tests and record the scores appropriately.
- Sort your information in ways appropriate to your interest. This may include sorting by category of observation, by event, by place, by individual, by group, by the time of observation, or by a combination or some other standard.
- When possible, necessary, and appropriate, transform qualitative into quantitative data. This might involve, for example, counting the number of times specific issues were mentioned in interviews, or how often certain behaviors were observed.

CONDUCT DATA GRAPHING, VISUAL INSPECTION, STATISTICAL ANALYSIS, OR OTHER OPERATIONS ON THE DATA AS APPROPRIATE

We've referred several times to statistical procedures that you can apply to quantitative data. If you have the right numbers, you can find out a great deal about whether your program is causing or contributing to change and improvement, what that change is, whether there are any expected or unexpected connections among variables, how your group compares to another you're measuring, etc.

There are other excellent possibilities for analysis besides statistical procedures, however. A few include:

- Simple counting, graphing and visual inspection of frequency or rates of behavior, events, etc., over time.
- Using visual inspection of patterns over time to identify discontinuities (marked increases, decreases) in the measures over time (sessions, weeks, months).
- Calculating the mean (average), median (midpoint),

and/or mode (most frequent) of a series of measurements or observations. What was the average blood pressure, for instance, of people who exercised 30 minutes a day at least five days a week, as opposed to that of people who exercised two days a week or less?

- Using qualitative interviews, conversations, and participant observation to observe (and track changes in) the people or situation. Journals can be particularly revealing in this area because they record people's experiences and reflections over time.
- Finding patterns in qualitative data. If many people refer to similar problems or barriers, these may be important in understanding the issue, determining what works or doesn't work and why, or more.
- Comparing actual results to previously determined goals or benchmarks. One measure of success might be meeting a goal for planning or program implementation, for example.

TAKE NOTE OF ANY SIGNIFICANT OR INTERESTING RESULTS

Depending on the nature of your research, results may be statistically significant (the 95% or better certainty that we discussed earlier), or simply important or unusual. They may or may not be socially significant (i.e., large enough to solve the problem).

There are a number of different kinds of results you might be looking for.

- Differences within people or groups. If you have repeated measurements for individuals/groups over time, we can see if there are marked increases/decreases in the (frequency, rate) of behavior (events, etc.) following introduction of the program or intervention. When the effects are seen when and only when the intervention is introduced – and if the intervention is staggered (delayed) across people or groups – this increases our confidence that the intervention, and not something else, is producing the observed effects.
- Differences between or among two or more groups. If you have one or more randomized control groups in a formal study (groups that are drawn at random from the same population as the group in your program, but are not getting the same program or intervention, or are getting none at all), then the statistical significance of differences between or among the groups should tell you whether your program has any more influence on the dependent variable(s) than what's experienced by the other groups.
- Results that show statistically significant changes. With or without a control or comparison group, many statistical procedures can tell you whether changes in dependent variables are truly significant (or not likely due to chance). These results may say nothing

about the causes of the change (or they may, depending on how you've structured your evaluation), but they do tell you what's happening, and give you a place to start.

- **Correlations.** Correlation means that there are connections between or among two or more variables. Correlations can sometimes point to important relationships you might not have predicted. Sometimes they can shed light on the issue itself, and sometimes on the effects of a group's cultural practices. In some cases, they can highlight potential causes of an issue or condition, and thus pave the way for future interventions.

Correlation between variables doesn't tell you that one necessarily causes the other, but simply that changes in one have a relationship to changes in the other. Among American teenagers, for instance, there is probably a fairly high correlation between an increase in body size and an understanding of algebra. This is not because one causes the other, but rather the result of the fact that American schools tend to begin teaching algebra in the seventh, eighth, or ninth grades, a time when many 12-, 13-, and 14-year-olds are naturally experiencing a growth spurt.

On the other hand, correlations can reveal important connections. A very high correlation between, for instance, the use of a particular medication and the onset of depression might lead to the withdrawal of that medication, or at least a study of its side effects, and increased awareness and caution among doctors who prescribe it. A very high correlation between gang membership and having a parent with a substance abuse problem may not reveal a direct cause-and-effect relationship, but may tell you something important about who is more at risk for substance abuse.

- **Patterns.** In both quantitative and qualitative information, patterns often emerge: certain health conditions seem to cluster in particular geographical areas; people from a particular group behave in similar ways; etc. These patterns may not be specifically what you were looking for or expected to find, but they may either be important in themselves or shed light on the areas you're interested in. In some cases, you may need to subject them to statistical procedures (regression analysis, for example) to see if, in fact, they're random, or if they constitute actual patterns.
- **Obvious important findings.** Whether as a result of statistical analysis, or of examination of your data and application of logic, some findings may stand out. If 70% of a group of overweight participants in a healthy eating and physical activity program lowered their weight and blood pressure significantly, compared to only 20% of a similar group not in the program, you can probably assume that program may have been effective. If there's no change whatsoever in education outcomes after two years of your education

program, then you're either running an ineffective program, or you're simply not reaching those who are most likely to have poorer outcomes (which can also be interpreted to mean you're running an ineffective program.)

Not all important findings will necessarily tell you whether your program worked, or what is the most effective method. It might be obvious from your data collection, for instance, that, while violence or roadway injuries may not be seen as a problem citywide, they are much higher in one or more particular areas, or that the rates of diabetes are markedly higher for particular groups or those living in areas with greater disparities of income. If you have the resources, it's wise to look at the results of your research in a number of different ways, both to find out how to improve your program, and to learn what else you might do to affect the issue.

INTERPRET THE RESULTS

Once you've organized your results and run them through whatever statistical or other analysis you've planned for, it's time to figure out what they mean for your evaluation. Probably the most common question that evaluation research is directed toward is whether the program being evaluated works or makes a difference. In research terms, that often translates to "What were the effects of the independent variable (the program, intervention, etc.) on the dependent variable(s) (the behavior, conditions, or other factors it was meant to change)?" There are a number of possible answers to this question:

- Your program had exactly the effects on the dependent variable(s) you expected and hoped it would. Statistics or other analysis showed clear positive effects at a high level of significance for the people in your program and – if you used a multiple-group design – none, or far fewer, of the same effects for a similar control group and/or for a group that received a different intervention with the same purpose. Your early childhood education program, for instance, greatly increased development outcomes for children in the community, and also contributed to an increase in the percentage of children succeeding in school.
- Your program had no effect. Your program produced no significant results on the dependent variable, whether alone or compared to other groups. This would mean no change as a result of your program or intervention.
- Your program had a negative effect. For instance, intimate partner violence increased (or at least appeared to) as a result of your intervention. (It is relatively common for reported events, such as violence or injury, to increase when the intervention results in improved surveillance and ease of reporting).
- Your program had the effects you hoped for and other effects as well.

- These effects might be positive. Your youth violence prevention program, for instance, might have resulted in greatly reduced violence among teens, and might also have resulted in significantly improved academic performance for the kids involved.
- These effects might be neutral. The same youth violence prevention program might somehow result in youth watching TV more often after school.
- These effects might be negative. (These effects are usually called unintended consequences.) Youth violence might decrease significantly, but the incidence of teen pregnancies or alcohol consumption among youth in the program might increase significantly at the same time.
- These effects might be multiple, or mixed. For instance, a program to reduce HIV/AIDS might lower rates of unprotected sex but might also increase conflict and instances of partner violence. Your program had no effect or a negative effect and other effects as well. As with programs with positive effects, these might be positive, neutral, or negative; single or multiple; or consistent or mixed.

If your analysis gives you a clear indication that what you're doing is accomplishing your purposes, interpretation is relatively simple: You should keep doing it, while trying out ways to make it even more effective, or while aiming at other related issues as well.

As we discuss elsewhere in the Community Tool Box, good programs are dynamic — constantly striving to improve, rather than assuming that what they're doing is as good as it can be.

If your analysis shows that your program is ineffective or negative, however — or, for that matter, if a positive analysis leaves you wondering how to make your successful efforts still more successful — interpretation becomes more complex. Are you using an absolutely wrong approach? Are you using an approach that could be effective, but is poorly implemented? Is there a particular contributing factor you're failing to take into account? Are there barriers to success — of culture, experience, personal characteristics, systematic discrimination — present in the population from which participants are drawn? Are there particular components or elements you can change to make your program more effective, or should you start again from scratch? What should you address to make a good program better?

Careful and insightful interpretation of your data may allow you to answer questions like these. You may be able to use correlations, for instance, to generate hypotheses about your results. If positive or negative changes in particular variables are consistently associated with positive or negative changes in other variables, the two may be connected. (The word “may” is important here. The two may be connected, but they may not, or both may be related to a third variable that you're not aware of or that you consider trivial.) Such a connection can point the way toward a factor (e.g., access to support) that is causing the changes in both variables, and that must be addressed to make your program successful. Correlations may also indicate patterns in your data, or may lead to an unexpected way of looking at the issue you're addressing.

DRAFT

Smart City - Identify solar Panel Basic Testing, characteristics, charge controller circuit

Test the android phone and its features, use of sensors and its usage

Objective:

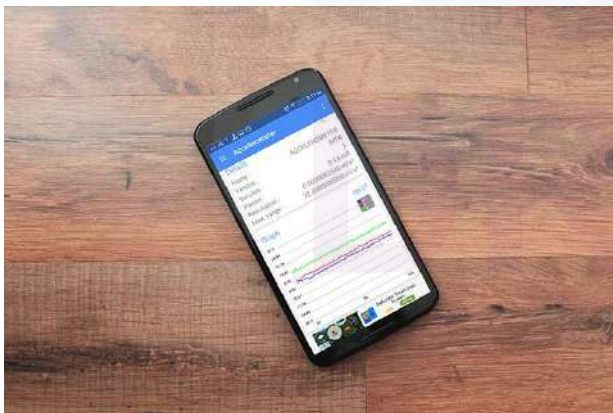
- to test the android phone and its features, use of sensors and its usage.

Requirements:

- Android phone
- Android sensors
- Gyroscope
- Magnetometer
- Proximity sensor
- Accelerometer
- Light sensor
- Barometer

PROCEDURE:

You must be knowing how much of RAM your Android device has, how much battery your device uses and how many processing cores it takes to make your device work. But, did you ever think how many sensors your devices has inbuilt? Okay, let's say you do. Now, do you know if they work correctly or not? Maybe they are damaged? Well, no worries you're about to find that out today.



You can perform these test not just on your phone but also on some other phone that you might buy from someone else. Testing used phones before buying is a must. But, before digging into the guide I'd like to share some knowledge of different sensors the new age android phones have.

IMPART THY KNOWLEDGE: TYPES OF SENSORS IN ANDROID SMARTPHONE

- **Accelerometer:** It detects the speed of your phone when moved. The readings of Accelerometer will spike up on every movement. Keep it flat on surface and readings will be stable. It is also used to determine where the phone is in 3 dimensions. (Upside-down or horizontal, for example).
- **Gyroscope:** Gyroscope is one step further to Accelerometer. As accelerometer shows where the phone is in three dimensions but it can't tell how is it rotating in those three dimensions. So, the Gyroscope helps to find out at what axis is the phone rotating. Helpful in playing FPS and Racing games.
- **Magnetometer:** Yes, your phone is able to detect magnetic fields. And, you must have guessed it right. It is used by Compass apps to detect the North pole of the planet.
- **Proximity Sensor:** It comprises of two elements, an LED and an IR light detector. It's placed near the earpiece of your phone. Most helpful in situations when you take a call and put your phone on your ear and the screen turns off and turns on when taken off the ear. It throws an infrared light (which is invisible to human eye) at the near object and the IR detector detects at what distance the object is. Actions are performed accordingly.
- **Light Sensor:** Detects how bright the surrounding light is, in which the phone is placed.
- **Barometer:** This is mostly found in high-end phones. You probably know this. It detects how high the phone above the sea level is. This gives better GPS accuracy.

Smart City - Identify solar Panel Basic Testing, characteristics, charge controller circuit

Check the Bluetooth module along and explore the possibility of pairing with Android smart phone

Objective:

- to check the Bluetooth module along and explore the possibility of pairing with Android smart phone.

Requirements:

- Arduino UNO
- Bluetooth device
- Android Smart phone

Procedure:

How to fix the Bluetooth on your Android

Having trouble establishing a Bluetooth connection on your Android cell phone? Was the Bluetooth on your phone working fine and now it won't even show an available connection? Is your Bluetooth having pairing issues? Will your Bluetooth not even switch on? Here are some suggestions and tricks that you can use to get your Bluetooth back up and running properly again.

Finding the proper solution

There are actually many things that can cause Bluetooth issues to occur on an Android smartphone and the fix that works for your phone is going to depend on what is actually causing your Bluetooth not to function properly.

This guide will start with the basics and provide troubleshooting that can be done to resolve the most common Bluetooth issues and will progress to more in-depth troubleshooting. So let's start from the basics and go from there.

Power cycle

If your cell phone started to suddenly have Bluetooth issues unexpectedly and out of the blue then try simply powering your Android cell phone off, wait a minute (30-60 seconds), and then power the phone back on. Then try to establish a connection again and see if this simple trick helped to fix it. If not then proceed to the next suggestion.

Battery pull

Much like powering your cell phone off and then back on, another form of soft reset for an Android device is to remove the battery while the device is powered on. Wait about 30 seconds, plug the battery back in, power the phone on and see if the Bluetooth will connect.

Remember that not all cell phones are designed to have a removable battery. If your cell phone does not allow you to remove its battery then you can skip this portion of troubleshooting.

Clear the Cache

If your Bluetooth was connected and working properly and you tried to adjust a setting on the phone only to find that your Bluetooth stopped connecting or pairing correctly then this (or the next suggestion) should be exactly what you are looking for.



There might be a setting in your phone which is now preventing your Android's Bluetooth from connecting to or even recognizing an available Bluetooth to sync to. So start by clearing the cache in your Bluetooth settings.

The method that you use to access the Bluetooth settings and clear its cache is going to depend on what make and model of cell phone you are currently using as well as what version of Android is currently installed on the device.

What you are going to need to locate is your phones "Application manager" or an option to "Manage applications". This feature will let you view all of the applications installed on your phone and you will be able to locate your Bluetooth feature and its settings.

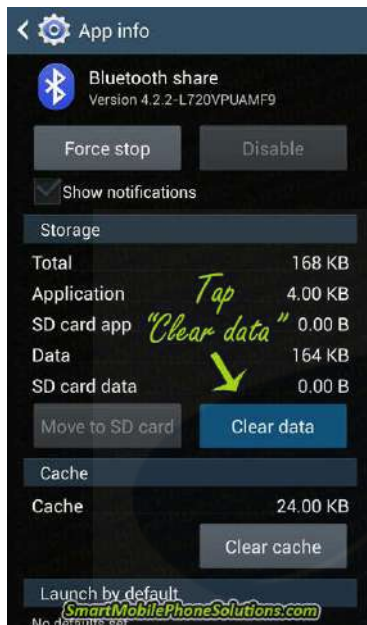
A common way to access this setting and clear the cache on Android 4.2 Jelly Bean is to open the phones **Settings**,

tap the **More** tab > tap **Application manager** > view **ALL** > **Bluetooth share** > tap **Clear cache**.

Note: "Bluetooth share" is a common name for the Bluetooth feature on an Android cell phone but if you cannot locate a "Bluetooth share" option then it might be labeled differently. Typically it will still have the word "Bluetooth" in its label.

Once you clear the cache, power cycle or restart your cell phone, try your Bluetooth and see if it's working properly. If not then along with the cache try clearing your Bluetooth Data as well.

Clear the data



Before clearing the Bluetooth data make sure to record any important Bluetooth connections that you do not think you will be able to remember or setup again. This step in troubleshooting will tell your phone to make your Bluetooth feature start fresh and clean again and as such it will erase any current connections saved in your Bluetooth's connection log/memory. This shouldn't be a big issue as you will be able to simply re-connect to those Bluetooth connections when you need to use them again but I thought I would mention it for good measure.

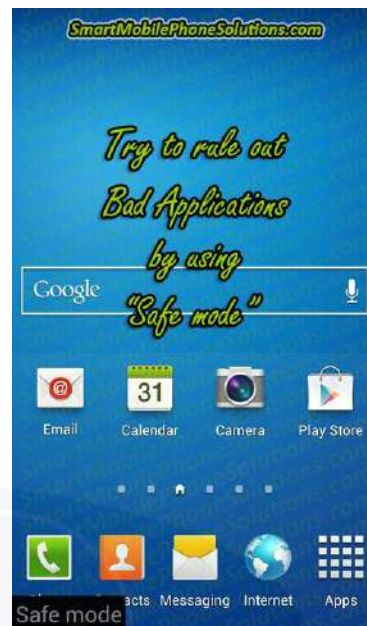
To clear the Bluetooth app data it's pretty much the same steps used when clearing its cache only instead of "Clear cache" you're going to select the "Clear data" option. So open the phones **Settings**, tap the **More** tab > tap **Application manager** > view **ALL** > **Bluetooth share** > **Clear data**.

Once the Data is also cleared then restart the phone and enjoy your working Bluetooth. Don't forget to let me know if this helped fix your cell phone by pressing the Facebook Like or Google + button(s) at the bottom of this article.

If for some reason your Bluetooth is still having trouble then keep reading for some additional suggestions.

Safe Mode

This issue with your Bluetooth might be the result of an application on your cell phone not playing nice with another application or conflicting with its software. To help rule out this scenario then you can try using your phone in Safe Mode.



Safe Mode allows the phone to run without any third party applications and can help you to locate apps that are causing mischief. If the phone runs properly while in Safe mode then it means that an app is in fact causing problems and you can start going through your recently installed or recently updated apps to locate the troublemaker. Just uninstall the app(s) that you believe might be the cause, reboot, test and if necessary repeat. Once you have found the app that's causing problems you can re-install it, see if that helps, or simply keep it uninstalled and off of your phone.

Check the other device

Usually pairing the Bluetooth on your cell phone with another device is pretty straightforward and very user friendly. If you are still having difficulties pairing to another device then try and make sure to check that device to make sure that it is functioning properly. It may not be an issue with your phone but an issue with that other device.

So make sure that the other unit is setup correctly and that you are trying to pair your phone to that device properly. If by the off chance another phone can pair to that other device but your phone is still having issues then there can still be a problem with the software on your device (which the next section of this article should help solve) or a hardware issue where an actual piece of the phone is defective or malfunctioning in which case

you will have to look into your repair or replacement options (which we'll cover later in the article as well).

Hard reset

Performing a factory data reset is usually a last resort when troubleshooting as it erases all user data from the cell phone in question. It is however a very powerful

troubleshooting tool as it removes everything from the phone which includes any viruses, software bugs or glitches, troublesome programs, bad apps, and also puts all of the phones settings back to factory default. So if there is something on the phone that's causing problems and you can't find what it is chances are a factory data reset can help you get rid of it.



Smart City - Identify solar Panel Basic Testing, characteristics, charge controller circuit

Test Bluetooth module with a micro controller and program to switch on/off an LED/Buzzer

Objective:

- to test Bluetooth module with a micro controller and program to switch on/off an LED/Buzzer.

Requirement:

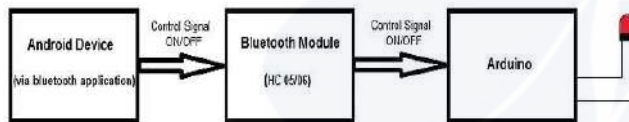
- Bluetooth module HC-05
- Arduino UNO
- 1K ohm resistor
- LED

PROCEDURE:

Android studio isn't really required here, since I will provide you with the Android application that I made. You can install the given .apk file to use the application.

How Does It Work?

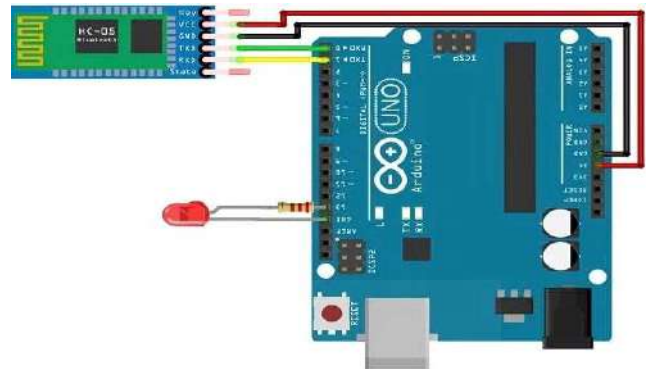
There are three main parts to this project. An Android smartphone, a Bluetooth transceiver, and an Arduino.



HC 05/06 works on serial communication. The Android app is designed to send serial data to the Arduino Bluetooth module when a button is pressed on the app. The Arduino Bluetooth module at the other end receives the data and sends it to the Arduino through the TX pin of the Bluetooth module (connected to RX pin of Arduino). The code uploaded to the Arduino checks the received data and compares it. If the received data is 1, the LED turns ON. The LED turns OFF when the received data is 0. You can open the serial monitor and watch the received data while connecting.

Connecting the Arduino Bluetooth Hardware

This circuit is simple and small. There are only four connections to be made between the Arduino and Bluetooth module!



Arduino Pins | Bluetooth Pins

RX (Pin 0) ———> TX

TX (Pin 1) ———> RX

5V ———> VCC

GND ———> GND

Connect an LED positive to pin 13 of the Arduino through a resistance (valued between 220?–1K?). Connect its negative to GND, and you're done with the circuit!

You can connect the Bluetooth module to the Arduino using a set of jumper wires and a connector.

Note: Don't connect RX to RX and TX to TX on the Bluetooth and Arduino. You will receive no data. Here, TX means transmit and RX means receive.

Uploading the Sketch to Arduino

```

char data = 0;           //Variable for storing received
data
void setup()
{
  Serial.begin(9600);     //Sets the data rate in bits per
                          //second (baud) for serial data transmission
  pinMode(13, OUTPUT);    //Sets digital pin 13 as
                          //output pin
}
  
```

```

}
void loop()
{
if(Serial.available() > 0) // Send data only when you
receive data:
{
data = Serial.read();    //Read the incoming data and
store it into variable data

Serial.print(data);      //Print Value inside data in Serial
monitor

Serial.print("\n");      //New line

if(data == '1')          //Checks whether value of data is
equal to 1

digitalWrite(13, HIGH); //If value is 1 then LED turns ON

else if(data == '0')     //Checks whether value of data is
equal to 0

digitalWrite(13, LOW);  //If value is 0 then LED turns
OFF

}
}

```

Upload the given sketch to Arduino using the Arduino IDE software, you can also get it from Github.

Installing the Android Application



In this tutorial, I will not cover Android app development. You can download the Android application from here and the source code of the entire project.

- Download the application from Amazon App Store or Github.
- Pair your device with the HC 05/06 Bluetooth module:
- Turn ON the HC 05/06 Bluetooth module by powering the Arduino.
- Scan your smartphone for available devices.3. Pair your smartphone to the HC 05/06 by entering default password 1234 OR 0000.

- Install the LED application on your Android device.
- Open the application.



- Press "paired devices".
- Select your Bluetooth module from the list (HC-05/06)



- After connecting successfully, press the ON button to turn the LED on and the OFF button to turn the LED off.
- Disconnect the button to disconnect the Bluetooth module.

Here's how it works!



This is just a basic tutorial on interfacing a Bluetooth module with Arduino. This project can be taken to a higher level, like home automation using a smartphone, smartphone-controlled robots, and much more! You can watch the video tutorial below:



Smart City - Identify solar Panel Basic Testing, characteristics, charge controller circuit

Check the GSM module and its interconnections

Objective:

- to check the GSM module and its interconnections.

Requirements:

- Atmega16
- GSM module (SIM900 or any other)
- LCD display
- Push buttons
- 10k resistors, Potentiometer
- Connecting wires
- 12V Adapter
- USBasp programmer
- 10 pin FRC cable

Procedure:

Introduction:

GSM modules are interesting to use especially when our project requires remote access. These modules could make all actions that our normal mobile phone could do, like making/receiving a call, sending/receiving a SMS, connecting to internet using GPRS etc. You can also connect a normal microphone and speaker to this module and converse on your mobile calls. This will open doors to lot of creative projects if it could be interfaced with a Microcontroller. Hence in this tutorial we will learn how we can Interface the GSM module (SIM900A) with AVR microcontroller ATmega16 and will demonstrate it by sending and receiving messages using GSM Module.

Software Used

We will use CodeVisionAVR software for writing our code and SinaProg software for uploading our code to Atmega16 using USBASP programmer.

You can download these softwares from the given links:

CodeVisionAVR : http://www.hpinfotech.ro/cvavr_download.html

SinaProg : http://www.ablab.in/sinaprog-hex-file-downloader-software-to-download-code-to-avr-microcontroller/#inline_content

GSM Module

Communicating with GSM module using AT commands

As you might have guessed it, the GSM module can communicate through Serial communication and could understand only one language and that is "AT

commands". Whatever that you might want to tell or ask to the GSM module it should only be via AT commands.



GSM Module SIM900A

For example if you want to know if your module is active. You should ask (send) a command like "AT" and your module will reply "OK".

AT	Replies with OK for Acknowledgement
AT+CPIN?	Check signal Quality
AT+COPS?	Find service provider name
ATD96XXXXXXX;	Call to the specific number, ends with semi-colon
AT+CNUM	Find the number of SIM card (might not work for some SIM)
ATA	Answer the Incoming Call
ATH	Hang off the current Incoming call
AT+COLP:	Show incoming call number
AT+VTS=(number)	Send DTMF number. You can use any number on your mobile keypad for (number)
AT+CMGR	AT+CMGR=1 reads message at first position
AT+CMGD=1	Delete message at first position

AT+CMGDA="DEL ALL"	Delete All messages from SIM
AT+CMGL="ALL"	Read all messaged from SIM
AT+CMGF=1	Set SMS configuration. "1" is for text only mode
AT+CMGS = "+91 968837XXXX"	Sends SMS to a particular number here 968837XXXX. When you see ">" start entering the text. Press Ctrl+Z to send the text.
AT+CGATT?	To check for internet connection on SIM card
AT+CIPSHUT	To close TCP connection, meaning to disconnect form Internet
AT+CSTT = "APN","username","Pass"	Connect to GPRS with your APN and Pass key. Can be obtained from Network Provider.
AT+CIICR	Check if SIM card has data pack
AT+CIFSR	Get IP of the SIM network

AT+CIPSTART = "TCP","SERVER IP","PORT"	Used to set a TCP IP connection
AT+CIPSEND	This command is used to send data to server.

If you have used GSM module with Arduino, while receiving messages you can use +CMT: command to view the mobile number and text message on serial monitor. Text message comes on second line as shown in picture.

We will scan this +CMT: command to check whether message is available or not.

```

COM14 (Arduino/Genuino Uno)
AT+CMGR=1,0
+CMGR: "REC READ", "DD-NICSMS", "", "15/07/30,20:04AT+CMGR=1,0
+CMGR: "REC READ", "DD-NICSMS", "", "15/07/30,20:04
+CMT: "+918368157093", "", "18/07/06,16:44:52+22"
hii
Autoscroll Both NL & CR 9600 baud Clear

```



Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Download mobile app from play store and control (ON/OFF) a simple LED via Bluetooth

Objective:

- to download mobile app from play store and control (ON/OFF) a simple LED via Bluetooth.

Requirements:

COMPONENTS AND SUPPLIES

- Arduino UNO & Genuino UNO
- HC-06 Bluetooth Module
- Breadboard (generic)
- LED (generic)
- Resistor 220 ohm
- 9V battery (generic)

PROCEDURE:

Step 1: Connecting HC-05 Bluetooth Module with Arduino

HC-05 is a serial port module which makes it very easy to use. If you see the pin configuration of HC-05, there are total 6 but we only need 4 middle ones for our set-up.

- Connect VCC with 3.3V of Arduino, please do not connect it with 5V as that can cook the module
- Connect GND with any GND of Arduino
- Connect Rx pin with Tx of Arduino
- Connect Tx pin with Rx of Arduino

Now power-up the Uno using USB cable, a red light LED on HC-05 will start blinking, means we are ready to go forward to the next step!

Step 2: Connect the LED and Control It Using Arduino Serial Monitor

I often refer to this step as Arduino 101, connecting the LED with Arduino on pin 13 is the most basic thing to do and you must be familiar with it if you came so far, just to help you again doing it:

- Connect the long end of LED with 220 ohm to 1K ohm resistor
- Connect the other end of resistor to the pin 13 of Arduino
- Connect the short leg of LED to GND of Arduino

If you prefer, you can skip this step and just use the built-in yellow LED of Arduino. Our circuit is now complete

and we will now move forward to building our software blocks!

Add TipAsk QuestionCommentDownload

Step 3: Send Serial Commands to Arduino Using Serial Monitor (or Any Terminal)

Upload the following sketch to the Arduino using USB cable.

Caution: Disconnect the HC-05 bluetooth module Rx and Tx pins from Arduino Uno as this particular board has only one hardware serial and connecting something to it while uploading a sketch will create conflict or you can use Arduino SoftwareSerial to avoid conflicts. Reconnect these pins once you are done uploading the sketch.

/*

Arduino Turn LED On/Off using Serial Commands

Created April 22, 2015

Hammad Tariq, Incubator (Pakistan)

It's a simple sketch which waits for a character on serial and in case of a desirable character, it turns an LED on/off.

Possible string values:

a (to turn the LED on)

b (to turn the LED off)

*/

char junk;

String inputString="";

void setup() // run once, when the sketch starts

{

Serial.begin(9600); // set the baud rate to 9600, same should be of your Serial Monitor

pinMode(13, OUTPUT);

}

void loop()

{

```

if(Serial.available()){
while(Serial.available())
{
char inChar = (char)Serial.read(); //read the input
inputString += inChar;           //make a string of the
characters coming on serial
}
Serial.println(inputString);
while (Serial.available() > 0)
{ junk = Serial.read() ; }    // clear the serial buffer
if(inputString == "a"){      //in case of 'a' turn the LED on
digitalWrite(13, HIGH);
}else if(inputString == "b"){ //incase of 'b' turn the LED
off
digitalWrite(13, LOW);
}
inputString = "";
}
}

```

After uploading, open Arduino Serial Monitor, set the baud rate to 9600 and command line ending drop down (the one next to baud rate) to "No line ending", that means we will not be sending and /r or /n characters with our serial command.

Now type character "a" in the serial monitor and press send, the your LED should turn on, then send "b", the LED should turn off.

I have also attached the sketch file with this step, this is all we need at Arduino sketch level for turning LED on/off even through the Bluetooth and mobile app. In next step, we will

Step 4: Getting HC-5 to Work with Arduino and Testing the Communication

In this step, we are focusing on getting HC-05 to work with Arduino and testing all the communication before we move towards building our mobile app.

Important: There is no extra step or coding required for HC-05 to work, it's a simple serial port module, which means if you pair it and then connect to it using any serial port terminal, it will work just like Arduino Serial Monitor.

Follow these mini-steps:

- Connect the Rx and Tx pins of HC-05 back to Arduino (if you haven't did this already).
- Pair the device with your mobile phone (in my case I am using Android OS and my phone is Samsung SII).
- Download "[Bluetooth Terminal](#)" app from Google Play Store.
- Open Bluetooth Terminal app.

- From the menu, tap on "Connect a device - Insecure".
- You will see a pop-up of "Paired Devices", tap on "HC-05", after a second you will get a toast notifying "Connected to HC-05".
- Now type "a" and send, the LED will turn on, similarly, send "b" to turn the LED off.

In the next step will start building our cross-platform HTML5 app by installing and using Cordova and Evothings Studio.

Step 5: Installing Evothings Studio

In this step, we will install Evothings Studio.

Evothings Studio uses Cordova which let's you develop applications in HTML5 and provides easy to use functions to interface with underlying OS architecture to use different functionalities of smartphone itself! Apps built with Cordova can be deployed on all kinds of modern mobile operating systems such as iOS, Android and Windows Phone etc. It's open-source and a large community is behind it, means you can easily find online help and plugins written for different use-cases and technologies.

Similarly, Evothings Studio is also open-source and it's main advantage is it's "reload-on-save" functionality, which means you don't need to rebuild your app all the time. If Evothings client and studio are installed and working, you can just edit your source HTML file and can see the latest changes in your mobile phone instantly.

Now quickly, follow these steps:

- Install [Evothings Workbench](#).
- Install Evothings Client app ([Android](#), [iOS](#))
- Connect Evothings Client with Evothings Workbench using your local WiFi IP address.
- Download and save the complete example app code from my github [repository](#).
- Drag & drop the app's "index.html" file from the example app code folder to the Evothings Workbench, that will creat a new project entry in the Evothings Workbench.
- Click the "Run" button alongside the new project entry, that will load the example app inside the Evothings Client.
- Now click the "Code" button to see the location of the code, open the "index.html" file in your favourite IDE and change something, like the colour of a button, the Evothings Studio will immediately pick the change and will reload the app inside the Evothings Client, this functionality allows you to rapidly prototype your web app for IoT devices.

Step 6: Introducing BluetoothSerial

Evothings comes pre-installed with BluetoothSerial plugin of Cordova, that means we can straight away start implementing bluetooth to serial

communication in our project.

The most important function of BluetoothSerial plugin is:

```
bluetoothSerial.write();
```

Whatever we will send in write function, the plugin will write that on serial output of bluetooth module.

So, in this case, following code bit is doing all the work. When a user presses the “LED ON” button, the script will write “a” onto the Arduino serial using BluetoothSerial plugin and in the other case it will write the character “b”, which our Arduino sketch understands as the argument to switch the LED to off status.

```
app.initialize();  
app.ledOn = function(){  
  bluetoothSerial.write("a");  
}  
app.ledOff = function()
```

```
  bluetoothSerial.write("b");  
}
```

Step 7: Developing Front-end Using Evothings

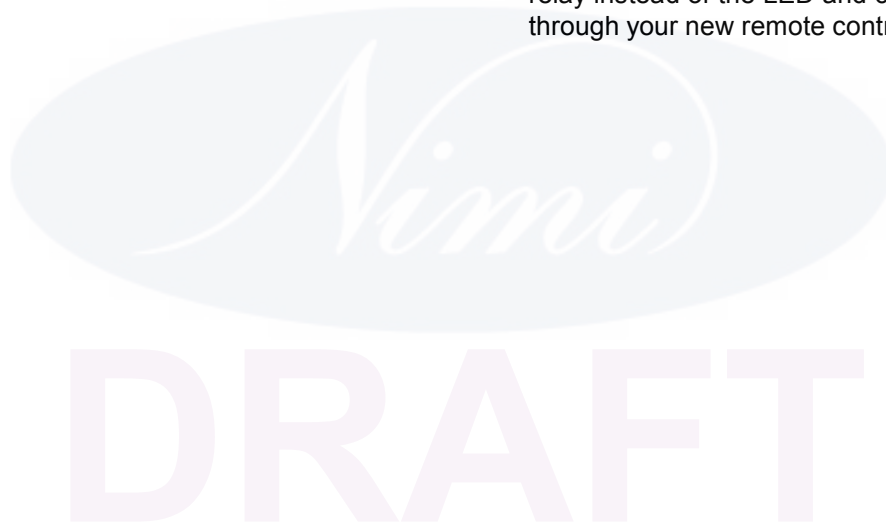
So, we have everything ready now, our circuit, Arduino sketch, example app code and Evothings Studio also knows about the location of the project and can “Run” it.

The plugin’s functionality code is hidden under /www/js/index.js and front-end is in /www/index.html, we have copied default CSS of Evothings and project is also running using Evothings Studio, whatever we will change, it will automatically reload the app in the phone, giving us the instant preview.

The only bit required is a bit of JavaScript and HTML know-how.

Step 8: Reviewing the Completed Bluetooth Remote Control App

So, now we have our app and hardware working. Your app is already running in Evothings client, that is great, you have your 3 buttons and they switch the lights on and off and also let you manage your connection with the Bluetooth module. Play with these, your work is over. The next step will be to Google a tutorial about adding a relay instead of the LED and control your home’s lights through your new remote control.



Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Test GPS module

Objective:

- to test GPS module.

Requirements:

- Arduino UNO & Genuino UNO
- GPS Module (Generic) NEO-6M GPS Module
- LED (generic)
- Texas Instruments Dual H-Bridge motor drivers L293D
- Brown Dog Gadgets Solar Cockroach Vibrating Disc Motor
- Spark Fun Snappable Proto board
- 9V battery (generic)
- 9V to Barrel Jack Connector
- Jumper wires (generic)

PROCEDURE:**Introduction:**

Everyone knows that traffic jams can be a major time waster. And it is impossible to predict how long it would take from origin to destination.

The problem of traffic jams affected me when I came to a city two months ago. Every day I spend more than two hours stuck in jams. And I felt it like why can't I utilize this time to do something?

Note: I use public transport. :-)

There are plenty of other things you can do while being stuck in traffic jam!

Some of the ones below are not just fun, but productive too:

Use the time to think and plan, either for current and future projects.

And of course making electronics stuffs always inspires me. So I built a destination notifier using Arduino and GPS module. So what it does is whenever you are near to your destination, it notifies you by glowing LED or through vibration (by using mini vibrating motor). I have provided circuits for both LED and vibrating motor.

For that, first you need to find latitude and longitude to define the location. Once you find your location, you can use the latitude and longitude values to find distance to the location and by keeping a range you can turn on the notifier.

Parts and Tools

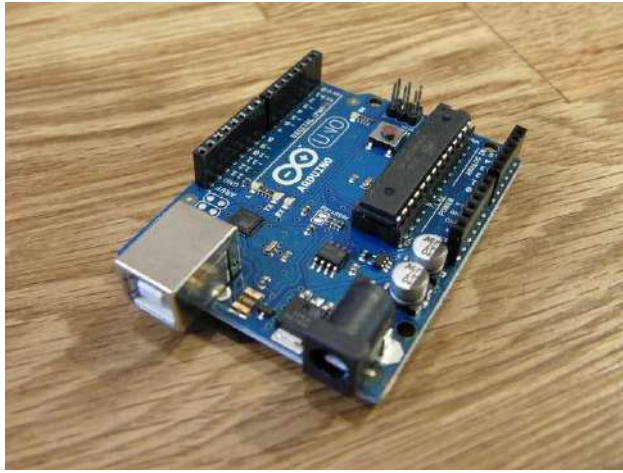
To get started with your destination notifier, here are the required parts:

- Arduino UNO
- NEO-6M GPS Module
- GPS stands for global positioning system and can be used to determine position, time and speed if you are travelling.
- This module has an external antenna and built-in EEPROM.
- Interface: RS232 TTL
- Power supply: 3V to 5V
- Default baudrate: 9600 bps
- Works with standard NMEA sentences

The NEO-6M GPS module has four pins: VCC, RX, TX, and GND. The module communicates with the Arduino via serial communication using the TX and RX pins, so the wiring couldn't be simpler:

NEO-6M GPS Module Wiring to Arduino UNO

- VCC VIN
- RX TX pin defined in the software serial
- TX RX pin defined in the software serial
- GND GND
- L293D IC



Brain of the project

The L293D is a 16-pin motor driver IC which can control upto two DC motors simultaneously in any direction.

Why to use L293D?

The input to the motor driver IC or motor driver is a low current signal. The function of the circuit is to convert the low current signal to a high current signal. This high current signal is then given to the motor.

NEO – 6M GPS Module

TinyGPS++ Library :

The TinyGPS++ library makes it simple to get information on location in a format that is useful and easy to understand.

The TinyGPS++ library allows you to get way more information than just the location, and in simple way, besides the location, you can get:

```
>>date
>>time
>>speed
>>course
>>altitude
>>satellites
>>hdop
```



Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Check Wifi module

Objective:

- to check WiFi module.

Requirements:

- Arduino UNO
- WiFi Module

PROCEDURE:

The ESP8266-01 is the first WiFi module of the ESP family from Espressif Systems.

It is a small and inexpensive WiFi transceiver that can be used in several different projects based on Arduino, Raspberry Pi or other applications.

Its specs are incredible for such a tiny module

- 802.11 b/g/n protocol
- Wi-Fi 2.4 GHz, support WPA/WPA2
- Super small module size (11.5mm x 11.5mm)
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack (ipv4 only at the moment)
- Deep sleep power <10uA, Power down leakage current < 5uA
- Integrated low power 32-bit MCU
- SDIO 2.0, SPI, UART, I2C
- Standby power consumption of < 1.0mW (DTIM3)

Testing the module

As the world is more and more interconnected it is crucial to allow your projects communicating with the rest of the network and the cloud.

As an example 5 years ago a typical hobbyist temperature logger would have used a SD card to store the temperature reads over time.

Today a modern temperature logger does not ship with local storage but it leverages any cloud service to store and visualize the temperature reads.

I have tried several type of RF 433Mhz module, infrared links, bluetooth transceiver and the good wireless Nrf24L01 modules. Each of this technology has its own pro and cons and I have achieved different level of satisfaction, but I am honest: none of them has been good and reliable as the great ESP8266-01 WiFi module.

The ESP8266-01 adds to my projects 802.11 b/g/n WiFi connectivity and allows me to control them over the Internet using standard technologies and applications on mobile phone.

BoM

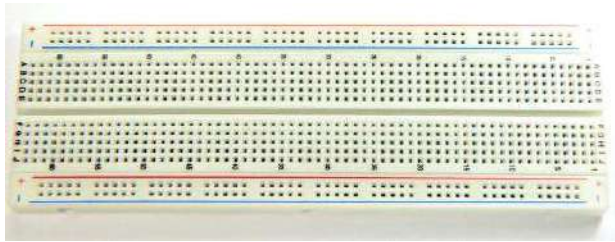
Below the components and material that is needed to test the module

- 1x ESP8266-01 WiFi module



- 1 x breadboard





- 1x PL2302HX USB to TTL converter or any other USB to TTL converter



- 1x AA battery holder (I have reused one from an old RC car) and 2x AA battery

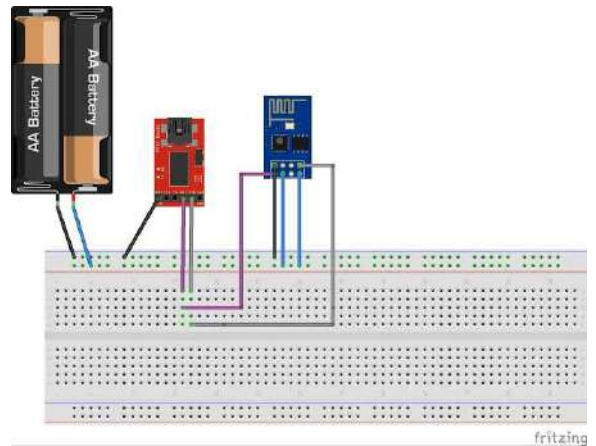


- 8x rainbow wire male-female



Wiring

I wire the module as depicted in the below figure



Power

The ESP8266-01 modules, according to its specifications, should be powered with a DC stable source at 3.3V.

Note: I really suggest to power it with a very stable 3.3V DC power source as the module is quite sensible to current fluctuation and it might be subject to auto-reset problem

In my very first test I have tried to power the module from an Arduino Uno but it was not providing enough Amps therefore I have switched to AA batteries

In the above schema the batteries must be wired in the following way:

- blue wire (+) is the positive from batteries
- black wire (-) is the negative or ground from batteries

Serial Interconnection

I use a USB to TTL converter which I have in my toolbox (PL2302HX). This module provides 3.3V power but also in this case the current it is not enough to properly power the ESP module.

Note: you can use any other USB to TTL converter, they all work in the same way

Wire and connect the USB to TTL module in the following way:

- black wire [1] to ground
- purple wire [1] to RX pin
- grey wire [1] to TX pin



Nimi
DRAFT

Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Identify cable and its pin mapping

Objective:

- to identify cable and its pin mapping.

Requirements:

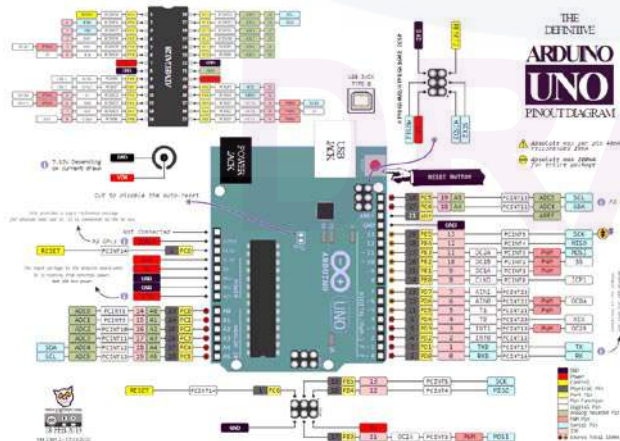
- Arduino UNO
- DC power jack
- USB
- VIN pin

PROCEDURE:

Introduction:

We'll be taking a closer look at the Arduino hardware, and more specifically, the Arduino Uno pinout. Arduino Uno is based on the ATmega328P by Atmel. The Arduino Uno pinout consists of 14 digital pins, 6 analog inputs, a power jack, USB connection and ICSP header. The versatility of the pinout provides many different options such as driving motors, LEDs, reading sensors and more. In this post, we'll go over the capabilities of the Arduino Uno pinout.

Arduino Uno Pinout – Diagram



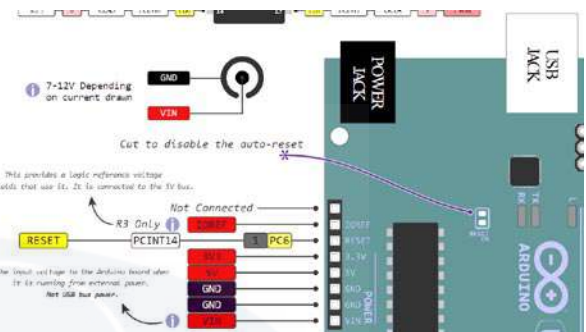
Arduino Uno pinout - Power Supply

There are 3 ways to power the Arduino Uno:

Barrel Jack - The Barrel jack, or DC Power Jack can be used to power your Arduino board. The barrel jack is usually connected to a wall adapter. The board can be powered by 5-20 volts but the manufacturer recommends to keep it between 7-12 volts. Above 12 volts, the regulators might overheat, and below 7 volts, might not suffice.

VIN Pin - This pin is used to power the Arduino Uno board using an external power source. The voltage should be within the range mentioned above.

USB cable - when connected to the computer, provides 5 volts at 500mA.



There is a polarity protection diode connecting between the positive of the barrel jack to the VIN pin, rated at 1 Ampere.

The power source you use determines the power you have available for your circuit. For instance, powering the circuit using the USB limits you to 500mA. Take into consideration that this is also used for powering the MCU, its peripherals, the on-board regulators, and the components connected to it. When powering your circuit through the barrel jack or VIN, the maximum capacity available is determined by the 5 and 3.3 volts regulators on-board the Arduino.

5v and 3v3

They provide regulated 5 and 3.3v to power external components according to manufacturer specifications.

GND

In the Arduino Uno pinout, you can find 5 GND pins, which are all interconnected.

The GND pins are used to close the electrical circuit and provide a common logic reference level throughout your circuit. Always make sure that all GNDs (of the Arduino, peripherals and components) are connected to one another and have a common ground.

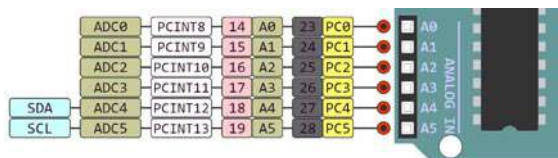
RESET - resets the Arduino

IOREF - This pin is the input/output reference. It provides the voltage reference with which the microcontroller operates.

Arduino Uno Pinout - Analog IN

The Arduino Uno has 6 analog pins, which utilize ADC (Analog to Digital converter).

These pins serve as analog inputs but can also function as digital inputs or digital outputs.



Analog to Digital Conversion

ADC stands for Analog to Digital Converter. ADC is an electronic circuit used to convert analog signals into digital signals. This digital representation of analog signals allows the processor (which is a digital device) to measure the analog signal and use it through its operation.

Arduino Pins A0-A5 are capable of reading analog voltages. On Arduino the ADC has 10-bit resolution, meaning it can represent analog voltage by 1,024 digital levels. The ADC converts voltage into bits which the microprocessor can understand.

One common example of an ADC is Voice over IP (VoIP). Every smartphone has a microphone that converts sound

waves (voice) into analog voltage. This goes through the device's ADC, gets converted into digital data, which is transmitted to the receiving side over the internet.

Arduino Uno Pinout - Digital Pins

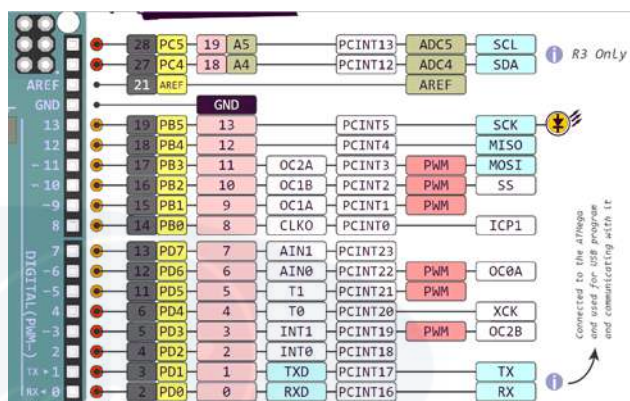
Pins 0-13 of the Arduino Uno serve as digital input/output pins.

Pin 13 of the Arduino Uno is connected to the built-in LED.

In the Arduino Uno - pins 3,5,6,9,10,11 have PWM capability.

It's important to note that:

- Each pin can provide/sink up to 40 mA max. But the recommended current is 20 mA.
- The absolute max current provided (or sank) from all pins together is 200mA.



Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Crimp and Test RJ9 / RJ11 / RJ45 connectors

Objective:

- to Crimp and Test RJ9/RJ11/RJ45 connectors.

Requirements:

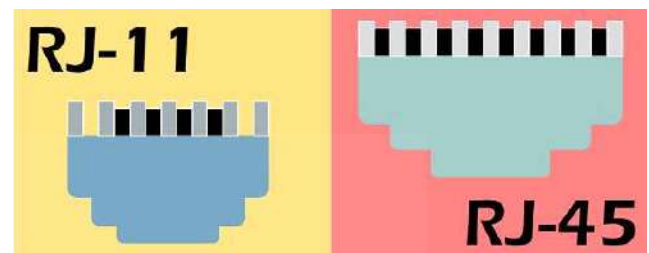
- Crimp tool
- RJ9/RJ11/RJ45 connectors

PROCEDURE:

The RJ (Registered Jack) connectors, RJ45 and RJ11 are prevalent connectors used with the UTP (Unshielded Twisted Pair) cables. These connectors can be differentiated by their size, the number of pin and structure where RJ45 has a larger size and number of pins as compared to the RJ11 connector.

RJ45 and RJ11 are the very common RJ ports which

are used for different purposes, RJ45 is newer, modular, self-securing and compact technology used for connecting the ethernet cables to different electronic devices. Conversely, RJ11 is used to terminate the conventional PSTN telephone networks.



Comparison Chart

BASIS FOR COMPARISON	RJ-45	RJ-11
Application	Used with ethernet cables	Used for connecting telephone wires
Number of wires	8	4
Type	8P8C	6P4C
Size	Large	Smaller comparatively
Bandwidth supported	10 Gbps over ethernet	24 Mbps

Definition of RJ45

The **RJ45** is an 8 pin connector used to attach the ethernet interfaces. It is known as an **8P8C connector**, which means that it contains 8 pin positions among which all of them are connectable. RJ45 describes the physical structure of the connector and how the links are wired from one end to the other.

The RJ45 connectors are used to link the Ethernet networks, it is signified by your ethernet NIC (Network Interface Controller) which have an RJ45 jack on it. We can not use coaxial connectors (RG6 and RG11) and non-modular connector DB25, instead, RJ45 is utilized by several network interface controller cards for attaching to networks like Ethernet. It falls in the class of category 5 cables, However; RJ45 does not match the TIA standard and said to be incorrect with respect to a generic 8P8C connector.

Structure

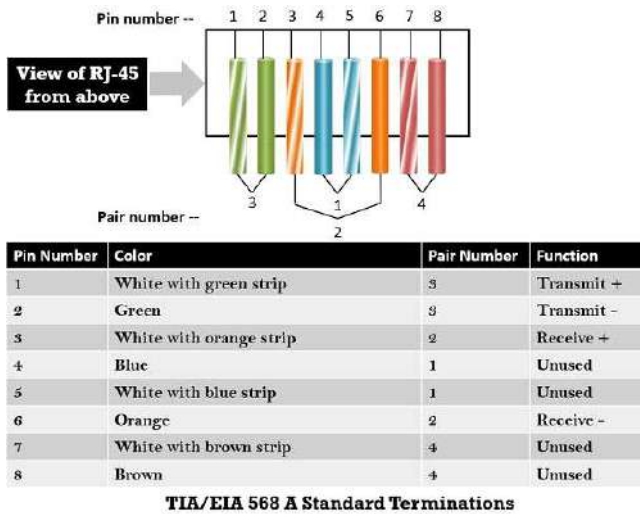
The RJ45 connector has 8 pins which are connected to the 8 coloured wires grouped into four pairs. Here, four wires (two pairs) are intended to carry the voltage and others are referred to as tip which is grounded. The connector is crimped over the end of the wire and the pin positions are numbered from the left 8 to 1.

Terminating twisted pair cable

TIA/EIA has defined two techniques for injecting the twisted pair cables into RJ45 plugs – **568A and 568B**. If these standards are followed in a correct manner then the data can be accurately transferred. The below-given diagram a and b illustrates the pin numbers and assignments for TIA/EIA 568A and TIA/EIA 568B respectively.

The networking experts consider the configuration shown in the diagram a and b as transmit and receive.

Traditionally, the T and R designation stands for tip and ring which are used when the RJ11 connectors are used to terminate old telephone lines.



Types of cables based on the termination

- **Straight-over cable** – When an identical configuration for RJ45 plugs is used to terminate a patch cable through following one of the standards shown above, then the generated cable would be known as straight-over cable. These straight-over cables are used to connect a switch to a router or

hub, a hub to a PC or server and switch to a PC or server.

- **Crossover cable** – When we terminate the RJ45 plugs at both ends of the patch cable in reverse order (i.e., Using one of the standards from TIA/EIA at one end and the other at the other end) then this cable connection is called a crossover cable. The crossover cable is utilized for linking hub to hub, switch to switch, PC to PC, router to router, etcetera.

Definition of RJ11

RJ11 is a four pins connector which is used for terminating the telephone wires. The other application of RJ11 is that it was also used with analog telephony and modems connected with computer serial ports to transfer modulated digital information using PSTN (Public Switched Telephone Network).

Structure

These are known as a 6P4C connector which signifies that it has 6 pin position but with four contacts or four wires connecting to the central junction box. Two contact positions are linked to the tip and ring while the other two contact positions are left as unused.

The other two remaining contacts from the two tip and ring contacts are used as a ground for selective ringers to avoid ringing of the bell over the other extensions through pulse dialling phones.

Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Understand Frequency Band, Gain, Antenna and Modulation for Wi-Fi

Objective:

- to understand Frequency Band, Gain, Antenna and Modulation for Wi-Fi.

Requirements:

- Wi-Fi IEEE 802.11
- WLAN
- Smart phones
- System
- Router

PROCEDURE:

Introduction

This document covers the basics of how wireless technology works, and how it is used to create networks. Wireless technology is used in many types of communication. We use it for networking because it is cheaper and more flexible than running cables. While wireless networks can be just as fast and powerful as wired networks, they do have some drawbacks.

Reading and working through **Learn Networking Basics** before this document will help you with some of the concepts used in wireless networks.

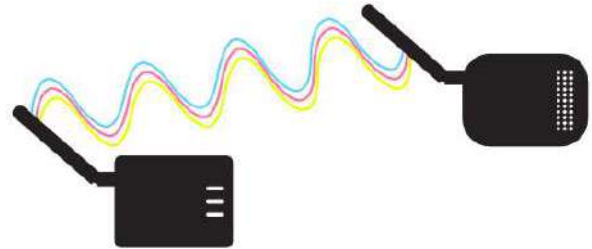
In addition to some background information, this document covers six basic concepts:

- **Wireless signals** - what they are and how signals can differ.
- **Wireless devices** - the differences and uses for receivers and transmitters.
- **Wi-Fi Modes** - how networks are made up of clients, access points, or ad-hoc devices.
- **Wi-Fi Signals** - the unique characteristics of Wi-Fi, and how signals are organized.
- **Power and Receiver sensitivity** - how far each wireless device can go, and how well a router can listen and filter out interference and noise.
- **Antennas** - how the type of antenna changes the way the router broadcasts.

Reading through this material should take about an hour. Working through the activities, or diving deeper into the subject with a group may take longer.

What is a wireless signal?

Wireless signals are important because they can transfer information — audio, video, our voices, data — without the use of wires, and that makes them very useful.



Wireless signals are **electromagnetic waves** travelling through the air. These are formed when electric energy travels through a piece of metal — for example a wire or antenna — and waves are formed around that piece of metal. These waves can travel some distance depending on the strength of that energy.

For more on how electromagnetic signals work, check the #External Resources section at the end of this document.

Types of Wireless Signals

There are many, many types of wireless technologies. You may be familiar with AM and FM radio, Television, Cellular phones, Wi-Fi, Satellite signals such as GPS and television, two-way radio, and Bluetooth. These are some of the most common signals, but what makes them different?

Frequency

First of all, wireless signals occupy a spectrum, or wide range, of frequencies: the rate at which a signal vibrates. If the signal vibrates very slowly, it has a low frequency. If the signal vibrates very quickly, it has a high frequency. Frequency is measured in Hertz, which is the count of how quickly a signal changes every second. As an example, FM radio signals vibrate around 100 million times every second! Since communications signals are often very high in frequency, we abbreviate the measurements for the frequencies - millions of vibrations a second is Megahertz (MHz), and billions of vibrations a second is Gigahertz (GHz). One thousand Megahertz is one Gigahertz.

Example Frequency Ranges

Below we can see the span of frequencies that are commonly used in communications. Broadcast transmitters for AM, FM and Television use frequencies below 1000 MHz, Wi-Fi uses two bands at higher

frequencies - 2.4 and 5GHz. Cellular phones use many different frequencies.

The frequencies from left to right:

- AM Radio: Around 10MHz
- FM Radio: Around 100MHz
- Television: Many frequencies from 470MHz to 800MHz, and others
- Cellular phones: 850MHz, 1900MHz, and others
- Wi-Fi: 2.4GHz
- Satellite: 3.5GHz
- Wi-Fi: 5GHz

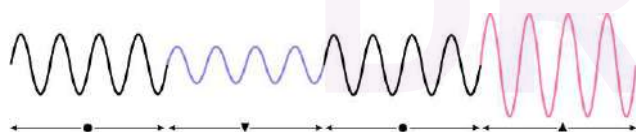
Modulation

In addition to having different frequencies, wireless signals can be different in the way they convey information. A wireless signal needs to be modulated—or changed—to send information. There are many types of modulation, and different technologies can use one or more types to send and receive information. In the two examples below — AM and FM radio — the M stands for modulation. The type of modulation is what makes them different.

Example one: AM radio. The A in AM comes from Amplitude - the energy or strength of the signal, operating at a single frequency. An un-modulated AM wave might look like:

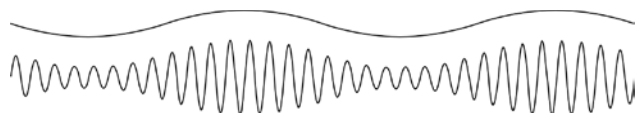


And a modulated AM radio wave has higher and lower energy (amplitude) waves indicating higher and lower audio frequencies in the signal:



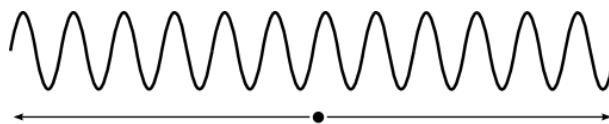
From left to right, we have the normal, un-modulated wave, then the lower amplitude wave (representing low points in audio waves), then the higher amplitude wave (representing crests or high points in audio waves).

A more detailed version of an AM signal is below:

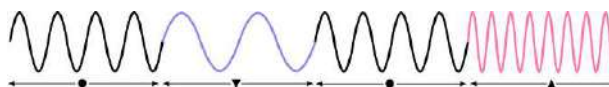


The audio signal is the wave on the top, with the corresponding Amplitude Modulated wave below it.

Example two: FM radio. The F in FM comes from Frequency - defined by how quickly the wave vibrates every second. An un-modulated FM wave might look like:



And a modulated FM radio wave has higher and lower frequencies indicating higher and lower audio frequencies in the signal:



From left to right, we have the normal, un-modulated wave, then the lower frequency wave (representing lower audio amplitudes), then the higher frequency wave (representing higher audio amplitudes).

The type of modulation various technologies use to communicate can be very different, and are often not compatible. Satellite equipment cannot speak directly to your laptop or smartphone, which uses Wi-Fi to send and receive information. This is because the radios in different devices can listen only to certain types of modulations and frequencies.



As an example, some broadcast radio receivers have a switch to select between AM and FM signals, for two reasons: they use different frequencies to transmit, and they use different modulation types. If you try and listen to an AM signal with a radio in FM mode, it won't work. The opposite is also true - in AM mode, an FM signal doesn't make sense to the receiver. It is important that transmitters and receivers use the same frequencies and modulation types to communicate.

Devices in your daily life use many types of wireless signals. Look at the table below to see the various frequencies and types of modulation each uses:

Technology or device

Type of wireless signal

Analog video - Amplitude modulated from 50MHz to 800MHz

Digital video - complex modulation from 200MHz to 800MHz

Voice - analog or digital modulation from 800MHz to 900MHz

3G, 4G or LTE - digital modulation from 1700MHz to 1900MHz and others

Bluetooth - digital modulation at 2400MHz

Walkie-talkie / two-way radio - analog AM, FM or digital modulation over many frequencies

Many types of signals - voice, audio, video, data

Many modulation types - analog and digital

Many, many frequencies - 3400MHz, 5900MHz, 10.7GHz, 14.5GHz, 23GHz, and many others.

Wi-Fi - digital modulation at 2400MHz or 5000 to 5800MHz

Bluetooth - digital modulation at 2400MHz

AM Radio - AM modulation from 0.6MHz to 1.6MHz

FM Radio - FM modulation from 88MHz to 108MHz

Nearly every device or technology uses a different wireless frequency and modulation. This means most devices can only understand a very specific kind of wireless signal.

Receivers and Transmitters

When a device sends out a wireless signal, it is called a transmitter. When another device picks up that wireless signal and understands the information, it is called a receiver. In the case of FM radio, there is one transmitter—owned and operated by the radio station—and many receivers that people listen to the station with. When a device has both a transmitter and a receiver, it is sometimes called a transceiver. Devices such as routers can both transmit and receive, which is what makes them useful for building networks—you probably want to be able to send messages to your neighbors and out to the world, as well as receive messages!

Quick Activity: What devices do you own or use frequently that are transmitters, receivers or transceivers? Fill in some examples below each type:

Transmitter	Receiver	Transceiver

Examples:	Examples:	Examples:

Do you use more transmitters, receivers, or transceivers throughout the day? What is different about the way you use each of these?		

Wi-Fi Signals

When building a network, you will be using Wi-Fi technology, which has some unique characteristics you will need to know.

There are two types of Wi-Fi signal, based on the frequencies they use:

- **2.4GHz** - A lower frequency, this is the more common Wi-Fi technology in use today. Many devices use it, so the signals can become more crowded and interfere with each other. It can pass through walls and windows fairly well.
- **5GHz** - This higher frequency technology is used by fewer devices, and can sometimes achieve higher speeds because the frequencies are less crowded. It cannot pass through walls and windows as well as the 2.4GHz band signals, so the range of 5GHz technology is often shorter.

These two types of Wi-Fi are called the **Frequency Bands**, or just **Bands** for short.

Each frequency band used in Wi-Fi is divided up into multiple “channels”. Each channel is similar to rooms at a party - if one room is crowded it is hard to carry on a conversation. You can move to the next room, but that might get crowded as well. As soon as the building is full, it becomes difficult to carry on a conversation at the party.

2.4GHz Band

For the 2.4GHz band, there are 14 channels total. Unfortunately, these channels overlap, so they aren't all usable at the same time. If you are setting up a mesh network — all of the mesh links will need to be on the same channel.

The available channels vary depending on where you are in the world. For example, in the United States channels 12, 13 and 14 are not allowed for Wi-Fi, as those frequencies are used by TV and satellite services.

If you are building networks in the United States, you can only use channels 1 through 11. In the rest of the world, channels 1 through 13 are generally usable, and in a few places channel 14 is available.

Despite that, the best channels in the United States and most of the world to use for 2.4GHz band equipment are channels 1, 6, and 11. This will minimize interference caused by partially overlapping Wi-Fi signals:

You could use other sets of Wi-Fi channels, as long as they are 5 channels apart - for instance 3, 8 and 13. This may not be optimal though, as channels 1 and 2 would be unused, and in many places in the world channel 13 is not available. Wherever you are, try and check what channels are most in use, and plan your network to use a channel that doesn't overlap.

5GHz Band

The 5GHz frequency band is much wider and has more channels, so the diagram is a bit more extensive. Fortunately, these channels do not overlap, so you don't have to worry about picking non-standard channels like in the 2.4GHz band.

There are many more channels available in the 5GHz band, so it should be easier to select a channel in this band that doesn't cause interference. This may not always be true — more and more wireless equipment is starting to use the 5GHz

In the United States, only channels available for building mesh networks are 36, 40, 44, 48, 149, 153, 157, 161, and 165. There are other channels available for Access Points or other types of community networks, but those channels won't work with mesh wireless. The best place to check what is allowed in your area is online. Links are provided in [External Resources](#) at the end of this document.

When setting up your wireless network, you will need to think about what frequency band to use, and what channel to use.

Power and Receiver Sensitivity

Many people want to know how far wireless signals will go. Knowing this is important for planning a network, as the power of the routers will affect the design of the network, and how much equipment is needed.

Different Wi-Fi routers can have very different power levels. Some are much stronger: they have more speaking or transmitting power than others. Some are very good listeners: they have what is called a better receive sensitivity. These two elements define how well wireless devices will connect, and how far away a receiving Wi-Fi router can be.

Manufacturers do not usually publish information about their router's transmit power or receive sensitivity. Instead, the manufacturer will give a generic "range" rating to their routers, usually relative to each other. In some cases, usually with more business or professional oriented equipment you can find the information for transmit power and receive sensitivity.

A router's transmit power can be measured with two scales — milliwatts (mW) or dBm:

- A **milliwatt** is one thousandth (that's 1/1000) of a single watt - which is a generic measurement of power. For instance, a light bulb might be 40 watts. A router will have an output power of 100mW, which is 400 times less!
- A **dBm** is a relative measurement using logarithms. One milliwatt is 0 dBm. 10 milliwatts is 10 dBm; 100 milliwatts is 20 dBm, and so on. This is the scale that many network designers use to calculate if longer wireless links will work.

A few examples of the transmit power levels in common Wi-Fi hardware is below:

10mW (10dBm): Laptop or smartphone, or very low cost Wi-Fi router.

About 25 to 50 meters

100mW (20dBm): Indoor home or office router.

About 50 to 100 meters

100mW (20dBm): Outdoor sector router.

About 5 to 10 kilometers

500mW (1/2 Watt or 27dBm): Outdoor, long distance focused routers.

About 10 to 20 kilometers or more

Wireless transmitter power is only one half of the connection. The Wi-Fi receiver has a range of power levels it can hear—the "listen power" in the diagram above. This is also known as the **receive sensitivity**. The receive sensitivity values are generally rated in dBm, and are usually in the range of -40dBm to -80dBm. The negative number indicates a very small signal — tiny fractions of a milliwatt.

Below we have an example of two routers in relatively close range. They have a good connection because the signal strength between them is strong.

As a receiver moves away from a wireless router, the signal it hears will get "quieter" — in other words, the power it receives will go down. Below, we can see the same routers, but with more distance between them. In this case, the routers have a weaker connection because the signal is near the limit of what the routers can hear. The speed between the routers will be less.

If the router moves too far away from the transmitter, it won't be able to receive any signal, either due to the signal being too weak or other signals interfering, and the routers will disconnect. Below we can see the two routers have disconnected, as there isn't enough signal.

The optimal signal range for outdoor wireless equipment is between -40dBm and -60dBm. This will ensure the connection can maintain the highest bandwidth possible.

Antennas

Wireless routers have different types of antennas. Some

routers will have antennas built in, and sometimes the routers will have a choice of antenna you can attach to the router. There are many specific types of antennas, but three basic types are used most of the time, and will be useful in building a wireless network. The first type of antenna is also the most common—omnidirectional.

Omnidirectional Antennas

An omnidirectional antenna sends a signal out equally in all directions around it.

Using omnidirectional antennas has the benefit of creating connections in any direction. You don't have to do as much planning to connect with multiple neighbors or buildings. If there is enough signal between nodes, they should connect.

The all-direction strength of these antennas comes with the drawback of transmitting a weaker signal. Since the signal is going in all directions, it spreads out and gets weaker with distance very fast. If nodes or clients are far away, they may not connect well.

Also, if there are only nodes or clients in one direction of the router, then the signals going in the opposite direction are wasted:

Directional Antennas

The next type of antenna is known as directional—it sends out a signal in a more focused way. There are two main types of directional antennas:

Sector Antenna

Focused Antenna

Sector antennas send out a pie-shaped wedge of signal - it can be anywhere between 30 degrees and 120 degrees wide. These are often long, rectangular antennas that are separate or integrated in to a router. A focused antenna sends out a narrow beam of signal - it is normally around 5 to 10 degrees wide, but it can be a little wider as well. These are often dishes or have a mesh bowl reflecting signal behind them.

Using directional antennas has the benefit of increasing the distance a signal will travel in one direction, while reducing it in all other directions. Since the signal is all going one way, the power that would be sent out in all directions with omnidirectional nodes is now focused, increasing the power in that direction.

It can also decrease the interference received at the node. There are fewer signals coming in to the antenna, since the node is only listening to signals from the direction it is pointing. It won't hear signals behind it or to the sides as well or at all. This reduces the signals it needs to sort out, and allows it to focus on other signals more, increasing the quality of those connections.

However, directional antennas also have the drawback of requiring more planning to create links in your neighborhood. Since you are defining and limiting the areas where wireless signals go, you need to think about how those signals cover your neighborhood. If there are areas that are then left out, how will those areas be included in the network?

Also, the node has a very powerful signal in a single direction. If omnidirectional units, or lower power units such as laptops, are connecting to the node, they may not connect properly. The laptop will hear the node very well, but the directional node may not hear the laptop. This will create the situation where it looks like there is a strong signal, but you cannot connect.

Quick Activity: What are the best uses for the different kinds of antennas?

Antenna Type Best Uses

Antenna Type	Best Uses
Omnidirectional	Sector Focused

What would the best antennas to use for building a neighborhood network?

Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Understand Frequency Band, Gain, Antenna and Modulation for GPRS

Objective:

- to understand Frequency Band, Gain, Antenna and Modulation for GPRS.

Requirements:

- GPRS
- System

PROCEDURE:**Introduction:**

Wireless technology is used in many types of communication. We use it for networking because it is cheaper and more flexible than running cables. While wireless networks can be just as fast and powerful as wired networks, they do have some drawbacks.

What is a wireless signal?

Wireless signals are important because they can transfer information — audio, video, our voices, data — without the use of wires, and that makes them very useful.

Wireless signals are electromagnetic waves travelling through the air. These are formed when electric energy travels through a piece of metal — for example a wire or antenna — and waves are formed around that piece of metal. These waves can travel some distance depending on the strength of that energy.

Types of Wireless Signals

There are many, many types of wireless technologies. You may be familiar with AM and FM radio, Television, Cellular phones, Wi-Fi, Satellite signals such as GPS and television, two-way radio, and Bluetooth. These are some of the most common signals, but what makes them different?

Frequency

First of all, wireless signals occupy a spectrum, or wide range, of frequencies: the rate at which a signal vibrates. If the signal vibrates very slowly, it has a low frequency. If the signal vibrates very quickly, it has a high frequency. Frequency is measured in Hertz, which is the count of how quickly a signal changes every second. As an example, FM radio signals vibrate around 100 million times every second! Since communications signals are often very high in frequency, we abbreviate the measurements for the frequencies - millions of vibrations a second is Megahertz (MHz), and billions of vibrations a second is Gigahertz (GHz). One thousand Megahertz is one Gigahertz.

Example Frequency Ranges

Below we can see the span of frequencies that are commonly used in communications. Broadcast transmitters for AM, FM and Television use frequencies below 1000 MHz, Wi-Fi uses two bands at higher frequencies - 2.4 and 5GHz. Cellular phones use many different frequencies.

The frequencies from left to right:

- AM Radio: Around 10MHz
- FM Radio: Around 100MHz
- Television: Many frequencies from 470MHz to 800MHz, and others.
- Cellular phones: 850MHz, 1900MHz, and others
- Wi-Fi: 2.4GHz
- Satellite: 3.5GHz
- Wi-Fi: 5GHz

Modulation

In addition to having different frequencies, wireless signals can be different in the way they convey information. A wireless signal needs to be modulated—or changed—to send information. There are many types of modulation, and different technologies can use one or more types to send and receive information. In the two examples below — AM and FM radio — the M stands for modulation. The type of modulation is what makes them different.

Example one: AM radio. The A in AM comes from Amplitude - the energy or strength of the signal, operating at a single frequency.

An un-modulated AM wave might look like:

And a modulated AM radio wave has higher and lower energy (amplitude) waves indicating higher and lower audio frequencies in the signal:

From left to right, we have the normal, un-modulated wave, then the lower amplitude wave (representing low points in audio waves), then the higher amplitude wave (representing crests or high points in audio waves).

A more detailed version of an AM signal is below:

The audio signal is the wave on the top, with the corresponding Amplitude Modulated wave below it.

Example two: FM radio. The F in FM comes from Frequency - defined by how quickly the wave vibrates every second. An un-modulated FM wave might look like:

And a modulated FM radio wave has higher and lower frequencies indicating higher and lower audio frequencies in the signal:

From left to right, we have the normal, un-modulated wave, then the lower frequency wave (representing lower audio amplitudes), then the higher frequency wave (representing higher audio amplitudes).

The type of modulation various technologies use to communicate can be very different, and are often not compatible. Satellite equipment cannot speak directly to your laptop or smartphone, which uses Wi-Fi to send and receive information. This is because the radios in different devices can listen only to certain types of modulations and frequencies.

As an example, some broadcast radio receivers have a switch to select between AM and FM signals, for two reasons: they use different frequencies to transmit, and they use different modulation types. If you try and listen to an AM signal with a radio in FM mode, it won't work. The opposite is also true - in AM mode, an FM signal doesn't make sense to the receiver. It is important that transmitters and receivers use the same frequencies and modulation types to communicate.

Devices in your daily life use many types of wireless signals. Look at the table below to see the various frequencies and types of modulation each uses:

Receivers and Transmitters

When a device sends out a wireless signal, it is called a transmitter. When another device picks up that wireless signal and understands the information, it is called a receiver. In the case of FM radio, there is one transmitter—owned and operated by the radio station—and many receivers that people listen to the station with. When a device has both a transmitter and a receiver, it is sometimes called a transceiver. Devices such as routers can both transmit and receive, which is what makes them useful for building networks—you probably want to be able to send messages to your neighbors and out to the world, as well as receive messages!

Wi-Fi Signals

When building a network, you will be using Wi-Fi technology, which has some unique characteristics you will need to know.

There are two types of Wi-Fi signal, based on the frequencies they use:

2.4GHz - A lower frequency, this is the more common Wi-Fi technology in use today. Many devices use it, so the signals can become more crowded and interfere with each other. It can pass through walls and windows fairly well.

5GHz - This higher frequency technology is used by fewer devices, and can sometimes achieve higher speeds because the frequencies are less crowded. It cannot pass through walls and windows as well as the 2.4GHz band signals, so the range of 5GHz technology is often shorter.

These two types of Wi-Fi are called the Frequency Bands, or just Bands for short.

Each frequency band used in Wi-Fi is divided up into multiple “channels”. Each channel is similar to rooms at a party - if one room is crowded it is hard to carry on a conversation. You can move to the next room, but that might get crowded as well. As soon as the building is full, it becomes difficult to carry on a conversation at the party.

2.4GHz Band

For the 2.4GHz band, there are 14 channels total. Unfortunately, these channels overlap, so they aren't all usable at the same time. If you are setting up a mesh network — all of the mesh links will need to be on the same channel.

The available channels vary depending on where you are in the world. For example, in the United States channels 12, 13 and 14 are not allowed for Wi-Fi, as those frequencies are used by TV and satellite services. If you are building networks in the United States, you can only use channels 1 through 11. In the rest of the world, channels 1 through 13 are generally usable, and in a few places channel 14 is available.

Despite that, the best channels in the United States and most of the world to use for 2.4GHz band equipment are channels 1, 6, and 11. This will minimize interference caused by partially overlapping Wi-Fi signals:

You could use other sets of Wi-Fi channels, as long as they are 5 channels apart - for instance 3, 8 and 13. This may not be optimal though, as channels 1 and 2 would be unused, and in many places in the world channel 13 is not available. Wherever you are, try and check what channels are most in use, and plan your network to use a channel that doesn't overlap.

5GHz Band

The 5GHz frequency band is much wider and has more channels, so the diagram is a bit more extensive. Fortunately, these channels do not overlap, so you don't have to worry about picking non-standard channels like in the 2.4GHz band.

There are many more channels available in the 5GHz band, so it should be easier to select a channel in this band that doesn't cause interference. This may not always be true — more and more wireless equipment is starting to use the 5GHz

In the United States, only channels available for building mesh networks are 36, 40, 44, 48, 149, 153, 157, 161, and 165. There are other channels available for Access Points or other types of community networks, but those channels won't work with mesh wireless.

Power and Receiver Sensitivity

Many people want to know how far wireless signals will go. Knowing this is important for planning a network, as the power of the routers will affect the design of the network, and how much equipment is needed.

Different Wi-Fi routers can have very different power levels. Some are much stronger: they have more speaking or transmitting power than others. Some are very good listeners: they have what is called a better receive sensitivity. These two elements define how well wireless devices will connect, and how far away a receiving Wi-Fi router can be.

Manufacturers do not usually publish information about their router's transmit power or receive sensitivity. Instead, the manufacturer will give a generic "range" rating to their routers, usually relative to each other. In

some cases, usually with more business or professional oriented equipment you can find the information for transmit power and receive sensitivity.

A router's transmit power can be measured with two scales — milliwatts (mW) or dBm:

- A milliwatt is one thousandth (that's 1/1000) of a single watt - which is a generic measurement of power. For instance, a light bulb might be 40 watts. A router will have an output power of 100mW, which is 400 times less!
- A dBm is a relative measurement using logarithms. One milliwatt is 0 dBm. 10 milliwatts is 10 dBm; 100 milliwatts is 20 dBm, and so on. This is the scale that many network designers use to calculate if longer wireless links will work.



Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Understand Frequency Band, Gain, Antenna and Modulation for RF

Objective:

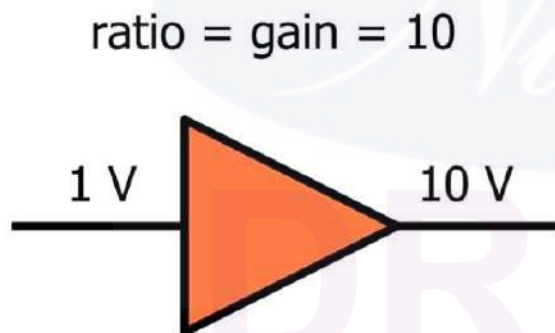
- Know the relationship of carrier frequency, modulation frequency and modulation index to efficiency and bandwidth
- Compare FM systems to AM systems with regard to efficiency, bandwidth and noise.

PROCEDURE:

Learn about the decibel and its variants in the context of RF design and testing.

RF engineering, like all scientific disciplines and subdisciplines, involves quite a bit of specialized terminology. One of the most important words that you will need when working in the world of RF is “dB” (and some variants thereof). If you become deeply entrenched in an RF project, you may find that the word “dB” becomes as familiar to you as your own name.

As you probably know, dB stands for decibel. It’s a logarithmic unit that provides a convenient way of referring to ratios, such as the ratio between the amplitudes of an input signal and an output signal.



We won’t cover the generic details of decibels because they are already available on [this page](#) of the AAC [Electric Circuits textbook](#). Instead, we will focus on practical aspects of the decibel in the specific context of RF systems.

Relative, Not Absolute

It is easy to forget that dB is a relative unit. You cannot say, “The output power is 10 dB.”

Voltage is an absolute measurement because we always speak of a potential difference, i.e., the difference in potential between two points; usually we are referring to the potential of one node with respect to a 0 V ground node. Current is also an absolute measurement because the unit (amperes) involves a specific amount of charge with respect to a specific amount of time. In contrast, dB is a unit that involves the logarithm of a ratio between

two numbers. A straightforward example is amplifier gain: If the power of the input signal is 1 W and the power of the output signal is 5 W, we have a ratio of 5:

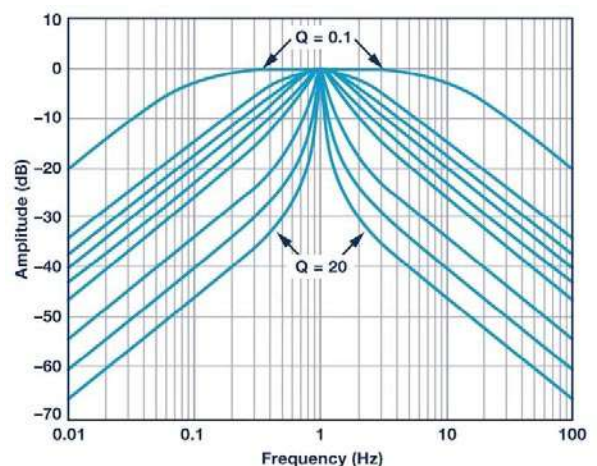
$$10 \log_{10}(\text{POUTPIN}) = 10 \log_{10}(5) \text{ dB} \quad 10 \log_{10} a (\text{POUTPIN}) = 10 \log_{10} a (5) \text{ dB}$$

Thus, this amplifier provides 7 dB of power gain—i.e., the ratio between the output signal strength and the input signal strength can be expressed as 7 dB.

Why dB?

It would certainly be possible to design and test RF systems without the use of dB, but in practice dB’s are everywhere. One advantage is that the dB scale allows us to express very large ratios without using very large numbers: a power gain of 1,000,000 is only 60 dB. Also, the total gain or loss of a signal chain is easily computed in the dB domain, because the individual dB figures are simply added (whereas multiplication would be required if we were working with ordinary ratios).

Another advantage is something that we’re familiar with from our experience with filters. RF systems revolve around frequencies and the various ways in which frequencies are generated, controlled, or affected by components and parasitic circuit elements. The dB scale is convenient in a context such as this because frequency response plots are intuitive and visually informative when the frequency axis uses a logarithmic scale and the amplitude axis uses a dB scale.



A Bode plot showing the magnitude response of different band-pass filters. Image courtesy of AnalogDialogue.

When dB Is Absolute

We've established that dB is a ratio and thus cannot describe the absolute power or amplitude of a signal. However, it would be awkward to be constantly switching back and forth between dB and non-dB values, and perhaps this is why RF engineers developed the dBm unit.

We can avoid the "ratios only" problem by simply creating a new unit that always includes a reference value. In the case of dBm, the reference value is 1 mW. Thus, if we have a 5 mW signal and we want to stay within the realm of dB, we can describe this signal as having a power of 7 dBm:

$$10\log_{10}(5 \text{ mW} / 1 \text{ mW}) = 10\log_{10}(5) \approx 7 \text{ dBm}$$
$$10\log_{10}(5 \text{ mW} / 1 \text{ mW}) = 10\log_{10}(5) \approx 7 \text{ dBm}$$

You definitely want to familiarize yourself with the concept of dBm. This is a standard unit used in real-life RF system development, and it's very convenient when, for example, you are calculating a [link budget](#), because gains and losses expressed in dB can simply be added to or subtracted from the output power expressed in dBm.

There is also a dBW unit; this uses 1 W for the reference value instead of 1 mW. Nowadays most RF engineers are working with relatively low-power systems, and this probably explains why dBm is more common.

More dB Variants

Two other dB-based units are dBc and dBi.

Instead of a fixed value such as 1 mW, dBc uses the strength of the carrier signal as the reference. For example, phase noise (discussed in page 2 of this chapter) is reported in units of dBc/Hz; the first part of this unit indicates that the phase noise power at a specific frequency is being measured with respect to the power of the carrier (in this case "carrier" refers to the signal strength at the nominal frequency).

An idealized point-source [antenna](#) receives a certain amount of energy from the transmitter circuit and radiates it equally in all directions. These "isotropic" antennas are considered to have zero gain and zero loss.

Other antennas, however, can be designed to concentrate radiated energy in certain directions, and in this sense an antenna can have "gain." The antenna is not actually adding power to the signal, but it effectively increases the transmitted power by concentrating electromagnetic radiation according to the orientation of the communication system (obviously this is more practical when the antenna designer knows the spatial relationship between the transmitter and receiver).

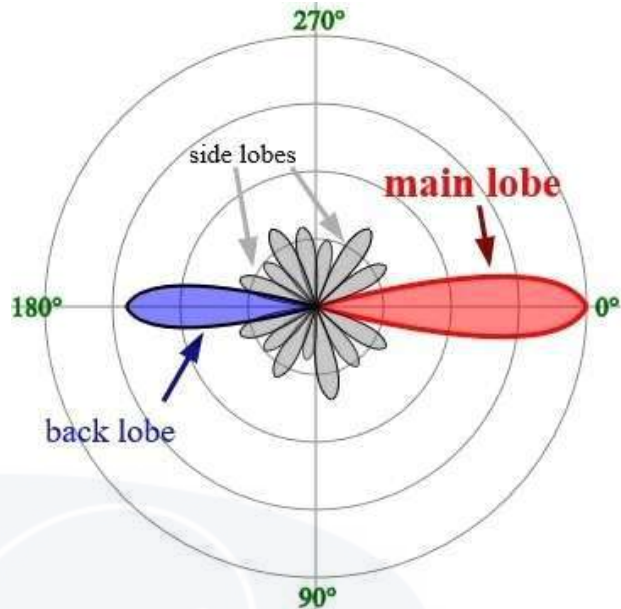


Image from Timothy Truckle (own work) [GFDL]. Here you can see the unequal distribution of radiated energy that results in gain in the forward direction (i.e., 0°).

The dBi unit allows antenna manufacturers to specify a "gain" figure that uses the ever-popular dB scale. As always, we need a ratio when we're working with dB, and in the case of dBi, the antenna gain is given with reference to the gain of an isotropic antenna.

Some antennas (such as those accompanied by a parabolic dish) have significant amounts of gain, and thus they can make a nontrivial contribution to the range or performance of an RF system.

Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Design and Test Local Area Networks over Ethernet & Wi-Fi. Selecting Appropriate LAN Technology

Objective:

- to design and test Local Area Networks over Ethernet & Wi-Fi.

Requirements:

- System
- Local Area Network
- Ethernet
- Wi-Fi

PROCEDURE:

- You should consider four main factors when selecting a LAN technology:
- Cost efficiency
- Installed base
- Maintainability
- Performance

Cost Efficiency

One of my central assumptions throughout this book is that the network is built for some business reason. It may not directly involve making money, but there must be some benefit to having the network that justifies the expense of building it. Clearly, the benefit is never infinite, so as network designers, we have a responsibility to build a network that meets the requirements for the lowest possible cost.

This problem is particularly important in the selection of network technologies. The classic example is that Token Ring cards for PCs are more expensive than the equivalent Ethernet cards. This fact alone has explained why so many organizations have undergone expensive changes in their LAN infrastructure to use more cost-effective options. As discussed previously, Token Ring has many performance benefits over Ethernet. But if the cost of Ethernet is low enough and the cost of Token Ring is high enough, then you can engineer around the performance benefits to build an Ethernet network that is at least as good as Token Ring, but less expensive. Or, you may decide to spend more money on Token Ring and get better performance.

Similarly, you could get a high-performance network by running Gigabit Ethernet to every desk. But the cost of doing this would be orders of magnitude higher than the same network using Fast Ethernet. There may still be

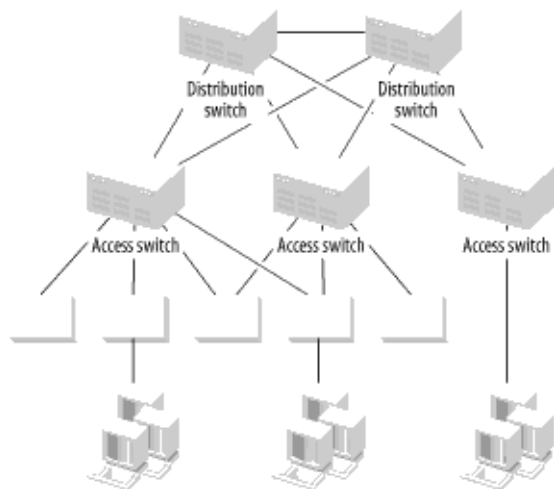
valid business reasons for wanting to build the faster network. However, it is more likely that a hybrid of the two approaches would meet all of the business requirements with a much more attractive budget.

In general, faster technology is more expensive. This is not universally true, however. Fast Ethernet equipment has become nearly ubiquitous, making the cost of building a Fast Ethernet network similar to the cost of building a regular 10Mbps Ethernet. This is even truer of the 4Mbps and 16Mbps Token Ring—it is now difficult to find Token Ring equipment that doesn't support both standards.

The other important cost/performance decision in both Ethernet- and Token Ring-based networks is the granularity of shared and switched segments. The finest granularity network has a switch port for every end device, which has significant performance benefits—particularly because it allows full-duplex operation. However, switch ports are generally more expensive than hub ports. A more cost-effective solution might involve a hybrid network in which some important end devices are directly attached to switch ports, while others are grouped in small numbers on hubs.

Another important economy involves the use of unmanageable Access devices. Small workgroup hubs and switches with no management capabilities are available for remarkably low prices. In the same vein, it is still possible to build an old-fashioned 10Base2 network, using a long piece of coax cable (often called "thin-net"), for almost nothing.

These inexpensive Access options definitely have their place. They may be ideal for the home or small office LAN. They can also be used to increase the effective port density of the network's Access Level by allowing small groups of users to share ports, as shown in Figure. This figure shows a Distribution Area containing two Distribution switches and three Access switches. Workgroup hubs and workgroup switches are connected to these Access switches. Some users are connected through the workgroup devices and some are connected directly to the Access switches. Note that I have shown some of these workgroup devices with dual attachments to the Access switches to provide extra redundancy.



Increasing effective port density of the LAN Access Level with unmanageable devices

This approach works well, but two main structural disadvantages should be considered. First, even if the end devices are able to connect to a workgroup switch at full-duplex Fast Ethernet speeds, they are still constrained by the uplink speed to the Access switch. If the Access switch is also Fast Ethernet, then remember that these end devices must share that link. This option may or may not be acceptable, given the application traffic patterns.

The second disadvantage is the increased probability of failure. The diagram shows that some of the workgroup devices have dual connections to the Access switches, and having these connections is a good way of helping to reduce the net probability of failure. However, workgroup devices are generally not built for the same level of serious use as the chassis switches that I prefer for the Access switches. Specifically, they often have external power supplies of similar quality to those used for low-end consumer electronics.

Augmenting the network's Access Level with workgroup hubs or switches (or passive MAUs in Token Ring networks) is sometimes a reasonable way to reduce costs. Giving up manageability can be dangerous, though, or at least inconvenient. Connecting end devices directly to Access switches allows control over their VLAN membership. Connecting these devices through an intermediate workgroup hub or switch, however, generally means that every device on the workgroup hub or switch must be part of the same VLAN. This requirement affects flexibility.

A more serious problem is the loss of fault management information. An unmanageable workgroup hub or switch cannot tell you when one of the devices misbehaves or when a cable is faulty. It can't tell you when its power supply is overheating. You might be able to get some information about an ill-behaved device somewhere on a workgroup hub by looking at the more complete management information on the Access switch. It can

be difficult to narrow down which device is in trouble, though.

Most seriously, if there are problems with one or more devices connected to a workgroup switch, then the only noticeable result will be performance problems for the other devices in that workgroup. The workgroup switch will not pass bad frames to the Access switch,[1] and it can't complain about the bad frames it receives from its end devices. It is possible to have a serious problem that simply will never be seen unless the users are diligent about complaining.

Installed Base

Installed base is another facet of cost effectiveness. The chances are slim that you are building a new network from scratch. In most cases, there is existing equipment, existing applications, servers, and a cable plant. A significantly cheaper alternative network technology may be available. If migrating to that means that you have to absorb a high cost in changing your installed base, then simply staying with the existing technology may be more cost-effective.

For example, a large company may make extensive use of native Token Ring protocols to connect to legacy mainframe equipment. Token Ring equipment is more expensive than Ethernet equipment, but after factoring in the cost of replacing the mainframe, rewriting the applications to use TCP/IP, and changing every end device to use this new application, they probably won't want to make the change.

This is where it is useful to have a long-term strategic information technology vision for the entire organization. If you have a long-term goal to phase out these legacy applications, then you need to build a network that can accommodate a phased-in migration to the target technology. Perhaps you will migrate the Core of the network from Token Ring to Fast and Gigabit Ethernet with TCP/IP routing and use DLSw to tunnel the native Token Ring protocols. Then, when the new servers and applications are available, you can migrate user devices in relatively small groups. Maintainability

One of the biggest potential hidden costs in a network is maintenance. I have mentioned how using unmanageable workgroup devices in the Access Level of the network can make it harder to find problems. I previously mentioned that the design principle of simplicity makes network maintenance easier. Remember that these are not just annoyance factors for the engineer who gets stuck with the ultimate responsibility for running the network. There are costs associated with these issues.

The trouble is that quantifying these costs can be extremely difficult. How can you tell, for example, that cutting a particular corner will result in needing an extra staff member to keep the network going? Only experience can tell you what the hidden costs are. In general, since your design goals are centered on reliability, the more corners you cut, the less reliable the

results will be. Lower reliability generally translates into higher maintenance costs.

Performance

And this topic brings us to performance considerations. You always want to build the fastest and best network you can for the money. Of course, by "best," I mean that the network best fulfills the business application requirements. A brilliant network with unbelievable throughput and low latency is useless if it doesn't support the applications for which it was built.

I mention performance last because it is far too easy to get absorbed in abstract issues of technology improvement. You always have to bear in mind that a network is built for a business reason. It has a budget that is based on how much money this business goal is worth to the organization. If you spend more on building and maintaining the network than it is worth to the organization, either through money saved or revenue earned, then the network actually hurts the organization more than it helps.

Within these limitations, your goal is to build the best network that you can. That also implies that you have to select technology that is appropriate to what you want to accomplish. Part of a LAN may serve an environment where cabling is impossible, so wireless technology could be a natural fit. But wireless technology tends to offer relatively poor bandwidth and latency compared to a similar network built with Fast Ethernet. When selecting appropriate technology, you have to be sensitive to these trade-offs and understand the strengths and weaknesses of the different options available to you.

Ethernet and Fast Ethernet

Ethernet is a bus topology LAN technology with a collision-based mechanism for dealing with contention. Physically, there are several different options for implementing an Ethernet network. I am generally including Fast Ethernet in these comments because the similarities between Ethernet and Fast Ethernet are strong. I will explicitly note where the comments do not apply to both.

Physical implementations of Ethernet and Fast Ethernet are generally determined by their IEEE designations. For 10Mbps standard Ethernet, the most common option today is 10BaseT. This option uses standard twisted pair cabling, such as Category 5 (although 10BaseT also works well over Category 3 cable plants). Other options include 10Base2 and 10Base5, which implement the LAN bus with an extended cable.

In 10Base2 (also called "thin-net"), the cable is an inexpensive 50 Ω impedance coaxial cable that is terminated at both ends with an impedance-matching resistor. Devices connect to the wire by means of T-junction connectors along the length of the cable. Some end devices are equipped with 10Base2 connectors, but a transceiver is frequently required. I will discuss transceivers later in this section.

10Base5 (also called "thick-net") is less common these

days because it is difficult to maintain and considerably more expensive than higher speed options. This system uses a thick coaxial cable with considerably longer distance limitations than 10Base2 (500 meters for 10Base5 versus 185 meters for 10Base2). Devices connect to the wire using a "vampire tap," which uses a retractable spike to connect to the wire in the middle of the cable. A transceiver is then required to connect this tap connector to the end device.

It is safe to consider both 10Base2 and 10Base5 as essentially obsolete technology, but they are still in use in some older networks, which is why I mention them here.

Besides copper-based Ethernet technologies, several different fiber optic systems are grouped together under the general IEEE title of 10BaseF. The most common 10Mbps fiber optic Ethernet standard is 10BaseFL. Other options exist, such as 10BaseFB and 10BaseFP. The term FOIRL (Fiber Optic Inter-Repeater Link) is often used generically to describe any 10BaseF transceiver, although technically, FOIRL describes an earlier standard.

Since the same fiber optic cabling is capable of transmitting Gigabit Ethernet, there is seldom much point in installing new 10BaseF systems. It is still used primarily in places where the distance limitations on copper Ethernet standards make it necessary to use fiber optic cable, which has much longer distance capabilities. The dominant flavors of Fast Ethernet are 100BaseTX, which runs over standard Category 5 twisted pair cabling, and 100BaseFX, which uses a fiber optic cable.

Designations such as 10BaseT may appear mysterious and arbitrary, but they have simple logic. The first part of the designation refers to the theoretical peak bandwidth—in this case, it is 10Mbps. For 100BaseT, it is 100Mbps. The word "Base" signifies baseband rather than broadband signaling. Baseband simply means that there is just one carrier frequency. Broadband, on the other hand, can multiplex several different signals on the same medium by transmitting them with different carrier frequencies.

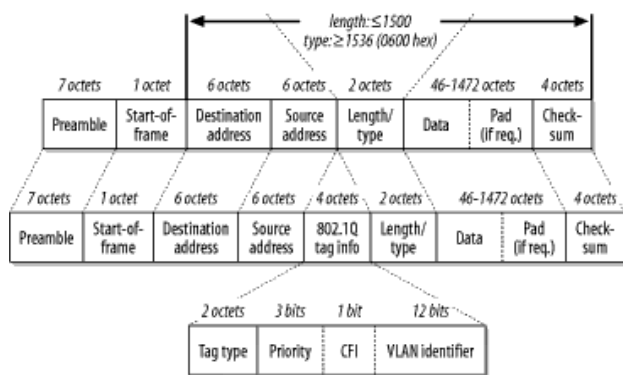
The last part is used inconsistently. The "2" in 10Base2 means 200 meters for the maximum distance of a segment, while the "5" in 10Base5 stands for 500 meters. When twisted pair standards such as 10BaseT came along, the developers probably felt that designating the type of medium was more important. Instead of calling the new twisted pair Ethernet standard 10Base1 to show that it has a 100-meter distance limit, it was called 10BaseT to designate that it operates over twisted pair cabling. Similarly, when the fiber optic standards were developed, the letter "F" was adopted to designate this different cabling standard.

The naming standards start to get a little strange when we get to names like 100VG-AnyLAN (actually, 100VG-AnyLAN isn't really Ethernet at all, because it doesn't use collisions to control contention). If the reader wants more details on these standards and the naming

conventions, it is best to look at the reference section of this book to find other books that focus more specifically on these matters.

Ethernet Framing Standards

Figure shows the standard 802.3 Ethernet frame structure. Several standard fields are defined, and they must all be present in some form. An installed base doesn't need to cripple a network, but it can limit your design options temporarily.



Ethernet framing formats, including 802.1Q VLAN tagging

The frame starts with a "preamble." The preamble consists of a string of seven bytes of the binary pattern "10101010" to indicate that the device is about to start sending a frame. Then the eighth byte, called the "start of frame delimiter," is nearly the same as the preamble except for the last bit: "10101011". The preamble and the start of frame delimiter are not included in the frame length counter. Once you get past the preamble and start of frame delimiter, you get into the interesting parts of the Ethernet frame. Three important fields are in the frame header: the source and destination MAC addresses and the length/type field.

All Ethernet MAC addresses are 6 bytes long. Every network interface card (NIC) has a globally unique address "burned-in" to it. It is possible to override this burned-in address (BIA) to create a locally administered address (LAA). However, there are also more special-purpose MAC addresses, such as multicast and broadcast addresses. I will discuss these special-purpose addresses later in this book.

The destination MAC address is always first. This gives the network devices every possible advantage in forwarding packets as quickly as possible. Modern high-speed networking equipment is able to read the frame as it is received. Since the network usually only need to look at where the packet is going, if the destination address is first, it is often possible to start directing the frame to the appropriate destination port just from this information.

The source MAC address comes next. This is the address of the device that sent the frame. Note that it is not necessarily the originator of the packet. If the packet

came from an intermediate device such as a router, then the source address will be that of the router. This address is included mostly for the benefit of the recipient device, which needs to know where to send its responses. If the return path needs to pass through a router, then the router's address needs to be here.

The third important field in the Ethernet frame header is the multipurpose "length/type" field (also called Ethertype). This 2-byte number could either be a length or a type. The only way to tell the difference is that the maximum valid length is 1500 bytes.[2] If the value in this field is less than or equal to 1500, it is interpreted as a length.

Similarly, anything larger than 1500 must be a type. Just to be absolutely certain, there is a small gap to the nearest "round" number in hexadecimal, 0600, which is 1536 in decimal. The actual values in the type field represent different protocols, and the IEEE keeps track of these values. An up-to-date list of assigned values is available online from the IEEE web site at <http://standards.ieee.org/regauth/ethertype/type-pub.html>. This list includes a very large number of companies that have registered particular Ethernet protocol types, although only a handful of types are commonly seen in most production LANs.

Novell reserves Ethernet types 8137 and 8138 for IPX. Type 8137 designates an older version of IPX that is not widely used anymore, while 8138 is the most typical for modern IPX installations. Apple's Ethernet protocol uses type code 809B. The Banyan Network Operating System uses 0BAD and 0BAF, and 8191 is reserved for NetBEUI, which is frequently used for PC file-sharing systems. The most common type field values are 0800, used for frames containing standard IP packets, and 0806, used for IP ARP packets.

In most LANs, IPX uses the length rather than the type. If you look at a protocol analyzer, you will probably see all of the IPX frames with a length/type value of something less than 05DC (the hex value of the decimal number 1500).

TCP/IP, on the other hand, almost universally uses the type rather than length. The reason for the difference is largely historical. The type interpretation is used by an earlier standard for defining Ethernet frames, called Ethernet II. The length interpretation, on the other hand, is the method employed by the newer IEEE 802.3 standard.

It should be obvious why it is more efficient to use the type field as a type rather than as a length. If any one protocol prefers to use its length, then that protocol has effectively trampled over 1500 possible type codes. Furthermore, it is much more efficient if the protocol stacks of both the end and network devices don't have to read into the data portion of the frame before they can figure out what type of packet it is. Not every device cares about every protocol (particularly when the packets are received as broadcasts), so knowing whether they should bother decoding any given packet is useful. But there are other benefits to this system.

For protocols that use the 802.3 standard, it is necessary to use another method for identifying the type. Using this method is done by adding Sub-Network Access Protocol (SNAP) information to the packet. SNAP is part of the LLC (Logical Link Control) sublayer of the Data Link Layer. It is defined as an extension to the 802.2 standard. The presence of a type rather than a length value in the "length/type" field automatically tells the receiving station to look for LLC information later in the packet.

This process may sound complicated, but it allows greater flexibility in the protocol. Rather than a single type field, 802.2 allows the creation of an arbitrary Protocol Data Unit (PDU), which can be used to contain a huge variety of extensions to the protocol. This LLC PDU information is tacked on to the start of the data portion of the packet, immediately after the standard Ethernet header information. In effect, it looks like another type of header, placed after the MAC header.

Also note that the 802.2 LLC sublayer is not unique to Ethernet. Exactly the same SNAP PDU that defines IPX in an Ethernet frame can be used to define IPX in a Token Ring frame.

SNAP is just one simple example of this type of a PDU. Inside the SNAP PDU is a field that defines that protocol type.

At the end of every 802.3 Ethernet frame is a 4-byte checksum called Frame Check Sequence (FCS). This checksum is a relatively simple method of ensuring that the packet was not damaged as it crossed through the network. Generally, one doesn't expect to see very many checksum errors in a stable Ethernet network. Those

that are seen are usually caused by other problems, such as late collisions. However, when random electrical problems are on a link, these checksums are useful in finding them.

This checksum is calculated on the entire Ethernet frame from the Destination Address right up to the Data (and possible padding). If the payload protocol contains another checksum, it provides an extra layer of certainty. When there are checksum failures, it can also be used to investigate which layers of the protocol see the problem. For example, if the Ethernet level FCS field is good, but the TCP checksum is bad, then the problem must have existed before the packet hit this part of the network.

The same 802.3 Ethernet frame used for 10Mbps Ethernet is also used for 100Mbps, Gigabit, and 10 Gigabit Ethernet. The same MAC multicast and broadcast addresses are used by all of these standards. The use of these addresses makes life much easier for the network designer because it means that you can freely mix these different standards to fit your immediate needs.

For example, you can have Gigabit Ethernet trunks connecting your Distribution switch to your Access switches. Then you can have 100Mbps Fast Ethernet links to some workstations, and even step down to workgroup hubs of standard half-duplex 10BaseT for the less active user workstations. Throughout this complex hybrid of media types, the same Ethernet frames can be transmitted without change.

DRAFT

Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Design and Test Local Area Networks over Ethernet & Wi-Fi. Selecting Appropriate LAN Technology

Objective:

- to design and test Local Area Networks over Ethernet & Wi-Fi.

Requirements:

- System
- Local Area Network
- Ethernet
- Wi-Fi

PROCEDURE:

Introduction

The Service Managed Gateway (SMG) contains a GSM (Global System for Mobile communications) 3G wireless modem. The modem uses a SIM card to create a dial-up WAN connection to the Internet.

Typical uses of the GSM (3G) modem

- You can use the GSM (3G) modem on the GW Series router in two ways.
- You can create a primary WAN link over the GSM (3G) modem.



A primary WAN link over the 3G modem

You can use the GSM (3G) modem as a backup WAN link. If the primary interface link fails, the GW Series router uses its GSM (3G) modem to automatically route traffic across the Internet via its alternative interface. This set up is not covered in this document. For further information contact Virtual Access.



primary link using the 3G modem on the GW SERIES ROUTER SMG

Types of data bearers that the GSM (3G) modem supports

Data bearers are frame protocols that transport data streams. The GSM (3G) modem supports GPRS, HSDPA, and CSD.

GPRS

GPRS (General Packet Radio Service) uses packet-switched technology to transmit small bursts of data. The 3G modem on the GW Series router supports multi-slot class 6, which allows a maximum data rate of 384Kbps class 3. The maximum data rate available to the 3G modem depends on provisioning by the network operator.

HSDPA

HSDPA (High-Speed Downlink Packet Access) uses circuit-switched technology to transmit data at speeds of up to 1.8Mbps.

CSD in EGSM900/GSM1800 MHz

CSD (Circuit-Switched Data) mode enables the 3G modem to transmit data over the EGSM (Extended Global System for Mobile communications) frequency or the GSM frequency. The modem behaves as a dual-band mobile phone behaves.

Configuring the SMG

The Service Managed Gateway (SMG) contains an internal web server that is used to configure the SMG. Before you can access the internal web server and start the SMG configuration, you must ensure that your PC has the correct networking set up.

To enable and configure connections on your SMG, the gateway must be correctly installed, and a valid service must be configured on it.

When your Service Managed Gateway is correctly connected to your PC, type fast. start into the URL line of your browser to display the Start page.



The SMG start page

If a login page appears type in the login password you received from your administrator.

If you have not received a password, contact the Virtual Access Support team.

Access the Fast Start Wizard by clicking the Fast. Start icon on the Start page of the embedded web.

The Fast Start Wizard will guide you through a series of forms that you must complete to configure your SMG.

Setting up the GSM (3G) modem for a primary WAN link

You set up the GSM (3G) modem for a primary WAN link in the Expert View of the Advanced section of the web of the SMG.

On the SMG Start page, click Advanced.

In the Advanced menu, click Expert View.

Configuring the GW Series router to use the GSM (3G) modem interface

To configure the GW Series router to use the GSM (3G) modem interface, follow the steps given below. You do not have to save each configuration until you have

updated all your settings. When you have completed all your configuration changes,

step 1: Configure the IP default route

In the Expert View menu, select **system -> ip -> default route**. The IP Default Route page appears.

Default IP Route page

Field name	Description	Command line
Configured	Enables or disables the default route. Set Configured to Yes .	Set Ip Route Default Configured
Route Type	Indicates whether the route type is over a numbered or unnumbered link. A numbered link is a link that has been assigned an IP address. Set Route Type to unnumbered .	Set Ip Route Default Numbered
Next Hop For Numbered Interfaces	When Route Type is set to numbered , enter the IP address in dotted-decimal notation, of an adjacent router. The local device sends traffic to this router when a route to a destination is not known. Set Next Hop For Numbered Interfaces fields to 0.0.0.0 .	Set Ip Route Default Next Hop Ip
Next Hop For Unnumbered Interfaces	Selects the interface as the default route. Set Next Hop For Unnumbered Interfaces to ppp-2 .	Set Ip Route Default Next Hop Interface

The IP default route fields, their descriptions and command lines

Click Update. The Configuration Update Result page appears.

Smart City - Identify and test Wired and Wireless communication medium such as RS232, RS485, Ethernet, Fibre optics, Wi-Fi, GSM, GPRS, RF etc and communication protocol

Design and Test Personal Area Networks over RF

Objective:

- to design and test Personal Area Network over RF.

Requirements:

- System
- ZigBee

PROCEDURE:

- WLAN Radio Frequency Design Considerations

Introduction

This chapter describes the basic radio frequency (RF) information necessary to understand RF considerations in various wireless local area network (WLAN) environments. This chapter includes information on the following topics:

- Regulatory domains and frequencies
- Understanding the IEEE 802.11 standards
- RF spectrum implementations including 802.11b/g and 802.11a
- Planning for RF deployment
- Manually fine-tuning WLAN coverage
- Radio Resource Management (RRM), also known as Auto-RF

RF Basics

This section provides a summary of regulatory domains and their operating frequencies.

Regulatory Domains

Devices that operate in unlicensed bands do not require any formal licensing process, but when operating in these bands, the user is obligated to follow the government regulations for that region. The regulatory domains in different parts of the world monitor these bands according to different criteria, and the WLAN devices used in these domains must comply with the specifications of the relevant governing regulatory domain. Although the regulatory requirements do not affect the interoperability of IEEE 802.11b/g and 802.11a-compliant products, the regulatory agencies do set certain criteria in the standard. For example, the emission requirements for WLAN to minimize the amount of interference a radio can generate or receive from another radio in the same proximity. It is the responsibility of the vendor to get the product certified from the relevant regulatory body. Table 3-1 summarizes the current regulatory domains for Wi-Fi products. The main regulatory domains are FCC, ETSI, and the MKK.

Besides following the requirements of the regulatory agencies, many vendors also ensure compatibility with other vendors through the Wi-Fi certification program (www.wi-fi.org).

Table 3-1 Regulatory Domains

Regulatory Domain	Geographic Area
Americas or FCC (United States Federal Communication Commission)	North, South, and Central America, Australia and New Zealand, various parts of Asia and Oceania
Europe or ETSI (European Telecommunications Standards Institute)	Europe (both EU and non EU countries), Middle East, Africa, various parts of Asia and Oceania
Japan (MKK)	Japan
China	People's Republic of China (Mainland China)
Israel	Israel
Singapore ¹	Singapore
Taiwan ¹	Republic of China (Taiwan) ¹

The regulations of Singapore and Taiwan for wireless LANs are particular to these countries only for operation in the 5 GHz band. Singapore and Taiwan are therefore only regulatory domains for 5 GHz operation; for operation in 2.4 GHz, they fall into the ETSI and FCC domains, respectively.

Note See the Cisco website for compliance information and also check with your local regulatory authority to find out what is permitted within your country. The information provided in Table 3-2 and Table 3-3 should be used as a general guideline.

Operating Frequencies

The 802.11b/g band regulations have been relatively constant, given the length of time it has been operating. The FCC allows for 11 channels, ETSI allows for up to 13 channels, and Japan allows up to 14 channels, but requires a special license to operate in channel 14.

For 802.11a, countries are moving to open the frequency range 5.250-5.350 GHz (UNII-2) and the frequency range

5.470 to 5.780 GHz for additional 802.11a channels. These various frequencies are covered in more detail in the specific 802.11 sections in this chapter.

802.11b/g Operating Frequencies and Data Rates

Ratified in September 1999, the 802.11b standard operates in the 2.4 GHz spectrum and supports data rates of 1, 2, 5.5, and 11 Mbps. 802.11b enjoys broad user acceptance and vendor support. 802.11b technology has been deployed by thousands of enterprise organizations, which typically find its speed and performance acceptable for their current applications.

The 802.11g standard, which was ratified in June 2003, operates in the same spectrum as 802.11b and is backward-compatible with the 802.11b standard. 802.11g supports the additional data rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mbps. 802.11g delivers the same 54 Mbps maximum data rate as 802.11a, but operates in the same 2.4 GHz band as 802.11b. It also provides backward compatibility with existing 802.11b devices.

Table 3-2 lists the various 802.11b/g channel frequencies and specifies whether a regulatory agency allows their use in their domain. Note that not all of these frequencies are available for use in all regulatory domains.

Table 3-2 Operating Frequency Range for 802.11b and 802.11g

Channel Identifier	Center Frequency	FCC (America)	ESTI (EMEA)	TELEC (Japan)	MOC (Israel Outdoor) ¹
1	2412	X	X	X	
2	2417	X	X	X	
3	2422	X	X	X	
4	2427	X	X	X	
5	2432	X	X	X	X
6	2437	X	X	X	X
7	2442	X	X	X	X
8	2447	X	X	X	X
9	2452	X	X	X	X
10	2457	X	X	X	X
11	2462	X	X	X	X
12	2467		X	X	X
13	2472		X	X	X
14 ²	2484			X	

¹ Israel allows channels 1 through 13 indoors.

² Japan requires a special license for channel 14.

802.11a Operating Frequencies and Data Rates

Operating in the unlicensed portion of the 5 GHz radio band, 802.11a is immune to interference from devices that operate in the 2.4 GHz band, such as microwave ovens, cordless phones, and Bluetooth (a short-range, low-speed, point-to-point, personal-area-network wireless standard). Because the 802.11a standard operates in a different frequency range, it is not compatible with existing 802.11b or 802.11g-compliant wireless devices, but it does mean that 2.4-GHz and 5-GHz equipment can operate in the same physical environment without interference.

Choosing between these two technologies (802.11b/g and 802.11a) does not involve a one-for-one trade-off. They are complementary technologies and will continue to coexist in future enterprise environments. Those

responsible for implementing these technologies must be able to make an educated choice between deploying 2.4 GHz-only networks, 5 GHz-only networks, or a combination of both. Organizations with existing 802.11b networks cannot simply deploy a new 802.11a network for existing APs and expect to have their 802.11a 54 Mbps coverage in the same areas as their 11Mbps 802.11b coverage. The technical characteristics of both these bands simply do not allow for this kind of coverage interchangeability.

802.11a provides data rates of 6, 9, 12, 18, 24, 36, 48, with a maximum data rate of 54 Mbps, though generally at shorter ranges for a given power and gain, but it has up to 23 nonoverlapping frequency channels (depending on the geographic area) compared to the three nonoverlapping channels of 802.11b/g, which results in increased network capacity, improved scalability, and the ability to create microcellular deployments without interference from adjacent cells.

The 5 GHz band in which 802.11a operates is divided into several different sections. Each of the Unlicensed National Information Infrastructure (UNII) bands presented in Table 3-3 was originally intended for different uses, but all can currently be used by indoor 802.11a with appropriate power restrictions. Initially, the FCC defined only the UNII-1, UNII-2, and UNII-3 bands, each of which had four channels. The channels were spaced 20 MHz apart with an RF spectrum bandwidth of 20 MHz, thereby providing nonoverlapping channels.

There are differing limitations on these three UNII bands. Restrictions vary between them for transmit power, antenna gain, antenna styles, and usage. The UNII-1 band is designated for indoor operations, and initially had a restriction of permanently attached antennas. The UNII-2 band was designated for indoor or outdoor operations, and permitted external antennas. The UNII-3 band was intended for outdoor bridge products and permitted

external antennas, but the UNII-3 band can now be used for indoor or outdoor 802.11a WLANs as well.

The channels in UNII-1 (5.150 to 5.250 GHz) are 34, 36, 38, 40, 42, 44, 46, and 48. The channels in UNII-2 (5.250-5.350 GHz) are 52, 56, 60, 64 and require Dynamic Frequency Selection (DFS) and Transmitter Power Control (TPC). The channels in the new frequency range (5.470-5.725 GHz) are 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140 and require DFS and TPC. The channels in UNII-3 are 149, 153, 157, 161, 165 and require DFS and TPC. Not all channels in a given range can be used in all of the regulatory domains. Figure 3-1 shows the various channels in the UNII-1, 2, and 3 bands, along with the additional 11 new channels.

For more information on FCC regulation updates, see the following URL: http://www.cisco.com/en/US/products/hw/wireless/ps469/products_white_paper0900aecd801c4a88.shtml

Table 3-3 Operating Frequency Range for 802.11a

Channel Identifier	36	40	44	48	52	56	60	64	149	153	157	161
Center Frequency	5180	5200	5220	5240	5260	5280	5300	5320	5745	5765	5785	5805
Band	UNII-1				UNII-2				UNII-3			

Table 3-3 shows the standard 802.11a frequencies. Table 3-4 shows the specific frequency bands and channel numbers for a few specific regulatory domains.

Table 3-4 Additional Frequency Bands and Channel Numbers for Other Regulatory Domains

Regulatory Domain	Frequency Band	Channel Number	Center Frequency
Japan1	U-NII lower bands	36	5.180
		40	5.200
		44	5.220
		48	5.240
Singapore	U-NII lower band	36	5.180
		40	5.200
		44	5.220
		48	5.240
Taiwan		52	5260
		56	5280
		60	5300
		64	5320
EMEA 1 Australia New Zealand	Same as USA	Same as USA	Same as USA
EMEA 2 ²	U-NII lower band	36	5.180
		40	5.200
		44	5.220

1 Japan is changing from channels 34,38,42,46 to 36,40,44,48 and adding channels 52, 56, 60, and 64 but required Dynamic Frequency Selection on channels 52 - 64. Cisco Equipment with the old channels are designated with a -J, Cisco equipment with the new channels are designated with a -P.

2 Some EMEA countries, such as Denmark and Germany, are limited to 20 mW.

In February of 2004, the FCC released a revision to the regulations covering the 5 GHz 802.11a channel usage. This revision added 11 additional channels, bringing the available channels capacity to 23 channels (see Figure 3-1). The new additional 11 channels will be for Indoor/Outdoor use. To use the 11 new channels, however, radios must comply with two features that are part of the 802.11h specification: transmit power control (TPC) and dynamic frequency selection (DFS). DFS is required to avoid radar that operates in this frequency range, but it can also be used for other purposes, such as dynamic frequency planning. 802.11h has been supported since Cisco Unified Wireless Network Software Release 3.1.

Figure 3-1 802.11 Channel Capacity

Understanding the IEEE 802.11 Standards

IEEE 802.11 is the working group within the Institute for Electrical and Electronics Engineers (IEEE) responsible for wireless LAN standards at the physical and link layer (Layer 1 and Layer 2) of the OSI model, as compared to the Internet Engineering Task Force (IETF, which works on network layer (Layer 3) protocols. Within the 802.11 working group are a number of task groups that are responsible for elements of the 802.11 WLAN standard. Table 3-5 summarizes some of the task group initiatives.

For more information on these working groups, see the following URL:

<http://www.ieee802.org/11/>

Table 3-5 IEEE 802.11 Task Group Activities

Task Group	Project
MAC	To develop one common MAC for WLANs in conjunction with a physical layer entity (PHY) task group
PHY	To develop three WLAN PHYs—Infrared, 2.4 GHz FHSS, 2.4 GHz DSSS
a	To develop PHY for 5 GHz UNII band
b	To develop higher rate PHY in 2.4 GHz band
c	To cover bridge operation with 802.11 MACs (spanning tree)
d	To define physical layer requirements for 802.11 operation in other regulatory domains (countries)
e	To enhance 802.11 MAC for QoS

f	To develop recommended practices for Inter Access Point Protocol (IAPP) for multi-vendor use
g	To develop higher speed PHY extension to 802.11b (54 Mbps)
h	To enhance 802.11 MAC and 802.11a PHY-Dynamic Frequency selection (DFS), Transmit Power control (TPC)
i	To enhance 802.11 MAC security and authentication mechanisms
j	To enhance the 802.11 standard and amendments to add channel selection for 4.9 GHz and 5 GHz in Japan
k	To define RRM enhancements to provide interfaces to higher layers for radio and network measurements
l	To define Radio Resource Measurement enhancements to provide interfaces to higher layers for radio and network measurements
m	To perform editorial maintenance, corrections, improvements, clarifications, and interpretations relevant to documentation for 802.11 family specifications
n	Focus on high throughput extensions (>100MB/s at MAC SAP) in 2.4GHz and/or 5GHz bands
o	To provide Fast Handoffs in Voice over WLAN (goal is around 50ms)
p	Focus on vehicular communications protocol aimed at vehicles, such as toll collection, vehicle safety services, and commerce transactions via cars
r	To develop a standard specifying fast BSS transitions and fast roaming
s	To define a MAC and PHY for meshed networks that improves coverage with no single point of failure
t	To provide a set of performance metrics, measurement methodologies, and test conditions to enable manufacturers, test labs, service providers, and users to measure the performance of 802.11 WLAN devices and networks at the component and application level
u	To provide functionality and interface between an IEEE 802.11 access network (Hotspot) and any external network
v	To provide extensions to the 802.11 MAC/PHY to provide network management for stations (STAs)

- w To provide mechanisms that enable data integrity, data origin authenticity, replay protection, and data confidentiality for selected IEEE 802.11 management frames including but not limited to: action management frames, deauthentication and disassociation frames

RF Spectrum Implementations

In the United States, three bands are defined as unlicensed: industrial, scientific, and medical (ISM) bands. The ISM bands are as follows:

- 900 MHz (902-to-928 MHz)
- 2.4 GHz (2.4-to-2.4835 GHz) (IEEE 802.11b/g operates in this frequency range)
- 5 GHz (5.15-to-5.35 and 5.725-to-5.825 GHz) (IEEE 802.11a operates in this frequency range)

Each range has different characteristics. The lower frequencies exhibit better range, but with limited bandwidth and thus lower data rates. The higher frequencies exhibit less range and are subject to greater attenuation from solid objects.

The following sections cover some of the specific RF characteristics that the various 802.11 radios use for improving communications in the 2.4 and 5 GHz frequency ranges.

Direct Sequence Spread Spectrum

Direct sequence spread spectrum (DSSS) encodes redundant information into the RF signal. This provides the 802.11 radio with a greater chance of understanding the reception of a packet, given background noise or interference on the channel. Every data bit is expanded into a string of bits, or chips, called a chipping sequence or barker sequence. The chipping rate mandated by IEEE 802.11 is 11 chips per bit. It uses binary phase-shift keying (BPSK)/quadrature phase-shift keying (QPSK) at the 1

and 2 Mbps rates and 8 chips (complimentary code keying-CCK) at the 11 and 5.5 Mbps rate. This means that at 11 Mbps, 8 bits are transmitted for every one bit of data. The chipping sequence is transmitted in parallel across the spread spectrum frequency range.

IEEE 802.11b Direct Sequence Channels

14 channels are defined in the IEEE 802.11b direct sequence (DS) channel set. Each DS channel transmitted is 22 MHz wide, but the channel separation is only 5 MHz. This leads to channel overlap such that signals from neighboring channels can interfere with each other. In a 14-channel DS system (11 usable channels in the US), only three nonoverlapping (and thus, non-interfering) channels 25 MHz apart are possible (channels 1, 6, and 11).

This channel spacing governs the use and allocation of channels in a multi-AP environment, such as an office or campus. APs are usually deployed in a cellular fashion within an enterprise, where adjacent APs are allocated nonoverlapping channels. Alternatively, APs can be co-located using channels 1, 6, and 11 to deliver 33 Mbps bandwidth to a single area (but only 11 Mbps to a single client). The channel allocation scheme is illustrated in Figure 3-2.

Figure 3-2 IEEE 802.11 DSS Channel Allocations

IEEE 802.11g

802.11g provides for a higher data rate (up to 54 Mbps) in the 2.4-GHz band, the same spectrum as 802.11b. 802.11g is backward-compatible with 802.11b and provides additional data rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mbps. At higher data rates, 802.11g uses the same modulation technique, orthogonal frequency division multiplexing (OFDM), as 802.11a (see IEEE 802.11a OFDM Physical Layer).

Table 3-6 lists 802.11g modulation and transmission types for the various data rates.

Table 3-6 802.11g Modulation and Transmission Types

Modulation	Transmission Type	Bits per Subchannel	Data Rate (Mbps)
BPSK	DSSS	NA	1
QPSK	DSSS	NA	2
CCK	DSSS	NA	5.5
BPSK	OFDM	125	6
BPSK	OFDM	187.5	9
CCK	DSSS	NA	11
QPSK	OFDM	250	12
QPSK	OFDM	375	18
16-QAM	OFDM	500	24
16-QAM	OFDM	750	36
64-QAM	OFDM	1000	48
64-QAM	OFDM	1125	54

IEEE 802.11a OFDM Physical Layer

IEEE 802.11a defines requirements for the physical layer of the OSI model, operating in the 5.0 GHz UNII frequency, with data rates ranging from 6 Mbps to 54 Mbps. It uses Orthogonal Frequency Division Multiplexing (OFDM), which is a multi-carrier system (compared to single carrier systems). OFDM allows subchannels to overlap, providing a high spectral efficiency. The modulation technique allowed in OFDM is more efficient than spread spectrum techniques used with 802.11b.

IEEE 802.11a Channels

The 802.11a channel shows the center frequency of the channels. The frequency of the channel is 10 MHz on either side of the dotted line. There is 5 MHz of separation between channels, as shown in Figure 3-3.

Figure 3-3 Channel Set

For the US-based 802.11a standard, the 5 GHz unlicensed band covers 300 MHz of spectrum and supports 12 channels. As a result, the 5 GHz band is actually a conglomerate of three bands in the USA: 5.150-to-5.250 GHz (UNII 1), 5.250-to-5.350 GHz (UNII 2), and 5.725-to-5.875 GHz (UNII 3).

RF Power Terminology

RF-specific terms, such as dB, dBi, and dBm are used to describe the amount of change in power transmitted measured at any discrete point, as perceived by the radio. The following sections cover their differences and provide a rule of thumb for their use, in addition to providing an explanation of effective isotropic radiated power (EIRP).

dB

The term decibel (dB) is mainly used for attenuation or amplification of the power level. dB is a logarithmic ratio of a signal to another standardized value. For example, dBm is where the value is being compared to 1 milliWatt, and dBw is where the value is being compared to 1 Watt.

The math is as follows:

$$\text{Power (in dB)} = 10 * \log_{10} (\text{signal/reference})$$

Plugging in some numbers (signal 100mW, reference 1mW) gives a value in dB of 20 ($100 = 10$ squared; taking the exponent 2 and multiplying by 10 gives you 20).

Remember that it is logarithmic (meaning that it increases or decreases exponentially and not linearly), and it is a ratio of some value to a reference. Also, remember that it is multiplied by 10.

Given that it is logarithmic, there are some general rules of thumb. An increase or decrease of 3 dB means that the signal doubled (double the power) or reduced in strength by 1/2, respectively. An increase or decrease of 10dB means that the signal went up by 10 times or down to 1/10th the original value.

Indoor WLAN and outdoor WLAN deployments both offer separate challenges in RF deployments, and need to be analyzed separately. However, there are some rules of thumb for indoor use. For every increase of 9dB, the

indoor coverage area should double. For every decrease of 9dB, the indoor coverage area should be cut in half.

dBi

The term dBi is used to describe the power gain rating of antennas. The real antennas are compared to an isotropic antenna (a theoretical or imaginary antenna) that sends the same power density in all directions, thus the use of dBi.

Antennas are compared to this ideal measurement, and all FCC calculations use this measurement (dBi). For example, a Cisco omni-directional AIR-ANT4941 has a gain of 2.2 dBi, meaning that the maximum energy density of the antenna is 2.2 dB greater than an isotropic antenna.

dBm

The term dBm uses the same calculation as described in the dB section, but has a reference value of 1 milliWatt.

So, taking into consideration the example previously given in the dB section, if the power jumped from 1 mW to 100mW at the radio, the power level would jump from 0 dBm to 20 dBm.

Besides describing transmitter power, dBm can also describe receiver sensitivity. Receiver sensitivity is in minus dBm (-dBm), because the signal reduces in value from its point of transmission. The sensitivity indicates the lowest power the receiver can receive before it considers the signal unintelligible. The approximate receiver sensitivity of Cisco radios is -84dBm for a 1200 series "a" radio and -90 dBm for the 1200 "g" radio.

Note The "g" radio can recover a signal at half the strength of the "a" radio. It is generally true that for a given technology, lower frequency radios can achieve a better sensitivity than higher frequency radios.

Effective Isotropic Radiated Power

Although transmitted power based on the radio setting is rated in either dBm or Watts, the maximum energy density coming from an antenna from a complete system is measured as effective isotropic radiated power (EIRP), which is a summation of the dB values of the various components. EIRP is the value that regulatory agencies, such as the FCC or ETSI, use to determine and measure power limits, expressed in terms of maximum energy density within the first Fresnel of the radiating antenna. EIRP is calculated by adding the transmitter power (in dBm) to antenna gain (in dBi) and subtracting any cable losses (in dB). For example, if you have a Cisco Aironet 350 bridge connected to a solid dish antenna by a 50 foot length of coaxial cable, plugging in the numbers gives the following:

- Bridge—20 dBm
- 50 Foot Cable—3.3 dBm (negative because of cable loss)
- Dish Antenna—21 dBi

- EIRP—37.7dBm

Planning for RF Deployment

Many of the RF-design considerations are interdependent or implementation-dependent. As a result, there is no "one-size-fits-all" template for the majority of requirements and environments.

Different Deployment Types of Overlapping WLAN Coverage

How much overlapping WLAN coverage you set in your wireless network depends on the usage, though with limited exceptions, all designs should be deployed to enable low latency coverage. Wireless networks can be deployed for location management, voice, or data-only networks, or a combination of all three. The difference is in the pattern in which the APs are laid out, and the amount of RF overlap in the coverage area. When planning a WLAN deployment consideration should be given to future uses of the WLAN deployment.

Converting a WLAN deployment to support additional services beyond a data-only deployment is not simply a matter of adding APs; it can require an additional site survey and the possible relocation of existing APs.

Data-Only Deployment

Data-only deployments do not require a large amount of overlap. This is because 802.11 clients respond to a lower signal from a nearby AP, should one fail, by stepping down their rate and taking a longer time to transmit. The required overlap is determined by the WLAN data rate requirement described in WLAN Data Rate Requirements. Minimal overlap is required for data-only networks, which allow all data rates. For data-only networks, the rule of thumb for separation of APs is typically 120-130 feet, but, when making your estimation for AP separation, remember to factor in objects that affect RF coverage, such as wall densities, machinery, elevators, or even wide-open space with steel cages, because your results can vary depending on the RF environment. RRM has been developed for this type of deployment and it is very useful for controlling the RF coverage.

Voice/Deployment

Figure 3-4 shows the voice network pattern and overlap. The APs are grouped closer together and have more overlap than a data-only installation, because voice clients should roam to a better AP before dropping packets. You generally also want to run smaller cells than in the past, and ensure the overlapping cell edges at or above -67 dBm. This accomplishes a number of things including greater homogeneity across a single cell and reducing processor load in the handheld, which increases link stability and reduces latency. Although only one AP might be required for a defined area, Cisco recommends that you have two APs on nonoverlapping channels with a received signal strength indication (RSSI) above 35 at all times in your installation, for redundancy and load balancing purposes. For the 7920 voice deployment, Cisco recommends that you have a Received Signal

Strength Indication (RSSI) above 35 at all time in your installation. This is to ensure that the VoIP phone has good reception as well as allowing some over-subscription and enhances roaming choices for the phone.

Remember that designing for low noise background is as important as relatively high energy density within the cell. This means that a good baseline power setting for the AP is in the 35-50 mW range. This generally requires approximately 15 percent more APs than if you deployed a coverage model at 100 mW. The smaller the cell, generally the more homogenous the energy density across the cell.

100 mW designs also generate significantly greater noise load into a given area; 35-50 mW designs are commonly 10 dBm or lower in background noise, even in complex coverage areas.

Pre-site surveys are useful for identifying and characterizing certain challenging areas and potential sources for interference, such as existing WLANs, rogues, and non-802.11 interference from sources such as microwave ovens and cordless telephones. Following a design that should be reviewed and approved by all stakeholders, post-site surveys should be considered as an excellent audit mechanism to ensure that the coverage model complies with the intended functional requirements as set forth by the stakeholders.

When making your estimation for separation, remember to factor in objects that affect RF coverage such as wall densities, machinery, elevators, or even wide open spaces with steel cages, because your results may vary depending on the RF environment. Be sure to include transient dynamics such as forklifts, large groups of people, or large objects moved through the area by crane or similar load bearing devices. Characterizing for other wireless deployments such as robotic systems, cranes, and so on, are also a good idea. A WLC is often a very effective method for preliminary site evaluation, by allowing a fast deployment of a WLAN infrastructure that can then be used to make RF measurements of the area; a hand-walked site survey is also effective insurance for complex areas such as those commonly found in healthcare, retail, and manufacturing.

Figure 3-4 Single Floor Site Survey for Voice

For more information on a wireless voice deployment, see Chapter 11, "VoWLAN Design Recommendations," as well as the 7920 deployment guide at the following URL: http://www.cisco.com/en/US/docs/voice_ip_comm/cuipph/7920/5_0/english/design/guide/7920ddg.html.

Location-Based Services Deployments

The third type of deployment is the location-based services (LBS) deployments, which may be the most complex of current applications because it relies not only on excellent cell coverage, but optimal location of APs. Location management deployments can simultaneously track thousands of devices by using the WLAN

infrastructure. Examples include Wi-Fi tag type deployments or asset tracking deployments to locate equipment or devices via the wireless network and/or simply to indicate where wireless clients are throughout the wireless network in relation to a drawing or diagram. This can be used to make the wireless infrastructure more secure by providing the location of a rogue client or APs, and greatly improve client troubleshooting capabilities.

For a location management deployment, the APs are laid out in a staggered pattern. Figure 3-5 shows a typical pattern. The staggered pattern allows for more accurate estimation of the location of a device.

For a discussion of Location-Based Services, see Chapter 15, "Cisco Unified Wireless Location-Based Services," and the white paper entitled WiFi Location Based Services 4.1 Design Guide, which can be found at the following URL: <http://www.cisco.com/en/US/docs/solutions/Enterprise/Mobility/WiFiLBS-DG.html>.

Figure 3-5 Example of a Single Floor Location Management Deployment

WLAN Data Rate Requirements

Data rates affect AP coverage areas. Lower data rates (such as 1 Mbps) can extend the coverage area farther from the AP than higher data rates (such as 54 Mbps) as illustrated in Figure 3-6 (which is not drawn to scale). Therefore, the data rate (and power level) affects coverage and consequently the number of APs required for the installation, as illustrated in Figure 3-7 for different data rates. As part of the planning process, consider the required data rates, the required range, and the required reliability.

Data Rate Compared to Coverage Area

Different data rates are achieved by the AP sending a redundant signal on the wireless link, allowing data to be more easily recovered from noise. The number of symbols, or chips, sent out for a packet at the 1 Mbps data rate is greater than the number of symbols used for the same packet at 11 Mbps. This means that sending data at the lower bit rates takes more time than sending the equivalent data at a higher bit rate. And when there is more than one client associated to the radio, the lower rate client affects the higher rate clients' maximum data rate by taking longer to transmit a packet of the same length.

Figure 3-6 Data Rate Compared with Coverage

The actual diameter of the coverage depends on factors such as environment, power level, and antenna gain. For example, indoors using the standard antennas on the NIC card and APs, the diameter of the 1 Mbps circle is approximately 700 feet (210 m), and the diameter of the 11 Mbps circle is about 200 feet (60 m). Increasing the gain of the antenna can increase the distance and change the shape of the radiation pattern to something more directional.

AP Density for Different Data Rates

The minimum required reliable data rate has a direct impact upon the number of APs needed in the design, along with power setting, antenna gain, and location. Figure 3-7 shows coverage comparison and AP density for different data rates. Although six APs with a minimum data rate of 2 Mbps might adequately service an area, it might take twice as many APs to support a minimum data rate of 5 Mbps, and more again to support a minimum data rate of 11 Mbps for the same coverage area.

Figure 3-7 Coverage Comparison and AP Density for Different Data Rates

The data rate you choose depends on the type of application to be supported, but should not be greater than the typical requirements because there is tradeoff in coverage. In a typical WLAN environment, the higher data rates give maximum throughput and should minimize performance-related support issues. In a WLAN vertical application environment, the data rates selected are determined by the application requirements of speed and reliability as measured by delay and jitter. The physical facility and/or whether the network is client-centric generally dictates range requirements; some clients might not support the higher data rates, longer ranges, or the delay and jitter rates of an infrastructure element such as an AP.

It might seem logical to choose the default configuration of APs and clients, thereby allowing all data rates. However, there are three key reasons for limiting the data rate to the highest rate at which full coverage is obtained:

- Broadcast and multicast (if enabled) are sent at the lowest associated data rate (to ensure that all clients can receive the packets). This reduces the throughput of the WLAN because traffic must wait until frames are processed at the slower rate.
- Clients that are farther away, and therefore accessing the network at a lower data rate, decrease the overall throughput by causing delays while the lower bit rates are being serviced. It might be better to force the clients to roam to a closer AP so as not to impact the performance of the rest of the network.
- If a 54 Mbps service is specified and provisioned with APs to support all data rates, clients at lower rates can associate with the APs that can create a coverage area greater than planned, thereby increasing the security exposure (by allowing association from outside the building) and potentially interfering with other WLANs.

Client Density and Throughput Requirements

Wireless APs have two characteristics that make actual client data throughput slower than the data rate:

- APs have an aggregate throughput less than the data rate because 802.11 provides a reliable transport

mechanism that ACKs all packets, thereby halving the throughput on the channel.

- APs are similar to shared hubs. That is, the channel is shared by all the clients associated to that AP on that channel, thus collisions slow data throughput.

With this in mind, you must have the approximate estimate of the maximum number of active associations (active clients). This can be adjusted more or less according to the particular application.

Each cell provides an aggregate amount of throughput that is shared by all the client devices that are within the cell and associated to a given AP. This basically defines a cell as a collision domain. After deciding on the

minimum data rate, be sure to consider how much throughput should, on average, be provided to each user of the wireless LAN.

Take the example of barcode scanners; 25 Kbps may be more than sufficient bandwidth for such an application because using an 802.11b AP at 11 Mbps of data rate results in an aggregate throughput of 5-6 Mbps. A simple division results in a maximum number of 200 users that can theoretically be supported. This number cannot in fact be achieved because of the 802.11 management overhead associated with the large number of clients and packet collisions. For a 1 Mbps system, 20 users can use the same AP for similar bandwidth results.



Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Explore and test series combination of Solar PV Modules

Objective:

- to explore and test series combination of solar PV Modules.
-

Requirements:

- Photovoltaic panels
- Solar panels
- System

PROCEDURE:**Series and Parallel Wiring**

The following image is a great example of series and parallel wiring.

Series Wiring

Series wiring is when the voltage of a solar array is increased by wiring the positive of one solar module to

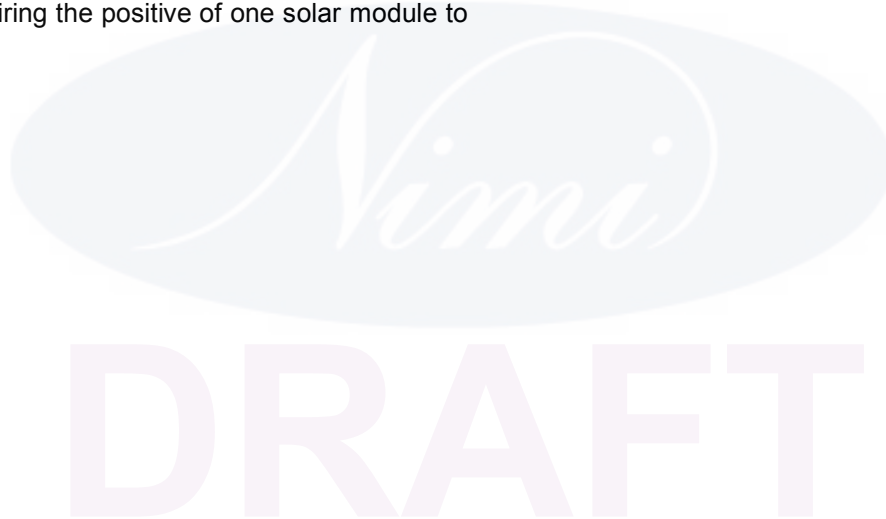
the negative of another solar module. This is similar to installing batteries in a flashlight. As you slide the batteries into the flashlight tube the voltage increases.

Parallel Wiring

Parallel wiring increases the current (amps) output of a solar array while keeping the voltage the same. Parallel wiring is when the positives of multiple modules are connected together and all the negatives for the same modules are connected together.

Series Parallel Combination

Here is an example of what is found in most large solar systems, a series and parallel wiring combination.



Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Test parallel combination of Solar PV Modules

Objective:

- To test parallel combination of solar PV Module.

Requirements:

- Photovoltaic panel
- PV modules
- Solar panel
- System

PROCEDURE:

Understand the difference between wiring your solar panels in series vs parallel.

You want your solar panels to deliver the maximum amount of energy possible, right? But did you know how your solar panels are connected within the electrical wiring of your house makes a difference in how well they work? Connecting your solar panel in series vs parallel affects current flow and is dictated by your installation's setup.

Warning: Science below! While we're not going to get too deep into the details, the difference between connecting solar panels in series vs in parallel is an intermediate level solar discussion. If you're looking for something a bit more on the beginner level, check out our articles [How Do Solar Panels Work?](#) and [Everything You Need to Know About Solar Inverter Types](#).

OK, now that the disclaimer is out of the way, we're ready!

What is a circuit?

Before we get into whether solar panels are better connected in series or in parallel, let's talk a little about wiring basics, starting with circuits. An electronic circuit is simply a path electrons can flow through. The simplest circuit is a battery, wires, and light bulb.

As electrons move through a circuit, they create **voltage** — the difference in charge between two points (measured in volts) — and **current** — the rate at which charge is flowing (measured in amps). Check out

this great 5 minute video to understand the difference between voltage and current – it's going to come in handy later on!

Each electrical component in a circuit affects current flow by changing its electrical properties. For example, resistors dissipate energy, while capacitors store electrical charge. Conductive wires enable current to flow. The configuration, or placement, of these components within the electrical circuit affects the flow of current and its electrical power.

What does “in series” and “in parallel” even mean?

Components connected in series looks like a string of Christmas lights – each piece is placed in a line, one after another, with each piece connected only to the one before and after. Since all the components are connected in a single line, the electrical current can only travel in one direction. If a circuit is disconnected at any point, current will not be able to flow through it. This will halt electrical activity in the entire circuit.

You experience this each year when you take those Christmas lights out of storage, then have to check every bulb to find the one that's burnt out and preventing the entire strand from lighting.

Fortunately, this isn't how the wiring in your house works. Most household electrical circuits are parallel circuits. Each component is connected to every other component. A parallel circuit will continue to work even if it contains a broken component because it has multiple paths for current to travel.

A broken component reduces the overall current, but the other components will stay operational. This why you can have 5 lights connected to the same circuit in your home, but choose which ones to turn on or off without affecting any of the others.

Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Check series-parallel combination of Solar PV Modules

Objective:

- to check series-parallel combination of solar PV Modules.
-

Requirements:

- Solar panels
- Different voltage
- System

PROCEDURE:

Selecting and connecting solar panels of assorted voltage or wattage in series and parallel configurations, and manufactured by different suppliers is generally a sought after question by most DIYers.

Even though connecting several solar panels is not encouraged, it's not prohibited either and everything could well be fine provided that each panel's electrical criteria (voltage, wattage, amps) are cautiously evaluated.

Should you wish to connect two solar panels manufactured by different companies in series or parallel configurations, the manufacturers are generally not the issue.

The issue remains in the conflicting electrical attributes of the solar panels, as well as their unique efficiency ratings.

If Photovoltaic devices are hooked up in series to accomplish increased output voltage. The optimum system voltage however should not be surpassed.

Having said that when panels are attached in series, one of the panel may carry a rated power below the other panel, because of the lower current spec of this solar panel with respect to the other modules in the chain, that unit could tend to drag down the existing system's output:

Combining different solar panels in series

Solar devices are normally attached in parallel to achieve greater output current. For Photo voltaic components attached in parallel absolute power is determined

Contrary to the combination in series, when solar panels are connected in parallel there may be one panel having power output below the spec of the other devices, this could perhaps not influence the total power output of the chain significantly only if this particular panel possesses voltage rating on par with the other modules voltage:

Optimum voltage on a series of modules should invariably be less than highest input DC voltage of the inverter.

While hooking up diverse solar modules, it's not the different power specifications that might be crucial, rather it's basically the current (for series connection) and voltage (for parallel connection) that might cause the draw down of the efficiency of the system.

Typically solar panels of specific or matching current needs to be connected with each other in series.

Should you connect a 3A solar panel to a 3.5A solar panel, the all round current will probably be pulled down to 3A.

This kind of a lowering of current would of course cause a loss of power output and eventually loss in equipment efficiency.

In the same way only solar panels of specific or matching voltage must be connected with each other in parallel.

Whenever you hook up a 15V panel to a 24 V panel, the overall voltage is going to be pulled down to 15 Volts. This kind of a lowering voltage will probably lead to a decrease in power output, and eventually deterioration in system efficiency.

In comparison to voltage and current, watt is not a major issue. Whenever you connect with each other a 60W solar panel to a 100W panel in series, the gross hooked up power is likely to be 160W, given that the two solar panels are of identical ampere rating.

At this point any specific difference in voltages is not crucial, voltages would simply add up and all you've might need to judge is the fact that the total voltage must come within the inverter voltage limit.

In case their current specs are not the same , just be willing to witness undesirable consequences, because the overall current could possibly be the lower of the two panel specifications, which usually shows that you're not gonna accomplish a maximum of 160W rather often way less. Just how much less - is relative to dissimilarity in specified currents.

Additionally if you connect collectively a 60W solar panels to a 100W panel in parallel, the absolute associated power is likely to be 160W, assuming that the two solar panels are of matching voltage.

Here any kind of difference in currents is not crucial, currents would simply add up and all you've need to take into account is that the absolute current must not extend past the highest inverter input current.

When their voltage specifications are not the same, you need to be ready to expect disagreeable situations, because the overall voltage could possibly be much lower than the individual units that could imply that you're not

likely to accomplish an overall of 160W rather consistently less.

Just how much less - boils down to the difference between the voltage ratings of the two.



Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Measure VI Characteristics of Solar PV Module

Objective:

- to measure VI characteristics of solar PV Module.

Requirements:

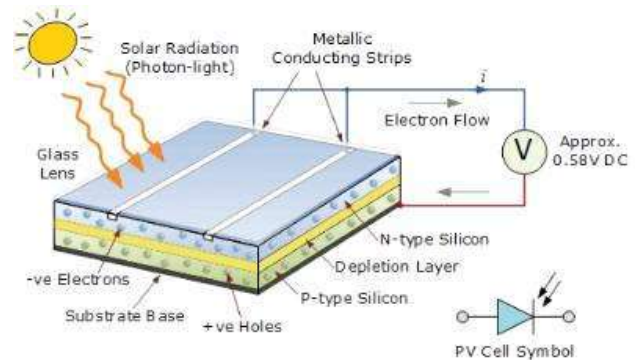
- Solar Panel
- System
- Internet Connectivity
- Potentiometer
- Multimeter
- Clamp
- Connecting wires

PROCEDURE:

Introduction:

Solar cell is the basic unit of solar energy generation system where electrical energy is extracted directly from light energy without any intermediate process. The working of a solar cell solely depends upon its photovoltaic effect, hence a solar cell also known as photovoltaic cell. A solar cell is basically a semiconductor p-n junction device. It is formed by joining p-type (high concentration of hole or deficiency of electron) and n-type (high concentration of electron) semiconductor material. At the junction excess electrons from n-type try to diffuse to p-side and vice-versa. Movement of electrons to the p-side exposes positive ion cores in n-side, while movement of holes to the n-side exposes negative ion cores in the p-side. This results in an electric field at the junction and forming the depletion region. When sunlight falls on the solar cell, photons with energy greater than band gap of the semiconductor are absorbed by the cell and generate electron-hole (e-h) pair. These e-h pairs migrate respectively to n- and p- side of the pn junction due to electrostatic force of the field across the junction. In this way a potential difference is established between two sides of the cell. Typically a solar or photovoltaic cell has negative front contact and positive back contact.

A semiconductor p-n junction is in the middle of these two contacts like a battery. If these two sides are connected by an external circuit, current will start flowing from positive to negative terminal of the solar cell. This is basic working principle of a solar cell. For silicon, the band gap at room temperature is $E_g = 1.1$ eV and the diffusion potential is $U_D = 0.5$ to 0.7 V.



Construction of a solar cell

Solar Cell I-V Characteristics Curve

Solar Cell I-V Characteristics Curve is the superposition of the I-V curves of the solar cell diode in absence (dark) and in presence of light. Illuminating a cell adds to the normal "dark"

$$I = I_0 \left[\exp \left(\frac{qV}{nkT} \right) - 1 \right] - I_L$$

where I_0 = "dark saturation current" or diode leakage current in absence of light

q = electronic charge

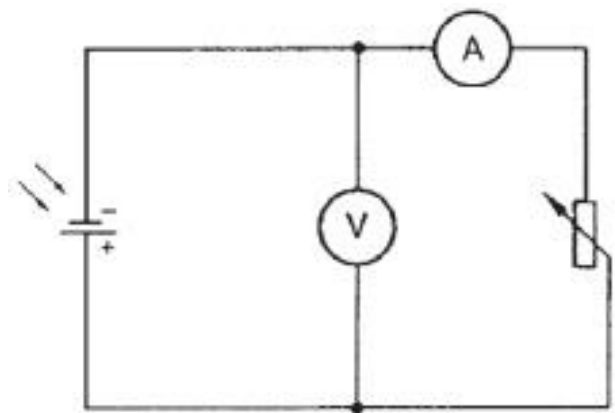
V = applied voltage across the terminals of the diode

n = ideality factor

k = Boltzmann's constant

T = temperature

I_L = light generated current.



Circuit for I-V characteristics of solar cell

A typical circuit for measuring I-V characteristics is shown in Figure. From this characteristics various parameters of the solar cell can be determined, such as: short-circuit current (I_{sc}), the open-circuit voltage (V_{oc}), the fill factor (FF) and the efficiency. The rating of a solar panel depends on these parameters.

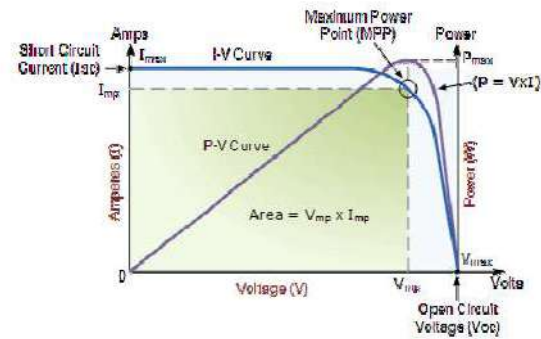
The short-circuit current is the current through the solar cell when the voltage across the solar cell is zero (i.e., when the solar cell is short circuited). is due to the generation and collection of light-generated carriers. For an ideal solar cell at most moderate resistive loss mechanisms, the short-circuit current and the light-generated current are identical. Therefore, the short-circuit current is the largest current which may be drawn from the solar cell.

The open-circuit voltage, V_{oc} , is the maximum voltage available from a solar cell, and this occurs at zero current. The open-circuit voltage corresponds to the amount of forward bias on the solar cell due to the bias of the solar cell junction with the light-generated current.

The "fill factor", more commonly known by its abbreviation "FF", is a parameter which, in conjunction with V_{oc} and I_{sc} , determines the maximum power from a solar cell. The FF is defined as the ratio of the maximum power

from the solar cell to the product of V_{oc} and I_{sc} . Graphically, the FF is a measure of the "squareness" of the solar cell and is also the area of the largest rectangle which will fit in the IV curve as shown in Figure.

The efficiency is the most commonly used parameter to compare the performance of one solar cell to another. Efficiency is defined as the ratio of energy output from the solar cell to input energy from the sun. In addition to reflecting the performance of the solar cell itself, the efficiency depends on the spectrum and intensity of the incident sunlight and the temperature of the solar cell.



A Typical I-V curve and power curve of a solar cell.

Vimi
DRAFT

Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Explore and test the Blocking Diode and Bypass Diode for solar panels

Objective:

- to explore and test the blocking diode and bypass diode for solar panels
-

Requirements:

- Blocking diode
- Battery
- Solar Panel
- Bypass diode
- Semiconductor material

PROCEDURE:**WHAT IS A DIODE?**

A diode is designed to let current flow in one direction. If you're familiar with plumbing, a diode is the electrical equivalent to a check valve. There are two purposes of diodes in a solar power system – bypass diodes and blocking diodes. The same type of diode is generally used for both, a Schottky barrier diode, but how they are wired and what they do is what makes them different.

BYPASS DIODES

Bypass diodes are used to reduce the power loss solar panels experience due to shading. Because current flows from high to low voltage, when a solar panel has cells that are partially shaded, the current is then forced through the low-voltage shaded cells. This causes the solar panel to heat up, and have severe power loss. Those shaded solar cells become consumers of electricity instead of producers.

Bypass diodes inside the junction box of a solar panel provide a low resistance path for the current to go around a series of solar cells that have been shaded. The diode is wired in parallel with the cells. Because electricity takes the path of least resistance, it is easier for the current to go through the diode than through the shaded cell, so it

does. This minimizes the heat gain, and reduces current loss.

Most solar panels have bypass diodes built in these days, so you typically won't have to worry about that anymore. However, if you have multiple solar panels wired together in series, and you consistently have shading on one or more of the solar panels, wiring a bypass diode in parallel across the shaded panel can prevent the current from being forced back through the shaded panel and cause it to heat and lose power. So, it acts the same as the internal bypass diodes, but bypasses the entire panel instead of the individual cells.

BLOCKING DIODES

Blocking diodes are used to prevent your batteries from discharging backwards through your solar panels at night. Again, current flows from high to low-voltage, so during a sunny day, the voltage of a solar panel will be higher than the voltage of a deep cycle battery and the current will naturally flow from the panel to the battery. But at night, if the solar panel is connected directly to a battery, the voltage of the solar panel is going to be lower than the voltage of the battery, so there is a possibility of some backwards flow, pulling power out of the battery. It won't be as much as the flow during the day, but there may be some.

As a result, in the days before charge controllers, people would put a blocking diode in series between the battery and the solar panel, only allowing power to go into the battery. Nowadays, most solar power systems have a charge controller between the solar panel and the battery, and this charge controller prevents this backflow of electricity, eliminating the need for a blocking diode.

Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Observe bypass diode and its working in Solar PV Module

Objective:

- to observe bypass diode and it's working in solar PV Module.

Requirements:

- Solar panel
- Bypass diode
- System

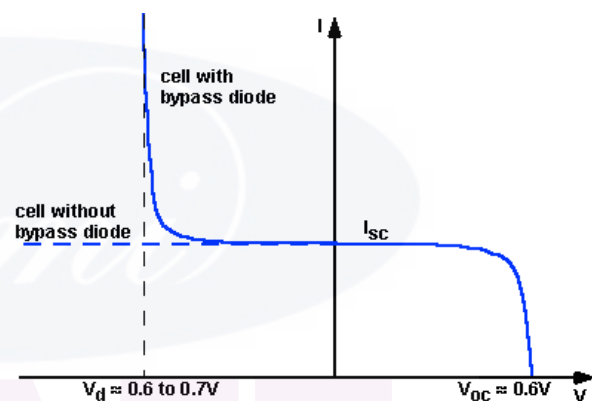
PROCEDURE:

Bypass Diodes

The destructive effects of hot-spot heating may be circumvented through the use of a bypass diode. A bypass diode is connected in parallel, but with opposite polarity, to a solar cell as shown below. Under normal operation, each solar cell will be forward biased and therefore the bypass diode will be reverse biased and will effectively be an open circuit. However, if a solar cell is reverse biased due to a mismatch in short-circuit current between several series connected cells, then the bypass diode conducts, thereby allowing the current from the good solar cells to flow in the external circuit rather than forward biasing each good cell. The maximum reverse bias across the poor cell is reduced to about a single diode drop, thus limiting the current and preventing hot-spot heating. The operation of a bypass diode and the effect on.

Current flow for two cells in series and the effect of a bypass diode. The animation progresses automatically from one condition to another.

The effect of a bypass diode on an IV curve can be determined by first finding the IV curve of a single solar cell with a bypass diode and then combining this curve with other solar cell IV curves. The bypass diode affects the solar cell only in reverse bias. If the reverse bias is greater than the knee voltage of the solar cell, then the diode turns on and conducts current. The combined IV curve is shown in the figure below.

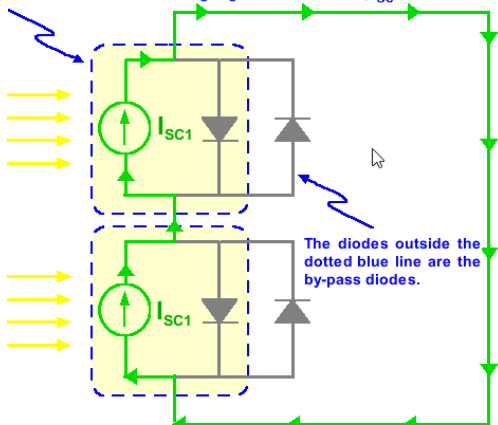


SERIES CONNECTED SOLAR CELLS WITH BYPASS DIODES

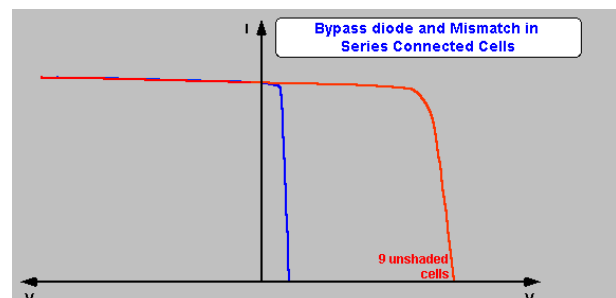
- ☒ Matched currents at short circuit
- ☐ Mismatched currents at short circuit
- ☐ Matched currents at open circuit
- ☐ Mismatched currents at open circuit

At short circuit conditions and with matched currents, the voltage across both the solar cells and the bypass diodes is zero. The bypass diodes have no effect.

The circuit elements contained within the blue dotted lines model a solar cell. The current source is the light generated current, I_{SC} .



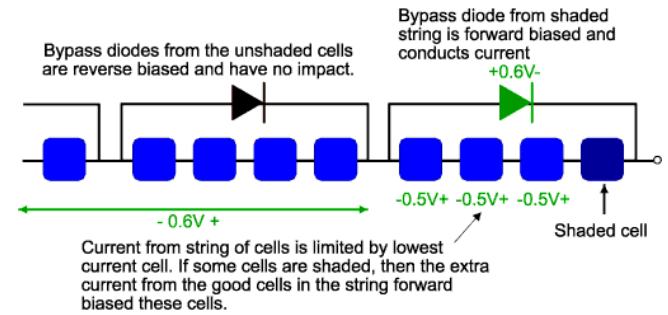
IV curve of solar cell with bypass diode.



Preventing hot-spot heating with a bypass diode. For clarity, the example uses a total of 10 cells with 9 unshaded and 1 shaded. A typical module contains 36 cells and the effects of current mismatch are even worse without the bypass diode, but are less important with the bypass diode. The animation moves automatically. You do not need to click to continue.

In practice, however, one bypass diode per solar cell is generally too expensive and instead bypass diodes are usually placed across groups of solar cells. The voltage across the shaded or low current solar cell is equal to the forward bias voltage of the other series cells which share the same bypass diode plus the voltage of the bypass diode. This is shown in the figure below. The voltage across the unshaded solar cells depends on the degree of shading on the low current cell. For example, if the cell is completely shaded, then the unshaded solar cells will be forward biased by their short circuit current and the voltage will be about 0.6V. If the poor cell is only partially shaded, then some of the current from the good cells can flow through the circuit, and the remainder is used to forward bias each solar cell junction, causing a lower forward bias voltage across each cell. The maximum power dissipation in the shaded cell is approximately equal to the generating capability of all cells in the group. The maximum group size per diode, without causing damage, is about 15 cells/bypass diode, for silicon cells. For a normal 36 cell module, therefore,

2 bypass diodes are used to ensure the module will not be vulnerable to "hot-spot" damage.



Bypass diodes across groups of solar cells. The voltage across the unshaded solar cells depends on the degree of shading of the poor cell. In the figure above, 0.5V is arbitrarily shown.



Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Measure effect of inclination angle of Solar PV Module

Objective:

- to measure effect of inclination angle of Solar PV Module

Requirements:

Solar panel

Photovoltaic cell

System

PROCEDURE:

Introduction:

A solar Photovoltaic is a method of generating electrical power by converting the solar radiation into direct current electricity using semiconductors that exhibit the photovoltaic effect.

Photovoltaic system installation has played a big role in renewable energy because PV systems are pollution free, economically reliable for long-term operation and secure energy source. The major obstruction of PV technology is its high capital costs compared to conventional energy sources and also it is the more widely used technology all over the world; there is 100GW of solar PV now installed in our world.

It is estimated that a total of some 5.3MWp of PV is now in use in Ethiopia. In April 2012 Ethiopia has established the first Polycrystalline solar PV module assembly plant with the cooperation of SKY Energy International, Metals and Engineering Corporation (METEC) and an Ethiopian state enterprise. The plant would expected to achieve 20% of Ethiopian power capacity coming from solar energy within the next five years including 3Million solar home systems distribution plan and may also be exported to a neighboring country .

Performance of these solar-photovoltaic (PV) system not only depends on its basic electrical characteristics; maximum power, tolerance rated value of percentage, maximum power voltage, maximum power current, open-circuit voltage (Voc), short-circuit current (Isc), maximum system voltage, but also is negatively influenced by several obstacles such as ambient temperature, relative humidity, dust storms and suspension in air, shading, global solar radiation intensity, spectrum and angle of irradiance.

The radiation received by cells in the PV module is lower than radiation reaching the module surface. The main causes of these energy difference are dirt accumulation on the surface of the modules, reflection and absorption losses by the materials covering the cells and the tilt angle of the PV module. To better understand the effect

of tilt angle on soiling, one has to know how it affects the current of the PV modules as it is directly proportional to the irradiance reaching the solar cells.

The incident irradiance on PV cells inside a PV module and the operating temperature of PV cells primarily dictate the power output of module. On a dual axis tracker, when module surface and the incident light rays are perpendicular to each other, the power output will be the highest. Therefore, this study aims to provide a better understanding of the extent at which tilt angle affects the dust, and hence the performance, of PV modules.

Methodology

Description of the study and set up of the PV Module:

Bahir Dar city is located at latitude of 11°36'N and longitude of 37°23'E respectively, and an elevation of 1,840 meters above sea level. Fig 1 shows four modules that, two modules were dirt and tilt different two angles and the other two modules were clean by washing water and detergent and tilt similar to the two dust modules and also due to the location of Bahir Dar those four modules face towards south by using compass, since more near to the equator and the system were installed at a minimum height of 1.5 meter from the ground.



Fig. 1. Clean and dust PV module at 21.50 and 250 tilt angle.

Calibration and Linearity check of the PV Module

The international Electronic Commissions (IEC) 60904-10 standard describes the procedures utilized for determination of the degree of linearity of any photovoltaic device parameter in relation to a test parameter. A device

is linear when it meets the requirements of IECat section 7.3, which is stated as follows. When some device is claimed to be linear, the applicable range of irradiance, voltage, temperatures, or other necessary conditions should also be stated. The requirements for the acceptable limits of non-linearity (variation) are:

- For the curve of short-circuit current versus irradiance, the maximum deviation from linearity should not exceed 2 %.
- For the curve of open-circuit voltage versus the irradiance logarithm, the maximum deviation from linearity should not exceed 5 %.
- If the temperature coefficient of short circuit current doesn't exceed 0.1 %/°K, the device can be regarded as linear in relation to this parameter [7, 8].

During the outdoor experiment, the temperature and the output voltage of all the modules were recorded simultaneously every 15 minute using a multi-meter and a digital thermometer. As well as the irradiance and the short circuit current flow that analysed the linearity of the experiment.

Properties of Dust Based on the Laboratory Testing

Due to small diameter and less density of dust, can be blows in the environment with natural force. This dust diameter that flows on Bahir Dar city that estimates around 0.5-150 microns. However it will vary from place to place. Thus, the average particle diameter is used for this experiment by sieving the dust with diameter of 0.075mm and it distributed 25 gram amount of dust to each tilt angle of PV module to make dirty.

Hydrometer Analysis: Composition of the Finer Dust Particles

According to ASTM D 422 clays are defined as particles diameter of between 0.075mm to 0.005mm and silt particles less than 0.005mm in diameter. Thus, from the 50 gram sample test of dust silt and clay amount were 63.62% and 36.38% respectively. Fig 2 shows that percentage of finer versus grain size diameter (mm).

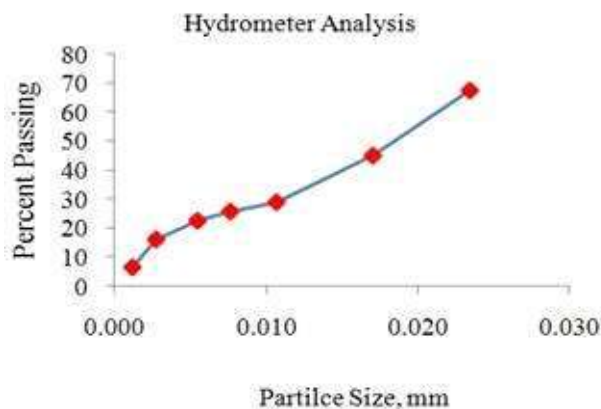


Fig. 2. Hydrometer analysis of percentage of passing versus diameter of particle size.

Experimental Set up and Data Collecting of Wind Speed

The vantage pro 2 instrument as shown below on Fig 3 that has been used to measure the weather condition of the environment (solar radiation, rainfall, etc). But, for this experiment it is only used to measure the wind speed of the environment close to the PV module installed. The instrument has been installed 2.5m away from the PV and at an average height of 1.7m. The wind data has been stored in the data logger of the wireless vantage pro 2 (Fig 4) devices and export the record data by connecting USB cable to the personal computer. The wind speed and direction data were record for eight days within 30 minutes time interval. During this time the tilt angles are changed in two days interval.



Fig. 3. Set up of vantage pro2 with clean and dusty PV module.



Fig. 4. Wireless data logger.

Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Explore and test different charging techniques

Objective:

- to Explore and test different charging techniques.

Requirements:

- Simple 1 or 2 stage controls
- PWM (pulse width modulated)
- Maximum power point tracking (MPPT)
- Indicator
- Battery
- Comparator
- Load
- System

PROCEDURE:**Charging Schemes**

The charger has three key functions

Getting the charge into the battery (Charging)

Optimising the charging rate (Stabilising)

Knowing when to stop (Terminating)

The charging scheme is a combination of the charging and termination methods.

Charge Termination

Once a battery is fully charged, the charging current has to be dissipated somehow. The result is the generation of heat and gasses both of which are bad for batteries. The essence of good charging is to be able to detect when the reconstitution of the active chemicals is complete and to stop the charging process before any damage is done while at all times maintaining the cell temperature within its safe limits. Detecting this cut off point and terminating the charge is critical in preserving battery life. In the simplest of chargers this is when a predetermined upper voltage limit, often called the termination voltage has been reached. This is particularly important with fast chargers where the danger of overcharging is greater.

Safe Charging

If for any reason there is a risk of overcharging the battery, either from errors in determining the cut-off point or from abuse this will normally be accompanied by a rise in temperature. Internal fault conditions within the battery or high ambient temperatures can also take a battery beyond its safe operating temperature limits. Elevated temperatures hasten the death of batteries and

monitoring the cell temperature is a good way of detecting signs of trouble from a variety of causes. The temperature signal, or a resettable fuse, can be used to turn off or disconnect the charger when danger signs appear to avoid damaging the battery. This simple additional safety precaution is particularly important for high power batteries where the consequences of failure can be both serious and expensive.

Charging Times

During fast charging it is possible to pump electrical energy into the battery faster than the chemical process can react to it, with damaging results.

The chemical action can not take place instantaneously and there will be a reaction gradient in the bulk of the electrolyte between the electrodes with the electrolyte nearest to the electrodes being converted or "charged" before the electrolyte further away. This is particularly noticeable in high capacity cells which contain a large volume of electrolyte.

There are in fact at least three key processes involved in the cell chemical conversions.

One is the "charge transfer", which is the actual chemical reaction taking place at the interface of the electrode with the electrolyte and this proceeds relatively quickly.

The second is the "mass transport" or "diffusion" process in which the materials transformed in the charge transfer process are moved on from the electrode surface, making way for further materials to reach the electrode to take part in the transformation process. This is a relatively slow process which continues until all the materials have been transformed.

The charging process may also be subject to other significant effects whose reaction time should also be taken into account such as the "intercalation process" by which Lithium cells are charged in which Lithium ions are inserted into the crystal lattice of the host electrode. See also Lithium Plating due to excessive charging rates or charging at low temperatures.

All of these processes are also temperature dependent.

In addition there may be other parasitic or side effects such as passivation of the electrodes, crystal formation and gas build up, which all affect charging times and efficiencies, but these may be relatively minor or infrequent, or may occur only during conditions of abuse. They are therefore not considered here.

The battery charging process thus has at least three characteristic time constants associated with achieving complete conversion of the active chemicals which depend on both the chemicals employed and on the cell construction. The time constant associated with the charge transfer could be one minute or less, whereas the mass transport time constant can be as high as several hours or more in a large high capacity cell. This is one of the reasons why cells can deliver or accept very high pulse currents, but much lower continuous currents. (Another major factor is the heat dissipation involved). These phenomena are non linear and apply to the discharging process as well as to charging. There is thus a limit to the charge acceptance rate of the cell. Continuing to pump energy into the cell faster than the chemicals can react to the charge can cause local overcharge conditions including polarisation, overheating as well as unwanted chemical reactions, near to the electrodes thus damaging the cell. Fast charging forces up the rate of chemical reaction in the cell (as does fast discharging) and it may be necessary to allow "rest periods" during the charging process for the chemical actions to propagate throughout the bulk of the chemical mass in the cell and to stabilise at progressive levels of charge.

See more about rest periods and how they can be used to increase battery life and to improve the accuracy of SOC measurements in the Software Configurable Battery page.

See also the affects of Chemical Changes and Charging Rate in the section on Battery Life.

A memorable though not quite equivalent phenomenon is the pouring of beer into a glass. Pouring very quickly results in a lot of froth and a small amount of beer at the bottom of the glass. Pouring slowly down the side of the glass or alternatively letting the beer settle till the froth disperses and then topping up allows the glass to be filled completely.

Hysteresis

The time constants and the phenomena mentioned above thus give rise to hysteresis in the battery. During charging the chemical reaction lags behind the application of the charging voltage and similarly, when a load is applied to the battery to discharge it, there is a delay before the full current can be delivered through the load. As with magnetic hysteresis, energy is lost during the charge discharge cycle due to the chemical hysteresis effect.

The diagram below shows the hysteresis effect in a Lithium battery.

Allowing short settling or rest periods during the charge discharge processes to accommodate the chemical reaction times will tend to reduce but not eliminate the voltage difference due to hysteresis.

The true battery voltage at any state of charge (SOC) when the battery is in its "at rest" or quiescent condition will be somewhere between the charge and discharge

curves. During charging the measured cell voltage during a rest period will migrate slowly downwards towards the quiescent condition as the chemical transformation in the cell stabilises. Similarly during discharging, the measured cell voltage during a rest period will migrate upwards towards the quiescent condition.

Fast charging also causes increased Joule heating of the cell because of the higher currents involved and the higher temperature in turn causes an increase in the rate of the chemical conversion processes.

The section on Discharge Rates shows how the effective cell capacity is affected by the discharge rates.

The section on Cell Construction describes how the cell designs can be optimised for fast charging.

Charge Efficiency

This refers to the properties of the battery itself and does not depend on the charger. It is the ratio (expressed as a percentage) between the energy removed from a battery during discharge compared with the energy used during charging to restore the original capacity. Also called the Coulombic Efficiency or Charge Acceptance.

Charge acceptance and charge time are considerably influenced by temperature as noted above. Lower temperature increases charge time and reduces charge acceptance.

Note that at low temperatures the battery will not necessarily receive a full charge even though the terminal voltage may indicate full charge. See Factors Influencing State of Charge.

Basic Charging Methods

Constant Voltage A constant voltage charger is basically a DC power supply which in its simplest form may consist of a step down transformer from the mains with a rectifier to provide the DC voltage to charge the battery. Such simple designs are often found in cheap car battery chargers. The lead-acid cells used for cars and backup power systems typically use constant voltage chargers. In addition, lithium-ion cells often use constant voltage systems, although these usually are more complex with added circuitry to protect both the batteries and the user safety.

Constant Current Constant current chargers vary the voltage they apply to the battery to maintain a constant current flow, switching off when the voltage reaches the level of a full charge. This design is usually used for nickel-cadmium and nickel-metal hydride cells or batteries.

Taper Current This is charging from a crude unregulated constant voltage source. It is not a controlled charge as in V Taper above. The current diminishes as the cell voltage (back emf) builds up. There is a serious danger of damaging the cells through overcharging. To avoid this the charging rate and duration should be limited. Suitable for SLA batteries only.

Pulsed charge Pulsed chargers feed the charge current to the battery in pulses. The charging rate (based on the average current) can be precisely controlled by varying the width of the pulses, typically about one second. During the charging process, short rest periods of 20 to 30 milliseconds, between pulses allow the chemical actions in the battery to stabilise by equalising the reaction throughout the bulk of the electrode before recommencing the charge. This enables the chemical reaction to keep pace with the rate of inputting the electrical energy. It is also claimed that this method can reduce unwanted chemical reactions at the electrode surface such as gas formation, crystal growth and passivation. (See also Pulsed Charger below). If required, it is also possible to sample the open circuit voltage of the battery during the rest period.

The optimum current profile depends on the cell chemistry and construction.

Burp charging also called Reflex or Negative Pulse Charging Used in conjunction with pulse charging, it applies a very short discharge pulse, typically 2 to 3 times the charging current for 5 milliseconds, during the charging rest period to depolarise the cell. These pulses dislodge any gas bubbles which have built up on the electrodes during fast charging, speeding up the stabilisation process and hence the overall charging process. The release and diffusion of the gas bubbles is known as "burping". Controversial claims have been made for the improvements in both the charge rate and the battery lifetime as well as for the removal of dendrites made possible by this technique. The least that can be said is that "it does not damage the battery".

IUI Charging This is a recently developed charging profile used for fast charging standard flooded lead acid batteries from particular manufacturers. It is not suitable for all lead acid batteries. Initially the battery is charged at a constant (I) rate until the cell voltage reaches a present value - normally a voltage near to that at which gassing occurs. This first part of the charging cycle is known as the bulk charge phase. When the present voltage has been reached, the charger switches into the constant voltage (U) phase and the current drawn by the battery will gradually drop until it reaches another present level. This second part of the cycle completes the normal charging of the battery at a slowly diminishing rate. Finally the charger switches again into the constant current mode (I) and the voltage continues to rise up to a new higher preset limit when the charger is switched off. This last phase is used to equalise the charge on the individual cells in the battery to maximise battery life. See Cell Balancing.

Trickle charge Trickle charging is designed to compensate for the self discharge of the battery. Continuous charge. Long term constant current charging for standby use. The charge rate varies according to the frequency of discharge. Not suitable for some battery chemistries, e.g. NiMH and Lithium, which are susceptible to damage from overcharging. In some

applications the charger is designed to switch to trickle charging when the battery is fully charged.

Float charge. The battery and the load are permanently connected in parallel across the DC charging source and held at a constant voltage below the battery's upper voltage limit. Used for emergency power back up systems. Mainly used with lead acid batteries.

Random charging All of the above applications involve controlled charge of the battery, however there are many applications where the energy to charge the battery is only available, or is delivered, in some random, uncontrolled way. This applies to automotive applications where the energy depends on the engine speed which is continuously changing. The problem is more acute in EV and HEV applications which use regenerative braking since this generates large power spikes during braking which the battery must absorb. More benign applications are in solar panel installations which can only be charged when the sun is shining. These all require special techniques to limit the charging current or voltage to levels which the battery can tolerate.

Charging Rates

Batteries can be charged at different rates depending on the requirement. Typical rates are shown below:

Slow Charge = Overnight or 14-16 hours charging at 0.1C rate

Quick Charge = 3 to 6 Hours charging at 0.3C rate

Fast Charge = Less than 1 hour charging at 1.0C rate

Slow charging

Slow charging can be carried out in relatively simple chargers and should not result in the battery overheating. When charging is complete batteries should be removed from the charger.

Nicads are generally the most robust type with respect to overcharging and can be left on trickle charge for very long periods since their recombination process tends to keep the voltage down to a safe level. The constant recombination keeps internal cell pressure high, so the seals gradually leak. It also keeps the cell temperature above ambient, and higher temperatures shorten life. So life is still better if you take it off the charger.

Lead acid batteries are slightly less robust but can tolerate a short duration trickle charge. Flooded batteries tend to use up their water, and SLAs tend to die early from grid corrosion. Lead-acids should either be left sitting, or float-charged (held at a constant voltage well below the gassing point).

NiMH cells on the other hand will be damaged by prolonged trickle charge.

Lithium ion cells however cannot tolerate overcharging or overvoltage and the charge should be terminated immediately when the upper voltage limit is reached.

Fast / Quick Charging

As the charging rate increases, so do the dangers of overcharging or overheating the battery. Preventing the battery from overheating and terminating the charge when the battery reaches full charge become much more critical. Each cell chemistry has its own characteristic charging curve and battery chargers must be designed to detect the end of charge conditions for the specific chemistry involved. In addition, some form of Temperature Cut Off (TCO) or Thermal Fuse must be incorporated to prevent the battery from overheating during the charging process.

Fast charging and quick charging require more complex chargers. Since these chargers must be designed for specific cell chemistries, it is not normally possible to charge one cell type in a charger that was designed for another cell chemistry and damage is likely to occur. Universal chargers, able to charge all cell types, must have sensing devices to identify the cell type and apply the appropriate charging profile.



Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Test Buck & Boost converter

Objective:

- to test Buck and Boost converter.

Requirements:

- Voltage
- Current
- Ripple
- System

PROCEDURE:

Introduction:

Solar panels are widely used as an environmentally-friendly electricity supply. It's very popular since it's simple to install and relatively cost-effective. Unfortunately, a problem is often found, where the voltage output produced is not constant.

A buck-boost converter is a component found in solar panels which is used to regulate the voltage output produced by these solar panels. This converter can be adjusted to produce voltage with a larger or smaller value than initial voltage. Here we aim to create a voltage output at a constant value 12 V, with a range of voltage input of 10 to 50 V.

The voltage value produced by the converter is controlled by the pulse width produced by the PWM (Pulse Width Modulation) generator. The pulse width produced by the PWM generator is controlled by a microcontroller. The microcontroller is programmed to control the pulse width and frequency produced. According to stated above, here we have created a circuit architecture as well as a controller scheme for buck-boost converters which may produce a constant voltage output, which is not affected by sunlight intensity.

The controller scheme used is the PID (proportional Integral Derivative). The converter circuit is connected to the Arduino microcontroller, and simulated in Proteus.

Pulse Width Modulation (PWM)

Pulse width modulation is a method used to adjust the width of a signal. This signal width is represented as a pulse width for one period of time. Commonly, a PWM signal has an equal basic amplitude and frequency, but with a varying pulse width. With the PWM technique, several on and off pulses will be formed. The percentage of on pulses are represented in duty cycles. A duty cycle has a range of 0 to 100%. From the calculation of duty

cycle, the resulting voltage output will be known. This is herein stated.

$$V_{in(min)} = -V_{out} \times \frac{(1-D)}{D}$$

Wherein:

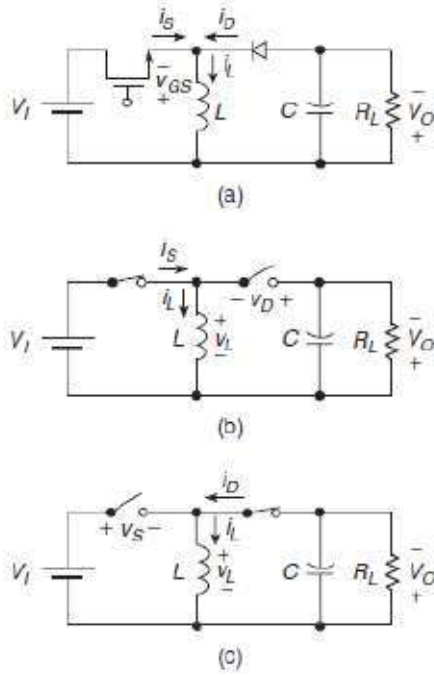
$V_{in(min)}$ = minimum voltage input
 V_{out} = maximum voltage output
 D = duty cycle

PWM buck-boost converter circuit on continuous current mode (CCM)

A buck-boost converter is a type of converter which has the ability to convert a voltage output into a larger or smaller output than the voltage input. The polarity of voltage output will be the opposite of the voltage input's polarity. The PWM buck-boost converter circuit is shown in Figure 1.a. The circuit consists of a MOSFET as a controlled switch, inductor L, filter capacitor C, and load resistor RL. The switch is turned on and off on the switching frequency ($F_s = 1/T$) with a duty cycle ratio ($D = t_{on}/T$), wherein t_{on} is a time interval when the switch is on. Figure 2.b. shows a buck-boost converter equivalent circuit for CCM when the switch is on and the diode is off, and when the switch is off and the diode is on.

During time interval $0 < t < DT$, the switch turns on and the diode turns off, as indicated in Figure 2.b. The diode voltage is $-(V_i + V_o)$ and this keeps the diode off. The inductor voltage is V_i and adds the linear increase on the inductor current with a steepness of V_i/L . During the $DT < t < T$ time interval, the switch is off and the diode is on, as shown in Figure 2.c. The inductor voltage is $-V_o$ and causes the inductor current to decrease linearly with a steepness of $-V_o/L$. The switch voltage is $V_i + V_o$. When $t = T$, the switch turns on, and a new cycle begins.

PWM buck-boost converter and its ideal equivalent circuit for CCM mode. (a) Buck-boost converter circuit. (b) Equivalent circuit during on switch and off diode. (c) Equivalent circuit during off switch and on diode.



Methodology

The designing of the buck-boost converter is done with first determining the types of component, component's value, as well as aligning the component's values with the ones available on market. Then, a programming is done on the Arduino microcontroller by referring to a previously made flowchart. This virtual circuit is then assembled on Proteus 8.4, and simulated by inserting the completed program.

Calculation of Components

Before component-calculating, the desired parameters are set, which are:

Parameters

Table 1. List of parameters

Legend	Parameter	Value
$V_{i(min)}$	minimum voltage input	10 V
$V_{i(max)}$	maximum voltage input	50 V
V_o	voltage output	12 V
$I_{o(min)}$	minimum output current	2 A
$I_{o(max)}$	maximum output current	5 A
$P_{o(min)}$	minimum output power	50 W
$P_{o(max)}$	maximum output power	100 W
F_s	switching frequency	7812.5 Hz
	ripple	< 1%
D_{min}	minimum duty cycle	0.193
D_{max}	maximum duty cycle	0.545

- Output current (I_o), output power (P_o) and output voltage (V_o)

The value of minimum power desired is 50 W. The initial voltage that enters is assumed as 25 V. Hence, the output current is $50 \text{ W} / 25 \text{ V} = 2 \text{ A}$. The input voltage is regulated, becoming 12 V. With the decrease in voltage, the current increases to 5 A. With this, the power is $12 \times 5 = 60 \text{ W}$. For accuracy in calculation, the current value set is into 3 A.

2) Load resistance (R_L)

The equation used is,

$$R_L = V_o / I_o \quad (2)$$

$$R_L = 12 / 3 = 4 \Omega$$

3) Voltage Transfer (M_{VDC})

M_{VDC} is a voltage transfer function in converters. The DC voltage transfer function is classified into minimum voltage transfer ($M_{VDC(min)}$), and maximum voltage transfer ($M_{VDC(max)}$). The equation used is,

$$M_{VDC} = V_o / V_i \quad (3)$$

Hence,

$$M_{VDC(min)} = 12 / 50 = 0.24, \text{ and } M_{VDC(max)} = 12 / 10 = 1.2$$

4) Minimum Inductance (L_{min})

The minimum inductance value is gained by the equation $L_{min} = R_L(1 - D_{min})^2 / 2F_s$ (4)

$L_{min} = 167 \mu\text{H}$. For the circuit to work well, the inductor value must be at least 25% larger than the chosen L_{min} .

The inductor has a value of 220 μH [2]. With a value of 220 μH , the values of I_o , P_o and R_L are recalculated.

$$1. \quad R_L = 5.2783 \Omega$$

$$2. \quad I_o = 2.2734 \text{ A}$$

$$3. \quad P_o = 56.8350 \text{ W}$$

I_o and $P_{o(max)}$ lies in the range of 2 to 5 A and 50 to 100 W, hence inductor 220 μH can be used in the circuit. To determine the suitable MOSFET and diode, a calculation is done with these parameters:

2) Peak-to-peak inductor current ($\Delta I_{L(peak)}$)

The peak-to-peak inductor current is,

$$\Delta I_{L(peak)} = V_o(1 - D_{max}) / F_s L \quad (5)$$

$$\Delta I_{L(peak)} = 3.1767 \text{ A}$$

4) Current stress ($I_{SD(max)}$)

Current stress is the current value that is able to be obtained by the MOSFET switch and the diode in an on condition. The current stress value is calculated with the equation,

$$I_{SD(max)} = I_{D(max)} + I_{S(max)} - \Delta I_{L(peak)} / 2 \quad (7)$$

$$I_{SD(max)} = I_{D(max)} = 12.5884 \text{ A}$$

5) Voltage stress ($V_{SD(max)}$)

Voltage stress is the voltage value which is able to be obtained by the MOSFET and diode in an on condition. The voltage stress value is calculated with the equation,

$$V_{SD(max)} = V_{DS(max)} + V_o - V_{SD(min)} \quad (8)$$

$$V_{SD(max)} = 62 \text{ V}$$

From the calculation results above, the IRE5210S MOSFET is used, with a $V_{DS} = -100 \text{ V}$ and $I_{D1} = -140 \text{ A}$. The diode used is 75LQ150 with $I_{FAV} = 75 \text{ A}$, $V_{KRO4} = 150 \text{ V}$ and $V_{F2} = 0.78 \text{ V}$. The negative value on the V_{DS} and I_{D1} MOSFET is caused by the voltage-polarity-reversing MOSFET.

6) Output capacitor (C_{out})

The C_{out} value is used to minimize the voltage ripple produced by the circuit. The voltage ripple can also be minimized by calculating the ESR (Equivalent Series Resistance) value. ESR is the value of series resistance on the capacitor, with a unit in $\text{ohm} (\Omega)$.

$$1. \quad \text{Voltage ripple } (V_r) = 0.12 \text{ V}$$

$$2. \quad \text{Peak-to-peak voltage ripple } (V_{rpp}) = 0.1 \text{ V}$$

$$3. \quad \text{ESR} = 7.9438 \text{ m}\Omega$$

The voltage ripple on the capacitor ripple (V_{rip}) is 0.02 V

7) Minimum capacitor (C_{min})

This is the minimum capacitor value for the circuit work as desired. The calculation used is,

$$C_{\text{min}} = \left(\frac{D_{\text{max}}}{F_s \times R_L} \right) \times \left(\frac{V_o}{V_{\text{rip}}} \right) \quad (9)$$

$$C_{\text{min}} = 7929 \mu\text{F}$$

By calculating the capacitor value available on market, a capacitor with a value 8200 μF [31] utilized.

With a capacitor value of 8200 μF , I_o , P_o and R_L are recalculated:

1. $R_L = 5.1044 \Omega$
2. $I_o = 2.3509 \text{ A}$
3. $P_o = 58.7725 \text{ W}$

$I_{o(\text{min})}$ and $P_{o(\text{max})}$ are in a range of 2 to 5 A and 50 to 100 W, hence the 8200 μF capacitor may be used in the circuit. The R_L component is adjusted with the resistor value that is market-available. The R_L used is $5.1 \Omega \pm 5\%$.

A buck-boost converter is aimed for battery-charging, hence an additional resistor is added with a total resistance of 5Ω , as the load. After it is simulated, the current produced ranges between 2.35 to 2.4 A.

The amount of diodes given are 2 to 3 diodes. This is to prevent heat from the real, implemented circuit. From the simulation result, the maximum current produced is 129 A. Meanwhile, one 75LQ150 diode can only withstand if passed by a current (I_{FAV}) to 75 A. In this circuit, 2 diodes are arranged in parallel so that the circuit may withstand a current of maximum 150 A.

Design of PWM MOSFET driver:

The PWM driver acts as a switching pengatur of a MOSFET. Without the PWM driver, a MOSFET may not be able to change from the on to off state, or the opposite. This will cause a circuit failure.



Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Check effect of change in solar radiation on Solar PV Module

Objective:

- to check effect of change in solar radiation on solar PV Module.

Requirement:

- Solar Panel
- System

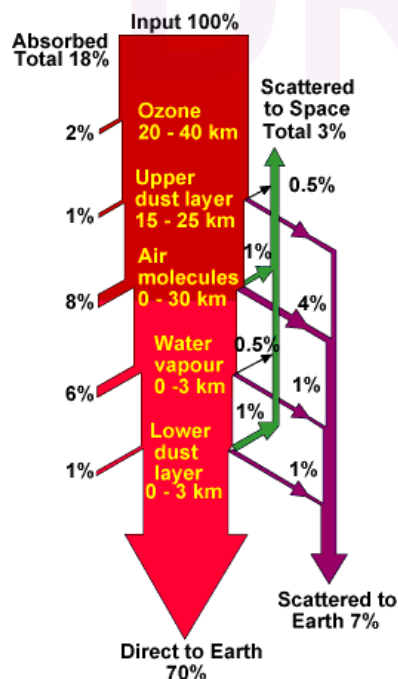
PROCEDURE:

Atmospheric Effects

Atmospheric effects have several impacts on the solar radiation at the Earth's surface. The major effects for photovoltaic applications are:

- A reduction in the power of the solar radiation due to absorption, scattering and reflection in the atmosphere;
- A change in the spectral content of the solar radiation due to greater absorption or scattering of some wavelengths;
- The introduction of a diffuse or indirect component into the solar radiation; and
- Local variations in the atmosphere (such as water vapor, clouds and pollution) which have additional effects on the incident power, spectrum and directionality.

These effects are summarized in the figure below.

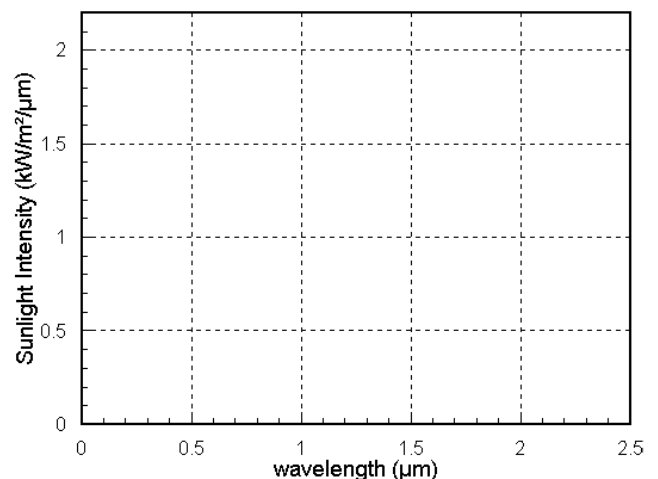


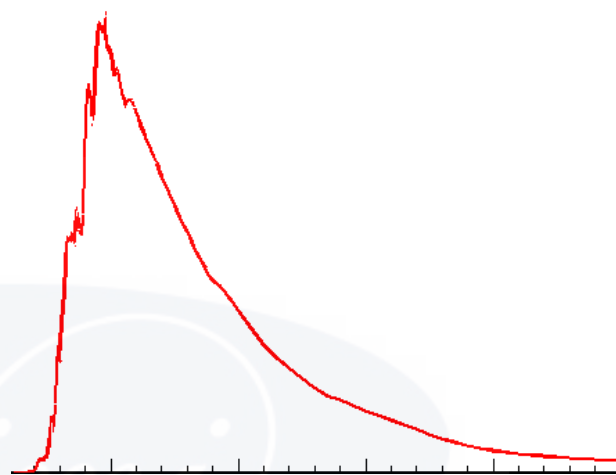
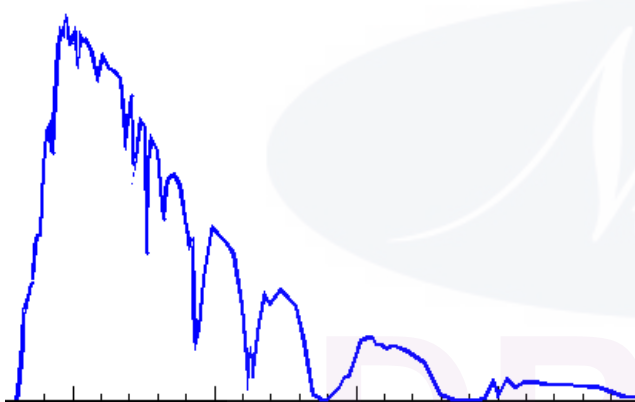
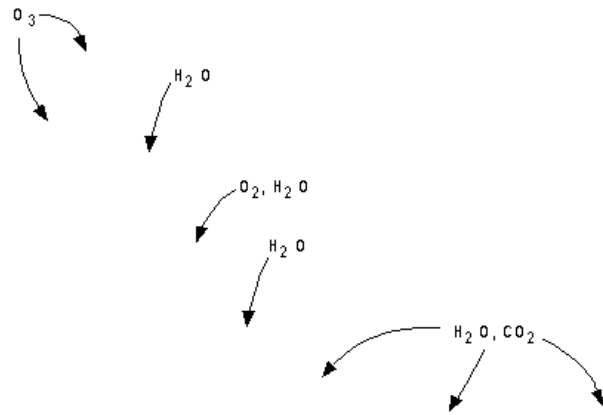
Typical clear sky absorption and scattering of incident sunlight 1.

Absorption in the Atmosphere

As solar radiation passes through the atmosphere, gasses, dust and aerosols absorb the incident photons. Specific gasses, notably ozone (O₃), carbon dioxide (CO₂), and water vapor (H₂O), have very high absorption of photons that have energies close to the bond energies of these atmospheric gases. This absorption yields deep troughs in the spectral radiation curve. For example, much of the far infrared light above 2 μ m is absorbed by water vapor and carbon dioxide. Similarly, most of the ultraviolet light below 0.3 μ m is absorbed by ozone (but not enough to completely prevent sunburn!).

While the absorption by specific gasses in the atmosphere change the spectral content of the terrestrial solar radiation, they have a relatively minor impact on the overall power. Instead, the major factor reducing the power from solar radiation is the absorption and scattering of light due to air molecules and dust. This absorption process does not produce the deep troughs in the spectral irradiance, but rather causes a power reduction dependent on the path length through the atmosphere. When the sun is overhead, the absorption due to these atmospheric elements causes a relatively uniform reduction across the visible spectrum, so the incident light appears white. However, for longer path lengths, higher energy (lower wavelength) light is more effectively absorbed and scattered. Hence in the morning and evening the sun appears much redder and has a lower intensity than in the middle of the day.





Show Blackbody Show AM0 Show AM1.5
Show Visible Wavelengths Show Notes

A comparison of solar radiation outside the Earth's atmosphere with the amount of solar radiation reaching the Earth itself. The human eye has evolved to the point where sensitivity is greatest at the most intense wavelengths 2.

The standard spectra given above and described in more detail in the appendices give a typical spectra for a sunlight. Computers models allow for more detailed models of the solar spectra for a particular location and time of day. The Simple Model of the Atmospheric Radiative Transfer of Sunshine, or SMARTS,³ is used to generate the standard solar spectra. The solar spectrum calculator at PV lighthouse also gives the solar spectrum as a function of location and time of the day. It uses a slightly simpler algorithm from Bird⁴

Direct and Diffuse Radiation Due to Scattering of Incident Light

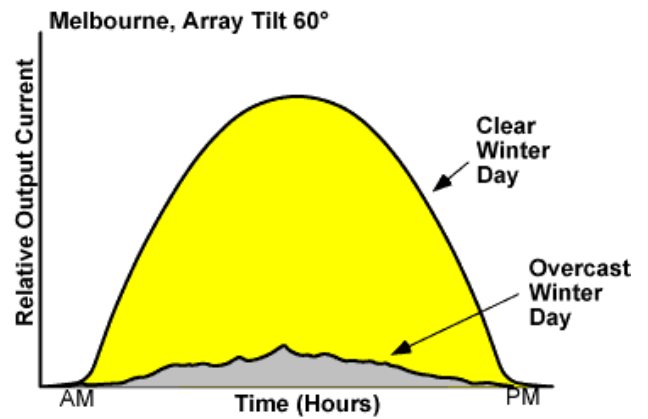
Light is absorbed as it passes through the atmosphere and at the same time it is subject to scattering. One of the mechanisms for light scattering in the atmosphere is known as Rayleigh scattering which is caused by molecules in the atmosphere. Rayleigh scattering is

particularly effective for short wavelength light (that is blue light) since it has a λ^{-4} dependence. In addition to Rayleigh scattering, aerosols and dust particles contribute to the scattering of incident light known as Mie scattering.

Scattered light is undirected, and so it appears to be coming from any region of the sky. This light is called "diffuse" light. Since diffuse light is primarily "blue" light, the light that comes from regions of the sky other than where the sun is, appears blue. In the absence of scattering in the atmosphere, the sky would appear black, and the sun would appear as a disk light source. On a clear day, about 10% of the total incident solar radiation is diffuse.

Effect of clouds and other local variations in the atmosphere

The final effect of the atmosphere on incident solar radiation is due to local variations in the atmosphere. Depending on the type of cloud cover, the incident power is severely reduced. An example of heavy cloud cover is shown below.



Relative output current from a photovoltaic array on a sunny and a cloudy winter's day in Melbourne with an array tilt angle of 60° 5.

Nimi
DRAFT

Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Explore and test running different applications i.e. LEDs, Dusk to Dawn sensing

Objective:

- to explore and test running different application like LEDs, Duck to Dawn sensing.

Requirements:

- Solar Panel
- Solar power controller
- Battery
- System

PROCEDURE:

I try to lean towards the "make do with what I can re-purpose or get cheaply" mindset. You can buy outdoor solar lighting setups that are turn key and ready to go but I've found I can build my own for much less and make it better from a utilitarian point of view.

Different municipalities have different building codes. Some places have stricter codes than others. This is all low voltage (12v) stuff. It's up to the reader / builder to adhere to any local building codes that may apply.

What's needed on your part to duplicate what I have done?

Basic knowledge of hand tools, power tools, electricity, wiring, and of course that ever elusive common sense. Gear wise you'll need solar panels, batteries, a charge controller, and cable to hook it all up with. For years I have been a big fan of using 12-14ga outdoor lighting cable like what's used in Malibu light setups. The stuff has a heavy sun resistant insulation and will last for years outdoors.

What to expect from one of these systems?

A fairly reliable solar lighting system that can illuminate part of your yard without reliance on the commercial power grid.

What limitations will a solar powered lighting system have?

To put it simply, no sun = no power. Without getting into an oversized panel and battery setup, you just have to accept that on cloudy days your system may not run as many hours as expected. In this instructable we're all about keeping it simple enough so it works reliably on sunny days.

The battery is your "gas tank". It stores the power you have generated during the day. For a simple dusk till dawn setup (on at night, off during the day), 10Ah per 10w of LED lighting is right at the minimum. If you can afford a little more battery capacity then go for it! Is 10w

of LED lighting bright? Yes, a modern 10w LED floodlight can easily light a driveway for example. A 10w LED bulb in a lamp can illuminate a room.

The type of battery I use is called an SLA, Sealed Lead Acid (pics 1-3). This battery is commonly found in UPS systems for computers, adult mobility scooters, and kids ride on toys like Power Wheels. 6V and 12V are common. They will be rated in voltage and amperage. For example an SLA battery could be marked 12V 10.5Ah (pic 1). This means it's a 12v battery that can hold up to 10.5Ah of current. If you are pulling 1 amp from this battery and it's fully charged and in good shape, you should be able to run up to 10 hours off it. The higher the Ah rating, the more capacity the battery has. These types of batteries are typically rectangular or little cubes. They can be mounted upright or on their sides but I prefer to run them upright just in case the seals aren't up to snuff.

Take a multi meter and measure the voltage on the remaining batteries. 12v batteries showing less than 10.5v are probably done for and not worth messing with. On 6v batteries pass on anything reading under 5v. Once batteries are deeply discharged and left that way they can be damaged.

Take what's left in your "possibly good" pile and charge them on a slow charger. Small batteries in the 20Ah size or less should be charged at around 2Ah. All the small car battery chargers I have seen have a setting that is comparable to this. Any higher and you risk cooking the battery. If you have any 6v batteries you'll need a 6v capable charger.

Charge a used battery on slow charge over night. Take it off the charger and let it sit for about an hour to stabilize. Measure the voltage, write it on the battery with a permanent marker and let the battery sit disconnected over night and re-measure it's voltage the following day. If it's dropped under 12v it's bad and has an internal short of some kind. If it's voltage is above 12v chances are it's usable. On 6v batteries the test is the same but pass on any that drop below 6v after sitting for a day. There are actual battery capacity testers for this purpose but we're keeping it cheap and simple here.

How much battery capacity do you need?

Take the number of lights running, add their wattage, multiply by the number of hours they will be on, finally divide by 12 which is our system voltage. this will give us the amount of amps we will draw for one day of operation.

There's always that one battery question everybody getting into solar always asks... Can I use a car battery? Yes you can but car batteries have some shortcomings. They are intended for a quick massive draw from the starter motor and then immediately recharged. They don't work well for long discharge applications. Car and truck batteries are also what is referred to as a wet cell battery. They are filled with liquid acid so must remain upright at all times and emit fumes when charged. SLA's are much better suited for enclosed electronics boxes and indoor applications.



Solar panels come in all shapes, sizes, ratings, and prices. You can spend a bunch of money but don't need to if you're an educated shopper. We're keeping it thrifty remember?

Amorphous Vs Silicon

Amorphous solar panels (pic 1) are what's usually found in little solar powered yard lights and trinkets. The panels have a brownish look. This is a low cost technology that is the most inefficient (requires more panel to produce the same power when compared to silicon) and has a shorter lifespan. There are some discount tool vendors that sell solar kits made up from amorphous panels. They do have one benefit which is they perform better in "shaded" applications, cloudy days for example. Pass on these unless you got them for free or darn near free. The one benefit doesn't outweigh all the negatives.

Silicon panels (pic 2) have a blue look to them and that's what you want. Silicon solar cells can be made from mono or poly crystals. This results in a dark blue even color or a blue color that seems to be made up of slivers. Either one is fine for our needs. There is a difference in mono Vs poly crystal panels but for our purposes either one is fine. Mono crystal panels tend to produce more power per square inch but it's a minimal difference on the scale that we're working at.

So we're looking for silicon cell panels, what else do you need to physically look for?

Solar panels can be framed or unframed. They can be rigid or flexible. they can have a glass face or an epoxy sealed face. Aluminium framed, glass faced silicon

panels have the longest lifespan. It's not uncommon for one to last 20+ years as long as it's not physically abused.

Unframed panels need to be framed to protect them and give you a way to mount them. Epoxy faced panels work well for a few years but the epoxy sealant will slowly begin to darken over time and diminish the panels output. Eventually the epoxy gets so dark, the panel is useless. Flexible panels are convenient for portable installations but have short lifespans due to the flexing of electrical contacts and the epoxy sealer they use to keep the whole assembly flexible. Glass faced aluminium framed silicon is the way to go.

Solar panels come in many different output variations. In order to keep it simple, stay with panels that have an open circuit voltage in the 17-20v range. These would be considered 12v system panels and will match up nicely with the ultra cheap regulators we'll be talking about later.

Can you use panels with higher voltages? Yes but it will require a more expensive MPPT regulator.

Can you use panels with lower open circuit voltages? Not without some electronics ingenuity that is simply not worth the effort. The exception is if you have multiples of the same panel and you connect them in series to raise their operating voltage. There's an additional electrical spec to worry about but we'll come back to that.

Figuring out how much solar panel capacity you need isn't hard but you do need to know some data about how much of an electrical load you're going to run and how good the sunlight is in your area. Solar panels are typically rated in watts. This number is derived by multiplying the panel's operating voltage times it's current. This info is found on the label on the backside of the panel.

This instructable is intended to help folks set up a cheap lighting solution while keeping things simple. So far we've been sticking with 12v 10w LED bulbs as a load. This gives us an easy number to work with and those bulbs are cheap and easy to find online.

Calculating the long way...

On a dusk till dawn system we're looking for a operating time of about 10-12hrs every day for most places. This means our solar panels have to produce enough electricity to run the lights overnight and overcome any losses from charging. No system is 100% efficient.

- 1 10w bulb running for 12hrs = 120w of power consumption every night.
- 2 10w bulbs running for 12hrs = 240w of power consumption every night

So, if we want to run (2) 10w LED floodlights for 12hrs every night we need to produce 240w of power from our system. This number is in a perfect world though. No battery charging system is 100% efficient. Add to this you may have a cloudy day or a bird that decided to camp out on your panels. A good rule of thumb in sunny places is to consider that your system 80% efficient in a perfect world.

So our math tells us we need 240w of power for (2) 10w LED lights. Our system is 80% efficient at best so we need to come up with 300w per day to keep the battery charged enough to run our lights that evening. This is cutting it close though. One day of lousy weather and you have taken a big hit on how much power you put into your battery for the day. I prefer to have a bigger safety blanket and at least double the number of what I need to produce. So for (2) 10w LED lights running 12hrs I'd like to have 480w minimum of daily production. This looks like a big number but it really isn't as you will see.

Anywhere you go in the world, you can look up what an area's average solar insolation number is. This is the number of hours of good strong sunlight an area averages. Where I'm at, that magic number is 7.5. So knowing that number I can easily calculate what my panels will produce on a daily basis in clear weather.

30w panel multiplied by 7.5 = 225w of production from good strong sunlight. This is close enough to my target of 240w (120w x 2 safety margin) for a single 10w LED lamp to run 12hrs a day reliably.

Why am I calculating solely on the solar isolation number? This leaves me a little safety margin at the start and the end of the day where the solar panel is still producing something but it's not at peak output.

Solar capacity needed for dusk till dawn system = wattage of lamps multiplied times hours of operation and then doubled.

To calculate what a panel produces for the day find out solar insolation number and multiply that by panel's rated wattage.

There's lot's of variables here. Do your panels really produce what they claim? If your system more inefficient than typical? etc. This will get you in the ballpark.

Now the easy way based on my experiences in sunny South Florida...

30w of panel and 10Ah battery will run a 10w LED floodlight all night but you have no safety margin.

20w of panel and a 10Ah battery will run a 9w LED lamp all night but you have no safety margin.

A word about system reliability...

In order to keep costs and size down I've kept things to a scale that will be very reliable on sunny days and somewhat tolerant of a partially cloudy day. As I said before,

no sun = no power.

If you are absolutely in need of a setup that can handle a completely overcast day without leaving you in the dark, I suggest you multiply your battery and solar capacity by the number of cloudy days you expect to get through before the system has a chance to recharge.

Don't forget to upgrade your regulator as well. A 10A charge controller (regulator) can only handle 120w of panel in a 12v system.

Next to the solar panels themselves, the next biggest price drop in solar has been the cost of regulators. The regulator or charge controller is what takes the power coming from the panel and keeps it within a safe range to charge your battery with. Some charge controllers can also provide you with a low voltage disconnect (LVD) and the ability to automatically turn the lights on at dusk and back off at dawn.





Avoid older shunt type regulators as they can wreak havoc on electronics. The way they work is by connecting the panels directly to the battery and watching the voltage. Once it hits a predetermined value they shut off and wait for the battery voltage to drop again. This on/off up and down voltage swing can upset some electronics.

There's one more charge controller tech called MPPT which is the most efficient and has some benefits over the PWM type but they do cost more money, hence out of the scope of this instructable.

In pics 1, & 3-5) you see a few different options from China. These range from 10a to 20a. They all have LVD and dusk till dawn light control ability.

What is the LVD?

The Low Voltage Disconnect is a circuit within the charge controller that detects when your batteries have run very low and disconnects the load (whatever is being powered) from the battery. This is important because running a lead acid battery dead will drastically shorten its life. This is a failsafe to prevent battery damage. Once the battery is charged back up the LVD resets and everything works again.

What is the dusk till dawn feature?

This feature uses the solar panels as light sensors. If the panels aren't seeing light, they aren't producing power. If no power is being produced, this means it's dark and we should turn on the lights. That's how it works. In the case of a lighting system being used inside a cabin or shed for example, you'll want control of the light via a

switch. The dusk till dawn feature can be bypassed via button programming on every single charge controller I have seen that has an LCD display of some sort. Some very basic charge controllers retain the LVD but do away with the dusk till dawn feature.

Resist the urge to bypass the LVD by connecting your lights straight to the battery. If your battery is run flat, it will ruin it.

My experience with the recent crop of Chinese charge controllers (regulators) is you connect the battery FIRST, then the panels, then the load. Some of these controllers can operate on 24v systems and connecting the battery first is what tells it that you are using a 12v or 24v system.

Charge controllers are rated in amps (pic 2) they can handle. A little overkill is not a big deal but for a basic light setup like we are building here, a 10A unit is plenty. A 20A unit is more than you'll need for a basic system like we're discussing here.

USB charging - added bonus!

Seems the latest trend in Chinese charge controllers is to provide you with a USB charge port for phones and such. For folks building a setup for use in a cabin or workshop this charge port can be a welcome addition.



When it comes to solar lighting projects, there's one way to go and that's LED!

LED tech has really been one of the best things to happen to off grid power. It's efficient, rugged, cheap, and there's a ton of options ranging from sterile bright white to bulbs that can change colour via remote control. If it wasn't for LED bulbs, none of my setups would be worth doing.

In pic 1 you see my latest floodlight setup for the backyard. Those two lights are enough to illuminate a sizable portion of the yard yet only consume 20w of

power. In pics 2 and 3 you can see what the fixtures look like when they aren't powered up.

In pic 4 you see another wonderful option for retrofitting conventional 110v fixtures. These are 12v LED Edison based bulbs. They screw into a regular fixture so you can re-purpose nice looking yard fixtures intended for 110v bulbs.

In pics 3-5 you see the e27 base bulb inside a regular fixture and a couple of shots of the fixture. Since they can fit any standard fixture, you can do like I did and grab some nice fixtures off the clearance table at the local home improvement store and use them with your solar setup without any mods. The same wiring that works for 110v can be used for 12v since the current draw is so minimal in our application.

One thing to consider with these bargain bulbs though is they are typically not weather proof. Whatever you put them in must shield the bulb from the environment. The exception is outdoor rated floodlights you see in pics 2-3.

My preference for white light is warm white (aprox 3000k on the color temp scale) which closely matches the light from a conventional incandescent. The comforting warm white color comes at the expense of slightly reduced light output. Cold white bulbs (6000-7000k) produce a sterile bright white light and can achieve more lumens (light output) per watt but some folks find it annoying for anything other than task lighting.

Remember LED bulbs are much more efficient at producing light than their old incandescent predecessors. A 10w LED floodlight can easily light up a 2 car driveway. The 9w bulbs I used in the outdoor fixtures do a great job of lighting the yard without being obnoxiously bright.

LED's are polarity sensitive. If your light doesn't work, flip around the wires to the lamp. I haven't seen any that get damaged by being connected backwards but most DC LED bulbs won't turn on if the polarity is incorrect. The rare exception is some 12V ac/dc bulbs that don't care about polarity.



Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Explore the use of PV Analyser

Objective:

- to explore the use of PV Analyser.

Requirements:

- PV Analyser
- Arduino UNO
- Photo resistor
- Current chip

PROCEDURE:

Introduction:

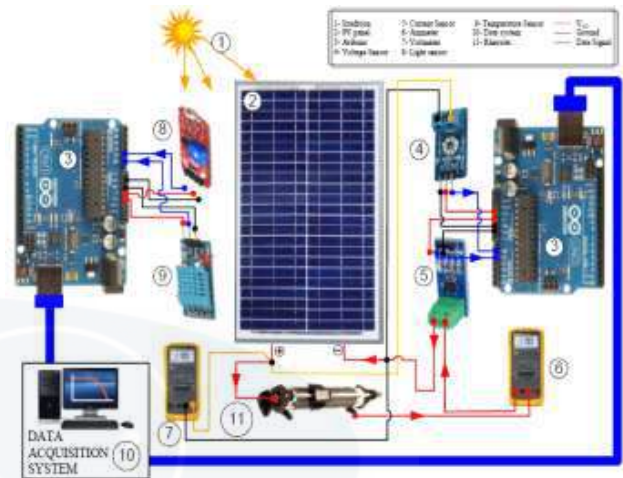
Among the renewable energy sources of power generation, the solar energy is most demanding due to its various advantageous features such as environmental friendly and easily availability. The solar photovoltaic system is more economic in comparison to other renewable energy source for most of the house hold applications. The solar PV systems are used for power generation. Here we have designed a low cost data logger using Arduino for monitoring the parameters of solar PV system. A low cost microcontroller based solar radiation measurement and data acquisition system has been designed and another design which is a computer based monitoring acquisition system for measuring of humidity, temperature, voltage, and current parameters, in renewable energy applications is also available. A hardware system is developed to plot the V-I characteristics of PV system in real time in. A system is designed to monitor the PV array characteristics and solar irradiance and temperature in. With the motivation of above literature review, the research novelty of this paper is that a low cost PV analyzer is designed to monitor the PV module performance.

LAYOUT OF PV ANALYZER

Apparatus Description In this designed low cost PV analyzer system, various components are used such as, Two Arduino (ATmega328P, ATmega16U2), Photo resistor Light sensor (VEE00067), Temperature and Humidity Sensor Module (DHT11), Current sensor (5A range ACS712), and Voltage sensor module (KG045). A monocrystalline PV module, which comprises 4x4 cell in series-parallel connection for the system investigation. The layout of designed system is shown in Fig. 1 as,

Specifications of System Components and Cost Analysis In this designed "Low cost PV analyzer using Arduino UNO" system, various components are used to investigate the PV performance in terms of I-V and P-V

characteristics and PV module specifications for the validation. All the used components are discussed with their specifications as, J) Arduino (Atmega 328)



- Microcontroller Atmega 328 Operating Voltage 5V Input Voltage (recommended) 7-12V
- Input voltage (limits) 6-20V
- Digital output Pins 14 (of which 6 provide PWM output)
- Analog input pins 6
- DC current per output pin 40 mA
- DC current for 3.3V pin 50 mA
- Flash memory- 32 KB (Atmega 328) of which 0.5 KB used by boot loader
- SRAM- 2 KB (Atmega 328)
- EEPROM- 1 KB (Atmega 328)
- Clock speed- 16 MHz

2) Photoresistor Light Sensor

- Photoresistor sense intensity of light
- Input voltage-3.3V to 6V in DC
- Supply current-0.5-3mA
- Sense irradiation range 0-1023W/m²
- Peak wavelength-540nm

3) Temperature and Humidity Sensor

- Power supply- 3-5.5V DC Output signal- digital signal via single- bus sensing element
- Polymer resistor measuring range- humidity
- 20-90% RH
- Temperature 0-50°C humidity hysteresis
- $\pm 1\%$ RH
- Long-term stability- $\pm 0.5\%$ RH/year
- Sensing period average: 2s

4) Current sensor

- Current sensor chip: ACS712ELC-05B
- Pin 5V power supply, on board indicator

- Modules can be measured plus (+) or minus (-) 5A, corresponding to the analog output 185mV/A.
- 5) Voltage sensor
- Input DC voltage range- 0-25V
- DC voltage detection range- 0.02445V-25V
- Voltage analog resolution- 0.00489V
- DC Input: positive terminal connected to VCC, GND negative pole

Output interface: "+" then 5/3.3V, "-" then GND, "s" then the AO pin

- 5V maximum voltage on the ADC side of the module
- 25V maximum voltage on the input side



Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

On Grid Smart Energy Management

Objective:

- to grid Smart Energy Management

Requirement:

- Grid Connected Invertor
- Sunlight
- Solar Panel Convertor
- Electric Power Network

PROCEDURE:**HOME ENERGY MANAGEMENT SYSTEM (HEMS)**

Utility sector is eyeing its current customer base to be equipped with smart home system. Home Energy Management System (HEMS) is a part of this initiative. HEMS is a monitoring system for client's residential electricity usage and is accessible through customer's PC / Smartphone / Tablet for energy optimization. Its online energy monitoring system enables customers to monitor electricity bill, automate appliances and keep their home safe with security features. A smart meter, smart plug and a dashboard is required for HEMS and Smart Grid provides HEMS along with SG® Smartplug.

Utility companies intend to increase their revenue by making the electricity usage visible to their consumers

and making efficiency a catalyst. HEMS will work on its own, cut the unnecessary electricity wastages and enable the energy efficiency perspective. Users can pick and choose what they want to use depending on the control, how they use and when they want to use it.

Smart Grid offers Mesh networking for HEMS and it operates on Bluetooth Low Energy (LE). SG® Smartplug is an Advanced Smart Plug solution for your home users enabling them to control their voltage usage according to their budget. It is easy to see the current usage of each appliance connected to a Smartplug. An easy to understand dashboard for mobile/web will display the below information:

- Current usage of each appliance connected to the Smartplug for a certain period of time
- Cost of the current usage for a particular appliance of a certain period of time
- Daily, weekly, and monthly total usage
- ON/OFF control of appliances connected to the Smartplug

DRAFT

Smart City - Demonstrate and deploy responsive Web Application using APIs and generate reports using templates

Install Linux Operating system porting

Objective:

- to install Linux Operating system porting.

Requirement:

- System
- Linux

PROCEDURE:

Linux. What is it? At one point in time it was a niche operating system run by those who wanted to show off their PC prowess and feel more alternative and less than the rest. But something happened on the way to the convention -- Linux became accepted. Not only did this platform become accepted, it was adopted as a must-have technology by enterprise-level businesses, where reliability, flexibility, and security are key.

But for the masses, those that have really only ever known either Windows or Mac, the question still remains. What is Linux?

I want to answer that question for you. The answer will not be exhaustive (for that you'll need a book) and will target those that have had zero exposure to the operating system. I'm not going to pull it back so far as to answer the question, "What is an operating system?" I'm fairly confident the masses have that one down; so we'll focus on only this one particular operating system.

Linux

And with that said, let me dive into the answer.

It's really a kernel

If you ask the question, "What is Linux?" to a purist, they'll tell answer, "It's a modular kernel, created by Linus Torvalds." That's not an answer that helps us much, as the average user would then need to understand what a kernel is and why it's relevant to the issue at hand. Instead, I want to go by the broader definition of Linux, and say it's an operating system powered by the Linux kernel.

But what makes it so special?

To understand why Linux is so special, you have to first understand what open source is. Open source is software for which the original source code is made freely available to the public. With this source code you can modify and

redistribute the software in any way you see fit (so long as you retain attribution to the original creator). In the modern vernacular, open source applies to much more than software. Nearly anything can be "open sourced" these days. The important factor in this is that the original designs, plans, schematics, etc. must be made available for the public to use.

I can already hear your next question: But if anyone can view the source code, how do companies sell their software? You'll be surprised to hear that most open source software is given away for free. That's right, adopting open source software can mean you're entire software library can be had free of charge. And most of the open source software you can find is very high quality. Outside of the operating system, you'll find free software such as:

- LibreOffice - a full-blown office suite
- The GIMP - an image editing tool similar to Photoshop
- Evolution - a groupware tool similar to Microsoft Outlook
- Audacity - a powerful audio recording software
- Open Shot - a user-friendly video editing tool
- Firefox - a widely-used web browser

That list goes on and on. Couple that with plenty of music and video players, even more browsers and email clients, just about every productivity tool you can think of, and you have the making for a computing platform that does everything your current operating system will do.

But why bother? If it will only do what my current operating system will do, what would make me want to use Linux? Therein lies the crux of the issue. There are three ideals that Linux offers that are unmatched by any other platform:

- Reliability
- Flexibility
- Security

Smart City - Demonstrate and deploy responsive Web Application using APIs and generate reports using templates

Configure load cloud and server

Objective:

- to configure load cloud and server.

Requirement:

- Raspberry Pi
- System
- Micro USB
- Internet Connection

PROCEDURE:**Introduction**

With the introduction of the Digital Ocean metadata service, it is possible to start configuring your servers before you even log in. In short, the metadata service is an HTTP location that your server can access during the boot process.

The metadata location will contain basic data about the server's configuration and environment, such as network addresses, hostname, etc. During the initial setup, these values can be pulled down by a program called cloud-init to help configure essential services.

The most powerful feature is that you can pass a script to the metadata service when you create a server using a field called user-data. This will be run during the initial boot process and is very flexible, allowing you to accomplish whatever you can script.

The most common type of script to pass in is called a cloud-config script. This is a YAML formatted file that provides simple, readable methods of setting up common configuration items by declaration. It also has the ability to run arbitrary commands for other tasks.

In this guide, we will get familiar with the DigitalOcean metadata service and cloud-config files by trying out a simple example. We will be recreating some of the steps outlined in the initial server setup for Ubuntu 14.04 guide in order to demonstrate.

Goals

In order to successfully replicate the steps in the Ubuntu 14.04 initial server setup guide, our script would have to do a number of tasks.

These are the basic tasks that the above guide accomplishes:

- Change the root user's password
- Create a new user
- Create the new user's password

- Give the new user root privileges
- (Optional) Change the port the SSH daemon listens on
- (Optional) Restrict root SSH login
- (Optional) Explicitly permit our new user

Modifying the Goals to Address the Environment

For our servers created with a cloud-config file, we will have to modify our goals a bit. Any information passed in through a cloud-config file is accessible to any user of the system for the entire life of the server.

This introduces a number of security issues that are important to understand and address. Some things to keep in mind are:

- Any information passed in your cloud-config is accessible to every user on the system. Do not place anything confidential in your cloud-config files.
- You can set the password for existing users, but you must pass it in plain text.
- For new users, you can pass in a hashed version of a password, but these hashes can be broken very easily with modern hardware.

With these things in mind, our setup should do everything possible to avoid committing passwords, in any form, within the cloud-config file. We can adjust our goals to accommodate the specific needs of our deployment environment.

Our adjusted strategy will look something like this:

- Set no password and provide no SSH keys for the root account through cloud-config (any SSH keys added through the DigitalOcean interface will still be added as usual)
- Create a new user
- Set no password for the new user account
- Set up SSH access for the new user account
- Give the new user password-less sudo privileges to make administrative changes.
- (Optional) Change the port the SSH daemon listens on
- (Optional) Restrict root SSH login (especially if you do not include SSH keys through the DigitalOcean interface)

- (Optional) Explicitly permit our new user

Besides removing the passwords for both accounts, the most drastic change here is that the new account will be allowed to use sudo without entering an account password. This is necessary since we are not allowing root logins and we are not setting an account password for our new user.

After the new user logs in, they will have freedom to set a password for themselves securely and modify the sudo privileges to require a password, if desired.

With these adjusted goals in mind, let's get started.

Using Cloud-Config Files

A cloud-config file is basically a YAML file that understands certain directives. YAML is a data serialization format that is meant to be very human readable, making it simple to understand and edit.

YAML files rely on a few rules for formatting:

- Indentation with whitespace indicates the structure and relationship of the items to one another. Items that are more indented are sub-items of the first item with a lower level of indentation above them.
- List members can be identified by a leading dash.
- Associative array entries are created by using a colon (:) followed by a space and the value.
- Blocks of text are indented. To indicate that the block should be read as-is, with the formatting maintained, use the pipe character (|) before the block.

The first line of a cloud-config file must contain a special identifier so that the cloud-init program knows that the file is a cloud-config file. This looks like this:

```
#cloud-config
```

This must be placed alone on the very first line. The cloud-config file must be provided when the server is created. This can be accomplished in a two different ways.



Smart City - Demonstrate and deploy responsive Web Application using APIs and generate reports using templates

Configure over the air (OTA) Node

Objective:

- to configure over the air (OTA) Node

Requirement:

- Sensor
- Micro-controller
- System

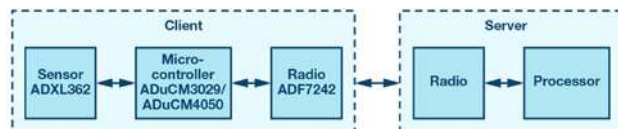
PROCEDURE:

Introduction:

An OTA update replaces the current software on a device with new software, with the new software being downloaded wirelessly. In an embedded system, the device that runs this software is typically a microcontroller. A microcontroller is a small computing device with limited memory, speed, and power consumption. A microcontroller typically contains a microprocessor (core) as well as digital hardware blocks for specific operations (peripherals). Ultra low power microcontrollers that typically consume 30 μ A/MHz to 40 μ A/MHz in active mode are ideal for this type of application. Using specific hardware peripherals on these microcontrollers and placing them into low power modes is an important part of the OTA update software design.

An example of an embedded system that might require OTA updates is shown in Figure 1. Here we see a microcontroller connected with a radio and sensor, which may be used in an IoT application that gathers data about the environment using the sensor and reports it periodically using the radio. This portion of the system is referred to as the edge node or client and is the target of the OTA update. The other portion of the system is referred to as the cloud or server and is the provider of the new software. The server and client communicate over a wireless connection using transceivers (radios).

Figure 1. Server/client architecture in an example embedded system.



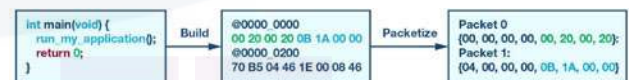
What Makes a Software Application?

Much of the OTA update process is the act of transferring the new software from the server to the client. The software is transferred as a sequence of bytes, after it has been converted into a binary format from the source format. The conversion process compiles the source

code files (for example, c, cpp), links them together into an executable file (for example, exe, elf), and then the executable is converted into a portable binary file format (for example, bin, hex). At a high level, these file formats contain a sequence of bytes that belong at a specific address of memory in the microcontroller.

Typically, we conceptualize the information being sent over a wireless link as data, such as a command to change the system's state or sensor data collected by the system. In the case of the OTA update, the data is the new software in binary format. In many cases, the binary file will be too large to send in a single transfer from the server to the client, meaning that the binary file will need to be placed into separate packets, in a process called packetizing. To visualize this process better, Figure 2 demonstrates how different versions of the software will produce different binary files, and thus different packets to be sent during the OTA update. In this simple example, each packet contains 8 bytes of data, with the first 4 bytes representing the address in the client's memory to store the next 4 bytes.

Figure 2. Binary conversion and packetization process of a software application.



Major Challenges

Based on this high level description of the OTA update process, three major challenges arise that the OTA update solution must address. The first challenge relates to memory. The software solution must organize the new software application into volatile or nonvolatile memory of the client device so that it can be executed when the update process completes. The solution must ensure that a previous version of the software is kept as a fallback application in case the new software has problems. Also, we must retain the state of the client device between resets and power cycles, such as the version of the software we are currently running, and where it is in memory. The second major challenge is communication.

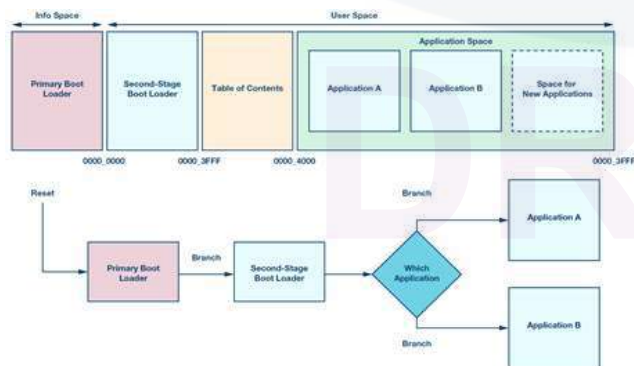
The new software must be sent from the server to the client in discrete packets, each targeting a specific address in the client's memory. The scheme for packetizing, the packet structure, and the protocol used to transfer the data must all be accounted for in the software design. The final major challenge is security. With the new software being sent wirelessly from the

server to the client, we must ensure that the server is a trusted party. This security challenge is known as authentication. We also must ensure that the new software is obfuscated to any observers, since it may contain sensitive information. This security challenge is known as confidentiality. The final element of security is integrity, ensuring that the new software is not corrupted when it is sent over the air. The Second-Stage Boot Loader (SSBL)

Understanding the Boot Sequence

The primary boot loader is a software application that permanently resides on the microcontroller in read-only memory. The region of memory the primary boot loader resides in is known as info space and is sometimes not accessible to users. This application executes every time a reset occurs, generally performing some essential hardware initializations, and may load user software into memory. However, if the microcontroller contains on-chip nonvolatile memory, like flash memory, the boot loader does not need to do any loading and simply transfers control to the program in flash memory. If the primary boot loader does not have any support for OTA updates, it is necessary to have a second-stage boot loader. Like the primary boot loader, the SSBL will run every time a reset occurs, but will implement a portion of the OTA update process. This boot sequence is illustrated in Figure 3. In this section, we will describe why a second-stage boot loader is necessary and describe how specifying the role of this application is a key design trade-off.

Figure 3. An example of a memory map and boot flow with SSBL.



Lesson Learned: Always Have an SSBL

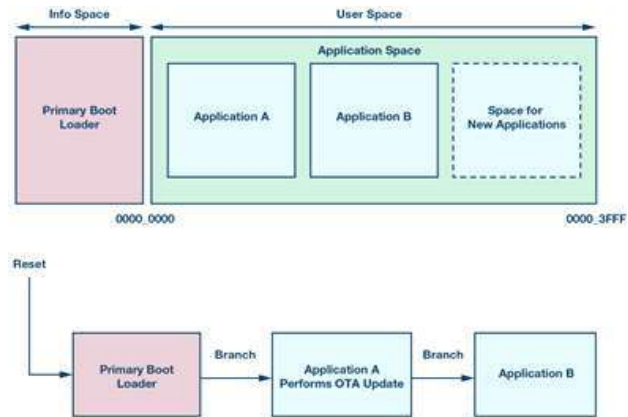
Conceptually, it may seem simpler to omit the SSBL and place all the OTA update functionality into the user application, as it would allow an existing software framework, operating system, and device drivers to be seamlessly leveraged for the OTA process.

The memory map and boot sequence of a system that chose this approach is illustrated in Figure 4.

Figure 4. Example memory map and boot flow without SSBL

Application A is the original application that is deployed on the microcontroller in the field. This application

contains the OTA update-related software, which is leveraged to download Application B when requested by the server. After this download is complete and



Application B has been verified, Application A will transfer control to Application B by performing a branch instruction to the reset handler of Application B. The reset handler is a small piece of code that is the entry point of the software application and runs on reset. In this case, the reset is mimicked by performing a branch, which is equivalent to a function call. There are two major issues with this approach:

- Many embedded software applications employ a real-time operating system (RTOS), which allows the software to be split into concurrent tasks, each with different responsibilities in the system. For instance, the application presented in Figure 1 may have RTOS tasks for reading the sensor, running an algorithm on the sensor data, and interfacing with the radio. The RTOS itself is always active and is responsible for switching between these tasks based on asynchronous events or specific time-based delays. As a result, it is not safe to branch to a new program from an RTOS task, since other tasks will remain running in the background. The only safe way to terminate a program with a real-time operating system is through a reset.
- Based on Figure 4, a solution to the previous issue would be to have the primary boot loader branch to Application B instead of Application A. However, on some microcontrollers, the primary boot loader always runs the program that has its interrupt vector table (IVT), a key portion of the application that describes interrupt handling functions, located at address 0. This means that some form of IVT relocation is necessary to have a reset map to Application B. If a power cycle occurs during this IVT relocation, it could leave the system in a permanently broken state.

These issues are mitigated by having an SSBL fixed at address 0, as illustrated in Figure 3. Since the SSBL is a non-RTOS program, it can safely branch to a new application. There is no concern of a power cycle placing the system in a catastrophic state since the IVT of the SSBL at address 0 is never relocated.

Smart City - Demonstrate and deploy responsive Web Application using APIs and generate reports using templates

Explore GUI based parameter configuration, GUI based IOT application

Objective:

- to explore GUI based parameter configuration, GUI based IOT application.
-

Requirements:

- System
- IoT devices
- Graphical User Interface

PROCEDURE:

The Internet of Things (IoT) technologies allow devices to communicate, share information, and coordinate decisions with no human intervention. IoT is present in various everyday situations, such as transportation, healthcare, industrial, and home automation. IoT projects usually share some essential features: connect different devices, process data, a user management system, display information to the user, e.g., tables, charts or maps, and so on [2]. Thus, when a team of developers starts to work on an IoT project, it needs to implement all features above listed for each new project. To cope with this problem, we propose a fast way to configure and generate Internet of Things systems: the AutoIoT platform. Our goal is to allow individuals with minimum programming experience create and deploy a server-side IoT application that communicates and interacts with devices. AutoIoT is a web-based platform that provides a Graphical User Interface (GUI), which allows programmers to deploy and configure an IoT system quickly. This way, users do not need to deal with domain-specific frameworks or to begin a project from scratch. The resulting system generated by AutoIoT is a Flask [3] project (a python micro-framework). It can be executed as it is or developers that need a more complex system can extend the generated code to fulfill their needs. The resulting system can connect to an MQTT Broker, store and query information in a database, present data to users, and exchange data with another system. Moreover, AutoIoT also provides a simulator that communicates with real devices. Therefore, users with no access to real hardware can research on IoT. The present research work also describes how to create an Assembly Line [4] server-side application using AutoIoT. This application receives data from RFID readers and shows detailed information about the status of the assembling process on a GUI.

The main contributions of this work are the following:

- A working platform that quickly deploys server-side IoT applications.
- A demonstration of how to use AutoIoT in a real-world scenario: an Assembly Line.

- The present research work also shows that AutoIoT is an effective tool for prototyping IoT systems.

BACKGROUND

The MQTT Protocol A generic IoT system can be composed of IoT devices, MQTT Broker, and a server-side application, as shown in Figure 1. The MQTT is the communication protocol usually applied by IoT systems to connect devices and the server-side application. MQTT is a machine-to-machine (M2M)/IoT connectivity protocol that is on top of the TCP/IP protocol stack. It was designed as an extremely lightweight publish/subscribe messaging protocol. In turn, the MQTT Broker is the application that connects devices using the MQTT protocol. Devices send messages to a topic in the MQTT Broker, and all applications listening to this particular topic receive the message, the opposite direction is also valid. A device is any gadget that communicates using the MQTT protocol: it can be a tag with sensors and actuators or even a smartphone. In real-world scenarios, devices may have restricted resources and cannot communicate directly to the MQTT Broker. In such situations, a gateway is used: the device communicates with the gateway, and the gateway communicates with the MQTT Broker

Assembly Line An assembly line is a production system where different tasks of the production process are concentrated at different workstations. It consists of various workstations where production tasks are carried out and a transportation system that moves the workpieces between the workstations. During the production process, a workpiece passes through all the workstations in a given order until the final product comes out at the last station. The workpieces can be moved between the stations by various location systems like conveyor belts or automatic guided vehicles. To synchronize the workstations, the assembly line is organized with prescribed cycle time. The cycle time is the time a workpiece stays at one workstation. All the tasks of the workstation have to be completed in the given cycle time and each workpiece moves to the next station after the cycle time is over. The cycle time is the same for all workstations.

Figure 2 illustrates an example of an assembly line with three workstations located along a conveyor belt. The workpieces are moved along the conveyor belt. They pass through station 1, 2 and 3 in the given order. The time that they spend on one workstation is exactly the cycle time

THE AUTOIOT PLATFORM

IoT platforms are the central component of an IoT system. They provide a set of fundamental applications that are the building blocks of more complex IoT applications. AutoIoT is a platform that automatically generates IoT server-side application through a visual tool. The generated system can communicate with IoT devices and third-party software, store information on a database, and provide visualizations of data. In the following, we detail four components of the AutoIoT platform: Modules Manager, Communication Manager, Database Manager, Builder. Due that the Simulator is not used in this paper, it will be detailed only in future works.

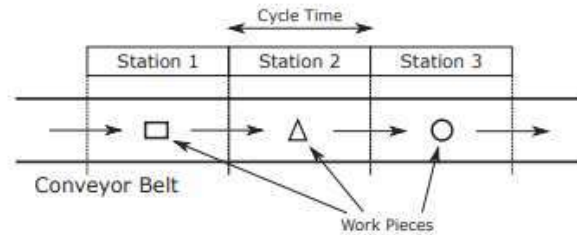


Fig. 2. Illustration of a typical assembly line. It consists of three work stations and a conveyor belt. The conveyor belt moves work pieces (illustrated as a square, a triangle and a circle) from work station to work station in the given order. The time a work piece spends at one work station is exactly the cycle time.

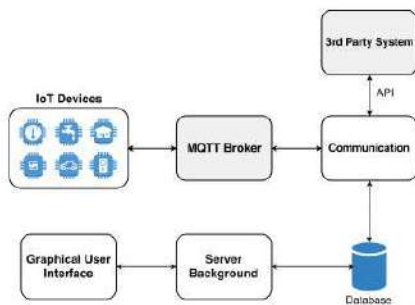


Fig. 1. Illustration of a generic IoT system [15]. It presents a basic architecture connecting devices, database, and application.

Nimi

DRAFT

Smart City - Demonstrate and deploy responsive Web Application using APIs and generate reports using templates

Check IoT Gateway using Wi-Fi and Ethernet

Objective:

- to check IoT Gateway using Wi-Fi and Ethernet.

Requirement:

- Raspberry Pi (any model will do, but you'll see faster results with a model 3).
- Power adapter for your Pi.
- SD Card to install the operating system on (you could use an 8GB card up to whatever size you want. I usually go with a 32GB card in case I want to add more options to the RPi).
- Wifi adapter
- Ethernet cable
- HDMI cable (I have an old DVI monitor so I use this HDMI to DVI cable).
- Keyboard and mouse
- Card reader or computer with built in card reader.
- A case for the Pi (optional)

PROCEDURE:

As IoT grows and billions of connected devices enter the world, one of the most critical components of future internet of things systems may be a device known as an IoT gateway. An IoT gateway aggregates sensor data, translates between sensor protocols, processes sensor data before sending it onward and more.

The importance of IoT gateways is understandable when you consider the explosion in connected "things" that has occurred over the past few years. With scores of protocols, connectivity models and energy profiles and the highly dispersed nature of IoT systems, gateways are needed to manage and control these complex environments.

Consider the use case of an IoT-connected office building environment. Sensors are the IoT equivalent of our five senses. But instead of five senses, there may be hundreds or thousands of sensors with dozens of different functions, measuring temperature, light, noise, position of people and equipment, particles in the air, building systems operations, security systems, factory machines and more. But IoT is not just about sensing; it's also about controlling systems. Turning on and off lights, HVAC, networks and more can be done through connected systems.

Each of these IoT devices may use different protocols to connect -- such as Wi-Fi; Bluetooth; serial ports (for example, RS-232); Ethernet; MQ Telemetry Transport, or MQTT; Zigbee; and others. And each of them may

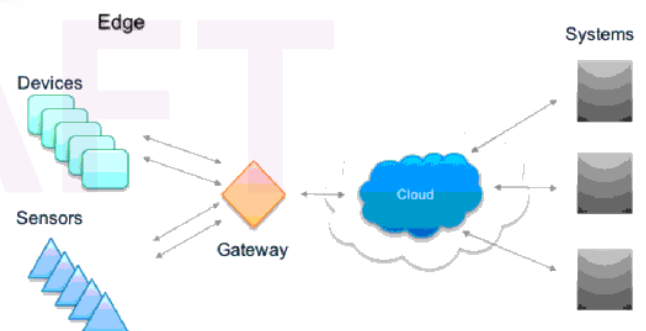
connect to different control environments and have different models for management and security.

As connected devices, protocols and needs proliferate, having components connect individually back to the systems that need their data is not often possible. Some sensors and controllers use very low energy and don't support energy-intensive protocols like Wi-Fi or Bluetooth, and, therefore, can't connect directly. Some edge devices generate so much data that, in aggregate, the data is overwhelming and not all that valuable in its raw form.

Enter the IoT edge gateway

IoT gateways perform several critical functions such as device connectivity, protocol translation, data filtering and processing, security, updating, management and more. Newer IoT gateways also operate as platforms for application code that processes data and becomes an intelligent part of an edge device-enabled system.

IoT gateways sit at the intersection of edge systems -- connected devices, controllers and sensors -- and the cloud.



A gateway's position in the IoT ecosystem

Traditional network gateways have mostly performed protocol translation and device management functions. They were not intelligent, programmable devices that could perform in-depth and complex processing on IoT data. Today's "smart" IoT gateways -- delivered by companies such as Dell Technologies, Wind River/Intel, Nexcom and others -- are full-fledged computing platforms running modern operating systems (for example, Linux or Windows). These systems are

sometimes also called intelligent gateways or edge gateways, but the line is blurring and the nonintelligent gateway market is likely to become largely irrelevant in coming years.

Next-generation IoT gateways open up huge opportunities to push processing closer to the edge, improving responsiveness and supporting new operating models. Consider the building systems use case above: A building management company may control millions of square feet of office and industrial space from a remote location using a distributed IoT network of sensors and controllers connected through the cloud. However, transmitting every routine packet of information generated by the sensors from dozens of facilities would quickly overwhelm the headquarters systems of the management company. They care about serious issues, out-of-bounds environmental conditions and other factors worthy of additional attention.

Adding new IoT edge gateways with full onboard processing would allow the gateways to filter out routine information and pass through the alerts that are worthy of attention. Going even further, these gateways can perform their own analytics and operations on the data - such as determining that the temperature is too high in part of the building -- and can then use that insight to take independent action, such as turning on the air conditioning or opening vents.

Recent product announcements from Wind River and Dell show the direction the market is heading. Wind River (an Intel Company) ships a gateway platform -- Wind River Intelligent Device Platform XT -- as part of its Helix portfolio of IoT systems. This platform allows OEMs to build their own gateway devices that are customized for their needs.

The Dell Edge Gateway 5000 Series, announced in October 2015, is a turnkey device targeting multiple use cases for companies that don't want to build their own gateways. Dell's gateways ship with the Wind River platform, as do many other of the newer IoT gateway platforms.

Dell believes that its gateways will appeal to users in three major use cases: industrial automation, building automation and transportation. One of Dell's early clients is Eigen Innovations, a supplier of intelligent industrial vision systems. Eigen's infrared sensors detect quality problems in manufacturing and assembly operations. The data is transmitted to the Dell IoT gateway which runs a rules engine and can shut down the factory if issues are discovered. Since speed and availability are

critical to its clients, the ability to process the data locally and ensure that quality issues are responded to quickly is critical to Eigen's model.

While these intelligent edge gateways are just rolling out to the market, the advantages of having a fully programmable and manageable platform closer to the IoT devices they support is clear.

Bridging the operations/IT gap

According to Wind River's Dinyar Dastoor, vice president and general manager of operating system platforms and IoT, an IoT edge gateway also serves as an important bridge between operations and IT. On the operations side -- factory, building management, etc. -- the focus is on delivering the business capability for which they are responsible, meaning getting the data from where it's created into systems that can act on it. IT, on the other hand, must be concerned with the security and integrity of the networks, the supportability of the system and how all of the pieces are managed.

"The huge problem is human beings. On the edge side you have people connected to the device itself -- OT, or operational technology," Dastoor said. "Trying to get these devices securely connected to the world is where the CIO comes in. Who is going to step into the others' shoes?"

The IoT gateway's role in security, data, analytics and more

To successfully bridge the gap between IT and OT, IoT gateways need to support the following functionality:

- **Trusted connectivity and security** -- ensuring the integrity of the network and system in both directions
- **Protocol and data bridge** -- being able to translate and transfer data among and between systems operating with different communications protocols and data formats, often requiring bidirectional communication capabilities
- **Storage and analysis** -- on-board application development platforms and storage to drive intelligence and decision-making closer to the edge device
- **Management** -- the ability to provision, update and control access of connected devices to the system as well as policy-based permissions

As IoT networks proliferate and use cases expand, the demands on IoT gateways for these capabilities will only intensify.

Smart City - Demonstrate and deploy responsive Web Application using APIs and generate reports using templates

Work with the command line and the shell

Objective:

- to work with the command line and the shell.

Requirement:

Raspberry Pi

Linux

Win32DiskImager

System

PROCEDURE:**Preparing the OS**

The Raspberry Pi is nothing more than a CPU, GPU, and some peripheral devices. To bring the Raspberry Pi into a usable state, it requires an Operating System - software that schedules and executes essential routines and user programs. The Raspberry Pi is configured to boot from an SD card - that is to say, it reads a bootable image from an SD card. Essentially, it reads in CPU instructions from the SD card that further instruct it to load up the rest of the Operating System software.

To put an Operating System on an SD card, you typically download a compiled image and flash it to the SD card. The procedure (for a computer running Windows) is as follows:

- Download the Raspbian image from http://downloads.raspberrypi.org/raspbian_latest
- Extract the .zip file
- Install Win32DiskImager
- Run Win32DiskImager as Administrator (from right-click menu)
- Insert the SD card into your SD card reader, if you have not done so
- In Win32DiskImager, "Browse" to the extracted .img file

- Select the SD card reader as the destination device - make sure you don't select your hard drive!

For Linux, Mac OSX, and Unix-variants, the procedure is slightly different:

- Download the Raspbian image from http://downloads.raspberrypi.org/raspbian_latest
- Extract the .zip file (unzip -x file.zip)
- Insert the SD card if you have not done so
- use `dmesg | tail` to determine the path to your SD card (should be `/dev/sdX`)
- use `sudo dd bs=4M if=/path/to/rpi.img of=/dev/(SD Card)` to flash the image - this will take a while!

Finally, plug your Raspberry Pi into a power supply (micro-B USB cable), keyboard, and monitor, and you should see the Raspberry Pi logo and some scrolling text on the screen. If this did not work, please consult one of the volunteers.

Shell Basics

This section describes some basic commands for interacting with a Linux shell - specifically, we will be using the bash shell. This is the default shell on many popular Linux distributions. If you have not yet logged in to your raspberry pi, do so now - the username is pi, and the password is raspberry. Plug your raspberry pi into a monitor, power source (USB Cable) and keyboard, and you will eventually be greeted by a login prompt.

Listing 1 gives some examples on "getting around" in the shell. The commands used - for example `cd`, are what you would type at the command-line prompt after logging in to the raspberry pi.

Smart City - Demonstrate and deploy responsive Web Application using APIs and generate reports using templates

Manage directories and files

Objective:

- to manage directories and files.

Requirements:

System

Files

PROCEDURE:**Managing Files And Folders**

A file is a collection of numbers which have been written to your computer's hard drive. These numbers can be converted into a picture, a sound, text, or a set of instructions for a program to perform certain actions. A clue to a file's type is given by its Icon or by its file extension. Folders are containers for anything on a computer including files and other folders. A path lists the folders that have to be opened to get to the required file.

Folders were called directories on older Microsoft systems.

These are some typical file icons. They help the user to identify the file type. There are innumerable file types, some of the common ones are represented here:

- .bmp – a bitmap image
- .doc – a Word document
- .wav – a sound file
- .ppt – animated slides
- .txt – plain text
- .xls – a spreadsheet
- .dbf – a database file
- A shortcut (note the arrow)
- .exe – an application (a program)

Windows allows you to view information about files in different ways. the icon view – the default used by Windows XP.

To change the icon view, click on View on the menu bar. Select the required view from the available list.

By default if a file type is a known one, such as a Microsoft Word Document, Windows won't display its

file extension. To view all file extensions click on Tools on the menu bar.

Various options can be configured. e.g. Display compressed files and folders with alternate colours. To display all file extensions, untick the Hide file extensions for known file types box.

File extensions are best left alone. Opening a file with the wrong application can sometimes damage the file. However you may at some stage need to change a file's extension.

Folders

Each of these is a folder. They may contain files or other folders (called subfolders) or both. There may be many "nestings" of folders within folders.

Files and folders are located on the computer by using a file path. The "James" folder is located inside a folder called "Home", which is located inside a folder called "es-net", which is located on the "C:" drive. The file path will be "C:\es-net\Home\James".

Moving and Copying

To move a file or folder, either right click on its icon OR left click on the Edit option on the toolbar. Choose cut to move or copy to copy!

At this point the item has been placed onto a clipboard – an area of memory accessible from nearly any application in Windows. Right click (or open Edit in the toolbar) in an open destination folder and choose "Paste" (or use drag and drop) .

When an attempt is made to move an item between volumes, it is effectively copied, and the original remains.

Creating Files and Folders

This is mercifully easy. Simply right-click on some empty space in any suitable folder or the desktop and choose to create a new object from the choices offered.

Be careful not to alter the file extension, as this can render the file unreadable. File extensions are usually hidden for this reason.

Smart City - Demonstrate and deploy responsive Web Application using APIs and generate reports using templates

Manage user access and data security by cryptography

Objective:

- to manage user access and data security by cryptography.

Requirements:

- System
- IoT devices

PROCEDURE:

How secure?

First let's qualify this post by acknowledging that security is a moving target and that motivated hackers with unbound resources will likely find new vulnerabilities: this should not deter us from implementing basic good practices, and even more advanced practices when the applications demands. A burglar is easily deterred from the average property that has an obvious security alarm, but a determined burglar on a mission to breach a high value property with known riches inside will not be stopped by an alarm.

And so it is with IoT security: raising the bar has value, how high depends on the consequences of a breach.

Why secure?

Security is generally driven by the need to protect valuable digital assets and individual privacy. In the IoT applications many devices will interact with the physical world and the consequences of a security breach can cause physical damage and threaten lives. Here are a couple of examples to get you thinking responsibly about security and cyber-physical security:

- Johnson & Johnson warns of insulin pump hack risk
- Cybersecurity vulnerabilities identified implantable cardiac devices
- Six cyber-physical attacks the world could live without

Security is often considered a 'cost' of doing business but it does not need to be so: in a growing number of applications the quality of the data is important and 'trusted data' can command a premium. Here is one great example:

Companies may be willing to pay for data from Fitbits

Here we will discuss three areas to consider when designing an IoT device or system:

- Ensure that data comes from devices of known provenance and is unaltered – trusted data.
- Protect devices so they don't become a host for malware or an access point into your network.

- Protect intellectual property that's embedded within or accessed by an IoT device.

Whats different about IoT?

Cyber security is a \$100 billion dollar industry that's focussed on protecting servers and client devices within a mature and homeogenous ecosystem: windows, linux, ios, intel, ARM, and a few others. Client devices are well resourced and protected by caring users, server farms are protected beneath layers of physical, electrical and software security, and dedicated teams monitor and manage security 24/7.

In contrast, IoT devices are generally light on resources, diverse in functionality and live at the edge of the network in unknown territory – perfect ingredients for digital chaos! Additional challenges with IoT security include:

- Often design to interact with unknown parties in the physical world
- No coherent IoT security standards
- Complex technology stack that spans hardware, firmware, software, cloudware
- Organizational responsibility not clear
- Not high on management's priority list
- Customers won't pay a premium for security

Fortunately many of the building blocks required to secure IoT devices and system are already well established and widely used across the internet, connecting servers, PC's and mobile devices. There is no need to 'invent' new technology; what's needed is an intelligent approach to applying the existing building blocks in ways that are appropriate to IoT devices and systems.

Here we will review some of these building blocks and where they apply. More detailed reviews of each block will be available in related posts.

Trusted data is the currency of IoT

For the internet of things to work the data coming from the edge devices (the Things) must be 'trusted' all the way from sensor to subscriber.

Without trust the data is at best useless, perhaps misleading and in some cases can result in life-impacting consequences. Cloud applications, in particular Big Data and AI, depend upon trusted data.

Fundamental ingredients of trusted data are:

- Authenticity – data comes from a known (unique Identity) and authenticated (verified to match Identity) source.
- Integrity – data has not been changed and can be verified to be so.
- Confidentiality – data cannot be observed by an unauthorized party.

Let's explore these terms more detail.

Device authentication

Most data transactions between parties or devices assume that you know the data truly came from the presumed party or device. But how can you be sure?

The generally accepted solution comes in two parts:

- Give the device a unique and robust Identity
- Use a cryptographic process to authenticate the device against its known Identity

Data that originates from an authenticated device, with the additional requirement of integrity, is sometimes referred to as 'authenticated data'. More on that in a moment.

Data integrity

Knowing that data has not been tampered with is fundamental to most transactions (yes there are a few where it does not really matter). If the data represents currency then integrity seems an obvious requirement. If that data represents air quality sensors, then integrity matters because it might negatively impact lives if an emergency response is compromised in a material way.

Data confidentiality

Confidentiality or privacy of data is not required by all applications – municipality transit data should be public for example – but keeping data confidential is important in matters ranging from an individual's privacy rights to a corporation's need to protect proprietary information. In each case the data must be protected from prying eyes using encryption techniques that extend from the publishing source – the IoT sensor or edge device- to the cloud and onwards to subscribers.

Using standard encryption algorithms is a simple and effective way to ensure privacy. See Cryptography section.



Set up a Linux File system

Objective:

- to set up a Linux File System.
-

Requirements:

- Raspberry Pi 3B/4B
- Power Adaptor for Raspberry Pi 3B/4B
- LCD/LED Monitor
- HDMI Cable
- USB Keyboard
- USB Mouse
- MicroSD Card
- Raspberry Pi 3B/4B
- Power Adaptor for Raspberry Pi 3B/4B
- LCD/LED Monitor
- HDMI Cable
- USB Keyboard
- USB Mouse
- MicroSD Card

PROCEDURE:**Setting up Raspberry Pi Linux computer**

Using Raspberry Pi as a Linux machine is popular. As noted earlier, OSS developers generate revenues by selling hardware that runs open-source software by providing customer support for OSS and selling closed source software that runs on OSS. Now think about how the Raspberry Pi Foundation generates revenues.

Raspberry Pi as Linux machine

We will be using Raspberry Pi as a Linux machine. Several Linux distributions can run on RPi without any hassle like Ubuntu Mate, Ubuntu Core, Ubuntu Server, Kali Linux, and Centos, etc. Though any Linux Distro can be used, we will stick to Raspbian, the official OS for Raspberry Pi. Raspbian is a Debian based Linux distribution. It comes with Bash Shell as a default shell and its desktop environment. The latest version of Raspbian (released on September 26, 2019) is based on Linux Kernel 4.19.

As we will focus on Python programming to control electronics, we simply need to set up Raspberry Pi as a Linux machine where we can code, test, and run Python applications. Python is the official programming language for Raspberry Pi that comes with IDLE (integrated development environment) by default. There are two versions of Python that can run on Raspberry Pi – version 2.7 and version 3.x. The latest version of Raspbian has IDLE only for Python 3. If you wish to code your applications in Python 2.7, its IDE can be installed as well. Both versions are incompatible with one another. This can be a source of confusion. So, we will stick to using Python 3.5.

Setting up Raspberry Pi desktop

Before anything else, we first need to set up our Raspberry Pi Linux computer. Setting up RPi desktop is similar to setting up any desktop computer except that we don't need to assemble CPU. RPi is a single board computer with on-board RAM, processor, input/output, and network interfaces

Perform system initialization

Objective:

- to perform system initialization.
-

Requirements:

- Raspberry Pi
- System

PROCEDURE:

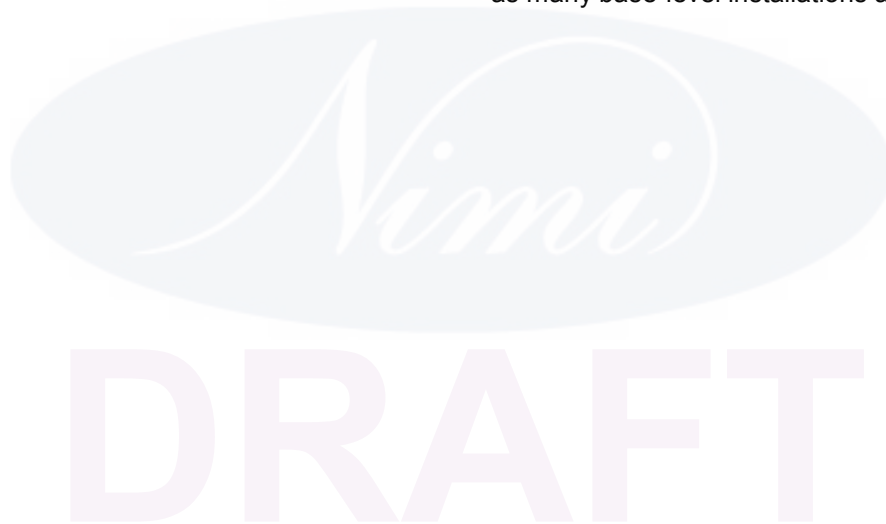
The Raspberry Pi is a great thing: it is real computer, it is cheap (\$40), it can interface with electronics, talk to the web and has full HDMI support.

However it runs on Linux, which I have a love-hate relationship with. I love the idea of Linux, but when I start messing around the command line and downloading packages and installing things, I often get lost.

I've assembled bits and pieces from various online posts and guides into this Instructable, which is what I call the "Ultimate Raspberry Pi Configuration Guide".

What this Instructable does is to set up a wireless Raspberry Pi that allows you to:

- ssh into from the Terminal window on the Mac (or equivalent on another machine).
- run wirelessly with a static IP for each SD card.
- automatically startup, no log in
- set your the time zone
- skip the GUI of the Raspberry Pi, which bogs things down and is unnecessary for most tasks
- minimize any external monitor use, specifically never having to lug a monitor over to where the ethernet router lives
- clone a "basic settings" SD card so that we can have as many base-level installations as we want.



Smart City - Identify and test smart Lightening system and its components

Connect a system to a network

Objective:

- to connect a system to a network.
-

Requirements:

- Raspberry Pi
- Ethernet Cable
- Arduino UNO
- System

PROCEDURE:

1. Make sure your network is running. In order to connect to an Internet network, the network's modem must be connected to an Internet output (e.g., a cable outlet), the network's router must be connected to the modem, and both the router and the modem must be turned on.

- Some modems include a built-in router.
- You can check the status of your Internet connection by looking at the status lights on the modem and/or router. If the Internet connection looks like it's faltering, you may want to use Ethernet instead of Wi-Fi
- Click the "Wi-Fi"



2. icon. It's in the bottom-right corner of the screen, on the right side of the taskbar. Clicking the Wi-Fi icon prompts a pop-up menu with a list of available networks.

- You may first have to click

- in the lower-right corner of the screen to see the Wi-Fi icon

3 Turn on Wi-Fi if necessary. If you see a message that says "Wi-Fi Turned off" at the top of the pop-up menu, click the Wi-Fi box in the bottom-left corner of the pop-up menu before proceeding.

4 Select your wireless network's name. Click the name of the wireless network to which you want to connect. The network's name will expand.

- If you don't see your network's name, you may need to move closer to your router/modem.
- Your network's name will most likely have your router/modem name, model number, and/or manufacturer name in its title if the network hasn't been set up before.

5 Click Connect. It's below the network's expanded name in the pop-up menu.

6 Enter the network's password. In the text box that appears, type in the password used to log into your network.

- If you haven't changed the network's password from the factory default, the password can be found on a sticker that's usually on the bottom or back of the router (or modem/router combination).
- If the network doesn't have a password, clicking Connect in the previous step will connect you to the network.



Install and Configure Linux

Objective:

- to install and configure Linux.

Requirements:

- Micro SD card
- Computer System
- Raspberry Pi
- Micro USB
- Internet Connectivity

PROCEDURE:

What you'll learn

- How to create a bootable Ubuntu Server microSD card
- How to setup internet connectivity on the Raspberry Pi
- How to access your Raspberry Pi remotely

What you'll need

- A microSD card (4GB minimum, 8GB recommended)
- A computer with a microSD card drive
- A Raspberry Pi 2, 3 or 4
- A micro-USB power cable (USB-C for the Pi 4)
- A Wi-Fi network or an ethernet cable with an internet connection

And optionally:

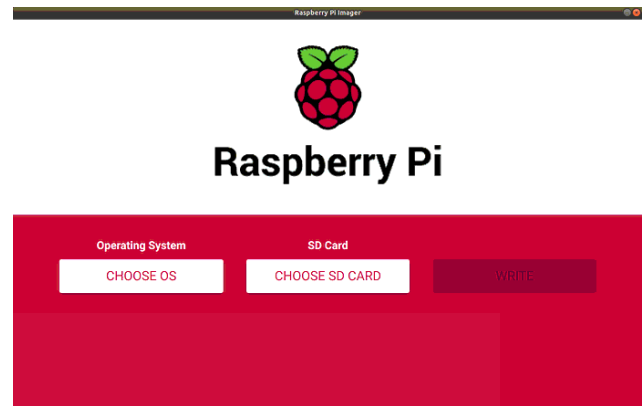
- A monitor with an HDMI interface
- An HDMI cable for the Pi 2 & 3 and a micro HDMI cable for the Pi 4
- A USB keyboard

First, insert the microSD card into your computer.

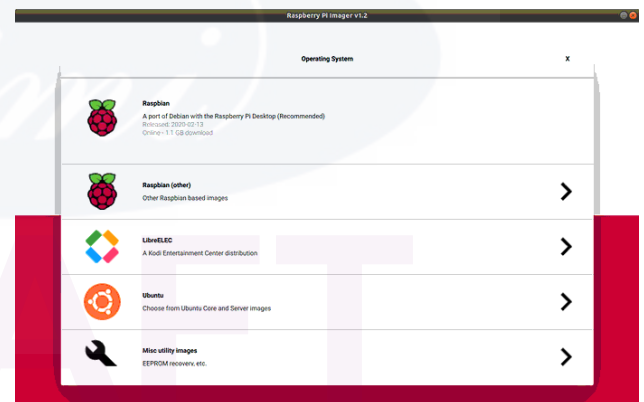
Now you need to install the right Raspberry Pi Imager for your operating system. You can do this with the following links:

- [Raspberry Pi Imager for Ubuntu](#)
- [Raspberry Pi Imager for Windows](#)
- [Raspberry Pi Imager for macOS](#)

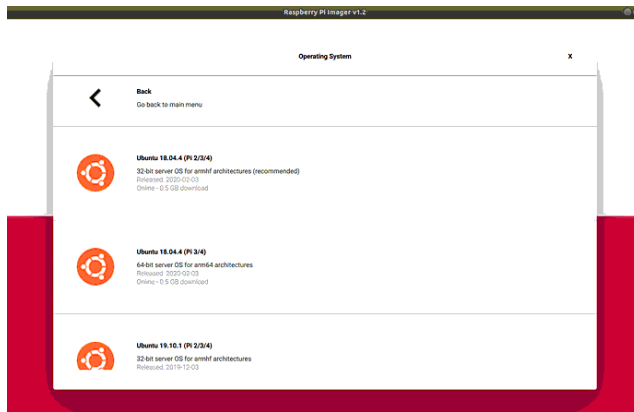
Once this is done, start the Imager and open the "CHOOSE OS" menu.



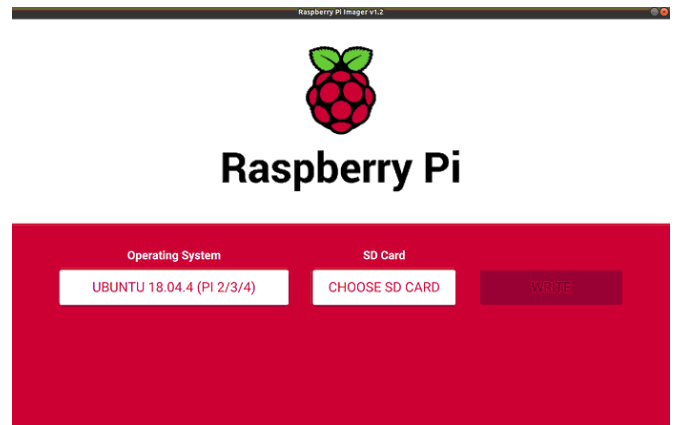
Scroll down the menu click "Ubuntu".



You will then be able to see a list of Ubuntu downloads to choose from. For this tutorial we recommend you select the Ubuntu 18.04 download. As indicated in the imager this will work for the Raspberry Pi 2,3, 3+ and any of the 4's.



Select the image and open the “SD Card” menu. Select the microSD card you have inserted.



Finally, click “WRITE” and wait for the magic to happen... (This magic might take a few minutes)



Create Shell scripts, Flow control in shell, Advanced Shell features

Objective:

- to create shell scripts, Flow control in shell, advanced shell features.
-

Requirements:

Computer System

Shell Language

PROCEDURE:

WHAT IS A SHELL SCRIPT?

Shell scripts allow you to automate pretty much anything you can do on the Linux command line. A shell script is simply a text file containing a series of shell (command line) commands that are executed one by one, as if they were input directly on the command line. They can save time if you need to execute lots of commands at once, or if you want to execute commands automatically, like when the Pi boots up.

The shell language is a high level programming language. It's more removed from system and memory processes than lower level languages like assembly or C. This basically means that it has commands and functions to perform tasks that would otherwise take a lot of code to execute.

Shell scripts can make use of variables, if-then statements, loops, and pipes (see below).

WHAT CAN A SHELL SCRIPT DO?

Shell scripts are great if you need to enter long sequences of commands into the command line to do something. Most operations can be accomplished with a single command if you know how to write a shell script for it. For example, they can be used to:

- Control what happens when the computer boots up
- Start applications when an event occurs
- Use the output of one command as the input for another command
- Batch rename and move files

- Batch convert file formats

One very useful feature of a shell script is the ability to create pipes. A pipe allows the output of one command to be forwarded to the input of the next command. Pipes can be used with as many commands as you want. The basic syntax for a pipe is:

command 1 | command 2

HOW TO CREATE AND RUN A SHELL SCRIPT

I'll just cover the basics of writing shell scripts here,

CREATE THE SCRIPT FILE

Shell scripts are simply an executable text file with the extension ".sh". In this example we'll write a simple "hello world" script to demonstrate how to create and run them.

To begin, log in to your Raspberry Pi, and navigate to the directory where you want to save the script. Then open the Nano text editor and create a new file named hello-world.sh by entering this at the command prompt:

sudo nano hello-world.sh

Now, enter this code into the text editor:

The first line of this program, `#!/bin/sh`, is called a shebang. This tells the BASH shell to execute the commands in the script. Every shell script you create will need this on the first line of the script.

Exit and save the file in Nano by pressing Ctrl-X to save and exit.

MAKE THE FILE EXECUTABLE

Next, we'll need to make the hello-world.sh file executable. To do that, enter this at the command prompt:

sudo chmod +x hello-world.sh

Smart City - Identify and test smart Lightening system and its components

Explore Database Management system

Objective:

- to explore Database Management system.
-

Requirement:

Raspbian

MariaDB

DBMS

Computer System

PROCEDURE:**How to install MariaDB on Raspbian and the Raspberry Pi****What is MariaDB?**

Before we go to the installation, let's talk a little more about MariaDB, what it is and what it serves.

Until now, we talked about MySQL, which is probably the most used DBMS (at least 56% market share on free technologies). But MySQL is not alone in the world of free DBMS and all its competitors, the most serious is undoubtedly MariaDB, whether by its market shares (18%), by its functionalities, very similar to those of MySQL, Or by its perfect compatibility with MySQL. Not surprisingly, since the creator of MariaDB (Michael Widenius) is also the creator of MySQL.

Why did Michael Widenius create MariaDB instead of continuing to improve MySQL?

To find the answer to this question, you have to make a small jump back in the history of MySQL.

- In 1995 Michael Widenius founded MySQL AB, an open source software company, and soon became a major player in the database industry.

- In 2008, MySQL was bought for \$ 1 billion by Sun Microsystems, another major player in the field of open source software.
- In 2009, Sun Microsystems was acquired by Oracle, and MySQL with it. It is at this moment that everything will change. There were real concerns about Oracle's willingness to maintain and evolve MySQL in an open source and free way, with Oracle already offering a competing, paying, proprietary solution for large companies, the well-known Oracle Database.
- In 2009, Michael Widenius leaves Sun on his redemption, get back the source code of MySQL (free software) and launches a petition to pressure Oracle to continue to develop MySQL.
- In order to ensure that the code of MySQL remains free, and that it does not disappear, Michael Widenius will then create a new DBMS, fork of MySQL (a fork is a software created from the source code of another).

He will name this DBMS after his second daughter name, Maria, as he had done with MySQL and his first daughter, My.

A foundation will then be create, the MariaDB Foundation, to ensure the governance of the project. Since then, MariaDB develops more and more every day, continuing to ensure compatibility with MySQL and improving it.

Test Cloud and Server Configuration for IoT

Objective:

- to Test Cloud and Server Configuration for IoT.

Requirements:

- Raspberry Pi
- Computer System
- Internet connectivity
- Power Cable

PROCEDURE:

In addition to setting up your own VPS, we will show you how to install a simple example that lets you control LEDs, lights, etc., by using a browser. The following figure shows how any number of devices can be controlled in real-time by navigating to your own VPS using a browser. The online VPS functions as a proxy and makes it possible for any number of users to control the devices via the online server.



Some of the benefits of using an online server and operating it as a proxy for routing messages between users and the devices are:

- Enables users on one network to control devices on another network. For example, a device connected to a home network can be controlled via a smartphone connected to the cellular network.
- Devices operate as network (TCP/IP) clients, thus no router port forwarding or other tricks are required for gaining access to the devices via the online server.
- Devices acting as network clients operate in stealth mode, thus greatly improving the security of the devices. We will get more into security later.
- Small devices typically do not have the resources for providing a direct web interface. The online server manages all the heavy lifting of providing a web interface for the devices, thus greatly minimizing the code complexity in the devices.

Virtual Private Server

Setting up your own VPS may sound complicated and expensive, but the reality is that setting up a VPS is very easy and the cost can be minimal. It is possible to find VPS services as low as \$8 a year. My favorite site for

shopping for a budget VPS is lowendbox.com and one of my favorite VPS providers is securedragon, where you can get a three-month lease for less than \$6 and a yearly lease for \$12.

Memory Limitation

A limitation with a budget VPS is the amount of memory available. The budget VPS we are using in this article has only 64Mbyte of memory, thus limiting the selection of IoT server solutions that can be used. The IoT solution we will use in this article is an application server framework that includes an IoT protocol, a web server, a server-side scripting language, and a database in one unit. This software package can easily run on a budget VPS, and in my test environment, I was able to connect 10,000 devices to the online server. I performed the test by using a number of additional Virtual Private Servers, where I ran a number of simulated devices.

Linux

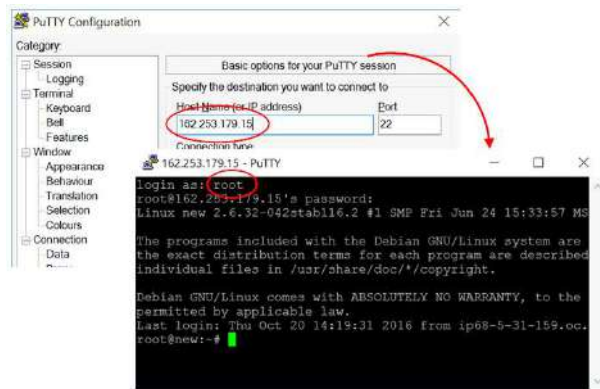
A budget VPS will be running Linux. If you have never used Linux, you may think that this is not for you; however, nothing can be further from the truth. An online VPS is the best starting point for anyone new to learning command-line Linux because no matter what you do with the VPS, an easy-to-use web-based VPS control interface will let you restore the VPS to its original configuration by a button click.

For this reason, playing around with an online VPS is a great way to learn command line Linux. If you get your Linux VPS into a state where you are unable to recover it, log into the VPS web control panel and click the re-install button. A fresh Linux VPS will then be available in one to two minutes, and you can start over.

Remote Login via SSH

An online VPS enables you to remotely login by using SSH (Secure Shell). If your own computer is running Windows, download the SSH client Putty and start the executable from any directory. Mac and Windows computers typically include a command-line SSH client, thus you can simply type ssh in a command line on Linux and Mac.

When you sign up for a VPS service, such as the budget securedragon VPS (O64), you will receive an email with instructions for how to remotely log into your VPS by using SSH. The instructions include the IP address of your new VPS and the root (admin) password. The following figure shows how to enter the IP address in Putty and how to log in as user root in the command line window.



At this point, feel free to try a few Linux commands. Don't worry about corrupting the VPS since a simple button click in the VPS control panel will re-install the Linux operating system.

Installing the IoT Server

To install the server software and the example program, copy all of the following and paste the commands into the SSH console window.

```
wget http://makoserver.net/install/brokerX86/install.sh;
```

```
chmod +x install.sh;
```

```
./install.sh
```



Smart City - Identify and test smart Lightening system and its components

Test QT based GUI for sensor output

Objective:

- to test QT based GUI for sensor output.
-

Requirement:

- Laptop
- Graphical User Interface
- Windows
- MySQL
- Raspberry pi
- Internet Connectivity

PROCEDURE:**Introduction**

If someone is running a Windows laptop for long-term testing and wishing they could have something small-scale and cheaper, this problem solves that. This is done by eliminating the laptops and replacing them with single board computers, running Linux. The Raspberry Pi foundation is responsible for the single board computer series Raspberry Pi, which are high performance / low price computers. These small computers are powerful enough to replace larger, more powerful computers in certain tasks. In this project we have replaced laptops that are used to store and visualize data that come from sensors in long-term tests. This will be done in Raspbian, the officially supported operating system for Raspberry Pi hardware. For data storage, two databases using MySQL is implemented. One of these will be used for local storage that will save local measurements, and another database that will be used to store all measurements from multiple local databases. Visualizing the measurements is done using a graphical user interface (GUI), that will be able to display the measurements in e.g. a graph over time. The user will also be able to change which measurements that are shown by changing time interval, sensor and type of graph. The Qt framework will be used to design the GUI.

Background

APR Technologies (henceforth called APRtec), a company based in Enköping, is the "external party for this project. The company develops embedded systems solutions for the space industry. Their primary product areas are fire alerts, heat dissipation, sensors and surface materials. The main areas that this project will incorporate are database management, data visualization, and single board computers, namely the Raspberry Pi single board computer. A basic understanding of each of these areas is useful when reading this report and therefore a brief introduction will be given to each of them. Furthermore, the measuring

units used to collect data will be explained, as well as our choice of operating system for the single board computers.

Database management

A database is, simply put, a structured collection of data. The data in a database is useful in some way, represents some real-world entity and can be anything from phone numbers or names to images or books. In this sense, a library, images on a hard drive or even a collection of physical movies can be seen as databases, but in the field of computing this refers to data stored on a memory unit. The area of database management involves the creation, deletion, maintenance etc. of databases using a database management system. A database management system is defined as a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications. In other words, a database management system can be seen as an interface to an actual physical database, allowing users to interact with it in different ways. There are a number of various database management systems available for use such as MySQL, SQLite and Oracle.

Data visualization

Data visualization concerns the area of taking data and presenting it in a concise and useful way, depending on requirements. The simplest example are bar charts to determine some effect, to more complex visualizations such as three-dimensional scatter plots. There are many tools and frameworks for this available for Python, such as Matplotlib, Chaco or PyQtgraph. There are possibly performance issues to consider when selecting one or more of these. The basic requirement for visualization is to integrate it in a touch screen GUI.

Single board computers

Single board computers are small computers contained on a single circuit. They contain one or more CPUs along with memory and programmable input/output peripherals. Single board computers are used in embedded applications, in contrast to the microprocessors used in personal computers, ranging from basics such as equalizers to more advanced uses like cars or satellites. Their main advantage is cost-effectiveness and ability to perform many different tasks depending on user requirements. The Raspberry Pi Foundation is a charity that was founded in 2009. The foundation has developed a series of single-board computers to promote computer science in schools.

These single-board computers have a high performance/cost ratio, are as big as a credit card, and are provided by the foundation. Like other single board computers the Pi can be used in many embedded systems areas, such as scientific experiments, home automation and data analysis.

Operating system

Linux is a Unix-like operating system that contains free software. The group has chosen the Linux distribution called Raspbian for the project. This is the officially supported operating system for the Raspberry Pi. Raspbian is a modified version of Debian, which is an operating system that uses the Linux kernel. The main reasons for using Raspbian is threefold: it comes with most or all of the software packages needed, built-in support for the Raspberry touch screen, and in the group's opinion a larger chance of long-term support.

Measuring unit

In this project, the measuring unit will be considered a "box" that is a collection of sensors that sends its collected data via USB. In other words, we are not concerned about the internal functionality of the unit. The data sent by the measuring unit will be handled by our software, which is run on the Raspberry Pi's. In total, sixty measuring channels are available. Each channel is connected to a sensor, that has the ability to measure real world physical phenomena.

Purpose, aims, and motivation

APRtec currently uses Windows laptops for their work.

There are ready-made solutions available for these systems, but APRtec is unsatisfied with these due to stability concerns and general clunkiness of Windows. This is due to there being so much pre-installed software that comes with each installation. Every manufacturer likes to provide their own software for what they consider added functionality. Even if removing all of it is possible, there exists the possibility that one missed something. This could then interfere with system operation over time. The company has given examples of some program installing an update without asking after several weeks and thus interrupting the work the unit was doing. A single board computer would be more appropriate to use in terms of the above, physical size and with Linux being more stable. Linux also does not include bloatware to the same degree (depending on distribution), and automatic software upgrades must be enabled manually by the user. The overall question that this project aims to answer is how the Raspberry Pi single board computer can be used to store and visualize different types of measurement data. The project is important because it would result in a general purpose solution to the problem of storing and visualizing data. Many different solutions exist for the purpose of visualizing and storing data using conventional computers, but there is a lack of a general purpose solution using a cheap, small, portable system like the Raspberry Pi. Reaching the goal of this project would therefore allow individuals with a small budget to visualize and store data, something that is useful in many different fields of work.

APRtec
DRAFT

Test IoT Web and Application Development Tools for IoT

Objective:

- to Test IoT Web and Application Development Tools for IoT.
-

Requirements:

- Computer System
- Internet Connectivity
- Eclipse
- Arduino

PROCEDURE:**IoT Development Tools/Platforms****Eclipse IoT <https://iot.eclipse.org>**

Eclipse IoT provides technology needed to build IoT Devices, Gateways, and Cloud Platforms. Basically, Eclipse IoT is an ecosystem of companies and individuals collaborating to establish an Internet of Things based on open technologies. This collaboration helps to completely focus on the development, promotion and adoption of open source IoT technology. Many of our members provide a wide range of projects and services built on top of Eclipse IoT technology.

Node-Red

<https://nodered.org>

Node-Red is a browser-based editor which makes it very easy to wire together flows using nodes in the palette that can be deployed to its runtime in a single-click. Node-Red provides a built-in library that let you save helpful functions or templates for re-use. With 225,000+ modules in Node's package repository, it is easy to extend the range of palette nodes to add new capabilities.

Dronesmith

Drones are one of most emerging areas in the Internet of Things (IoT) space. This path-breaking technology has the potential to change each and every domain across the world. Dronesmith aims at making use of the drones much easier.

Arduino

<https://www.arduino.cc>

An Open-source prototyping platform, Arduino is based on easy-to-use hardware and software. Arduino is a hardware specification for interactive electronics and a set of software which includes an IDE and the Arduino programming language. Arduino is "a tool for making computers than can sense and control more of the physical world than your desktop computer."

IFTTT

When it comes to home automation, IFTTT provides robust support to several products. This makes it easy to connect and control IoT automation devices.

M2MLabs Mainspring

<http://www.m2mlabs.com>

M2MLabs Mainspring is an open source application framework to develop machine to machine (M2M) applications for remote monitoring, fleet management or smart grid. It covers different functionalities needed by M2M devices such as flexible modeling of devices, device configuration, communication between devices and applications, validation and normalization of data, long-term data storage, and data retrieval functions.

Canopy

<http://canopy.link>

Canopy simplifies the IoT cloud by acting as a cloud relay between IoT devices and applications. Canopy allows quick and seamless development of IoT solutions for diverse markets ranging from consumer, commercial and industrial. Canopy's server-side component (the "Canopy Cloud Service") is open source and runs anywhere, including the public cloud, private cloud, hybrid cloud, LAN, or even on your laptop.

Intel® Edison Development

<https://software.intel.com/en-us/iot/hardware/edison>

Intel® Edison Development Platform is the first in a series of low-cost, product-ready, general purpose compute platforms that help lower the barriers to entry for entrepreneurs of all sizes—from pro-makers to consumer electronics and companies working on the Internet of Things (IoT).

Tessel 2

<https://tessel.io>

Tessel 2 is a robust IoT and robotics development platform. Leverage all the libraries of Node.JS to create useful devices in minutes with Tessel. Interact with the physical world from sensing to actuation to connecting with other devices. Each module has an open source library on NPM, with instructions and tutorials available online. It's literally plug, npm install, and play.

Smart City - Identify and test smart Lightening system and its components

Select and Install Carbon dioxide sensor

Objective:

- to select and Install Carbon dioxide sensor.
-

Requirement:

- Raspberry Pi UART
- COZIR Co2 sensor
- Internet connectivity
- HDMI Cable
- Computer System
- USB Cable

PROCEDURE:

Carbon monoxide poisoning takes the lives of around 400 people every year in non-fire related cases. It is quite important that you protect your family and employees from becoming another statistic. Apart from being aware of the symptoms of poisoning in progress and ensuring that your fire and heating sources are checked, one sure way of preventing carbon monoxide poisoning is to install a carbon monoxide detector in your home, office or workshop.

Steps:

- 1 Do some research on the right carbon monoxide detector as per your needs:** There are two basic types of detectors available in the market. One is battery powered while the other is AC powered. Do some research on the best option for your home, office or wherever you may want to install it. It is quite important to stick to AC powered detectors if you realize you might have a problem remembering to change the batteries. You may also want to buy a detector whose alarm goes louder and faster as carbon monoxide levels increase.
- 2 Decide the installation location:** Deciding on the location of your detector is quite crucial. Since carbon monoxide is less dense than air, it tends to rise. This makes the best location to be near the ceiling and away from fuel burning appliances. It should at least 15 feet (4.6 m) away from heating and cooking appliances and humid areas. Care should also be taken to ensure that it is not covered by drapery, furniture or anything else. Recommendations by the consumer product safety commission (CPSC) are that the detector should be placed near your bedroom so

that it wakes you up if it goes off. For multi-level apartments it is recommended that you place detectors on every floor.

- 3 Unpack product and read installation instructions:** Make sure that the packaging has everything. Battery powered units usually come with screws and anchors while AC powered units simply need to be plugged in.
- 4 Mark installation holes:** Take off the twist off base and line it with the wall in the position that you have chosen for installation. Mark the holes with a pencil dot.
- 5 Punch holes on the dots:** With the aid of your wall punch and hammer, make holes on the marked spots. Care should be taken to make holes that are not larger than the provided screw anchors in order to achieve a perfect fit.
- 6 Install the screw anchors:** Take your screw anchors and place them on top of the holes one at a time and gently tap them into position with the aid of the hammer.
- 7 Install the detector base into the wall:** If your carbon monoxide detector has a twist of base, take it off and screw it into position using your screwdriver and the supplied screws. In case your detector does not have a twist off base, place the screws into the anchors and screw then in without tightening. Make sure that they protrude enough to let the detector fit over them.
- 8 Install batteries and fit detector into position.**
- 9 Test it:** You need to make sure that your detector is working by testing it. this will also give you an opportunity to familiarize with the way it sounds.
- 10 Schedule battery replacement:** You will need to mark on your calendar or schedule an electronic reminder on your computer, mobile etc. for the replacement of the batteries twice every year. You will also need to check the chemical that enables the detection process. It should be replenished periodically as well.

Identify and install oxygen sensor

Objective:

- to identify and install oxygen sensor.

Requirement:

- Grove board
- Raspberry Pi
- Oxygen sensor
- Computer system

PROCEDURE:

we talked about Grove connectors and how to use them with the Raspberry Pi. You will be using three Grove devices for this great experiment to create a sophisticated Oxygen sensor. We have previously used an ADC with the Raspberry Pi in our Air Quality Sensor column in Raspberry Pi.

Interface Board

The Pi2Grove board (Figure 1) provides 15 Grove connectors: 10 Grove Digital, 4 Grove I2C, and one Grove UART. Fast prototyping with many, many Grove devices is supported. No software drivers are required, so you are right next to the hardware.



Figure 1: The Pi2Grove Board.

The board plugs right into your Raspberry Pi with a special connector that supports installing other boards or HATs on top of the Grove board itself.

The Grove 4 Channel 16 Bit ADC Board for the Raspberry Pi is based on the industry standard ADS1115 (Figure 2). It supports voltages of both 3.3V and 5V and also uses Grove connectors. The board has four 16 bit Analog-to-Digital Channels, which can be programmed in many different ways.



Figure 2: Grove 4 Channel 16 Bit ADC Board.

The O2 Grove Gas Sensor (Figure 3) works by outputting voltage values proportional to the current oxygen concentration in the air around you. You can interpret these numbers by referring to the oxygen concentration linear characteristic graph.



Figure 3: Grove Oxygen Sensor.

This sensor value only reflects the approximate trend of gas concentration in a permissible error range, so it is not exact. Detecting the specific makeup components in the air usually requires a more expensive device, which is far more precise than a simple gas sensor. This sensor also requires around half an hour to warm up (Figure 4).

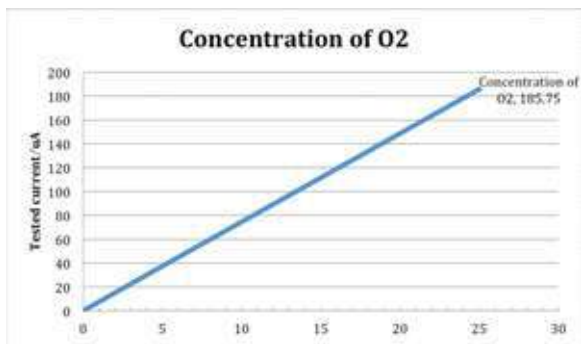


Figure 4: Grove Oxygen Sensor Curve takes about 30 minutes to reach the correct level.

All of the connections save one Jumper are Grove connections (Figure 5). All Grove connections use the same cable to connect to one another.

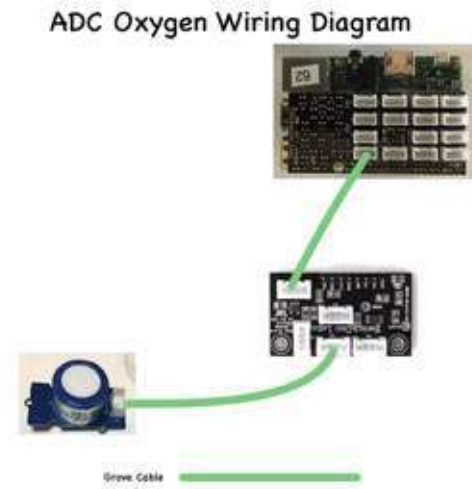


Figure 5: Project wiring diagram.

Nimi
DRAFT

Smart City - Identify and test smart Lightning system and its components

Select and install volatile organic compound sensor

Objective:

- to select and install volatile organic compound sensor.

Requirement:

- Computer System
- Volatile organic compound sensor
- USB Cable
- Internet connectivity

PROCEDURE:



Volatile organic compounds (VOCs) are emitted as gases from certain solids or liquids. VOCs include a variety of chemicals, some of which may have short- and long-term adverse health effects. Concentrations of many VOCs are consistently higher indoors (up to ten times higher) than outdoors. VOCs are emitted by a wide array of products numbering in the thousands.

Organic chemicals are widely used as ingredients in household products. Paints, varnishes and wax all contain organic solvents, as do many cleaning, disinfecting, cosmetic, degreasing and hobby products.

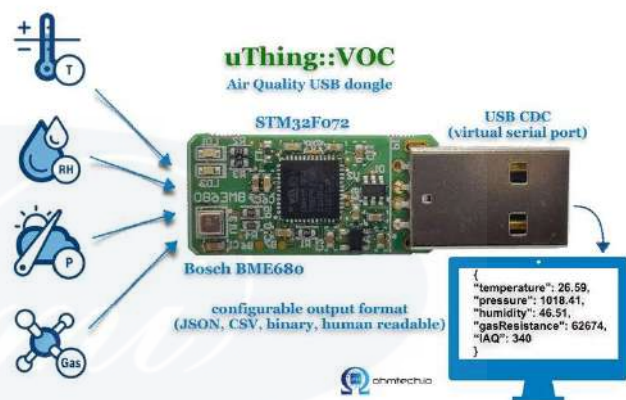
Since many people spend much of their time indoors, long-term exposure to VOCs in the indoor environment can contribute to serious health issues.

Good ventilation and air-conditioning systems are helpful at reducing VOCs in the indoor environment.

In this project, we will learn how to measure VOC levels using a USB sensor dongle connected to a Raspberry Pi, store the data into an InfluxDB time-series database and use the Grafana application to quickly build an air-quality monitoring dashboard.

Measuring VOCs indoor levels

OhmTech's uThing::VOC USB air-quality sensor dongle provides a quick way of obtaining data from the Bosch Sensortec BME680 sensor module.



The STM32 MCU integrated in the USB dongle captures raw data from the sensor and uses the Bosch's proprietary Air Quality calculation algorithms to obtain equivalent air-quality (IAQ Index), temperature, relative humidity and atmospheric pressure values. These values are then output over a simple Virtual Serial Comm on the USB connector in different configurable formats (JSON, CSV, Human readable or binary).

The uThing::VOC can be configured for different output data rates and formats. By default the sensor data is output every 3 seconds in JSON format. We will use the default settings for this project, but they can be changed easily, check the datasheet for more information.

Smart City - Identify and test smart Lightening system and its components**Identify and install Air temperature, Air humidity, and Atmospheric pressure and UV sensor****Objective:**

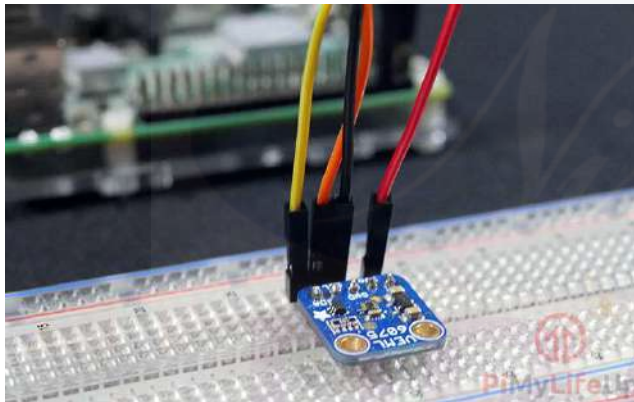
- to identify and install Air temperature, Air humidity, Atmospheric pressure and UV Sensor.

Requirements:

- Raspberry Pi
- Ultra Violet UV sensor
- USB cable
- Computer System
- Internet connectivity

PROCEDURE:**Raspberry Pi UV Sensor using the VEML6075**

In this Raspberry Pi UV sensor project, we will be walking you through the process of connecting the VEML6075 ultraviolet (UV) sensor to the Raspberry Pi.



Alongside wiring the UV sensor to the Raspberry Pi we will be showing you how you can interact with the sensor by utilizing the Python programming language.

To make this entire tutorial work, we will be making use of Adafruit's VEML6075 library. This library handles most of the grunt work by doing all the necessary calculations for working out the UV index and reading the UVA and UVB values from the UV sensor.

The VEML6075 UV Sensor is a dual-band sensor that senses both UVA (ultraviolet A/longwave) and UVB (ultraviolet B/shortwave) light bands.

One of the best things about this sensor is that it provides its data over the I2C serial protocol, making it very simple to interact with.

You can bundle this sensor with the many other sensors that work with the Raspberry Pi. For example, using a variety of sensors, you could build your own weather station.

Below is a list of equipment that you may need for setting up the VEML6075 UV sensor with the Raspberry Pi.

Recommended

Raspberry Pi

8GB SD Card or Micro SD Card if you're using a Raspberry Pi 2, 3 or B+

Breadboard wire

VEML6075 UV Sensor

Optional

Breadboard

Raspberry Pi Case

Smart City - Identify and test smart Lightning system and its components

Select and Install Nitric Oxide (NO), Hydrogen sulphide, Sulphur dioxide, Carbon Monoxide, Ozone soil Moisture and soil temperature sensor

Objective:

- to select and Install Nitric Oxide (NO), Hydrogen sulphide, Sulphur dioxide, Carbon Monoxide, Ozone soil Moisture and soil temperature sensor.

Requirements:

- Power supply
- Power controller
- Arduino mega
- LCD
- IoT sensors
- Internet connectivity

PROCEDURE:

INTRODUCTION

As per World Health Organization [WHO] air tainting is infectivity of the indoor or outside condition by any manufactured and normal administrator which changes qualities of the earth. Nuclear family start devices, vehicles and forest blazes are normal base of air sullyng and bustle pollution. Pollutions which are accountable for prosperity concern consolidate particulate issue, carbon monoxide, ozone, nitrogen dioxide and sulfur dioxide.

Air tainting cause respiratory and various sicknesses, which can be damaging. WHO has assessed nature of air in about 1500 urban territories and Indian capital city was the one of the most dirtied urban zones the world over. Delhi is having most raised gathering of particulate issue which is smaller than 2.5 micrometer .

Air pollution and nonattendance of air quality watching centers address regular and imaginative troubles for urban networks and circumstances around the world. To stand up to this issue, industry has focused its undertakings in finding an adaptable mechanical elective that allows the improvement of the air quality evaluating method and gives reference regards in framework goals where customary watching fails to cover appropriately. Appallingly, existing things and the created results don't address ease courses of action. Indian cultivating portion speaks to 18% of India's gross local generation (GDP) and offers work to half of country's workforce. India is second greatest producer of vegetables and natural items on the planet .

In India 43% of topographical locale is included by cultivating region . According to conjecture of FAO, one out of nine people far and wide need adequate sustenance expected to keep up a strong and dynamic lifestyle and says that this number is required to augment as masses continues rising. Not simply that, a couple

inspects have shown that the enthusiasm for the sustenance will be duplicated by 2050 due to augment in people. It will be furthermore moving errand to fulfill the requirements of masses in perspective on withdrawing water level, changes in climatic changes and contracting number of cultivating fields . The entrenched methods for creating harvests won't help in extending the agrarian yield with all the recently referenced issues. These assurances demands for more research headway in rustic region. As shown by late reports, around 70-80% of agricultural ozone draining substance releases, for instance, nitrous oxide start from use of nitrogen compost.

Thusly, one must apply the required proportion of nitrogen composts in the wake of knowing the essential of the yields. Despite giving water to agribusiness field should be in controlled manner with the objective that it won't wash away the minerals added to soil similarly as cause soil breaking down. All these requires steady checking of the cultivating field similarly as reap advancement. Hence it requires usage of current techniques in plant fields which associates in advancement of gather proficiency with sufficient use of advantages.

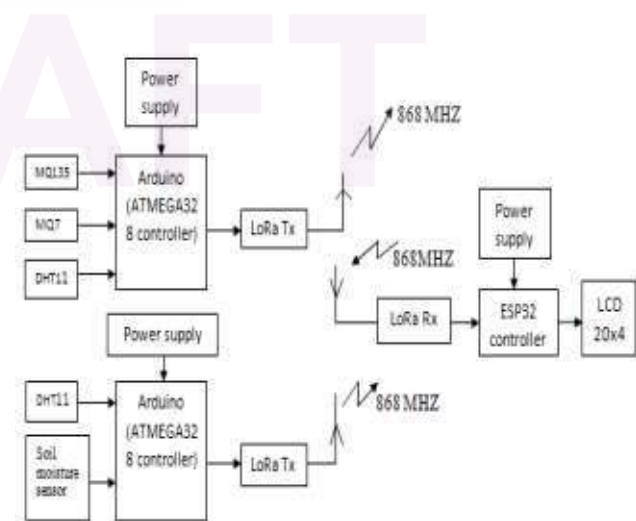


Figure 1. Block Diagram of the System

The working of this adroit city application is essentially established on the remote correspondence between the sensors and various devices. Along these lines for capability reason we are using LoRa Network system,

which will give compelling and streamlined results. The objective of this endeavor is to manufacture a smart city with less multifaceted nature, upgraded power supply, spending admirably arranged and capable delayed consequence of correspondence sort out. Here to perceive the climatic parameters we are setting remote sensors which will observe the sullyng gases like carbon mono oxide, hydrogen sulfide gases, and smoke. We will use soil soddenness and temperature sensor to identify the sogginess in the earth. Resulting to recording the data it will send to the entry where the pro is accessible. We are utilizing adruino uno stage for transmitting the information. Atmega328 microcontroller we are utilizing to gather the information from various sensors and set it up to transmit to the recipient area. Esp32 we are utilizing on the collector area to get the transmitted information. ESP32 is a movement of negligible exertion, low – control system on chip microcontrollers with consolidated Wi-Fi and twofold mode Bluetooth.

Data accumulated from various sensors will be then arranged to make the results. This made information is then appeared on the LCD. Above square outline demonstrates the working of the air checking and crop ranch framework. The framework is separated mostly into 3 section to be specific transmitter1 (Tx1), transmitter2 (Tx2) and beneficiary (Rx). Transmitter1 is utilized to gather and transmit the information from the sensors utilized for air checking while transmitter2 is utilized to gather and transmit information from the yield estate sensors. Recipient is utilized to get the transmitted information from the transmitters and procedure the information to give the ideal outcome. This ideal information is then shown in LCD. Transmitter 1 (environmental data transmitter) Transmitter1 is blend of MQ135, MQ7 gas sensor DHT11 Humidity, Temperature sensor, Lora module for remote transmission and Microcontroller. It is called as Environmental sensors. The gas sensor is interface to the simple stick which is in simple in structure. The DHT11 works sequentially so it is sequentially interface. The Lora module utilizes SPI convention to transmit and get the information. All the three sensors Gas sensors and temperature and stickiness are giving their information to the controller. The errand of controller to process these information and send it to the recipient side for air checking utilizing lora convention.

Figure 2. Flowchart of Transmitter of Environmental Sensors Algorithm of Transmitter (Environmental Node Circuit)

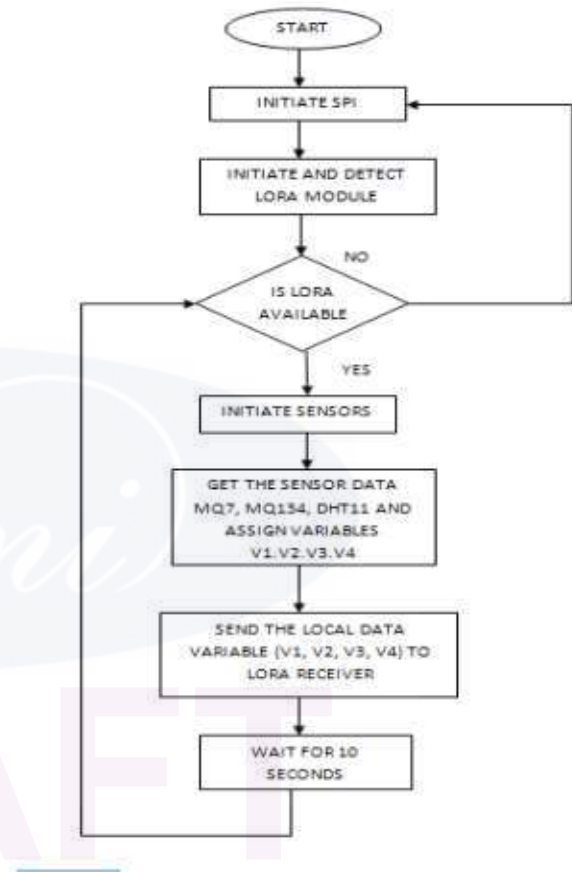
- Step 1. Start the circuit by connecting the power supply
- Step 2. Start the SPI protocol
- Step 3 detect the LoRa module (868 MHz)
- Step 4 If LoRa Available start all the sensor otherwise goto step 2

Step 5 Get the data from all the sensors (MQ7, MQ135 and DHT11)

Step 6 Send the data received from the sensor to LoRa receiver

Step 7 Wait for 10 seconds

Step 8 Goto step 4 Transmitter 2 (plantation data transmitter) Transmitter2 made up of DHT 11, Soil moisture sensor, Microcontroller which has a place with AVR family and Lora module. DHT 11 and Soil moisture sensors are consolidate called as crop plantation sensor.



Microcontroller gathers the information from crop plantation sensors and send it remotely through Lora module to the Receiver.

Figure 3 Flowchart of Transmitter of Plantation Sensors Algorithm of Transmitter (Plantation Node Circuit)

- Step 1. Start the circuit by connecting the power supply
- Step 2. Start the SPI protocol
- Step 3 detect the LoRa module(868 MHz)
- Step 4 If LoRa Available start all the sensor otherwise goto step 2

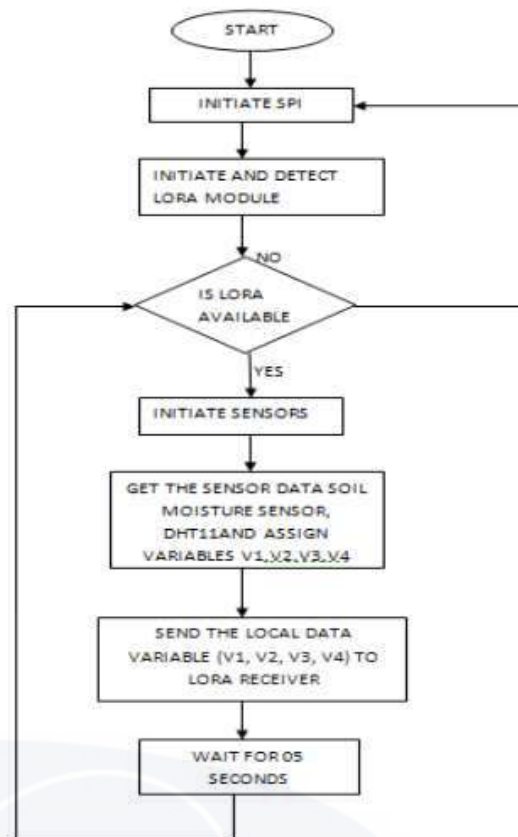
Step 5 Get the data from all the sensors (DHT11 and soil moisture sensor)

Step 6 Send the data received from the sensor to LoRa receiver

Step 7 Wait for 5 seconds

Step 8 Goto step 4 Receiver Recipient comprising of LCD module 20x4,ESP32 and Lora as a remote collector module.

The information got from Transmitter1 and Transmitter2 are gathered remotely process through ESP32 microcontroller and show on LCD. The information constantly send by crop plantation sensors and Air quality measure sensors are gotten it and procedure persistently on continuous premise and show on LCD.



Smart City - Identify and test smart Lightening system and its components**Check magnetic field for smart parking and IR for Human presence****Objective:**

- to check magnetic field for smart parking and IR for human presence.

Requirements:

- Computer System
- Arduino Mega
- Power Supply
- Power controller
- IoT sensors
- USB Cable

PROCEDURE:**INTRODUCTION**

Automation of devices has revolutionized the comfort levels of the human life. New devices are equipped with automated systems for reducing the need for human effort. The eco-friendly automation is relying on sensors, transducers, and control systems. Therefore researchers have focused much on developing sensors with new technologies for sensing and detection of the targeted objects. Human exhibits various unique characteristic like body temperature, shape and color, and release of CO₂. Our research work aims to create a system which can respond to the human body temperature for detection. IR sensors are best and accurate to measure radiated heat by human body. It is the most precise characteristic for human detection in sensitive locations. It is well known that anything that has a temperature above absolute zero emits infrared radiation.

Another crucial challenge is human body temperature is not constant which varies from 36.5 to 37.5 °C due to wearable clothes, external conditions like seasons and day and night timings. The primary factors for localization of human presence may include the distance between the heat source and sensor system. The body surface radiation is absorbed by the sensor can range from 20°C to 40°C depending on the external conditions. For indoor applications the sensor should be able to differentiate human from static objects/things like metal objects, wooden furniture and plastic objects even though the human is mobile or immobile. The parameters like distance, angle and coordinates are calculated using the position of the sensing device for accurate human location in the field of view (FOV) or targeted location. Therefore, it is highly essential to choose the IR sensors for concerning range, accuracy, and FOV.

This work has Omron D6T-44L-06 MEMS IR sensor which results 4x4 array pixel low-resolution thermal image as graphical output. The outcome of this work is to visualize the thermal signature of human and human-

object interaction in two dimension (2D) using thermopile modules of Omron D6T sensor. The position of sensors is also another crucial parameter to consider for better coverage of FOV. The number of required sensors is based on size of FOV and human count by avoid the interference range between multiple sensors. Unlike CCTV cameras, the low resolution 2D thermal signature output of the thermopile sensor is not smooth to interpret thereby reduces the privacy concerns of the users. The 2D array matrix output of Omron D6T MEMS series sensor identifies the pixels of relatively high temperature than the surrounding and gives the location and presence of the human in range. The PIR sensors were most commonly preferred to detect human presence before the thermal sensors due to low cost. However PIR sensor can able to identify the signals from the person in motion and simultaneously it results the false negative when the person was stationary. PIR sensor also has another limitation of time transmission delay of approximately 2 seconds to 9 minutes. On the other hand, the thermal sensor can continuously differentiate a stationary human from other static or dynamic objects.

The thermal sensor is more capable device for human detection than existing technologies like visual camera. Although visual camera gives the high quality images higher than 4x4 pixel sensing array, whereas using visual camera at night time required continuous illumination of light. The characteristic of the thermopile does not allow energy emitting to the surrounding which keeps the sensor unaffected. Hence, visual camera and thermopile sensor based technologies are failed to achieve human privacy and accuracy.

APPLICATIONS OF THE SYSTEM

The thermal sensory system is a new field has many revolutionary capabilities.

Security and Privacy

The human detection systems are very important in security and surveillance fields. Due to the limitations of PIR sensor, now the emerging applications are demanding thermal sensor based human/object detection systems. To further improve the security aspects, thermal sensors can be coupled with PIR and ultrasonic sensor to make the device more perfect for detecting each human activity.

Disaster Relief

Every year thousands of people die in natural disasters, a significant percentage of them can be saved if efficient detection devices are available. Therefore, smart thermal

sensing devices can help to detect the humans buried under the rubble. Currently, these thermal sensors are not able to penetrate deeply into the rubble, but they still can provide an extra 3-8m arm to the rescuers which indeed a significant role for saving the people.

Medical Treatments and Safety from Radiation

Thermal imaging plays a key role in medical treatments. If adequately calibrated sensors can detect the variations found in thermal signature of the patients which helps to detect the appropriate problem and diagnosis. One of the most recent utilization of these proposed sensors involves alert the people from radiation exposure.

Energy Management Systems

The automated devices can use the thermal sensors like Omron D6T to detect the presence of the human to control home appliances. Smart thermostats have developed to adjust the indoor temperature automatically and regulate the room environment through the mobile applications.

Activity Recognition and Localization

Another critical application of Omron D6T thermal sensor deals with assisting older and physically challenged people. The sensor was successfully able to provide positive results in detecting human activities like sitting, lying, falling, and many more. These sensors can also connect to the home network which can make home and human-activity monitoring by these category people.

F. Advanced driver assistance system (ADAS)

ADAS involves heterogeneous sensors which are installed in vehicle to assist the drivers while driving. The sensory system accurately alerted the driver about pedestrian near the vehicle through IR sensor.

G. Thermal Cameras and Fire Safety

Well-equipped hardware components and image processing algorithms results the quality thermal images. A thermal camera provides night vision which most of the visual cameras lack and therefore thermal camera is much more suitable for alarm.



Smart City - Identify and test smart Lightening system and its components

Measure the Hall Effect (doors and windows opening), Water presence, Liquid level, Liquid flow, Temperature and Humidity for smart security

Objective:

- to Measure the Hall Effect using water presence, Liquid level, Liquid flow, Temperature and Humidity for smart security.

Requirement:

- Raspberry Pi
- IoT sensors
- Jumper wires
- Pair of magnetic switches
- Computer System

PROCEDURE:

The Smart Security model is designed to monitor security parameters such as perimeter access control, liquid presence detection and doors and windows openings. Each model has six dedicated sockets to connect sensor probes. Besides, a relay system allows this model to interact with external electrical machines.

Presence and light sensors are installed around the home, monitoring activity in determined areas and if the lights are being used regularly. Temperature and humidity sensors are also employed, as well as door and window monitors. Some locations like the bathroom and the kitchen need specific monitoring. Water flow and water leakage sensors can be installed here to prevent water overflow or any other sink misuse.



Some of the Smart Security sensor applications are :

Perimeter Access Control

Access control to restricted areas and detection of people in non-authorized areas.

Liquid Presence

Liquid detection in data centers, warehouses and sensitive building grounds to prevent break downs and corrosion.

Radiation Levels

Distributed measurement of radiation levels in nuclear power stations surroundings to generate leakage alerts.

Explosive and Hazardous Gases

Detection of gas levels and leakages in industrial environments, surroundings of chemical factories and inside mines.

List of Sensors available :

- Relay to control external DC units. (<30VDC, 1A)
- IR human presence
- Hall effect (doors and windows openings)
- Water presence
- Liquid level
- Liquid flow
- Temperature
- Humidity
- Atmospheric pressure
- Luminosity (Luxes Accuracy) for Smart Lighting
- Ultrasound (distance measurement)

Libelium provides many different sensor probes depending on what is going to be measured. For Smart Security model following sensor probes can be used.

Liquid Flow Sensor Probes

Water Flow Small, YF-S402

- Flow rate: 0.3 ~ 6 L/Min
- Working voltage: +5 V ~ +24 V
- Working temperature: 0 °C ~ 80 °C
- Pipe connection: 1/8"

- Max rated current: 15 mA (DC 5 V)

Liquid Level Sensor Probes

PTFA3415 Sensor



Water Flow Medium, FS300A

- Flow rate: 1 ~ 60 L/Min
- Working voltage: +5 V ~ +24 V
- Working temperature: 0 °C ~ 80 °C
- Pipe connection: 3/4"
- Max rated current: 15 mA (DC 5 V)

- Measurement Level: Horizontal
- Liquids: Water
- Material (box): Propylene
- Material (float): Propylene
- Operating Temperature: -10 °C ~ +80 °C
- Minimum consumption: 0μA*

PTFA0100 Sensor

- Measurement Level: Horizontal
- Liquids: Heavy oils and combustibles



The liquid flow sensors output a signal that consists of a series of digital pulses whose frequency is proportional to the flow rate of the liquid through the sensor. That digital signal, whose frequency is in the range between 0Hz and 100Hz, is directly read through one of the digital input/output pins of the micro-controller.

- Material (box): Polyamide
- Material (float): Polyamide
- Operating temperature: -10 °C ~ +80 °C
- Minimum consumption: 0μA*

PTFA1103 Sensor

- Measurement Level: Vertical
- Liquids: Water
- Material (box): Propylene
- Material (float): Propylene
- Operating temperature: -10 °C ~ +80 °C
- Minimum consumption: 0 μ A*



There are three liquid level sensors whose operation is based on the status of a switch which can be opened and closed (depending on its placing in the container) as the level of liquid moves the float at its end. The main differences between the three sensors, regarding its use in Waspote, are to be found in their process for placing them in the container (horizontal in the case of the PTFA3415 and PTFA0100 sensors, vertical for the PTFA1103 sensor) and in the material they are made of (the PTFA1103 and PTFA3415 sensors recommended for edible liquids and certain acids and the PTFA0100 for heavy oils and combustibles, more specific information can be found in the sensors' manual).

Hall Effect Sensor Probe



- Length: 64 mm
- Width: 19 mm
- Height: 13 mm
- Maximum contact resistance (closed): 200 m Ω
- Minimum contact resistance (open): 100 G Ω
- Minimum consumption: 0 μ A*

This is a magnetic sensor based on the Hall effect. The sensor's switch remains closed in the presence of a magnetic field, opening up in its absence. Together with its complementary magnet it can be used in applications of monitoring proximity or opening mechanisms.

Water Leakage / Liquid Detection Sensor Probe (Line)

- Length: 5 meters sensor + 2 meters jumper wire
- Material: PE + alloy lend
- Weight: 18 g/meter
- Pull force limit: 60 kg
- Cable diameter: 5.5 mm
- Core resistance: 3 ohm/100 meters
- Maximum exposed temperature: 75 °C
- Detectable liquids: Water
- Minimum consumption: 0 μ A*



Test calibration kits for the sensor probes for water quality analysis

Objective:

- to test calibration kits for the sensor probes for water quality analysis.

Requirements:

- Computer system
- Arduino
- IoT sensors
- USB cable

- Difficult Process.
- Take More Time.

PROCEDURE:

NEED OF WATER QUALITY MONITORING IN SHRIMP CULTURE

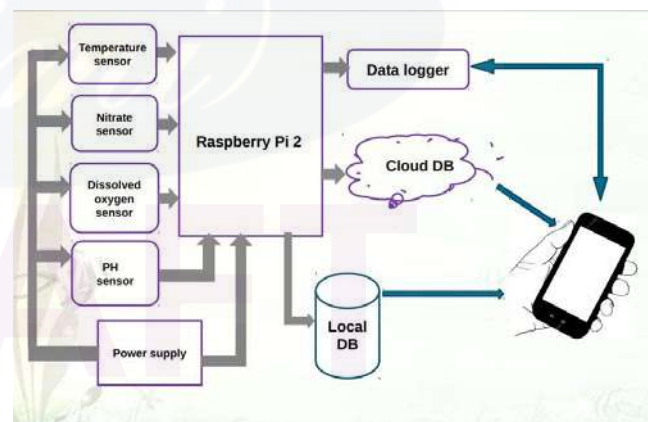
Need of Water Quality in Shrimp Ponds

Water Quality (WQ) determines the ultimate Success or Failure of an aqua culture. The farmer must measure, record and manage WQ all through the growing season. Water Quality parameters affect respiration, feeding, metabolism and reproduction of shrimps.

Water Parameter	Optimum level
Temperature	26-33C
Salinity	10-25 ppm
Dissolved oxygen	>3ppm
PH	7.5-8.5
Total Ammonia Nitrogen	<1.0ppm
Total Nitrate Nitrogen	<5.0ppm
Nitrite Nitrogen	<0.01ppm
Sulphide	<0.03ppm
Biological Oxygen Demand	<10ppm
Chemical Oxygen Demand	<70ppm
	(parts per million)



IMPLEMENTATION ARCHITECTURE



Water Quality Parameters

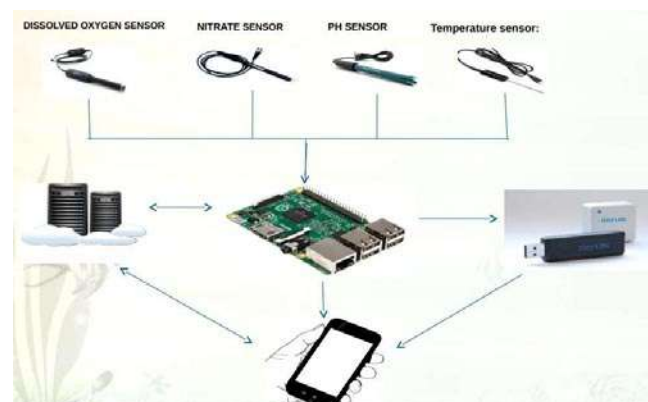
- pH
- Temperature
- Dissolved Oxygen
- Hardness
- Ammonia, Nitrite & Nitrate

Water Quality Parameters

Formers take water samples from pond.

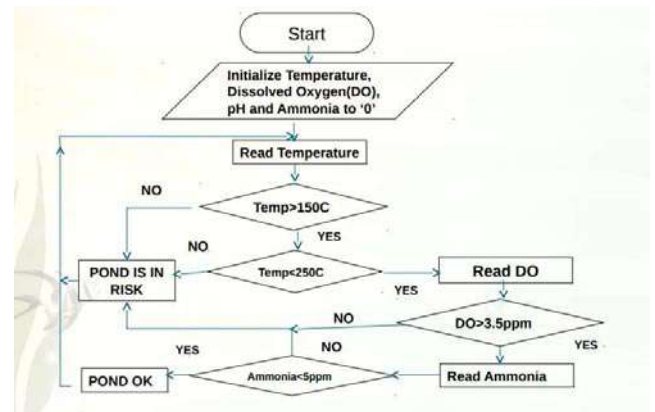
- Get results from Aqua Labs.
- Expensive.

SENSOR SUITE FOR WATER QUALITY SCREENING



Raspberry Pi 2

- Processor
- GPIO Pins
- SD Card Slot
- Ethernet Port
- USB Port
- HDMI Port 7.3.5mm Jack



Vimi
DRAFT

Power up the solar inverter as per the device manual

- Objective:
- to power up the solar inverter.

Requirement:

- Solar inverter
 - IoT devices
 - Computer System
- PROCEDURE:

About this Manual

This manual is to provide the explanation and procedures for installing, operating, maintaining, and troubleshooting the below solar inverters:

RPI M6A/ RPI M8A/ RPI M10A

Product Description

This device is a 3-phase grid-tied solar inverter which does not support off-grid functionality.

The operation of solar inverter is shown as figure 1-1. Inverters convert the DC input power supplied from the PV Array into 3-phase AC output power to Grid.

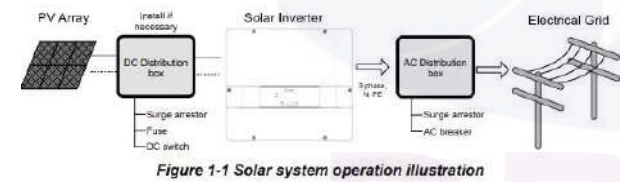


Figure 1-1 Solar system operation illustration

Checking Unit and Accessories

Unpredictable damages may occur during shipment. Check if all the accessories are in the package, the standard accessories are list as Table 2-1:

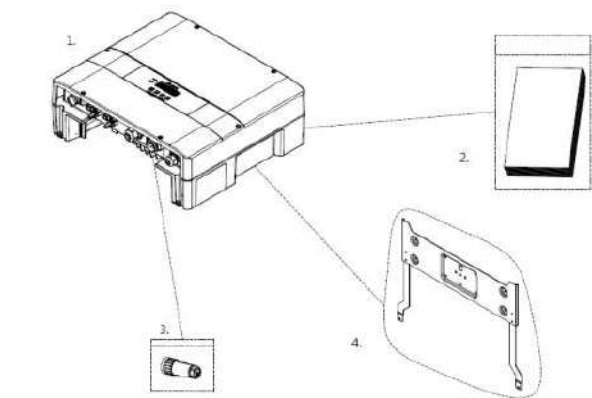


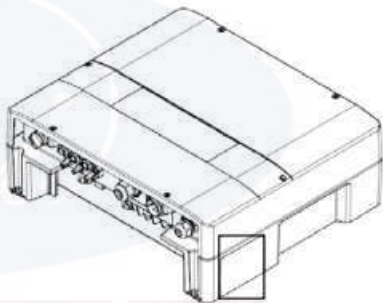
Figure 2-1 Packing listUnit

RPI M6A/ RPI M8A/ RPI M10A			
	Object	Qty	Description
1	PV Inverter	1	RPI M6A/ RPI M8A/ RPI M10A solar inverter
2	User Manual	1	The Instruction to provide the information of safety, Installation, specification, etc.
3	AC Plug	1	Connector for AC connection
4	Mounting Bracket	1	To mount solar inverters on the wall.

Table 2-1 Packing list

Product Label

Users can identify the model number and the specifications by the information on the product label. The location for the label please sees Figure 2-2.



RPI M6A/ M8A/ M10A

Figure 2-2 Product label

Exterior Objects

The Inverter's exterior objects are shown in Figure 2-3. The detailed input/output interfaces illustration is shown in Figure 2-4.

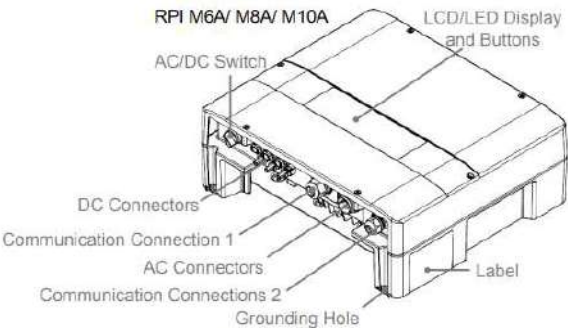


Figure 2-3 Inverter's exterior objects

Smart City - Identify, select, install and troubleshoot different module/ devices used in smart street light based on IoT and cloud Technology

Integrate solar inverter with serial protocol working on Modbus Master Software

Objective:

- to integrate solar inverter with serial protocol working on Modbus Master Software.

Requirements:

- Computer system
- Solar Inverter
- Modbus Master Software

PROCEDURE:

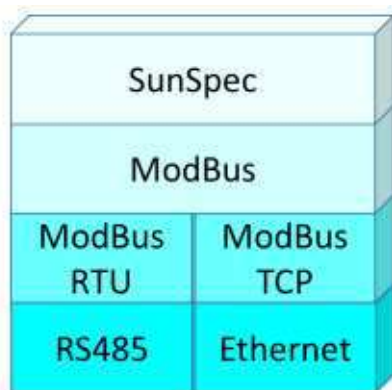
SolarEdge inverters support reading inverter-level monitoring data directly from the inverter to a local non-SolarEdge device, by implementing the SunSpec open protocol for interoperability between devices in renewable energy systems. This option can be used alongside the connection to the SolarEdge monitoring server. This document describes the connection method and the protocol and configurations needed to implement this feature. Direct connection to a monitoring device is useful when a network connection is unavailable, when extensive custom data processing is required, or when authorities require direct access to monitoring data. In many cases, it is possible – and recommended – to employ the direct connection alongside a SolarEdge monitoring platform connection. Connection to the monitoring platform enables all the monitoring benefits, primarily:

Proactive installer maintenance and real time troubleshooting by SolarEdge support, using with the physical mapping available only in the monitoring platform

Module-level monitoring

Communication Technologies

SolarEdge uses an open, industry-standard communications stack in order to provide efficient messaging between SolarEdge and third-party devices and applications.



The communications stack components are briefly described below.

SunSpec

SunSpec is an application-layer communications protocol designed to achieve interoperability between Distributed Energy Resource (DER) components and smart grid applications.

Modbus

Modbus is a serial communications protocol typically used to connect data collection terminals to a centralized processing unit. SolarEdge products use Modbus to perform SunSpec messaging over two types of physical/link-layer channels:

Modbus RTU: Remote Terminal Unit (RTU) Modbus over a serial RS485 connection

Modbus TCP: Modbus over an Ethernet connection
SolarEdge systems support a single Modbus master only – either single Modbus RTU or single Modbus TCP

SunSpec Supported Inverters

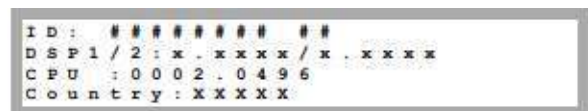
Depending on their type, SolarEdge devices may be configured in either of the two ways:

- Using SetApp
- Using the LCD

All SolarEdge inverters with SetApp configuration are SunSpec-supported. SolarEdge inverters with the LCD that have Firmware version 3.xxxx and above only are SunSpec-supported.

To check the inverter firmware versions (for inverters with the LCD):

- Short press the LCD light button until reaching the following screen:



- If required, upgrade to the latest available firmware, as described in

Use Cases for MODBUS over RS485

This section describes RS485 options to connect the inverter to a non-SolarEdge monitoring device.

Smart City - Identify, select, install and troubleshoot different module/ devices used in smart street light based on IoT and cloud Technology

Communicate and verify the parameters on DLMS server software

Objective:

- to communicate and verify the parameters on DLMS server software.

PROCEDURE:

Introduction:

The following test specification defines the tests used to declare conformance to the Modbus protocol specifications. This test specification defines the criteria for passing the Conformance Test and being declared conformant to the published protocol specification. Also included are the tests the laboratory will use when testing a Modbus server. At this time, a conformance test consists of a Protocol Test of a Modbus server only. The Protocol Test, acting as a Modbus Client, assesses Modbus conformance by checking the server's supported Modbus Function codes. Additionally it will test for message error handling and message format error handling. If the Server passes each required Criterion, it is declared a Conformant Server for the current test revision. The test setup is based on using Modbus TCP. For Modbus TCP devices, the connection is made over TCP/IP. For Modbus Serial devices, a properly configured Schneider Electric ETG-100 Modbus Ethernet-to-Serial Gateway is used.

Definitions

DUT – Device Under Test. The device being tested for conformance.

IP v4 Address. Internet Protocol Address. This is the 32-bit address assigned to a device.

All IP addresses on a network must be unique.

STP (Shielded Twisted-Pair). Cabling used for Ethernet, where strands of wire are twisted together and are shielded either individually or in pairs.

List of public Modbus function codes:

Function Code (in dec)	Subcode	Description	Data Type
1		Read Coils	0x
2		Read Discrete Inputs	1x
3		Read Holding Registers	4x
4		Read Input Register	3x
5		Write Single Coil	0x
6		Write Single Register	4x
7		Read Exception Status	Diagnostics
8	0-18, 20 (5-9, 19, and >20 RESERVED)	Diagnostics	Diagnostics
11		Get Com Event Counter	Diagnostics
12		Get Com Event Log	Diagnostics
15		Write Multiple Coils	0x
16		Write Multiple Registers	4x
17		Report Slave ID	Diagnostics
20		Read File Record	File
21		Write File Record	File
22		Mask Write Register	4x
23		Read/Write Multiple Registers	4x
24		Read FIFO Queue	4x
43	13	CANopen Interface Transport	Other
43	14	Read Device ID	Other

List of Valid Exception Responses

Modbus Exception Codes		
Code (hex)	Name	Meaning
01	Illegal Function	The Function Code received in the query is not an allowable action for the server/slave. This may be because the Function Code is only applicable to newer devices, and was not implemented in the unit selected. It could also indicate that the server/slave is in the wrong state to process a request of this type, for example because it is unconfigured and is being asked to return register values.
02	Illegal Data Address	The data address received in the query is not an allowable address for the server/slave. More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, the PDU addresses the first register as 0, and the last one as 99. If a request is submitted with a starting register address of 96 and a quantity of registers of 4, then this request will successfully operate (as far as address is concerned at least) on registers 96, 97, 98, 99. If a request is submitted with a starting register address of 96 and a quantity of registers of 5, then this request will fail with Exception Code 0x02 "Illegal Data Address," since it attempts to operate on registers 96, 97, 98, 99 and 100, and there is no register with address 100.
03	Illegal Data Value	A value contained in the query data field is not an allowable value for server/slave. This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does NOT mean that a data item submitted for storage in a register has a value outside the expectation of the application program, since the Modbus protocol is unaware of the significance of any particular value of any particular register.
04	Slave Device Failure	An unrecoverable error occurred while the server/slave was attempting to perform the requested action.
05	Acknowledge	Specialized use in conjunction with programming commands. The server/slave has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a timeout error from occurring in the client/master. The client/master can next issue a Poll Program Complete message to determine if processing is completed.
06	Slave Device Busy	Specialized use in conjunction with programming commands. The server/slave is engaged in processing a long-duration program command. The client/master should retransmit the message later when the server/slave is free.

08	Memory Parity Error	Specialized use in conjunction with Function Codes 20 and 21 and reference type 6, to indicate that the extended file area failed to pass a consistency check. The server/slave attempted to read record file, but detected a parity error in the memory. The client/master can retry the request, but service may be required on the server/slave device.
0A	Gateway Path Unavailable	Specialized use in conjunction with gateways to indicate that the gateway was unable to allocate an internal communication path from the input port to the output port for processing the request. Usually means that the gateway is misconfigured or overloaded.
0B	Gateway Target Device Failed to Respond	Specialized use in conjunction with gateways to indicate that no response was obtained from the target device. Usually means that the device is not present on the network.

Smart City - Identify, select, install and troubleshoot different module/ devices used in smart street light based on IoT and cloud Technology

Power up the energy meter as per the device manual

Objective:

- to Power up the energy meter as per the device manual.

PROCEDURE:

Caution - We are not responsible for any mishappening while reproducing this project by anyone.

Energy Meter using XMC1100 and TLI 4970 & Wi-Fi module NodeMcu(ESP8266)

The Energy meter is as an application of TLI4970 (Current Sensor) and XMC 2Go and is a plug and play device to any electrical socket with AC supply

In this application, Energy meter has the following features

- Displays Power, Energy Consumed by the appliances & an estimation of the bill one may incur.
- Remotely Monitor the energy of home appliances.

Power from the AC mains is drawn and passed through a fuse for avoiding any damage to the circuit board during the accidental short circuit.

Then the AC power line is distributed in two parts:

1. To the load through the current sensor (TLI4970).
2. 230V AC/5V DC Power Supply module.

The current sensor measures the amount of current passing through a load and sends a 16-bit SPI data (13-bit current value) to the XMC 2Go in which energy, power, and bill calibration takes place.

XMC 2Go sends the data to the cloud (Thingspeak) using Nodemcu and it also displays on an OLED.

For powering up the devices, Buck converter is used to step down 230v AC to 5v DC

Step 1: Components/hardware & Tools Used

TLI4970 :

The TLI4970 is a high-precision current sensor based on Infineon's proven Hall technology. Its AC & DC measurement range up to $\pm 50A$ and SPI output of 16bit (13-bit current value). It's an easy-to-use, fully digital solution that does not require external calibration or additional parts such as A/D converters, 0 pAmps or reference voltage.

It's has a ready to use Arduino library.

Please find the datasheet of the TLI4970 variant here.

XMC2Go:

The XMC 2Go Kit with XMC1100 is maybe the world's smallest, fully-featured Microcontroller Evaluation Kit

showcasing - XMC1100 (ARM® Cortex™-M0 based) - On-board J-Link Lite Debugger (Realized with XMC4200 Microcontroller) - Power over USB (Micro USB) - ESD and reverse current protection - 2 x user LED - Pin Header 2x8 Pins suitable for Breadboard.

It can be programmed using the Arduino IDE. Link

The user manual can be found here.

NodeMCU:

Wi-Fi board for further info link

AC-DC Dual Output:

Steps down 220v Ac to 5v Dc. Link

Oled I2C display:

Link

Prototype Board :

Link

5 in 1 Extension Box:

Link

Electrical wires

Tools used-

Small flat-head screwdriver

Soldering iron, desoldering braid

Wire cutters

Dremal or similar tool

Step 2: Install Arduino and Make It Ready to Compile the Examples

- Install the Arduino IDE. Link
- Install the Infineon board package to compile the example code.
- Follow the installation step one by one. Link
- Install the board package for ESP8266.
- Follow the installation steps one by one. Link
- Install the extra libraries required to compile the example code-
- TLI4970
- OLED Screen

Step 3: Connection Diagram

The Connection is as follows:

- **XMC 2Go ---- > Tli4970**

Vss ----->GND

Vdd ----- > 3.3V

P0_6 ----- > MISO

P0_8 ----- > SCK

P0_9 ----- > CS

- **XMC 2Go ----- > Nodemcu**

Vss ----- > GND

Vdd ----- > 3.3

VP2_0 ----- > D6

- **Nodemcu ----> OLED**

GND ----- > GND

3.3V ----- > 3.3V

D1 ----- > SCK

D2 ----- > SDA

Step 4: Setting Up ThingSpeak to Visualize the Data

Create an account in ThingSpeak

Create a channel in the ThingSpeak account

Take the credentials of ThingSpeak Channel and Write API Key and update the details in the secret file present along with .ino file that is to be flashed in the NodeMCU.

Step 5: Final Steps

Flash the code given in rar file after replacing the pins_arduino given in the package.

Note: Copy pins_arduino.h and replace with the pins_arduino.h present in path C:\Users\....\AppData\Local\Arduino15\packages\Infineon\hardware\arm\1.4.0\variants\XMC1100\config\XMC1100_XMC2GO\pins_arduino.h

Note: From the buck convertor take 5V output and power up both XMC2Go and NodeMcu.

Step 6: Flow Diagram & Circuit Connection

Flash the code check the connections, energy meter is ready to calculate the power consumed by any appliance connected to the energy meter.

In this project board with fuse is being taken which increases the price of this maker project, this thing could also be done just by using one socket in which load could be plugged in. But if you are using a single socket without fuse be double protective while handling the AC power supply.



Smart City - Identify, select, install and troubleshoot different module/ devices used in smart street light based on IoT and cloud Technology

Integrate energy meter with serial protocol working DLMS protocol

Objective:

- to integrate energy meter with serial protocol working DLMS protocol.

Requirements:

- Computer System
- Energy meter
- EEPROM

PROCEDURE:

Introduction:

Today, DLMS is the most popular communication protocol used in metering arena across the world. It has been accepted by most energy utilities in the wake of liberalized energy market. More over several countries including India have framed metering specification based on DLMS which will act as a guideline for utilities to specify their metering requirements and ultimately improve consumer data collection, billing and revenue. In this wider scheme of things, one vital component is DLMS powered inter operable meters.

Electronic energy meters with communication interface are not a rarity in this age. All large scale meter manufacturers and most medium scale manufacturers have this already in place with the only exception being that the communication protocol is proprietary or loose standards. It is therefore deducted that meters already use an embedded microcontroller platform with peripherals such as UART, EEPROM. In this backdrop the migration from proprietary protocol to DLMS is mainly a matter of proper planning in using right hardware and software components.

Planning:

As per Indian companion specification there are three categories of meters:

1. Energy account and Audit meters
2. Boundary (ABT) meters
3. Consumer meters
 - 3.1. Single phase meters
 - 3.2. Three phase meters

Meter manufacturer shall first decide the categories in which they are going to develop DLMS meters and then prepare detailed list of parameters and functionality for each category.

FORM THE PARAMETER, AND FUNCTIONALITY LIST

Indian companion specification has well documented the

complete list of parameters for each meter category and it also explains the basic minimum functionalities to be supported. This could be treated as a guideline for generating parameter list and functionality list. While doing this exercise meter manufacturer should have a fundamental knowledge of DLMS data models (especially Interface class and OBIS), since Indian companion specification widely uses DLMS terminologies. If sufficient DLMS expertise is not available with meter manufacturer, it is advised to develop the expertise by undergoing training or seek consultancy service from external industry experts.

Challenges while finalizing parameter, functionality list

Meter manufacturers who already have non-DLMS meters with fair enough parameters and functionality may face the challenge of finding new parameters in the companion specification which is not part of his existing meter implementation. In this situation, the meter manufacturer shall analyze if the newly found parameters are entirely not supported in his existing meter or if they are already implemented, but in a different way.

- a. Parameter not supported- In this case meter manufacturer shall start implementing those parameters in the metering IC parallel to DLMS implementation efforts. An example for this could be "Logical device name".
- b. Parameter supported in a different way - In this case meter manufacturers shall tweak his existing implementation of one or more parameters to perfectly align with Indian companion specification. An example for this situation could be "Activity calendar object" specified under "TOU metering" section in Indian companion specification. Meter manufacturer might have already implemented TOU in his meter, but the time zone information and corresponding tariff IDs may be stored in a different way from the DLMS model. In this case meter manufacturer only needs to modify his existing TOU data storage to DLMS's Activity calendar format (with season profile, week profile and day profile). The end result of this exercise is to generate one/ more parameter list tables as shown in Table 1.

The table shall be comprehensive capturing the finest details regarding parameters:

Parameter Name	OBIS code	IC	Attribute ID
L1 voltage	1.0.32.7.0.25 5	3	2
L1 current	1.0.31.7.0.25 5	3	2
Frequency	1.0.14.7.0.25 5	3	2

Table 1-Parameter List

Functionality list

Meter manufacturer shall now decide the DLMS services and other protocol configurations to be used. The choices offered by DLMS are:

- Application context
- Security level
- Services
- Communication profile
- Application Context - DLMS offers two referencing schemes viz., Logical name and Short name. Indian companion specification has advocated the mandatory usage of Logical name.
- Security Level – DLMS offers three levels of sign on authentication viz., lowest level, low level and high level.

Lowest level has in fact no authentication. Low level involves client sending a plain text password to meter, based on which meter authenticates client and accordingly accept or reject Client's associate request.

High level involves both parties (Client and Server) exchanging challenges in first round and then exchanging processed challenges in second round. In this method both parties authenticates each other and also the risk of security being hacked is eliminated as password or processing key is not exchanged through communication media.

Indian companion specification requires all three authentication levels to be supported in meters; each level will be advocated by each type of client (Public client use lowest level, Meter reader use low level and Utility will use at High level).

Services - DLMS offers variety of application layer services which is used by client to access server's (meter) data. Services mandatory as per Indian companion specification are:

- Get: Service used to read normal (small) data from meter.
- Get with block: Service used to read large data from meter.
- Set: Service used to write normal (small) data into meter.
- Set with block: Service used to write large data into meter.
- Action: Service used to execute methods inside meter.
- Selective Access: Service used to selectively access meter data (example – selective reading of load profile based on date-time)

Communication profile – Decide on the communication interfaces for the meter. Mandatory choices as per Indian companion specification are:

- RS485 (allowing multidrop) – primarily for utility software to communicate. This could also be used for remote connection using modem.
- Optical – for meter reading instrument.

Other communication options possible with DLMS are:

- TCP IP (Ethernet/GPRS modem)
- PLC

These communication profiles may require their own stack (example – TCP IR stack in the meter's communication chip).

Smart City - Identify, select, install and troubleshoot different module/ devices used in smart street light based on IoT and cloud Technology

Communicate and verify the parameters on DLMS server software

Objective:

- to communicate and verify the parameters on DLMS server software.

Requirements:

- Energy meter
- Test tool
- Computer System

PROCEDURE:**Introduction:**

The Device Language Message Specification (DLMS) and Companion Specification for Energy Metering (COSEM) form together the DLMS/COSEM application layer communication protocol and an interface model for metering applications. The requirement of an Open Protocol for energy metering was necessary for bringing in homogeneity among various makes of meters.



The openness of the standard will make the Automatic Meter Reading (AMR) and interoperability a reality. One among the many open protocols is the DLMS/COSEM. This open protocol based on IEC 62056 helps to overcome most of the challenges for data acquisition from energy meters. The difficulties faced by the utilities to have successful remote metering deployments and

the consequent difficulty in accounting energy losses due to proprietary protocols are well known. The Indian Companion Specification (ICS) is intended to provide a field level basis for efficient and secure transfer of electricity metering data in an open manner with judicious application of features and protocols of the International Standard. The Indian Companion Specification (ICS) was evolved by BIS as an Indian Standard IS 15959 which inherits the IEC 62056 series of standards. This article presents the qualitative testing of DLMS/COSEM ICS compliant Energy meters.

The advent of electronic energy meters with data download capability made Metering Billing Collection (MBC) process to move away from human to a machine dependent. Further the deregulation raised the need for integrating meter data with the business processes of stakeholders.

The MBC process which begins with meter reading / downloading was the starting point of contention due to variety of protocols. With different makes of meters, the data collection for an automated process was a challenge. In order to bring in homogeneous environment and to optimise the business processes, structured data and also embracing all makes of meters was found to be an urgent necessity. This directed to the quest for open communication protocols so that AMR / AMI systems can become a reality. The need for open protocol was very much felt. One among the many open protocols is the DLMS/COSEM which has evolved over a period of time and is adopted by IEC as IEC 62056 series of standards and are adopted in India.

DLMS/COSEM is based on a strict client-server structure. The philosophy is to “model”, “message” and “transport” with an object oriented approach. The server is meant to be within the meter while the client accessing the meter could be a gateway or the central office. Other use cases where the server is within the gateway and the client is in the central office are also feasible. Before the actual metering information can be exchanged an association has to be build up, which is initiated by the client. The DLMS client can then access the interface object model inside the server. Once an association exists the DLMS server is also able to send notifications to the client without an explicit request.

DLMS/COSEM supports clock synchronization and transmission of measurement data and profiles. The DLMS/COSEM has been improved and the current version facilitates digital signature and firmware upload.

Data objects for firmware updates are already part of the Blue Book Ed. Both are relevant for smart meters.

DLMS/COSEM includes authentication and confidentiality services based on symmetric encryption. The continuous development of DLMS/COSEM open communications protocols for data retrieval, updation and reconfiguration of metering devices has enabled diverse operators to access safely and quickly data from metering equipment provided by diverse manufacturers.

Among the many open protocols like MODBUS, ANSI, DLMS, the DLMS/COSEM (IEC 62056) was chosen by India. MODBUS was more towards process instrumentation and ANSI the meter protocol used in North America.

The IEC 62056 is an open protocol exclusively evolved for Metering purpose. It operates at meter level. A wide range of data types can be read either selectively or in groups. The parameters are identified through unique codes. The protocol can work with many common communication medium. The protocol implementation can be verified for conformance with the dedicated test tool.

India has been aligning with IEC for standardization. In the present effort to bring in uniform protocol and based on the relative merits and demerits, IEC 62056 protocol has become the natural choice for metering purposes in India.

The Indian Companion Specification (ICS) was formulated and released by Bureau of Indian Standards (BIS) as IS 15959 : 2011 "Data exchange for electricity meter reading, tariff and load control – Companion specification". This standard supplement the IEC 62056 series of standards. This companion specification spelt out the Indian specific requirements for designing open energy meters based on DLMS/COSEM. With this standardization a homogeneous environment would prevail at metering end for data collection.

The Companion Specification refers to latest updates of Interface classes and OBIS codes available in 12th edition (2014) of the Blue Book of DLMS UA. This also refers to updates of procedures and services available in 8th edition (2014) of the DLMS UA Green Book. These revised Technical Reports of DLMS UA, considered as pre-standards for several parts of IEC 62056, will be adopted by IEC in due course and subsequently by BIS.

Need for Protocol Testing

Testing of energy meters for communication protocol unlike any other equipment is very much essential to make sure that the meter is complying with the said protocol standards. This is to ensure interoperability and easy acquisition of data from the meters using any third party client tool which makes the device operate vendor

neutral. This enables every meter reading system to understand a value by its identification system, irrespective of the meter make. The DLMS/COSEM also allows for uploading the Objects supported by a meter, and then reading the meter values, which enable any DLMS/COSEM Compliant Client software to read any manufacturer meter.

Compliance Testing as per IS 15959: 2011 Indian Companion Specification

The ICS standard is intended for use as companion to IEC 62056 series of standards on "Electricity Metering – Data Exchange for Meter Reading, Tariff and Load Control", adopted from various parts of IEC 62056. The metering device is evaluated in order to gain confidence about implementation of declared features of ICS and generic standards IEC 62056.

The compliance testing evaluation is carried out to verify whether the DLMS protocol are implemented correctly in the metering device. The test setup is as shown in Fig. 1. Meters claiming compliance to this Companion Specification will be required to:

- Conform to DLMS/COSEM (IEC 62056) protocol requirements as certified by the Conformance Test Tool (CTT)
- Parameter verification as a part of this:
- All mandatory parameters applicable to the category of the meter under test are implemented.
- All data types where specified are conforming to ICS.
- All Application Associations are implemented as specified in ICS with all specified services supported.
- Association object lists conform to ICS with access rights and OBIS codes as specified here and
- Event related DLMS objects are implemented with Event identifiers as specified in the Event reference tables in ICS.

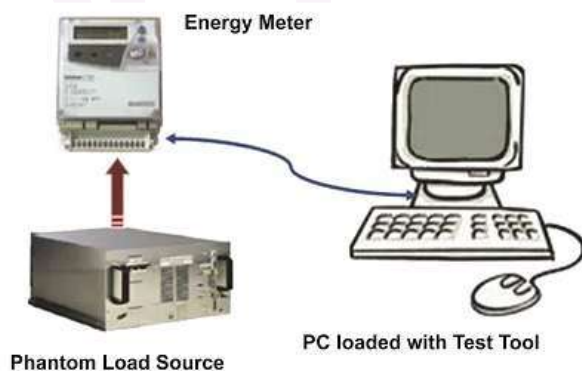


Fig 1: Compliance Test Setup

Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Setup wired local area network and wireless network

Objective:

- to Setup wired local area network and wireless network.

Requirements:

- Computer System
- UTP cable
- Switch
- USB cable

PROCEDURE:

Today almost every home and small office has a local network, and an Internet connection.

The home network or small office network enables multiple devices e.g. PCs, tablets etc to connect to each other, and also to connect to the internet.



Wired networks use Ethernet over UTP cable and tend to be faster than wireless networks, which is an important consideration if you are a gamer. The simple wired home network diagram below shows a minimum setup with a switch and broadband router.

Wired Network Advantages

- Fast typically 100 Mbps to 10 Gbps
- Secure and reliable.

Wired Network Dis-Advantages

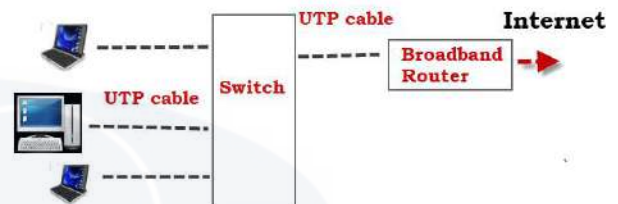
- Doesn't work with devices that don't have an Ethernet port e.g. tablets and smart phones.
- Not so easy and fast to setup as it requires running cables. However Homeplug or powerline adapters can be used instead.
- Not so easy for visitors and mobile devices (laptops) to connect to.

Main Uses

Used for network backbone i.e. connecting between router, network switches and wireless access points on different levels (floors).

Home Wireless Networks

Wireless networks use Wi-Fi. and are quick and easy to install, but are generally slower than wired networks. See Wi-Fi and Wireless networks for more details.



Simple Wired Home Network Diagram

Wireless Network Advantages

- Easy to setup from an end user perspective.
- Allows easy access to smart phones, Tablets and mobile devices.
- No cables to run.

Wireless Network Dis-Advantages

- Not as Secure as wired networks without proper configuration. and easy to setup insecurely.
- Not so as fast as wired networks.
- Not as reliable as wired networks.

Main Uses

- Connecting peripheral devices like computers, smart phone, tablets etc.
- Most home networks will use a mixture of wired and wireless.

Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Setup environment for Modbus TCPIP server client testing

Objective:

- to setup environment for Modbus TCPIP server client testing.

PROCEDURE:**MODBUS MASTER/SLAVE Serial and Ethernet Communication Server**

The MODBUS MASTER/SLAVE Serial and Ethernet Communication Server (hereafter referred to as the "MODBUS MASTER/SLAVE Server" or "MODBUS Server" or "MODBUS" or "Server") is a Microsoft Windows 32-bit application program that acts as a communication protocol Server and allows other Windows application programs access to data from MODBUS network using the RS-232 serial and/or TCP/IP interface. The Server can act either as a MODBUS master device or a MODBUS slave device. Any Microsoft Windows program that is capable of acting as a DDE, FastDDE, SuiteLink or OPC Client may use the MODBUS MASTER/SLAVE Server.

There are two different MODBUS Server versions described in this manual:

- Server version (ordering number DR 380 10), supporting SuiteLink, FastDDE and DDE protocols; this version hereafter is referred to as the "Suite Link & DDE" version.
- Server version (ordering number DR 380 11), supporting OPC and DDE protocols; this version hereafter is referred to as the "OPC & DDE" version;

The separate installation package is supplied for each version of the Server. In all cases the name of Server executable file is MODBUS.EXE. All further information in this manual is same for all versions of the Server, with the exception of few points where communication protocol specific features are explained.

Communication Protocols

Dynamic Data Exchange (DDE) is a communication protocol developed by Microsoft to allow applications in the Windows environment to send/receive data and instructions to/from each other. It implements a client-server relationship between two concurrently running applications. The server application provides the data and accepts requests from any other application interested in its data. Requesting applications are called clients. Some applications such as Wonderware InTouch and Microsoft Excel can simultaneously be both a client and a server.

FastDDE provides a means of packing many proprietary

Wonderware DDE messages into a single Microsoft DDE message. This packing improves efficiency and performance by reducing the total number of DDE transactions required between a client and a server. Although Wonderware's FastDDE has extended the usefulness of DDE for the industry, this extension is being pushed to its performance constraints in distributed environments. The MODBUS MASTER/SLAVE Server "Suite Link & DDE version" supports the FastDDE Version 3 - an extension to Wonderware's proprietary FastDDE Version 2. This extension supports the transfer of Value Time Quality (VTQ) information. The original DDE and FastDDE Version 2 formats are still supported, providing full backward compatibility with older DDE clients. FastDDE Version 3 works on Windows 9x systems as well as Windows NT systems.

NetDDE extends the standard Windows DDE functionality to include communication over local area networks and through serial ports. Network extensions are available to allow DDE links between applications running on different computers connected via networks or modems. For example, NetDDE supports DDE between applications running on IBM compatible computers connected via LAN or modem and DDE-aware applications running on non-PC based platforms under operating environments such as VMS and UNIX.

SuiteLink uses a TCP/IP based protocol and is designed by Wonderware specifically to meet industrial needs such as data integrity, high-throughput, and easier diagnostics. This protocol standard is only supported on Microsoft Windows NT 4.0 or higher. SuiteLink is not a replacement for DDE, FastDDE, or NetDDE. The protocol used between a client and a server depends on your network connections and configurations. SuiteLink was designed to be the industrial data network distribution standard and provides the following features:

Value Time Quality (VTQ) places a time stamp and quality indicator on all data values delivered to VTQ-aware clients. · Extensive diagnostics of the data throughput, server loading, computer resource consumption, and network transport are made accessible through the Microsoft Windows NT operating system Performance Monitor. This feature is critical for the scheme and maintenance of distributed industrial networks. · Consistent high data volumes can be maintained between applications regardless if the applications are on a single node or distributed over a large node count.

The network transport protocol is TCP/IP using Microsoft's standard WinSock interface.

OPC (OLE for Process Control) is an open interface standard to provide data from a data source and communicate the data to any client application in a common standard way. The OPC is based on Microsoft OLE, COM and DCOM technologies and enables simple and standardized data interchange between the industrial or office sector and the production sector. From general point of view many aspects of OPC are similar to DDE, but main difference is in the implementation by using Microsoft's COM (Component Object Model) technology. It enables fast exchange with process automation data and OPC open interface allows access to data from OPC Server in same standard way from OPC client applications supplied by different developers.

For more information on the basics of OPC, please refer to the OPC Specification. The OPC Data Access Custom Interface Specification is maintained by OPC Foundation, the current specification is 2.05a dated June 2002 (3.00 dated March 2003).

The OPC support for MODBUS MASTER/SLAVE Server "OPC & DDE" version is implemented based on FactorySoft OPC Server Development Toolkit and it conforms to OPC Data Access Custom Interface Specification 2.04. The MODBUS MASTER/SLAVE Server "OPC & DDE" version is tested for compliance and is compatible with OPC Foundation OPC Data Access Compliance Test Tool.

The Suite Link, FastDDE (Version 3) and DDE support for MODBUS MASTER/SLAVE Communication Server is implemented by Wonderware I/O Server Toolkit ver. 7,2,1,6.

The FastDDE (Version 2) and DDE support for MODBUS Server "OPC & DDE" version is implemented by Wonderware I/O Server Toolkit ver. 5.0 (008).

Accessing Remote Items via the Server

The communication protocol addresses an element of data in a conversation that uses a three-part naming convention that includes the application name, topic name and item name. The following briefly describes each portion of this naming convention:

Application name

The name of the Windows program (server) that will be accessing the data element. In the case of data coming from or going to a MODBUS node via this Server, the application portion of the address is MODBUS.

Topic name

Meaningful name of communication entity, that is referred also as "Logical device". Commonly the name considers entire MODBUS node but in more intricate applications separate data and control item groups on the same node can be accessed by different topics. For example, NODE5 is an ordinary topic name for the fifth node on the MODBUS bus.

Note: You can define multiple topic names for the same device (node) to poll different items at different rates.

Item name

A specific data element within the specified topic. For example, when using this Server, items can be individual operands (registers, binary inputs, binary outputs) in the MODBUS node. The term "point" is used interchangeably with the term "item" in this User Manual.

DRAFT

Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Communicate and configure Modbus devices through GSM and GPRS network

Objective:

- to communicate and configure Modbus through GSM and GPRS network.

Requirements:

- Computer system
- GSM
- GPRS
- Modbus device

PROCEDURE:

Introduction:

There are 7 example files in the iMod telemetric module, which help with a proper configuration of the device and help to understand the basic, necessary rules of creating configuration files:

- access to hardware resources,
- modbus proxy,
- SMS, e-mail event types,
- modbus events,
- script events,
- counters,
- SNMP.

Each protocol describes one of the basic functionalities available on the NPE/iMod device. The following example describes the simplest functionality - access to hardware resources. By using and analyzing the other examples, you can easily configure the NPE/iMod industrial computer as an alarm module (notifier), data logger or a fully functional PLC.

Access to hardware resources

Channel definition

The first example is built on the basis of the simplest configuration - it enables easy access to hardware resources of the device, readout of the status and control via the Modbus TCP protocol. In this example the digital input DI1 and the USER LED is used.

There is a video tutorial below, describing the use of the example configuration and its verification in order to check if it works properly. Additionally at the end of the article there is a direct link to the configuration file.



The iMod configuration file consists of channel and parameter groups. The channel groups are divided into three main channels:

- source channel - in this case it is a hardware resources access channel,
- data access channel - by using this channel you can read the parameter values located in the pre-defined data source channel,
- communication channel



Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Setup Serial to Ethernet protocol converter and verify

Objective:

- to setup serial to Ethernet protocol converter and verify.

Requirement:

- Ethernet converter
- Computer system
- Internet connectivity

PROCEDURE:**What is a serial to Ethernet converter and how does it work?**

A serial to Ethernet converter is a small electronic device which can convert Ethernet IP/TCP packets to either RS232, RS485 or RS422 serial data signals and vice versa. It is also sometimes called a serial device server and it comes in several different shapes and sizes; here are a few examples of what it can look like:



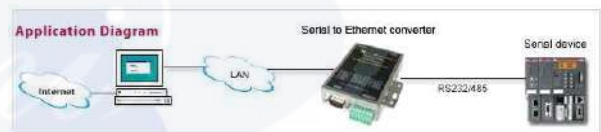
As you can see from above images a serial Ethernet converter can have one serial port or it can have multiple serial ports, up to 25 ports or more. The standard panel mount serial Ethernet converter is most common but it is also available with DIN mounts so it can easily be installed on a standard DIN rail. It is also available as a

wireless converter in which case the IP/TCP packets are transmitted over a IEEE 802.11a/b/g network link.

Some serial Ethernet converters only has a RS232 interface and some has a RS232 and a RS485 interface, yet others has all 3 interfaces (RS232, RS485 and RS422) built in.

So what is a serial Ethernet converter used for?

Most commonly it is used for connecting a serial RS232, RS485 or RS422 device such as a serial printer, barcode scanner, scale, GPS, sensor or any other consumer or industrial device with a serial interface, to a computer over a standard LAN network. The advantage of this is obvious; you will be able to control, monitor and communicate with your serial device remotely from a central computer. No need to walk all the way down to the other end of the factory to check your serial device.

**How does a serial Ethernet converter work?**

Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Setup serial to Wi-Fi protocol converter and verify

Objective:

- to Setup serial to Wi-Fi protocol converter and verify.

Requirement:

- Computer system
- Wi-Fi
- Serial device

PROCEDURE:

Hardware debugging and programming via UART/Serial connection has always been something so vital to hardware developers and even other team players like QA engineers. In certain circumstances, the device that needs to be debugged or programmed is not reachable by wires. I remember how QA engineers in my previous job had to come up with a Bluetooth adapter to bridge the serial debugging information of the tested products because it was fixed inside the car dashboard and can't be reached out. In this article, it's time to make some serial data fly over the air!

Serial Bridge Using ESP8266

One of the most well-known (appeared on Hackaday's blog) and well-designed projects to make ESP8266 as a WiFi-UART bridge is jeelabs's esp-link. Actually, this project is far beyond being a simple serial bridge as it also manages MQTT client pub/sub and REST HTTP requests in order to connect the MCU to the internet. Moreover, it can be used to flash the attached MCU. Esp-link has a very handy web interface stored inside the ESP.

To make the test, we will use ESP8266 development board, specifically NodeMCU.

Prepare The Software

First, we need to download one of the release images of the project from releases page. The compressed file (A .tgz file is beside the download word in bold) should have

5 bin files :blank.bin, boot_v1.x.bin, esp_init_data_default.bin, user1.bin and user2.bin. The ESP8266 board should be connect after downloading one of the available download tools like: Flash download tool from Espressif or Nodemcu flasher.

Now, follow the flashing guidelines available in Readme file according to your ESP8266 edition and set the bin files addresses according to your module memory size.

You can simply make the Flash download tool from Espressif find it for you by connecting it and pressing "start" to get the required specification from the boot message.

Alternatively, you can read the boot message using a serial console with 76600 baud.

After flashing the image files successfully, you should see a new WiFi network.

Prepare The Hardware

To test the serial bridge code, we will achieve the following connection: TX0 from ESP with RX of the device and RX0 with TX of the device that sends serial information and needed to be bridged via WiFi. For the sake of the test, the device here would be a USB-TTL cable (USB-UART converter) and the data will be sent and received from a console (PUTTY for instance).

Image Courtesy of ElecFreaks

Now, let's see some serial data fly over the air!

Finally, I think a considered amount of time is needed to explore the other options of this amazing project!

Serial Bridge Using ESP8266 (Simpler)

The previous project could be too complicated to do the simple job of converting Serial connection to a WiFi connection.

Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Set up serial to GPRS protocol converter and verify

Objective:

- to set up serial to GPRS protocol converter and verify.

Requirements:

- Computer system
- GPRS protocol
- Data server
- External devices

Programmable automation controller (PAC) with GPRS modem*



- Designed for the needs of automation, telecommunications, remote supervision, and monitoring
- Full range of communications interfaces, including built-in GPRS communication*
- Service of standard protocols (e.g. MODBUS, SNMP), possibility of installation of dedicated user protocols
- Exceptional capabilities of data logging thanks to the built-in 1GB FLASH memory and servicing of SD cards
- Capabilities of presentation of web pages (Scada web) directly from the device for the purpose of visualization of current and archived data as well as remote control
- Ready software for creation of PLC ladder algorithms

Basic information

- Ready-to-use autonomous platform, tried and tested in installations in many countries of the world

Hardware properties

- ▶ Adapted for constant operation under industrial conditions
- ▶ Efficient, energy-saving processor using RISC technology
- ▶ Large RAM memory for start-up of many independent services
- ▶ Rich set of I/O interfaces: including digital and analog inputs and outputs, RS232/RS485 serial ports, transmitter outputs
- ▶ Built-in Ethernet (LAN) and GPRS (WAN)* communication
- ▶ Servicing of the economic 1-Wire bus typically used for reading of temperature sensors*
- ▶ Possible hardware versions on special order (e.g. USB port), also with a log specified by the orderer (OEM versions)

Software properties

- ▶ New firmware based on Linux Kernel 2.6.28 ensuring stability and security of device operation
- ▶ Possibility of operation with expansion modules for the purpose of increasing the amount of available interfaces (see accessories section)
- ▶ Ready tools and pre-compiled packs, service of VPN, SSH, SQL, PHP, JAVA, C/C++
- ▶ Developer tools and support of their use, instructions, informational materials
- ▶ Remote software updates
- ▶ Possible upgrade to the innovative iMod platform and NX Dynamics web modules
- ▶ Full technical support through a dedicated portal, super-standard design cooperation thanks to TechBase Solution Partner

Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Set up Ethernet IoT Data Acquisition system, connect to cloud and verify

Objective:

- to setup Ethernet IoT Data Acquisition system, connect to cloud and verify.

Requirement:

- Computer system
- Ethernet
- Cloud
- Internet connectivity

PROCEDURE:

Introduction:

Industrial IoT requires a robust infrastructure comprised of sensor nodes, energy-efficient communication networks, and gateways that connect to the Internet and cloud. The datasets generated by the sensors pass through many phases before becoming valuable for businesses. This article introduces the concept of sensor nodes, low-power networks, and IoT Gateways used in industrial scenarios. We will use Arduino, XBee, Raspberry Pi 2 and open source software for prototyping an end-to-end solution.

The latest Raspberry Pi 2 packs a lot more punch than its predecessor. Powered by a 900 MHz quad-core ARM Cortex-A7 processor and 1GB RAM, Raspberry Pi 2 Model B can be used for a variety of scenarios. We will use an open source software stack to aggregate, store and visualize sensor data in real-time.

What is an IoT Gateway?

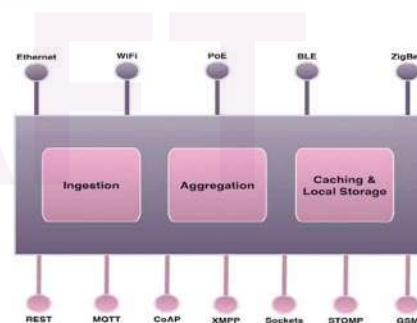
In an industrial IoT scenario, there are many sensors and actuators that interact with the machinery. Each machine would typically have multiple sensors tracking its health and monitoring the key parameters related to the production. Each sensor and actuator is attached to a microcontroller that is responsible for acquiring the data or controlling a switch through a pre-defined instruction set. The microcontroller — along with the sensors, power and a radio — is called a sensor node. It is a self-contained, deployable unit that captures the data generated by sensors. The sensor node doesn't have enough processing power, memory, and storage to deal with the data locally. It uses a low-energy radio communication network to send the data to a central location. The communication link between the sensors' nodes and the central hub is based on ZigBee, Bluetooth Low Energy (BLE), or Power over Ethernet (PoE). The hub that acts as an aggregator of multiple raw datasets generated by the sensor nodes is called an IoT gateway.

An IoT gateway has multiple roles to play. One of the

first jobs of the gateway is to transform and normalize the data. The datasets generated by the sensor nodes will be in disparate formats. Some of the legacy nodes use proprietary protocols, while the contemporary ones may rely on JSON or CSV. The gateway acquires heterogeneous datasets from multiple sensor nodes and converts them to a standard format that is understood by the next stage of the data processing pipeline.

The second role of an IoT gateway is protocol transformation. Since the sensor nodes cannot use power hungry Wi-Fi or ethernet, they use low-powered communication networks. A gateway supports multiple communication protocols for accepting the inbound data sent by the sensor nodes. It uses a variety of protocols for the outbound communication, which typically connects the gateway to a process running in the cloud. Some of the popular outbound protocols used in the context of IoT are REST, MQTT, CoAP, STOMP and even SMS. In some scenarios, the gateway may also process the data and raise alerts in real time. But this is best left to the powerful stream processing pipelines running in the cloud.

The diagram below represents the deployment architecture of an IoT gateway.



Gateways act as an edge device, obscuring the sensor nodes from the public internet. Though the sensor nodes can make outbound connections to the internet and cloud through the gateway, they cannot be accessed directly. Thus, gateways play the dual role of routers and firewall securing the sensor nodes and internal network.

Sensor nodes that are capable of connecting to the Internet still need a gateway for data aggregation and transformation. They connect to an appliance running in the cloud called a cloud gateway. The local edge device running on-premises is often referred to as a field gateway.

Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Set up Wi-Fi IoT Data Acquisition system, connect to cloud and verify

Objective:

- to Set up Wi-Fi IoT Data Acquisition system, connect to cloud and verify.

Requirement:

System

IoT Data Acquisition system

Cloud

Internet

PROCEDURE:

Introduction:

Internet of Things (IoT) is an emerging domain that promises ubiquitous connection to the Internet, turning common objects into connected devices. The IoT paradigm is changing the way people interact with things around them. It paves the way to creating pervasively connected infrastructures to support innovative services and promises better flexibility and efficiency. Such advantages are attractive not only for consumer applications, but also for the industrial domain. Over the last few years, we have been witnessing the IoT paradigm making its way into the industry marketplace with purposely designed solutions. This paper represents the implementation & result of Liquid or solid objects level monitoring and data acquisition using industrial level sensor and has been displayed on dashboard as well as from remote location using wireless network.

Data acquisition is method of gathering and organizing the information. In this, using analog to digital converter sampled signals are converted into digital form and these signals can be analysed, displayed and stored in computer. It is also process of measuring the physical or electrical terms like voltage, current, pressure, temperature, sound etc. with the help of computer. In this project, by using industrial level sensor which is work on radar principle, we calculate the level of liquid or solid objects and percentage of reservoir filled and distance of medium from sensor device, these distance can be calculated with the help of electromagnetic waves Emitted by sensor device, as a result of this we get an exact level of reservoir filled.

Following are the components used in this project:

Level Sensor

This is basically radar sensor which uses the principle of reflection, but it is slightly different from normal radar sensor used in industry. This is the first 80GHz radar developed according to international safety standard. It is based on frequency modulation continuous wave

method. It measures the level by using frequency difference i.e. difference between emitting wave and reflected wave frequencies.

This type of sensors are mainly used in non-contact level measurement of liquids, pastes and slurries and because of its higher frequency it comes under W band. Therefore it is perfect technique for high frequency radar. It not get affected by medium changes, temperature variation, gas blankets and vapors.

This sensor works with the HART communication. Because of industrial sensor, it gives the output in the form of current i.e. in between 4 to 20mA. These data can be displayed on computer by using HART communication device. From that we can see the distance, level and temperature of the device.

I to V Convertor(4-20mA to 1-5V)

The main advantage of a two-wire loop is that it minimizes the number of wires needed to run both power and signal. The use of a current loop to send the signal also has the advantages of reduced sensitivity to electrical noise and to loading effects. The electrical noise is reduced because the two wires are run as a twisted pair, ensuring that each of the two wires receives the same vector of energy from noise sources, such as electro-magnetic fields due to a changing current in a nearby conductor. The circuit compliance to handle a given voltage drop from additional loop devices depends on the transmitter output circuit and on the power supply voltage. The typical power supply for industrial transmitters is +24 VDC. Where the current loop signal is connected to the main receiving equipment or data acquisition system, a precision load resistor of 250 ohms is normally connected. This converts the 4 to 20 mA current signal into a 1 to 5 volt signal, since it is standard practice to configure the analog-to-digital converter of the receiving equipment as a voltage-sensing input.

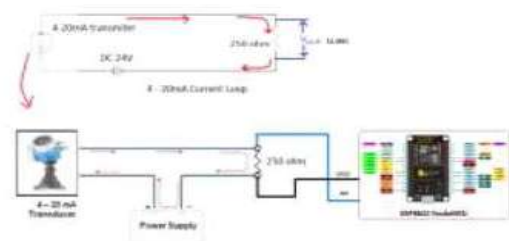


Fig -1: I to V conversion

Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Explore IoT Cloud Configuration Utility

Objective:

- to explore IoT Cloud Configuration Utility.

PROCEDURE:**What you do**

In this article, you do the following tasks:

- Use Cloud Explorer for Visual Studio to monitor device-to-cloud messages.
- Use Cloud Explorer for Visual Studio to send cloud-to-device messages.

What you need

You need the following prerequisites:

- An active Azure subscription.
- An Azure IoT Hub under your subscription.
- Microsoft Visual Studio 2017 Update 9 or later. This article uses Visual Studio 2019.
- The Cloud Explorer component from Visual Studio Installer, which is selected by default with Azure Workload.

Update Cloud Explorer to latest version

The Cloud Explorer component from Visual Studio Installer for Visual Studio 2017 only supports monitoring device-to-cloud and cloud-to-device messages. To use Visual Studio 2017, download and install the latest Cloud Explorer.

Sign in to access your hub

To access your hub, follow these steps:

- In Visual Studio, select View > Cloud Explorer to open Cloud Explorer.

- Select the Account Management icon to show your subscriptions.
- If you are signed in to Azure, your accounts appear. To sign into Azure for the first time, choose Add an account.
- Select the Azure subscriptions you want to use and choose Apply.
- Expand your subscription, then expand IoT Hubs. Under each hub, you can see your devices for that hub.

Monitor device-to-cloud messages

To monitor messages that are sent from your device to your IoT Hub, follow these steps:

- Right-click your IoT Hub or device and select Start Monitoring D2C Message.
- The monitored messages appear under Output.
- To stop monitoring, right-click on any IoT Hub or device and select Stop Monitoring D2C Message.

Send cloud-to-device messages

To send a message from your IoT Hub to your device, follow these steps:

- Right-click your device and select Send C2D Message.
- Enter the message in input box.

Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Create / modify organization, connect devices over cloud

Objective:

- to Create / modify organization, connect devices over cloud.

Requirement:

- Computer system
- Cloud
- IoT devices

PROCEDURE:**Creating registries and devices**

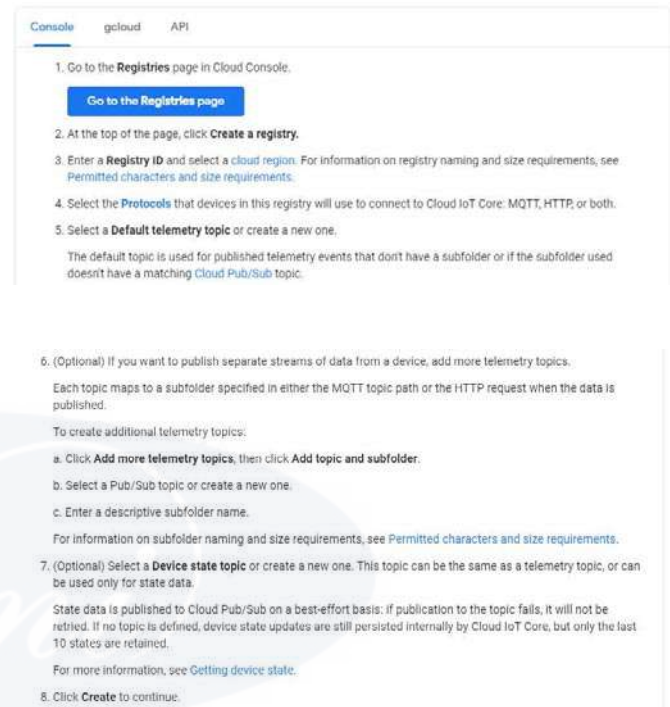
Here we will see how to create, edit, and delete device registries and devices within them.

A device is a "Thing" in "Internet of Things": a processing unit that is capable of connecting to the internet (directly or indirectly) and exchanging data with the cloud. A device registry is a container of devices with shared properties. For more details about devices and registries, see Devices.

If you haven't already, complete the Getting Started steps before proceeding.

Creating a device registry

To use Cloud IoT Core, you must create at least one device registry. You can create a registry using Cloud Console, the API, or gcloud.



The screenshot shows the Google Cloud Console interface with tabs for Console, gcloud, and API. The main content area displays a list of steps for creating a device registry:

1. Go to the **Registries** page in Cloud Console.
[Go to the Registries page](#)
2. At the top of the page, click **Create a registry**.
3. Enter a **Registry ID** and select a **cloud region**. For information on registry naming and size requirements, see [Permitted characters and size requirements](#).
4. Select the **Protocols** that devices in this registry will use to connect to Cloud IoT Core: MQTT, HTTP, or both.
5. Select a **Default telemetry topic** or create a new one.
The default topic is used for published telemetry events that don't have a subfolder or if the subfolder used doesn't have a matching **Cloud Pub/Sub** topic.
6. (Optional) If you want to publish separate streams of data from a device, add more telemetry topics.
Each topic maps to a subfolder specified in either the MQTT topic path or the HTTP request when the data is published.
To create additional telemetry topics:
 - a. Click **Add more telemetry topics**, then click **Add topic and subfolder**.
 - b. Select a Pub/Sub topic or create a new one.
 - c. Enter a descriptive subfolder name.For information on subfolder naming and size requirements, see [Permitted characters and size requirements](#).
7. (Optional) Select a **Device state topic** or create a new one. This topic can be the same as a telemetry topic, or can be used only for state data.
State data is published to Cloud Pub/Sub on a best-effort basis: if publication to the topic fails, it will not be retried. If no topic is defined, device state updates are still persisted internally by Cloud IoT Core, but only the last 10 states are retained.
For more information, see [Getting device state](#).
8. Click **Create** to continue.

Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Configuration of parameters, alarms, notification on cloud platforms

Objective:

- to configure the parameters, alarms, notification on cloud platforms.

Requirement:

- Cloud
- Computer system
- IoT devices

PROCEDURE:

Once you have a grasp of AWS CloudWatch terms and the basic flow of the tool, you're ready to create a new metric. To create a new CloudWatch metric, all you need to do is "put" data into it. There's no separate command for creating metrics. To begin working with CloudWatch, input metrics from the command line and view them using the AWS Management Console.

The following line uses the AWS unified command line to create a metric and give it an initial value. AWS used to have separate command-line tools for each product, but this was confusing and required admins to repeat the installation and configuration process. Recently, however, the company introduced a single unified command line that supports most products.

```
aws cloudwatch put-metric-data --metric-name myMetric
--value 40 --timestamp 2001-12-14 12:30
```

Figure 1 shows a script for monitoring a Cassandra cluster. The first four lines determine the number of servers in the "UN" (up/normal) state and set up a timestamp variable. The last line puts the value to CloudWatch.

Note: The first time you create a metric in AWS CloudWatch, it can take up to 15 minutes.

```
/opt/cassandra/nodetool -h localhost status > cassok
theCount=$(grep -w UN cassok | wc -l)
numServers=$((expr ${theCount} / 1))
DATE=$(TZ='Z' date +"%Y-%m-%dT%H:%M:%SZ")
aws cloudwatch put-metric-data --namespace Cassandra --metric-name ServerCount \
--value ${numServers} --timestamp $DATE
```

Figure 1. Monitor a Cassandra cluster in AWS CloudWatch.

To create a Topic, select SNS from the services list in the AWS Console and then pick "Create and Add."



Figure 2. Create a topic from the AWS Management Console.

Give the new topic a Topic Name and DisplayName. If you decide to send alarms as text messages, the DisplayName you create will be in each text message.

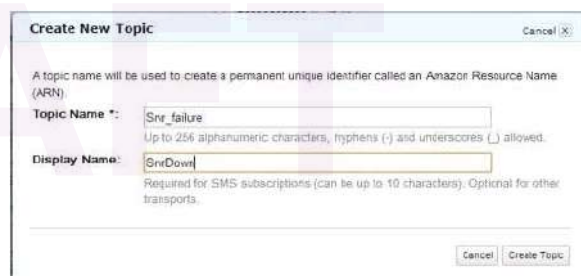


Figure 3. Designate a topic name in CloudWatch.

You now have a topic and an Amazon Resource Name (ARN). For now, choose Create Subscription. A subscription is an email address or cell phone to receive messages from the topic. Enter your email or cell phone number and press Subscribe. You'll get a confirmation mail/text confirmation.



Figure 4. Create a CloudWatch subscription.

Once you have a metric, a topic and a subscription, you're ready to link them to an alarm using Amazon Management Console and the command line.

From the console, select the CloudWatch service, then Alarms and then Create Alarm. Next, select the metric you created earlier and press Next. This brings up the Create Alarm dialog.

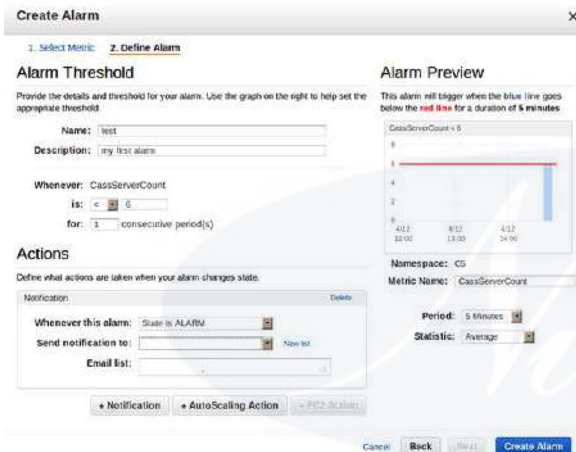


Figure 5. Set and create alarms.

Name the alarm and describe and define when it should alert an admin. In this example, we create an alarm that fires if the metric's value is less than six.

Next, select your topic in the Send notification to box and press Create alarm.

To do this using the command line, enter the following command:

```
aws cloudwatch put-metric-alarm --alarm-name myAlarm
--alarm-description "my new alarm" --metric-name
myMetric --namespace myNameSpace --statistic Average
--period 300 --threshold 6 --comparison-operator
LessThanThreshold --evaluation-periods 1 --alarm-
actions <that ARN you got when you created the topic>
```

Although this command has a lot of parameters, each part makes sense.

Essentially, the command says: Create an alarm with a name and namespace that fires when the average value of the metric is less than six for a single 300-second (five minute) time period, and notify my topic if it does.

Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Explore user management roles and security. Observer devise diagnostics for troubleshooting

Objective:

- to explore user management roles and security
- observe devise diagnostics for troubleshooting.

PROCEDURE:

To create a role definition

- In Object Explorer, expand a report server node.
- Expand the Security folder.
- If you're creating an item-level role definition, right-click Roles > New Role.

Or, if you're creating a system-level role definition, right-click System Roles > New System Role.

- Type a unique name for the role. A name must contain at least one character. It can also include spaces and certain symbols, but not the following characters [; : \ / @ & = + , \$ / * < > | "].
- Optionally type a description. In Management Studio, this description is visible only on this page. Users who view this item through the web portal can see this description in that tool.
- Select the tasks that members of this role can do.
- Select Ok.

To delete or modify a role definition

- In Object Explorer, expand a report server node.
- Expand the Security folder.
- To delete or modify an item-level role definition, expand the Roles folder, then do one of the following actions:
 - To delete a role definition, right-click the item > Delete. The Delete Catalog Items dialog box is displayed. Select OK to delete the role.
 - To modify a role definition, right-click the item > Properties. The General page of the User Role Properties dialog box is displayed.
- Select the tasks that members of this role can do, and then select OK.
- To delete or modify a system-level role definition, expand the System Roles folder. Do one of the following actions:
 - To delete a system role definition, right-click the item

and select Delete. The Delete Catalog Items dialog box is displayed. Select OK to delete the role.

- To modify a system role definition, right-click the item and select Properties. The General page of the System Role Properties dialog box is displayed. Select the tasks that members of this role can do, and select OK to apply the changes.

How to use predefined roles

- Review the predefined roles to determine whether you can use them as is. If you need to adjust the tasks or define additional roles, you should do this before you begin assigning users to specific roles. To create or edit custom roles use SQL Server Management Studio. For more information, see Create, Delete, or Modify a Role (Management Studio).
- Identify which users and groups require access to the report server, and at what level. Most users should be assigned to the Browser role or the Report Builder role. A smaller number of users should be assigned to the Publisher role. Very few users should be assigned to Content Manager.
- When you are ready to assign user and group accounts to specific roles, use the web portal. For more information, see Grant User Access to a Report Server.

Predefined role definitions

Predefined roles are defined by the tasks that it supports. You can modify these roles or replace them with custom roles.

Scope defines the boundaries within which roles are used. Item-level roles provide varying levels of access to report server items and operations that affect those items. Item-level roles are defined on the root node (Home) and all items throughout the report server folder hierarchy. System-level roles authorize access at the site level. Item and system-level roles are mutually exclusive but are used together to provide comprehensive permissions to report server content and operations.

The following table describes the predefined scope of the roles:

TABLE 1

Predefined role	Scope	Description
Content Manager Role	Item	May manage content in the Report Server. This includes folders, reports, and resources.
Publisher Role	Item	May publish reports and linked reports to the Report Server.
Browser Role	Item	May view folders, reports, and subscribe to reports.
Report Builder Role	Item	May view report definitions.
My Reports Role	Item	May publish reports and linked reports; manage folders, reports, and resources in a users My Reports folder.
System Administrator Role	System	View and modify system role assignments, system role definitions, system properties, and shared schedules, in addition to create role definitions, and manage jobs in Management Studio.
System User Role	System	View system properties, shared schedules, and allow use of Report Builder or other clients that execute report definitions.

Content manager role

The Content Manager role is a predefined role that includes tasks that are useful for a user who manages reports and Web content, but doesn't necessarily author reports or manage a Web server or SQL Server instance. A content manager deploys reports, manages report models and data source connections, and makes decisions about how reports are used. All item-level tasks are selected by default for the Content Manager role definition.

The Content Manager role is often used with the System Administrator role. Together, the two role definitions provide a complete set of tasks for users who require full access to all items on a report server. Although the Content Manager role provides full access to reports, report models, folders, and other items within the folder hierarchy, it doesn't provide access to site-level items or operations. Tasks such as creating and managing shared schedules, setting server properties, and managing role definitions are system-level tasks that are included in the System Administrator role. For this reason, we recommend that you create a second role assignment at the site level that provides access to shared schedules.

A role assignment maps a group or user account to a predefined role that defines the tasks that can be done. It determines the types of tasks that a user does to a folder, report, model, or other content type. To create, modify, or delete role assignments, you use the SSRS web portal. After you create a role assignment for a particular user or group, you can modify it later by selecting a different role. If you want to revoke permissions to a report server, you can delete a role assignment from the report server.

Depending on your objective, alternative approaches might be more appropriate. Examples include customizing or creating a new role definition, or modifying the membership of a group account in Active Directory.

For example, suppose you have a group of users who need to manage their content, but shouldn't have the full set of permissions associated with Content Manager. You could create a new role definition called Department Content Manager. It could include all of the tasks in Content Manager, except Set security policies for items.

Similarly, if you are a system or network administrator, it is probably easier for you to manage Active Directory group accounts than role assignments in the web portal. You can reduce the overhead of managing role assignments by creating a single role assignment for a group account. Then you can modify the group membership when users no longer require access to reports.

If you determine that modifying or deleting a role assignment is the best approach, remember to check for both system role and item role assignments. Each type of role assignment is configured through different pages in the web portal.

To modify or delete a system role assignment

- Access the web portal of a report server (SSRS Native Mode).
- Select Site Settings > Security. All system-level role assignments currently defined for the server or scale-out deployment are listed by account name.
- Find the role assignment that you want to modify or delete.
- To add or remove the role for a particular user or group, select Edit.
- To delete a role assignment, select the check box next to the user or group name, then select Delete.

To modify or delete an item role assignment

- Access the web portal and locate the item for which you want to edit or delete a role assignment.

- Hover over the item, and select the drop-down arrow.
- In the drop-down menu, select Security.
- Find the role assignment that you want to modify or delete.
- To add or remove the role for a particular user or group, select Edit.
- To delete a role assignment, select the check box next to the user or group name, then select Delete.



Smart City - Establish and troubleshoot IoT connectivity of devices to cloud having multiple communication medium, protocols and networking topology and device management and monitoring

Set up environment for embedded SCADA

Objective:

- to set up environment for embedded SCADA.

Requirement:

- SCADA software
- Computer system
- Server

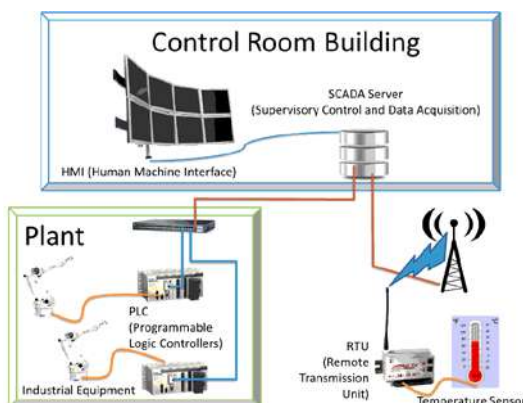
PROCEDURE:

Advancements in Intelligent Instrumentation and Remote Terminal Units (RTUs) / Programmable Logic Controllers (PLCs) have made the process-control solutions in many of the industries to be easily managed and operated by utilizing the benefits of a SCADA system. SCADA is popular in several applications like process industries, oil and gas, electric power generation, distribution and utilities, water and waste control, agriculture/irrigation, manufacturing, transportation systems, and so on. Let us know about the SCADA system's working principle in brief from this article.

What is SCADA System?

SCADA stands for Supervisory Control and Data Acquisition; it is an industrial computer-based control system employed to gather and analyze the real-time data to keep track, monitor and control industrial equipments in different types of industries. Consider the application of SCADA in power systems for operation and control.

SCADA in power system can be defined as the power distribution application which is typically based on the software package. The electrical distribution system consists of several substations; these substations will have multiple numbers of controllers, sensors and operator-interface points.



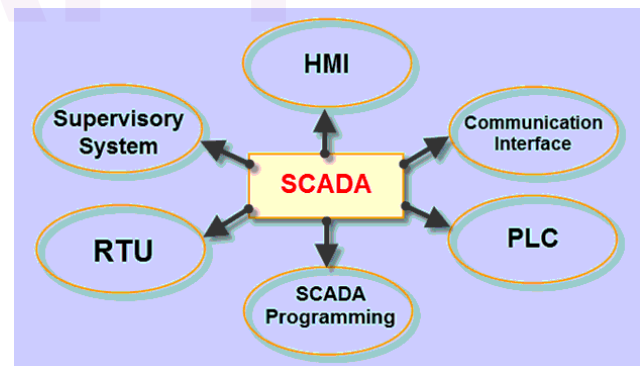
SCADA System

In general, for controlling and monitoring a substation in real time (PLCs) Programmable Logic Controllers, Circuit breakers and Power monitors are used. Data is transmitted from the PLCs and other devices to a computer-based-SCADA node located at each substation. One or more computers are located at different centralized control and monitoring points.

SCADA system usage have become popular from the 1960s with the increase in need of monitoring and controlling the equipment. Early systems built using mainframe computers were expensive as they were manually operated and monitored. But the recent advancements in technology have made-advanced, automated SCADA systems with maximum efficiency at reduced cost, according to the alarming requirements of the company.

SCADA Basics

Before discussing about the architecture of SCADA and different types of SCADA systems, primarily we must know a few SCADA basics. Consider the block diagram of SCADA system shown in the figure which consists of different blocks, namely Human-machine Interface (HMI), Supervisory system, Remote terminal units, PLCs, Communication infrastructure and SCADA Programming.



Basics of SCADA

1. Human-machine Interface (HMI)

It is an input-output device that presents the process data to be controlled by a human operator. It is used by linking to the SCADA system's software programs and

databases for providing the management information, including the scheduled maintenance procedures, detailed schematics, logistic information, trending and diagnostic data for a specific sensor or machine. HMI systems facilitate the operating personnel to see the information graphically.



Human-Machine Interface

2. Supervisory System

Supervisory system is used as server for communicating between the equipment of the SCADA system such as RTUs, PLCs and sensors, etc., and the HMI software used in the control room workstations. Master station or supervisory station comprises a single PC in smaller SCADA systems and, in case of larger SCADA systems, supervisory system comprises distributed software applications, disaster recovery sites and multiple servers. These multiple servers are configured in a hot-standby formation or dual-redundant, which continuously controls and monitors in case of a server failure for increasing the integrity of the system.

3. Remote Terminal Units

Physical objects in the SCADA systems are interfaced with the microprocessor controlled electronic devices called as Remote Terminal Units (RTUs). These units are used to transmit telemetry data to the supervisory system and receive the messages from the master system for controlling the connected objects. Hence, these are also called as Remote Telemetry Units.

4. Programmable Logic Controllers

In SCADA systems, PLCs are connected to the sensors for collecting the sensor output signals in order to convert the sensor signals into digital data. PLCs are used instead of RTUs because of the advantages of PLCs like flexibility, configuration, versatile and affordability compared to RTUs.



Programmable Logic Controllers

5. Communication Infrastructure

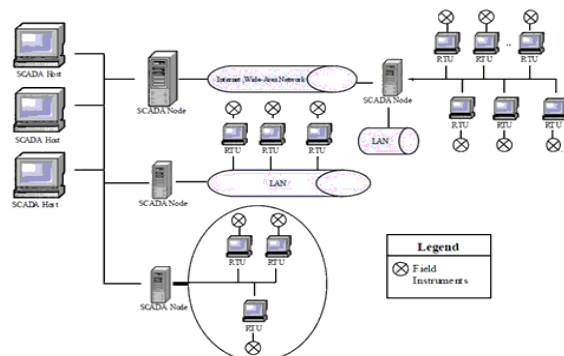
Generally the combination of radio and direct wired connections is used for SCADA systems, but in case of large systems like power stations and railways SONET/SDH are frequently used. Among the very compact SCADA protocols used in SCADA systems – a few communication protocols, which are standardized and recognized by SCADA vendors – send information only when the supervisory station polls the RTUs.

6. SCADA Programming

SCADA programming in a master or HMI is used for creating maps and diagrams which will give an important situational information in case of an event failure or process failure. Standard interfaces are used for programming most commercial SCADA systems. SCADA programming can be done using derived programming language or C language.

Architecture of SCADA

Generally the SCADA system includes the following components: local processors, operating equipment, PLCs, instruments, remote terminal unit, intelligent electronic device, master terminal unit or host computers and a PC with human machine interface.



Architecture of SCADA

The block diagram of SCADA system shown in the figure represents the basic SCADA architecture. The SCADA (supervisory control and data acquisition) systems are different from distributed control systems that are commonly found in plant sites. When distributed control systems cover the plant site, SCADA system cover much larger geographic areas.

Above figure depicts an integrated SCADA architecture which supports TCP/IP, UDP and other IP based communication protocols as well as industrial protocols like Modbus TCP, Modbus over TCP or Modbus over UDP. These all work over cellular, private radio or satellite networks.

In complex SCADA architectures, there are a variety of wired and wireless media & protocols involved in getting data back to the monitoring site. This allows implementation of powerful IP based SCADA networks over landline, mixed cellular and satellite systems. SCADA communications can utilize a diverse range of wired and wireless media.

The choice of the existing communication depends on the characterization of a number of factors. The factors are remoteness, available communications at the remote sites, existing communications infrastructure, polling frequency and data rates. These factors impact the final decision for SCADA architecture. Therefore, a review of SCADA systems evolution allows us to better understand many security concerns.

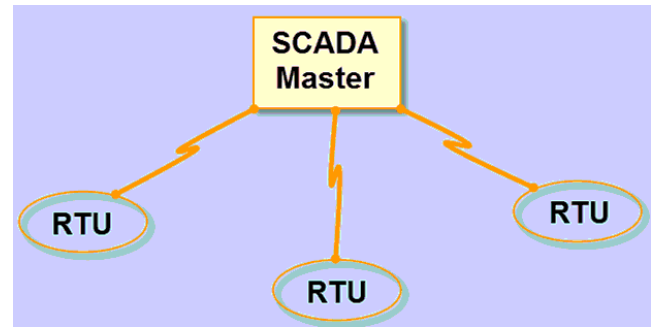
Four types of SCADA systems

There are different types of SCADA systems that can be considered as SCADA architectures of four different generations:

- First Generation: Monolithic or Early SCADA systems,
- Second Generation: Distributed SCADA systems,
- Third Generation: Networked SCADA systems and
- Fourth Generation: Internet of things technology, SCADA systems

1. Monolithic or Early SCADA Systems

Minicomputers are used earlier for computing the SCADA systems. In earlier times, during the time of first generation, monolithic SCADA systems were developed wherein the common network services were not available. Hence, these are independent systems without having any connectivity to other systems.

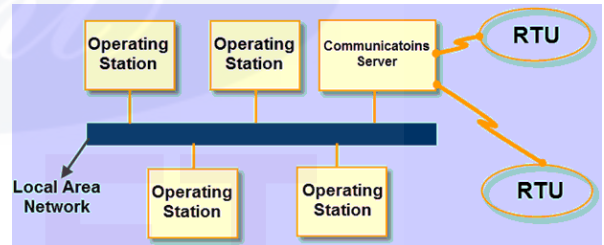


Monolithic or Early SCADA Systems

All the remote terminal unit sites would connect to a back-up mainframe system for achieving the first generation SCADA system redundancy, which was used in case of failure of the primary mainframe system. The functions of the monolithic SCADA systems in the early first generation were limited to monitoring sensors in the system and flagging any operations in case of surpassing programmed alarm levels.

2. Distributed SCADA Systems

In the second generation, the sharing of control functions is distributed across the multiple systems connected to each other using Local Area Network (LAN). Hence, these were termed as distributed SCADA systems. These individual stations were used to share real-time information and command processing for performing control tasks to trip the alarm levels of possible problems.



Distributed SCADA Systems

The cost and size of the station were reduced compared to the first generation system, as each system of the second generation was responsible for performing a particular task with reduced size and cost. But even in the second generation systems also the network protocols were not standardized. The security of the SCADA installation was determined by a very few people beyond the developers, as the protocols were proprietary. But generally the security of the SCADA installation was ignored.

3. Networked SCADA Systems

The current SCADA systems are generally networked and communicate using Wide Area Network (WAN) Systems over data lines or phone. These systems use Ethernet or Fiber Optic Connections for transmitting data between the nodes frequently. These third generation SCADA systems use Programmable Logic Controllers (PLC) for monitoring and adjusting the routine flagging operators only in case of major decisions requirement.



Nimi

DRAFT

Smart City - Demonstrate and deploy responsive Web Application using APIs and generate reports using templates

Explore Web API required input parameters and output

Objective:

- to explore Web API required input parameters and output.

Requirements:

- Computer System
- Web API
- Web application
- Mobile Application

PROCEDURE:

we understand what is Web API, let's see what is an API (Application Programming Interface).

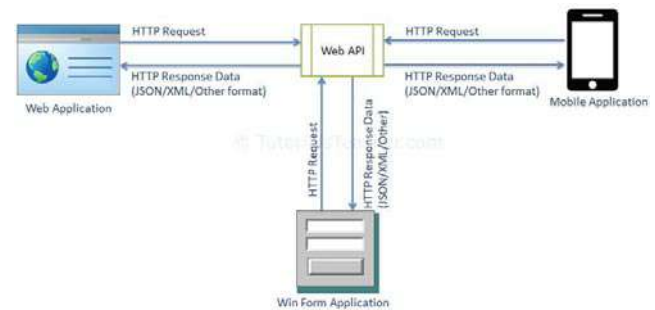
As per Wikipedia's Definition of API: In computer programming, an application programming interface (API) is a set of subroutine definitions, protocols, and tools for building software and applications.

To put it in simple terms, API is some kind of interface which has a set of functions that allow programmers to access specific features or data of an application, operating system or other services.

Web API as the name suggests, is an API over the web which can be accessed using HTTP protocol. It is a concept and not a technology. We can build Web API using different technologies such as Java, .NET etc. For example, Twitter's REST APIs provide programmatic access to read and write data using which we can integrate twitter's capabilities into our own application.

ASP.NET Web API

The ASP.NET Web API is an extensible framework for building HTTP based services that can be accessed in different applications on different platforms such as web, windows, mobile etc. It works more or less the same way as ASP.NET MVC web application except that it sends data as a response instead of html view. It is like a webservice or WCF service but the exception is that it only supports HTTP protocol.



Web API

ASP.NET Web API Characteristics

- ASP.NET Web API is an ideal platform for building RESTful services.
- ASP.NET Web API is built on top of ASP.NET and supports ASP.NET request/response pipeline
- ASP.NET Web API maps HTTP verbs to method names.
- ASP.NET Web API supports different formats of response data. Built-in support for JSON, XML, BSON format.
- ASP.NET Web API can be hosted in IIS, Self-hosted or other web server that supports .NET 4.0+.
- ASP.NET Web API framework includes new HttpClient to communicate with Web API server. HttpClient can be used in ASP.MVC server side, Windows Form application, Console application or other apps.

ASP.NET Web API Versions

Web API Version	Supported .NET Framework	Coincides with	Supported in
Web API 1.0	.NET Framework 4.0	ASP.NET MVC 4	VS 2010
Web API 2 - Current	.NET Framework 4.5	ASP.NET MVC 5	VS 2012, 2013

ASP.NET Web API vs WCF

Web API

Open source and ships with .NET framework.

Supports only HTTP protocol.

Maps http verbs to methods

Uses routing and controller concept similar to

Does not support Reliable Messaging and transaction. ASP.NET MVC.

Web API can be configured using `HttpConfiguration` class but not in `web.config`.

Ideal for building RESTful services.

WCF

Ships with .NET framework

Supports HTTP, TCP, UDP and custom transport protocol.

Uses attributes based programming model.

Uses Service, Operation and Data contracts.

Supports Reliable Messaging and Transactions.

Uses `web.config` and attributes to configure a service.

Supports RESTful services but with limitations.

When to choose WCF?

- Choose WCF if you use .NET Framework 3.5. Web API does not support .NET 3.5 or below.
- Choose WCF if your service needs to support multiple protocols such as HTTP, TCP, Named pipe.
- Choose WCF if you want to build service with WS-* standards like Reliable Messaging, Transactions, Message Security.
- Choose WCF if you want to use Request-Reply, One Way, and Duplex message exchange patterns.

When to choose ASP.NET Web API?

- Choose Web API if you are using .NET framework 4.0 or above.
- Choose Web API if you want to build a service that supports only HTTP protocol.
- Choose Web API to build RESTful HTTP based services.
- Choose Web API if you are familiar with ASP.NET MVC.
- Let's begin by creating a simple ASP.NET Web API project using Visual Studio in the next section.

DRAFT

Smart City - Demonstrate and deploy responsive Web Application using APIs and generate reports using templates

Map Web API to Widget/ Control/ plugin

Objective:

- to Map Web API to Widget/ control/ plugin.

Requirements:

- Computer system
- Web API
- Web application
- Mobile application

PROCEDURE:**Introduction**

This article is extending “Generate TypeScript Client API for ASP.NET Web API “ and is focused on Angular 2+ code examples and respective SDLC. If you are developing a .NET Core Web API backend, you may need to read Generate C# Client API for ASP.NET Core Web API.

Background

The support for Angular2 has been available since WebApiClientGen v1.9.0-beta in June 2016 when Angular 2 was still in RC2. And the support for Angular 2 production release has been available since WebApiClientGen v2.0.

A few weeks after the first production release of Angular 2 being released at the end of September 2016, I happened to start a major Web application project utilizing Angular2, so I have been using pretty much the

same WebApiClientGen for NG2 application development.

Presumptions

- You are developing ASP.NET Web API 2.x applications or ASP.NET Core applications, and will be developing the TypeScript libraries for a SPA based on Angular 2+.
- You and fellow developers highly prefer abstraction through strongly typed data and functions in both the server side and the client sides.
- The POCO classes are used by both Web API data serialization and Entity Framework Code First, and you may not want to publish all data classes and members to client programs.

And optionally, it is better that you or your team is endorsing Trunk based development, since the design of WebApiClientGen and the workflow of using WebApiClientGen were assuming Trunk based development which is more efficient for Continuous Integration than other branching strategies like Feature Branching and GitFlow, etc. for teams skillful at TDD.

For following up this new way of developing client programs, it is better to have an ASP.NET Web API project. You may use an existing project, or create a demo one.

Smart City - Demonstrate and deploy responsive Web Application using APIs and generate reports using templates

Display and Configure graphs, charts and other ready to use controls and widgets

Objective:

- to display and configure graphs, charts and other ready to use controls and widgets.

Requirements:

- Computer system
- Web application

PROCEDURE:

Gain visibility into your team's progress by adding one or more widgets or charts to your dashboard. Customizable, highly-configurable dashboards provide you and your teams with the flexibility to share information, monitor progress and trends, and improve your workflow processes. Each team can tailor their dashboards to share information and monitor their progress.

If you're just starting out, read [Add, rename, and delete dashboards](#). If you're looking for instructions on a specific task, in [context chart](#), [widget](#), or [report](#)—review [Dashboards and widgets quick reference](#). To learn about our reporting solutions, read [Reporting Roadmap](#).

Important**TABLE 1**

To view the content available for your platform, make sure that you select the correct version of this article from the version selector which is located above the table of contents. Feature support differs depending on whether you are working from Azure DevOps Services or an on-premises version of Azure DevOps Server, renamed from Team Foundation Server (TFS).

To learn which on-premises version you are using, see [What platform/version am I using?](#)

Key concepts

- Dashboards are customizable interactive signboards that provide real-time information. Dashboards are associated with a team or a project and display configurable charts and widgets.

- Charts are query-based status or trend charts derived from a work item query or test results.
- Widgets display configurable information and charts on dashboards. The widget catalog provides brief descriptions of those widgets available to you. In addition, you can add widgets provided through the Azure DevOps Marketplace.
- In-context reports are system-generated charts that support specific services. Examples are team velocity, sprint burndown, and the Cumulative Flow Diagram (CFD), and the Test Failures Report. These reports are displayed on the Analytics tab for a specific service and derive data from Analytics.
- Power BI reports allow users to create rich, customized Power BI reports or other reports using OData queries of Analytics data and the returned JSON data.

Configurable dashboards

With dashboards, you can configure an array of charts and widgets.

Each team can add and configure multiple dashboards to share information, view status, progress, and trends, and access quick links and other functions. Easily add and rearrange widgets on the dashboard to show recent changes made to view build status, bug trends, and more.

Sequence for adding and customizing a dashboard

Click a box below to open the corresponding article.

Charts: Work tracking status and trends

With flat-list queries, you can create various charts to monitor status, progress, and trends. To get started, you can open a shared query and create a chart based on your tracking interests. Chart types include status—pie, bar, column, stacked bar, and pivot—and trend—stacked area, line, and area—charts.

Smart City - Demonstrate and deploy responsive Web Application using APIs and generate reports using templates

Generate reports using readily available API, templates and to export it to excel, word pdf and other required formats

Objective:

- to generate reports using readily available API, templates and to export it to excel, word pdf and other required formats.
-

Requirements:

- WEB Application
- WEB API
- Computer System

PROCEDURE:

ADManager Plus offers more than 150 predefined reports on Active Directory including inactive users, locked out users, recently created groups, GPO-linked OUs, and more. You can even export these reports to the following formats for easy access:

- CSV
- PDF
- XLSX (Excel)
- HTML
- CSVDE
- XLS

The export option involves simple point-and-click tasks,

unlike the complex and tedious ones required with PowerShell or native AD tools. You can also customize the file heading and attributes included in the exported file. Having crucial AD information, such as computer name and description, email addresses, and distribution lists, readily available in the format of your choice makes communication with upper management easy. Needless to say, ADManager Plus' export option saves you a considerable amount of time and effort.

Export data from multiple platforms

Besides preconfigured AD reports, ADManager Plus provides reports for Office 365 (such as inactive users, distribution groups, and license reports) and Exchange Server (such as mailbox-enabled users, POP3-enabled users, and distribution list members). The tool also offers compliance reports, NTFS reports, and many more. All these reports can be exported to your desired format.

Print and schedule reports

Schedule reports to be directly emailed, in the desired format, as frequently as needed (hourly, daily, weekly, monthly, or custom). You can configure the generation schedule for multiple reports from a single screen and have the reports automatically delivered to specific email addresses.

Smart City - Identify and test smart Lightening system and its components

Rig up circuit to lighting system and measure different parameters such as voltage, current Lux using Multimeter and Lux meter

Objective:

- to Rig up circuit to lighting system and measure different parameter such as Voltage, current lux using Multimeter and Lux meter.

Requirement:

- 1 light dependent resistor
- 5 kilo ohm resistor
- Arduino
- 2x16 LCD shield
- Breadboard
- Digital Multimeter (DMM)
- Commercial lux meter (for characterizing your LDR)

PROCEDURE:

A multimeter by definition is a meter that can measure multiple parameters. We will be discussing how to choose an electrical multimeter, the different types and how to use it safely. I did go through a 5 year union electrical apprenticeship and was an active union electrician for 10 years (IBEW local #46, Seattle) before some nerve damage caught up to me. I also do a lot of electronics work so feel well qualified to make this guide.

Please keep in mind that I have a wide audience outside of r/spacebuckets to include botany students.

DC= direct current like from a battery or the secondary side of an LED power supply (the wires that go to the LED. The highest I've worked with professionally is 700 volts DC when spending 3 months rebuilding the Seattle Monorail trains in 1998.

AC= alternating current like you get from you wall receptacle. All line voltage you will encounter will be AC voltage to include the line side of an LED power supply. 480 volts AC is typically the highest the voltage an indoor wireman electrician will work with in the US but there are exceptions like doing some high voltage splicing. Linemen do all the high voltage stuff you see on the streets.

Line voltage/power= what you get from your receptacles/ plug ins. It's either 120 or 220/240 volts depending on where you live. What makes it dangerous is it's low impedance which means a lot of current can flow. Voltage hurts, current kills. 20-50mA (milliamps) can be lethal if you take a direct shot through the heart. With low voltage theses these current levels will never be hit in most situations through your body. That's why I push people to use lower voltage power supplies.

Ohm's Law- we will cover this later but is important to know.

Series/parallel circuit- will cover this more later but you measure voltage in parallel and current in series with most multimeters.

A lot of stuff in parentheses you can ignore.

The crappiest electrical multimeter you can buy

Why would I talk about the lowest price "junk" meter? Because a lot of people are on a budget, they're often given away for free so people may have them laying around and I want to ensure that people understand the safety risks involved. It's the same way I would recommend generic UFO lights for Space Buckets or junk Epistar/Epileds LEDs but never for professional use (Epistar LED dies are not bad per se, it's the Chinese manufactures making the LED and using the bottom of the barrel in terms of quality). They're cheap and the hobbyist is going to use them.

Here is an Amazon link (be sure to buy through the spacebuckets.com website if you can). They have different branding just like cheap UFO lights have different branding.

Rule#1- don't use these \$6 multimeters with line voltage. Period.

Reason#1- the probes are so cheap that they will easily snap on you. They are two piece pressure fitted so easily come apart. This can really suck if you unintentionally just grab the exposed energized connection to get it loose while perhaps standing on a wet concrete floor. Hopefully you have a GFCI. The proper way is to shut the power off at the circuit breaker and then pull the probe out. I only say this because despite my warnings some people are going to do what's the cheapest. Plug a lamp or something in the other sockets to make sure the power is actually off.

Reason#2- they are not fused and have a very thin set of strands in their leads so they might just melt on you for current measurements even though this meter says it's good for 10 amps. More on this below but with most meters the max current is only meant to be used intermittently unless designed otherwise.

Reason#3- is your life worth \$6? That meter says it's rated for 1000 volts DC or 750 volts AC and it has a "CE" marking. People might be fooled in to thinking that these meters are actually safe at line voltages. In 120 volt AC countries the peak voltage is about 170 volts. In 240 volt countries the peak voltage is about 340 volts. I would not want to be holding on to a \$6 meter at these voltages.

(BTW, in US/Canada the standard residential line voltage is 120 volts +/- 10% per code but typically 115-120 volts. This is the RMS voltage or the “effective” voltage. 120 volts DC would heat up a resistor as much as 120 volts AC RMS. RMS means “root mean square”. Higher quality meters will state that they are “true RMS” meters which means they can accurately measure AC voltages other than sine waves.)

These very low cost meters are OK to use when measuring low voltage (<60V DC or <30V AC) and when you are measuring something that is fairly low current limited (<100 volt-amps or less than 100 watts). Why 60 volts DC or 30 volts AC in particular? I have to say that for liability reasons alone and just a professional recommendation to someone just getting started in electrical/electronics.

How to use this multimeter:

For measuring voltage make sure that the black lead is in the “COM” receptacle. COM means common. Have the red lead in the receptacle that says “V” typically with also the omega symbol (the omega symbol is for measuring resistance). Set this multimeter's voltage to 200 volts DC. It should look like this. You should be good to go but should not expect stellar accuracy. If measuring below 20 volts then set the meter to 20 volts.

For measuring current set the meter to 10A (10 amps) and move the red lead to the 10ADC (amps DC) receptacle. It should look like this.. So, what does “111” mean? With this meter it means 1.11 amps. You get what you pay for.

And here is where the danger can kick in: you just measured current which is done in series with a circuit but voltage readings are done in parallel with a circuit. It's can be easy to forget what the meter settings are (just being absent minded or whatever) so you can run in to a dangerous situation like this where you want to then measure the line voltage but you meter is set to current. This is what it looks like.. That meter has no fuse so what's going to happen if you meter is setup for current but you try to read voltage? Hopefully the power mains circuit breaker trips before anything real bad happens and you just damage your meter. What if the 15 amp circuit breaker doesn't react fast enough and the main 200 amp breaker has to catch the short circuit that you just created? I have seen this happen once and no matter how you look at it something bad is going to happen particularly if you are holding that meter in your hand. As we said in the military, “better you than me”.

This type of low cost meter would also work well with small solar cells to make relative light measurements. Set the meter up to measure current, short the small solar cell in to the meter (just hook it up), and you'll have a relative light meter that is cosine correct and linear over 7-10 orders or magnitude. These \$2-3 solar outdoor LED lights will have the solar cell needed.

Get me off that line voltage

Just use a Kill A Watt for line voltage work. It's ETL

marked which is the same as UL listed that is the gold standard in electrical safety and you can just plug a power strip in to it. I would not max these out in terms of how much continuous current you put through the device. Pinching male socket plugs together can sometimes help to create a better connection from a cord to the meter. The 4460 linked to did indeed have less than 0.2% error as advertised.

The Kill A Watt meters are ETL marked. UL listed (US) is the same as ETL marked (US) or CSA marked (Canadian) but not the same as CE marked (European) as far as Washington State electrical code goes and I'm sure for every other state or Canadian province. UL/ETL/CSA are independent test laboratories but CE has self test provisions which opens up abuse for potential fraud with very low end electrical devices. I don't trust CE markings alone myself. This can be particularly important in commercial grow set ups where lab approved electrical devices must be installed in many cases. It costs about \$20,000 or so to get something lab tested and marked which can be harsh with a start-up making a LED grow light.

A little better quality multimeter

A high quality meter will have the fuse and tell you right on the screen that you're about to make a terrible mistake. The problem in the picture is that I have the probes hooked up to measure current, but the meter set up to measure voltage and hooked up to the line voltage power source in parallel as you should to measure voltage. This configuration with the meter actually electrically in series with no other load in series, such as a power supply, creates which is essentially a dead short circuit condition. This is bad.

The lowest priced multimeter I have seen so far that I would consider line voltage safe is this Amprobe AM-510 that has a high rupture capacity (HRC) fuse (see conversation below in comments). In no circumstance could I recommend a general purpose electrical multimeter for line voltage work without a HRC fuse.

Rule #2: No HRC fuse means no line voltage electrical work (there's a theme here). This axiom will help you select a meter most appropriate for you.

There's a lot of \$20 range meters that have a pretty good meter count (a higher resolution). Cheap meters have a 2000 count, high resolution meters perhaps a 50,000 count. As a layman these are just simple ratios in terms of resolution but a 50,000 count \$20 meter isn't quite the same as a \$600 50,000 count NIST traceable meter (NIST traceable means it's been highly calibrated to a very accurate source and costs extra. I can then calibrate cheap meters to a NIST traceable meter). Really high impedance meters can give different readings than lower impedance meters in some cases which is something to keep in mind. (The Fluke 287 has an input impedance of 100GOhm so it can measure down to 1uV, 10nA and up to 500MOhm. It works well).

\$50 or so is what you can expect to pay for a good working meter but I don't have a lot of experience with

mid level meters so there would be others in a better position to make a credible recommendation based on personal experience as to what meters to get. If in doubt, buy a Fluke meter. edit- /u/PedroDaGr8 has some good recommendations below.

The clamp on meter

A clamp on meter means you can put that clamp around a wire and measure the current. This is easy money to a commercial service truck electrician- you're called in to a job because a breaker keeps tripping. You put the clamp over the wire/circuit in question and measure about 9 amps. You know this means a "loose" breaker (the action of the loose breaker will typically also feel different). Install a new one and go on to the next job. A 30-45 minute job was just billed at the standard 2 hour minimum rate (you have to factor in travel time but I could bill 10-11 hours on some days. Service truck work sucks because you have to deal directly with the public instead of saying "go talk to the foreman" but you learn a hell of a lot. 18 months as an apprentice and journeyman was enough). Some buildings and a lot of industrial sites have maintenance electricians to deal with small stuff.

You generally only measure one wire at a time and is handy when probing deeper than a Kill A Watt. Notice how a blue wire is the ground wire (actually looking back at the picture you can't see it but it is) in this UFO light and not green or yellow/green? Keep stuff like that in mind and don't assume.

Clamp on meters measures the magnetic field (not the electric field- voltage ticks warn you if a wire is energized). The greater the magnetic field the more current that is flowing which is the principal behind the clamp on meter.

They are best used back at the electrical panel and are an important tool in 3 phase load balancing (3 phase electrical is used in almost all commercial/industrial set ups. Beware of the 3 phase delta high leg if you don't know what you're going. I've installed them, seen them elsewhere and you should call in an electrician if you have to work with them).

Some meters are DC only

You can get cheap DC meters that will read volts and amps at the same time. Thou shall not try to read an AC voltage with them particularly line voltage. You must measure the current from the load to ground with these types of meters and the voltage from the LEDs preferably. You can monitor the temperature of the LEDs if using a constant current power supply by watching the voltage drop across the LEDs- lower voltage means hotter LEDs.

Scope meters

Stay away from cheap scope meters. I have one somewhere but never use it. If you need an oscilloscope then just buy one. You pay for speed and however fast your scope is it's not fast enough (I use an older version of the Tektronix TDS 2012). Rigol has a good reputation as far as lower cost oscilloscopes. Save your money and get a faster one because the one you're looking at isn't fast enough (that's a bit tongue in cheek).

So this is just the basics on how to use a meter. Ohm's law will be covered in my next article on LED power supplies but you need to know how to use a meter first.

DRAFT

Smart City - Identify and test smart Lightening system and its components

Test different types of lighting system such as outdoor and indoor, street light etc

Objective:

- to Test different types of lighting system such as outdoor and indoor, street light.

Requirements:

- Different LED lights

PROCEDURE:

Energy conservation is becoming more and more of a concern. One solution is to use more energy efficient bulbs at home, schools, businesses and public places. This document will introduce key terms to use when talking about lighting as well as types of light bulbs in use.

Key Terms:

- Watt (abbreviated W) is the unit corresponding to the rate of energy consumption (or power) in an electric circuit needed in this case to light a light bulb.
- Lumens are the unit describing the amount of light (energy) as seen by the human eye that is given off by the light bulb.
- Lux is defined as being equivalent to one lumen spread over an area of one square meter. To put it another way, a measurement of lux (light intensity) tells you how many lumens are needed for the area being illuminated.
- Efficiency is the amount of light that comes out of a light bulb compared to the electrical energy that goes into it. Energy efficient bulbs waste less energy in producing light than less efficient bulbs. Efficiency is an output over an input.
- Efficacy is related to the efficiency of the light bulb. It is also an output over an input. The output is the lumens of light and the input in power in watts. So, the expression we use to describe the efficacy of our light is "Lumens Per Watt" or lumens divided by watts. Keep in mind, the bigger the efficacy, the more efficient the bulb.
- Some bulbs make things they are illuminating look a different color than they really are. This property is called Color Rendition. Generally, bulbs with good color rendition are used. Color rendition can be measured by the Color Rendering Index (CRI), which is a scale ranging from 0 (very poor color rendition) to 100 (nearly perfect color rendition). In cases of extremely poor color rendition, that bulb's CRI value can be negative. Incandescent bulbs are defined to have perfect color rendition.

Indoor Lighting

Some of the most common indoor light bulbs are incandescent bulbs, which look like a traditional light bulb. Generally, the input for these bulbs is either 40W or 60W. But there are other kinds of indoor light bulbs as well, such as CFLs and LEDs. Keep in mind that the wattages listed for the CFLs and LEDs correspond to the 40W and 60W incandescent bulbs. This means that for a lamp that takes a 40W incandescent, you could also use a 9W CFL or a 6W LED. This will allow you to easily compare the bulbs to one another.

Incandescent Bulbs:

Wattage	40	60
Lumens	290	840



The incandescent light bulb has had the same design for over 100 years since Thomas Edison invented it! It produces light when a thin wire called a tungsten filament is heated by electricity running through it making it so hot that it starts to glow brightly. This releases a lot of heat and the bulbs get hot to the touch, meaning this bulb is very inefficient. Many countries, including the United States, are currently passing legislation banning the sale of these light bulbs because they are so inefficient. CRI: 100

Compact Fluorescent Light Bulbs (CFL):

Wattage	9	13
Lumens	550	810



These spiraled light bulbs are far more efficient than the standard incandescent bulb. Compact Fluorescent Light bulbs (CFLs) work by running electricity through gas inside the coils, exciting that gas, and producing light. There is a coating on the spirals, which makes this light white. These bulbs do not get nearly as hot as the incandescent bulbs.

CRI: 50-80

Wattage	6	9.5
Lumens	450	800



DRAFT

Smart City - Identify and test smart Lightening system and its components**Check circuit to test and troubleshoot Sensor Integrated lighting system****Objective:**

- to check circuit to test and troubleshoot Sensor integrated lighting system.

Requirements:

- Computer system
- Lamp
- LED
- Jumper wires
- Arduino
- PIR Sensor

Procedure:**Automatic Room Lights using Arduino and PIR Sensor**

In this project, we will see the Automatic Room Lights using Arduino and PIR Sensor, where the lights in the room will automatically turn ON and OFF by detecting the presence of a human.

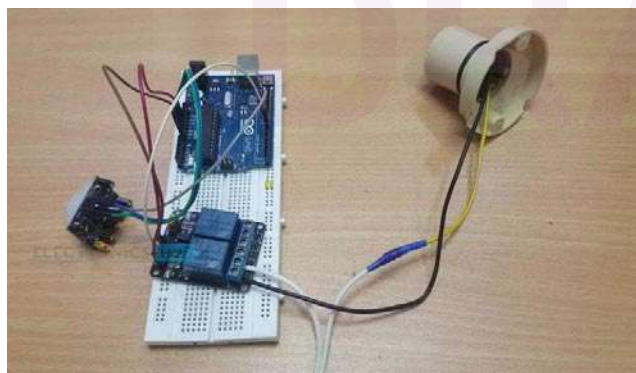
Such Automatic Room Lights can be implemented in your garages, staircases, bathrooms, etc. where we do not need continuous light but only when we are present. Also, with the help of an automatic room light control system, you need not worry about electricity as the lights get automatically off when there is no person.

So, in this DIY project, we have implemented Automatic Room Lights using Arduino and PIR Sensor.



Automatic Room Lights System using Arduino is a very useful project as you need not worry about turning on and off the switches every time you want to turn on the lights. The main components of the Automatic Room Lights project are Arduino, PIR Sensor and the Relay Module.

Out of the three components, the PIR Sensor is the one in focus as it is the main device that helps in detecting humans and human motion. In fact, the Automatic Room Lights project can be considered as one major application of the PIR Sensor. A similar concept is being already implemented in automatic toilet flush valves, hand dryers, etc.



Smart City - Identify and test smart Lightening system and its components

Apply non-sensor integrated lighting system

Objective:

- to apply non-sensor integrated lighting system.

Requirement:

- Computer system
- LED
- Lamp
- Jumper Wires
- Switch

PROCEDURE:

Introduction to Lighting Controls

A good lighting design includes a good controls design. Lighting controls play a critical role in lighting systems, enabling users manually or automatically to:

- turn the lights ON and OFF using a switch; and/or
- adjust light output up and down using a dimmer.

This basic functionality can be used to generate these benefits for the lighting owner:

- flexibility to satisfy user visual needs; and/or
- automation to reduce energy costs and improve sustainability.

In recent years, lighting controls have evolved two additional capabilities:

- adjust light source color, including shade of white light; and/or
- generate data via measuring and/or monitoring.

Based on an update to LCA's Education Express EE101: Introduction to Lighting Control, this article provides an overview of the basic functionality of today's lighting controls, benefits and the basic questions to ask when identifying an appropriate lighting control strategy.

Lighting Control Effects

Lighting controls provide the following basic functions. End-users harness these functions to support energy management and/or visual needs.

WHAT	HOW
Produce the right amount of light ...	Light output (intensity) dimming
... where the light is needed ...	Zoning of luminaires to controllers
... and when the light is needed.	Automatically reduce lighting when the space is unoccupied

Lighting controls are evolving to provide advanced functions, which vary in availability depending on system type and application need.

WHAT	HOW
Produce light at the right color or shade of white light ...	Separately dimming arrays of LEDs with different colors or white-light correlated color temperatures (CCTs)
... allow remote programming and control ...	Control systems with programming and lighting management capability
... and tell you how your lights are performing.	Centralized intelligent control systems with measuring and/or monitoring/alarm capability

Benefits: Visual Needs

By adjusting the intensity of one or more layers of lighting in a space, lighting controls can:

- change space appearance;
- facilitate different functions of the space;
- alter atmosphere and mood;
- reduce glare; and/or
- increase user satisfaction by providing users the ability to control their lighting.



Images courtesy of Finelite.

Benefits: Energy Management

By reducing lighting ON time, intensity or zoning, lighting controls reduce both demand and energy consumption. According to a Lawrence Berkeley National Laboratory

(LBNL) study, popular lighting control strategies produce 24-38% average lighting energy savings, which reduces building operating costs.

Because of strong energy savings, the majority of state commercial building energy codes require a wide range of controls in new construction. In existing construction, the controllability of LED lighting results in an ideal pairing with controls, combining to minimize energy costs.

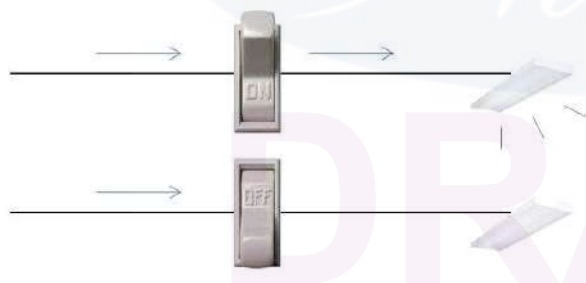
Basic Function

Lighting controls are input/output devices and systems. The control system receives information, decides what to do with it, and then adjusts lighting power accordingly. Here we see a basic lighting circuit (switch leg). Power travels along the circuit to energize a group of lights. This lighting system supplies illumination.



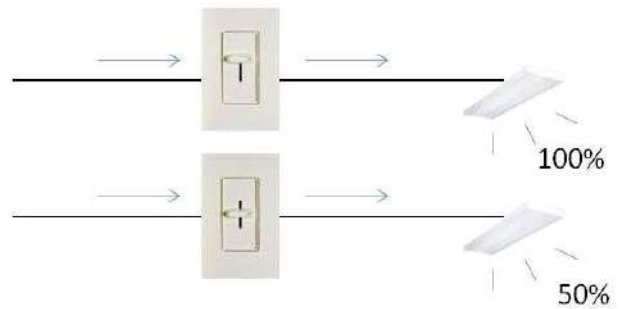
Switching

One basic output is switching. Here we see a switch placed on the line between the power source and the load. When the switch closes (i.e., switch is flipped "ON"), the circuit completes, allowing power to flow to the load. When it opens, the circuit breaks (switched flipped "OFF"), which disrupts power to the load. This makes the switch a power controller.



Dimming

Another basic output is dimming. If a dimmer-switch is used, in addition to ON/OFF, it can alter current flowing through the load during the ON state, which raises or lowers light output. Here we see a dimmer placed on the line, with the output being continuous dimming over the load's dimming range.



Color and CCT Control

With LED, it is relatively economical to provide users the ability to adjust lighting color and CCT.

With tunable-white LED products, separately dimming arrays of warm- and cool-white LEDs allows users to adjust light source CCT. Other colors may be added to enhance the available color spectrum and ensure good color rendering.

Two other approaches are dim-to-warm (LED products that dim to a very warm white similar to incandescent dimming) and full color tuning (separately dimmable red, green and blue LEDs plus amber or white and potentially other colors).



Image courtesy of USAI Lighting.

Smart City - Identify and test smart Lightening system and its components

Test different dimming control methods in lighting system

Objective:

- to test different dimming control methods in lighting system.

Requirements:

Dimmer switch

Switch plate

PROCEDURE:

What are the different methods used for dimming?

Several methods are available to dim lighting. These dimming methods are categorised into three groups:

- Dimming of electrical potential (decrease in power): phase control
- Dimming of control signal (analogue): 0-10V, 1-10V
- Dimming of control signal (digital): DALI

For the first type, dimming of electrical potential, a classic 3 core electrical cable will do. For the other two types of dimming, a 5 core electrical cable is required.

Phase control

Phase control is a dimming technique based on the electric wire that is often used for halogen and incandescent lamps. It “clips” part of the sine wave of the alternating current to dim the light. The following examples will make this clear.

Leading edge phase control

When a phase is cut (i.e. limited), the voltage will only flow at a certain amount of time after the zero crossing (i.e. the sine wave crossing the horizontal axis). Only the latter part of the wave is transmitted. This waiting time can be determined using a simple resistor-capacitor or digital switches. This dimming technique is suitable for both inductive and resistive loads (traditional magnetic ballast).

Trailing edge phase control

With phase control, the voltage is cut off before the end of the sine wave so that only the first part is transmitted. This dimming technique is used for capacitive loads (EVSA).

Phase control

Sometimes, both leading and trailing edge phase control is possible. This wave combines the aforementioned:

1-10 V

With a 1-10 V dimming technique, a signal is transmitted between 1 V and 10 V. 10 V is the maximum amount (100%) and 1 V is the minimum amount (10%).

0-10 V

Transmits a signal between 0 and 10 V. The output of the lamp is scaled such that a voltage of 10 V provides 100% light output. And, 0 V provides the least light output.

DALI

DALI stands for Digital Addressable Lighting Interface. It is an international standard that defines how a lighting installation should communicate with control and steering systems.

Important to know is that DALI is independent of manufacturers. This means that it is possible to use different brands of components in the same system.

Each system consists of a controller and a maximum of 64 lighting components, such as a ballast. Each of these components is given a unique address. The controller can control these components because the DALI system can transmit and receive data.

DALI is dimmable from 0-100%.

Built-in dimmers

There are two types of built-in dimmers: rotary or push button.

A rotary knob dimmer can be pressed to turn lights on or off. You turn the knob to select the light intensity.

A push button works according to the same on-off principle. However, to change the light intensity, you must hold in the button. Some push button dimmers alternate in their operation (brightness increases during the first long press, dimming occurs during the second long press). Other push button dimmers reach a specific percentage (brightness increases to a specific intensity when N percent is reached and then dims again).

Smart City - Identify and test smart Lightening system and its components

Rig up the circuit to interface microcontroller, LDR and light to vary brightness in accordance with illumination of the light. Upload the code to the microcontroller and test for proper operation

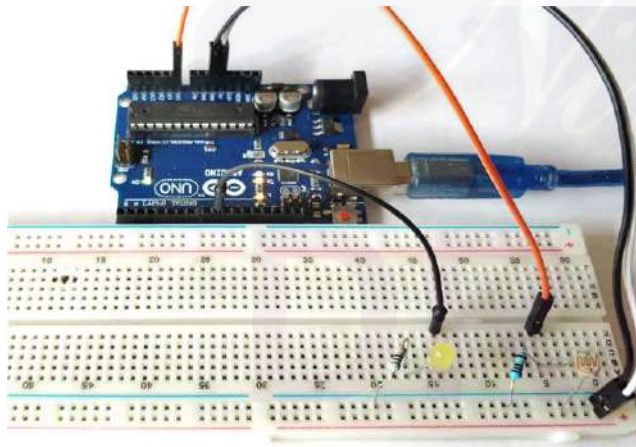
Objective:

- to Rig up the circuit to interface microcontroller, LDR and Light to vary brightness in accordance with illumination of the light
- to upload the code to the microcontroller and test for proper operation.

Requirement:

- Arduino UNO
- LDR (Light Dependent Resistor)
- Resistor (100k-1;330ohm-1)
- LED - 1
- Relay module - 5v
- Bulb/CFL
- Connecting wires
- Breadboard

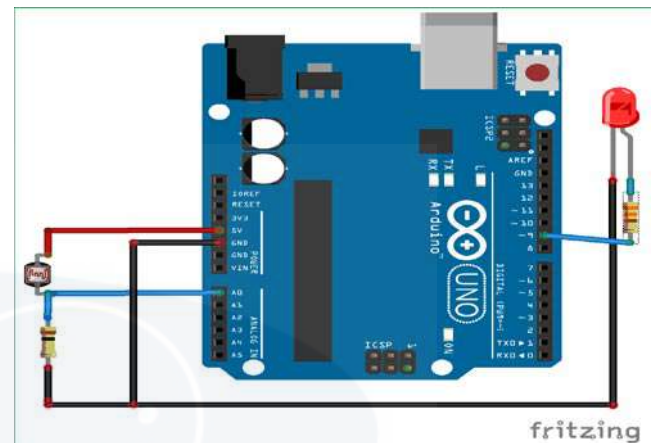
PROCEDURE:



We all want our home appliances to be controlled automatically based on some conditions and that's called Home automation. Today we are going to control the light based on darkness outside, the light turns ON automatically when it is dark outside and turns off when it gets bright. For this, we need a light sensor to detect the light condition and some circuitry to control the Light sensor. It's like Dark and light Detector circuit but this time we are using Arduino to get more control over light.

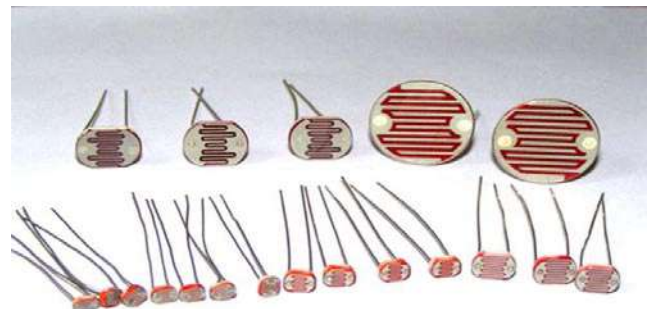
In this circuit, we are making a Light Sensor using LDR with Arduino to control a bulb/CFL as per light condition of the room or outside area.

Circuit Diagram



LDR

LDR is Light Dependent Resistor. LDRs are made from semiconductor materials to enable them to have their light-sensitive properties. There are many types but one material is popular and it is cadmium sulfide (CdS). These LDRs or PHOTO RESISTORS work on the principle of "Photo Conductivity". Now what this principle says is, whenever light falls on the surface of the LDR (in this case) the conductance of the element increases or in other words, the resistance of the LDR falls when the light falls on the surface of the LDR. This property of the decrease in resistance for the LDR is achieved because it is a property of semiconductor material used on the surface.



Check circuit to test and troubleshoot MQ135 pollution sensor module

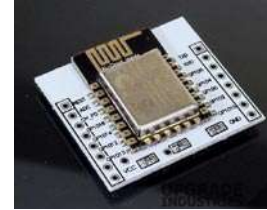
Objective:

- to check circuit to test and troubleshoot MQ135 pollution sensor module.

Requirements:

- MQ135 Gas sensor
- Arduino Nano
- LCD
- Breadboard
- 10K potentiometer
- 1K ohm and 220 ohm resistors
- Buzzer
- Jumpers

- ESP8266



- LCD Display



PROCEDURE:

Introduction

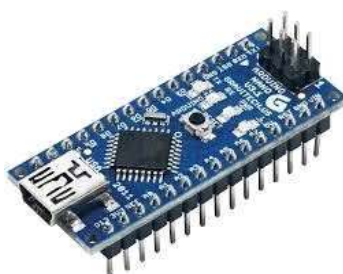
Air pollution sensors are devices that can detect the presence of air pollution in the surroundings, as the name suggests. The detection is in terms of a ppm value of the pollutants the sensor can detect, and then those ppm values can be used to monitor the environment, based on standard safety values. There are different types of sensors, depending on the usage. The major ones are specialized for one or more of the following components: Ozone, PM, Carbon Monoxide, Sulphur Dioxide and Nitrous Oxide. Our project is a simple demonstration to show that it can be easily done. We have used an MQ135 Gas Sensor, which is sensitive to gases like Ammonia, Benzene and Carbon Dioxide.

Required Components:

- MQ135 Gas sensor



- Arduino Nano



We are making an IOT based Air Pollution Monitoring System, in which we will monitor the air quality over a computer terminal, and an alarm will be triggered if the air quality goes beyond a certain level. The thresholds we have used are very general, so they may vary place to place. For example, as pointed out during evaluation of the project, concentration of Carbon Dioxide is high in households and closed places, and this may raise the overall PPM Value. We connect our sensor as an input to the arduino (Arduino Nano), and a buzzer, a LCD display and an ESP8266 as output. The LCD Display outputs the sensor data (in tangible language, of course). The voltage levels are converted into appropriate ppm values by calibration, as there are inbuilt functions in the MQ135 Library. The buzzer is used as an alarm and the ESP8266 Wi-Fi Module gives us the access to the Wi-Fi network. We have also connected a potentiometer along with the LCD Module so that the brightness can be varied, and kept less at night.

Smart City - Identify and test smart Lightening system and its components

Install CCTV Camera for building security and roadside safety

Objective:

- to Install CCTV Camera for building security and roadside safety.

Requirement:

System

Arduino

Camera

PROCEDURE:

History of IHDIT

Institute for Human Development and Information Technology (IHDIT) is a non-Governmental Organization, established in August 2005, at Malam Jamo House, last floor, opposite Government Secondary School Tarauni Kano State. Furthermore, this organization empowering people through Skills Acquisition Programs., Building Capacity through Training, Workshops, Seminars, Strategic meeting, Advocacy visitory, Media Campaigns, Drama/Theater Performance, Public Enlightenment and Awareness Campaigns, Design, Production and Distribution of IECs Material and Promoting Entrepreneurship Training in the Society as initiative aims raising a new class of Entrepreneur who can compete globally and succeed in managing Micro/Small and Medium Enterprises as well as provide channel for future Industrialization within the Community and Society at large.

Major Target

To develop the people at grass root level, from the inception of this institution to August 2017, they have successful train and graduated over thirty eight thousand (38,000) people as beneficiaries of training programmes. It is also become as open door programme policy that allow it to work with government, individuals, domestic and international organization that promote human development especially in the area of skills acquisition programmes and awareness campaign for the development of humanity.

Objective

- Promoting human development through coordination ideas, toward efficient and effective provision of skills acquisition programmes to the society for self-sustenance and development.
- Providing measures in discouraging corruption, child abuse, human trafficking with particular interest on disaster prevention, accident and emergency rescue assistance
- Promoting education, budget monitoring and tracking and volunteerism to the society.

- Empowering youths especially women and child in the society in order to become self-reliance
- Transforming hopeless-ness into hope and through human resource developmental activities

Components of a CCTV System

Cameras get most of the attention in the beginning, you also have other concerns, such as viewing, recording, and archiving the video footage, and the equipment required for carrying out those tasks. Here's a look at the basic components of a typical CCTV system.

Cameras

Security cameras are the starting point for most CCTV systems. There are endless possibilities when choosing CCTV cameras and lenses – everything from fixed models designed for monitoring very specific locations, to day/night cameras, and powerful PTZ domes for patrolling large areas.

Monitor

In a traditional CCTV setup, operators view footage from a central location on a monitor very much like a TV, but with higher lines of resolution for better picture quality. Monitors can be dedicated (meaning they display video from a single camera), or call-up (meaning operators can access multiple cameras at the same time).



Cable

With an analogue system, coaxial cable is required for transmitting video footage from the cameras. This is one of the drawbacks of analogue CCTV, as the cable can be expensive and difficult to install, especially for larger camera networks, and those where cameras must be positioned in difficult locations.

DVR

Most modern CCTV systems incorporate DVRs (digital video recorders) which enable operators to reap some of the benefits of a network-based surveillance setup. DVRs convert the analogue footage to digital, which helps to extend storage capacity, makes it much easier to search archived footage, and also allows users to stream video over a network for remote viewing from multiple locations.



Types of CCTV Security Cameras

Fixed

Fixed CCTV cameras point in a single direction, which makes them perfect for monitoring very specific areas of interest. For this reason, fixed cameras are quite effective not only for capturing footage of suspicious activity, but also for deterring criminals and vandals from carrying out their acts in the first place.

PTZ

PTZ cameras are ideal for wide-area surveillance. They give operators the ability to remotely control pan, tilt, and zoom functions to follow activity and to zoom in for detailed monitoring. This is an area where analogue CCTV cameras fall behind their IP camera counterparts. With IP cameras, the pan/tilt/zoom functions are controlled manually or automatically and delivered over a single network cable, while analogue cameras require additional wiring to perform similar functionality.

Understanding CCTV Image Resolution

The image detail of an analogue CCTV camera is usually conveyed in a form of measurement called TVL (or TV Lines). Think of the video picture as being composed of active horizontal lines. These lines are delivered to a monitor or recording device in two off-set fields.

One field contains even-numbered lines while the other contains odd-numbered lines. The viewer sees a complete picture because the lines are interlaced. Since the picture has a 3:4 aspect ratio, the amount of detail you can measure in 3/4 of the picture's width determines the horizontal TVL resolution. Generally, most standard CCTV cameras offer a TVL resolution of around 380, while high-resolution cameras will deliver something closer to 540 TVL.

Smart City - Identify and test smart Lightening system and its components

Rig up the circuit to interface Microcontroller, MQ135 pollution sensor and vary brightness of light in accordance with Fog/ Smog environment. Upload the code to microcontroller and test for proper operation

Objective:

- to Rig up the circuit to interface Microcontroller, MQ135 pollution sensor and vary brightness of light in accordance with Fog/Smog environment
- to upload the code to microcontroller and test for proper operation.

Requirements:

- Computer system
- Colour sensor
- LED
- Arduino UNO

PROCEDURE:

INTRODUCTION

Street lighting system is the bulkiest energy outlay for number of city. A smart street lighting system can minimize the municipal street lighting cost as much as 50% to 70%. An intelligent street light system is a system that calibrates the light output on the basis of utilisation and abundance. An accurate street light system intends the installation of the wireless system and overcome the definite energy consumption which will also control the intensity of the street light. The proposed smart street light system is a hardware application in which fog or rain sensor detects the fog and according to it the intensity of light can be controlled through controller. We are using methodology for intensity control is Arduino software and for power consumption MATLAB software is used to show the graphs that how much energy is consumed. Street lightening systems are important facilities for modern city. This system will sort night life more suitable, safe and eco- friendly. Currently, the integrity of the street lights and fog detection has been studied for the road safety and security. Firstly the fog sensor is made which can detect the fog during winter season and when the fog is detected the intensity of street light can be varied depending on the fog density. The variation of intensity of street light can be done by using potentiometer so that led cannot be fused .It is a fully automated system. Automated process is not very easy task to perform but it is very powerful technique .The main purpose of this project to increase the visibility on the roads so that the road accidents can be reduced. This project firstly develops the night or day fog detection using colour sensor and LED street light. The street light

controller should be installed on the pole lights and on the opposite pole one-color sensor and LDR is also installed so that by these two the fog detection process can be done properly so that required amount of led light can be glow on the road. Here the assimilation of new technologies proposing simplicity of preservation and comprise of spatially using independent devices using vibrant grid topology.

IDEA

In this energy efficient street light project cycle, first the requirement of the lightening system in the streets are examined. Then a best technology has been implemented so that a good energy efficient system can designed. The operation and maintenance is required for a system. The values obtained by the mechanism from the sensor can be measured and evaluated from the values that are already given. The values obtained from the sensor during the task performance and then these values are compared with the given values and the errors can be found out. The measured values show the efficiency of system. The projected intellectual LED system of street light for the detection of fog includes ARDUINO software strategy, LED and colour sensor. The inclusive action can be done by using MATLAB software. The data is received by sensor. The graph is shown between time and intensity. The prototype of intelligent LED street light system is developed. Means switching set of rules is intended to make the selecting system that usage of resources commencing the battery, but it correspondingly automatically switches to electricity network resources while battery capability is under the lesser threshold. The resources mechanism medley can be done by using the analog, the digital values and PWM features of the microcontroller. A large no. of serialized pile-up accidents takes place when visibility is low due to fog. The peril of these accidents can be reduced significantly with a system holding visibility sensors and flexible speed-limit. The present system is not very cost effective. So it is required to make a fog sensor which is cost effective at low budget and can install at every street of a country.

Smart City - Identify and test smart Lightening system and its components

Test system architecture of smart lighting and identify Wired DALI, GREEN BUS2, Wireless, Hybrid etc

Objective:

- to test system architecture of smart lighting.
-

Requirement:

- DHALI interface
- Antenna
- Power supply
- Arduino UNO

PROCEDURE:**Introduction**

Public lighting installations are important sources of energy consumption that are affected by certain factors, such as regulation and maintenance. This energy consumption can be reduced considerably by applying new communication and control technologies. The International Energy Agency (IEA) has estimated that the potential energy savings due to lighting technology can be approximately 133 to 212 TWh/y on a global level. The corresponding reductions in CO₂ emissions could equal about 86–137 MT/y for IEA countries. The European Commission carried out a recent study proving that between 30% and 50% of the electricity used for lighting could be saved through research into energy-efficient lighting systems. The use of low-consumption lamps, lighting fixtures and other types of outdoor luminary lighting fixtures is advisable, as is the use of cleaner methods of generating electric energy. This is not just a profitable investment but can also improve the quality of the lighting.

Two-thirds of the public lighting systems currently in use are still based on obsolete and inefficient technologies, with higher energy consumption than needed; therefore, there is great technological potential to renovate public lighting and reduce energy consumption. According to estimates, around 5% of the energy used in lighting is consumed by public lighting, which is the most important component of a city or town's energy consumption.

The minimum acceptable or legal requirements for controlling lighting are different for industry, offices, streets, and public lighting, but the concept is always the same: a fixed lighting intensity must be always achieved with energy efficiency and the lowest possible cost for energy consumption to increase the expected lifespan of the lamps. As a result, by using available lamps, ballasts and control systems, as well as techniques to develop an efficient lighting system, it is possible to produce quality energy in a more efficient and profitable way.

Traditional energy-saving techniques entail totally or partially turning off streetlights, but this implies a reduction in the uniformity of the lighting, which also has a considerable impact on the lifespan of lamps.

Several recommendations exist to reduce the electric energy consumption of public lighting. One of these actions is to control the on-off timespan of the lighting. This period can be controlled using photocell relays or astronomical time relays. The photocells are activated or deactivated depending on the level of environmental light. Their main disadvantage is their low sensitivity to dawn and the presence of clouds, which can cause the relay to be activated too soon. On the other hand, astronomical time relays use a program that contains the sunrise and sunset times in each time zone. A second action is to reduce light output (dimming) during hours with reduced traffic by reducing the voltage applied to each lamp, which reduces electric energy consumption. Lastly, another proposed action is to use high-performance lamps and lighting devices. This consists of installing smart, adjustable electronic ballasts that are able to automatically detect electric failures and lamp problems; measure and send data, such as on the state of the lighting, the lighting level, the energy consumption, voltage, current, and power factor; and receive interruption and lighting control commands.

Regarding the need for uniform lighting, the control systems have two possibilities: light posts with two lamps or with only one fully controllable lamp. In the first case, one of the two lamps can be turned off late at night and during the early morning hours. In the second case, the lamp needs to be connected to a regulation system. The last method has two practical options: dimmable ballasts or centralized control systems as line voltage regulators.

On the other hand, remote management systems also exist that allow the user to control and monitor each lamp individually. When using these systems, the operator can monitor the main parameters of any light from a control room or a mobile unit. The obtained data can be processed, allowing for statistical calculation of the consumption, state of the lamp, voltage, failures, etc. In this way, the average repair time can be reduced. Another interesting parameter could be the arc voltage level, which entails the change from corrective maintenance to predictive maintenance, meaning economic savings in maintenance costs.

To optimize the control, the over-the-air management system needs to support bidirectional communication between the user and the ballast. This management system is implemented using a communication system and a control protocol for the lighting. The communication system can be wired or wireless. The wired communication systems that are normally used include Ethernet-based systems and those based on optical fiber or power-line carrier (PLC). As wireless standards, we can find, among others, GSM/GPRS, RF, Wi-Fi™, IEEE 802.15.4™ and ZigBee™. The latter two are responsible for the increase of wireless sensor networks (WSNs).

Standard IEEE 802.15.4 is aimed at low-rate wireless personal area networks (WPANs). Its objective is to standardize the two lower layers of the communication protocols between open protocols, i.e., the physical layer and the MAC layer. On the other hand, ZigBee defines the upper layers, the network and the application layers. For the lighting-control protocol, it is possible to choose between open protocols, such as TCP/IP, BACNet™, DMX 512™, LONWorks™, X-10™, 0–10 V and DALI™, and proprietary protocols.

From the previously mentioned protocols, the system described in this work uses protocol

IEEE 802.15.4 for wireless communications and the DALI protocol to control the lamps. The

Digital

Addressable Lighting Interface (DALI) is designed to digitally control electronic ballasts and luminaires equipped with this technology [11]. The interface and protocol are defined by standard IEC 60929/EN 60929 Section E.4, and were modified by IEC 62386, which also includes other applications for DALI besides lighting and expands the types of lamps to high-intensity discharge (HID) lamps, halogen lamps, incandescent bulbs, LEDs, etc.

System Architecture

Figure 1 shows the general structure of the tele-management lighting system described in this work. It is a three-level hierarchical model that is widely used in lighting-control systems based on wireless communication. To complete the system, an information and failure location application was developed for mobile devices. This app can be easily used by maintenance technicians.

At the lowest level, directly controlling the luminaire, we can find the wireless nodes, which communicate in a wireless way using IEEE 802.15.4. At the intermediate level, managing the wireless networks of a lighting sector, we can find an industrial computer with 802.15.4 communication that also acts as the network's coordinator. This computer contains an application that enables the comfortable and user-friendly management of a given sector's lighting network. This local computer can also act as a gateway towards central control of the lighting network. The communication with the central control is executed through another communication

network that allows for greater distances, such as 3G/GPRS, WIFI or WIMAX. Lastly, the upper level contains the central control, which groups several sectors of the lighting network.

Figure 1. Architecture of the system.

The local monitoring of a lighting sector is essential, as the central control may be very far away or have obstacles exceeding the physical capacities of the wireless protocol. On the other hand, the local computer for each lighting sector can work in isolation, since it can work as a server database. This characteristic ensures easy deployment and scalability of the telemanagement system. New sectors of the lighting network that are added to the central control can be debugged. Then, a more detailed description of each part will be given.

Network Topology

The 802.15.4 standard is framed within the area of the wireless personal area networks (WPANs), with low consumption and cost. As a result, it is increasingly gaining relevance in the industrial domain. Standard 802.15.4 only defines the physical and media access control layers. A network and application layer has been developed on this standard, as shown by Figure 2.

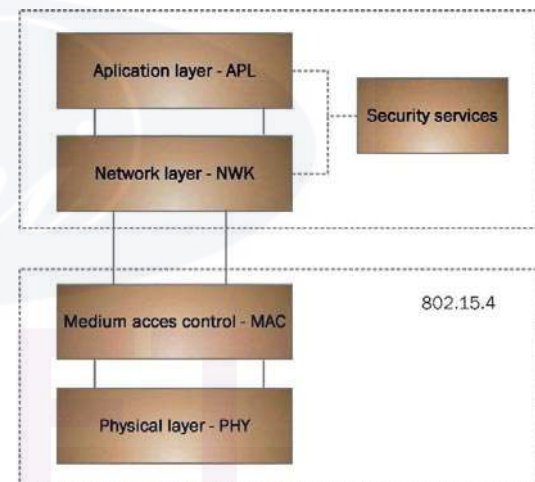


Figure 2. Layers model.

Standard 802.15.4 defines two types of topologies: star and peer-to-peer networks. The peer-to-peer topology allows each node of the network to communicate with any other node, provided that the node is within its range. The star topology allows the establishment of communication among the devices and of a sole central node that acts as the network coordinator. The network coordinator is responsible for starting and finishing the connections, and any communication between two nodes must compulsorily pass through the network coordinator.

The network layer must take the topology of the lighting network into account. On the one hand, each luminaire is controlled by a node with 802.15.4 communication capacities. On the other hand, the luminaires are often deployed along long streets, which results in a linear structure with slight branches. Taking this topology

formed by the luminaires into account, a tree topology created from the peer-to-peer connection possibilities of standard 802.15.4 has been chosen.

In the tree topology, the coordinator is located on the first level of the network, with several end devices and a router. The router forms a new network level consisting of several end devices and a new router, and so on until the desired number of levels required in each case is achieved. The number of leaps allowed by the network layer must be high. This type of structure does not adapt well to any usual topology in wireless networks or to any standard. For example, Zigbee Pro has a limited tree-type number of network layers, with a maximum depth of 15. The number of network hops allowable within ZigBee PRO is increased to 30 from the 10 allowed under ZigBee.

A tree topology with a high depth entails a serious routing problem, as almost all nodes need to have this ability. A novel routing algorithm is proposed for streetlight systems. Additionally, there are few alternatives in the event of an intermediate node failure. The allocation of addresses turns into a complex problem as well. A new address-allocation method that improves ZigBee tree addressing is proposed, which offers some results in simulations. For these reasons, we decided to create our own network layer that could adapt to a tree topology with a high number of levels: the computer that serves as the origin or destination of all exchanged information on the network. That is, we have implemented the communication by using IEEE 802.15.4 Std. and by creating our own (ad-hoc) Network layer (NWK).

Each node has two unique identifiers—the radio module's MAC address and a node identifier

(nodeID)—that allow it to be located by the application, similar to a lamp address. In this way, when a radio module is replaced, only the same nodeID needs to be selected in the node. This nodeID is easily set with a microswitch. The nodeID has a length of 16 bits and can select up to 65,536 ballasts, which increases the number of ballasts that can be routed by a DALI controller. The user application only needs to know the nodeID. The network layer maintains some tables with the MAC addresses where the frames need to be sent to get to their destination. These tables are dynamically created when the network is initialised, when a node is connected, etc.

Lighting Control: DALI

Most current digital ballasts implement the DALI protocol. Additionally, it is also implemented in low-voltage transformers, movement detectors, wall switches, etc.

The main reason for choosing DALI over other lighting control protocols is that it allows for bidirectional communication (unlike X-10, 0-10 V and DMX). On the other hand, its physical interface is very simple, as it uses only two wires for the control signals. Thanks to the use of this protocol, the following advantages are achieved:

- Failure detection for lamps or electric failures
- The ability to take and send data related to the lamp's status
- Level dimming
- Better energy consumption
- Information on the lamp's voltage, current and power factor
- The ability to execute other actions, such as turning the lights on/off, controlling them, etc.

The physical layer of the DALI protocol uses Differential Manchester encoding, in which the logic values are represented by transitions in the signal, as shown by Figure 3. Each node is attached to the bus line via its two bus terminals, which provide differential receiving and transmitting capability.

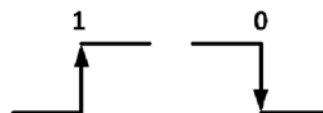


Figure 3. Differential Manchester encoding.

The DALI protocol is based on the master/slave model, in which one master can control up to 64 slaves. The master sends frames (forward messages) to any slave device of the system and receives a response frame (backward message) if the forward frame requested a response. As shown by Figure 4, a forward master–slave frame (ballast) consists of 19 bits, distributed as follows: one starting bit, eight bits for routing, another eight bits for data and two stop bits. Meanwhile, a backward response frame consists of 11 bits, a start bit, 8 bits for data and two stop bits. At the physical layer, DALI uses an effective data transfer rate of 1.2 kb/s and a tolerance in the timing bit of 10%.

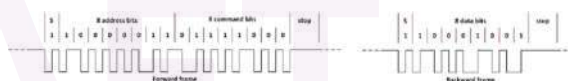


Figure 4. Format of the DALI frames.

The main disadvantage of the DALI protocol is that it is not implemented in most microcontrollers. In addition, the logic levels of the physical layer have incompatible voltage values with most microcontroller-based systems.

3.3. Wireless Node

An important aspect of developing streetlight control systems is deciding which components to develop and which ones to purchase from the market. To make this decision, it is important to study whether these components exist in the market, their price and the economic impact on the project, among other factors. In the presented system, we have to pay special attention to the wireless node (mote), as a mote needs to be installed in each luminaire. As a result, a high number of motes are needed. All of them show the following disadvantages in our system:

- They do not have a DALI interface.

- They are not properly powered. The input voltage is 230 VAC; hence, the voltages for the processing part and for the DALI interface must be generated, as explained below.
- The nodeID cannot be configured in an easy way by maintenance technicians if a mote needs to be replaced.
- No robust connectors are available.

We decided to design our own mote, since if we had opted to use commercial mote, we would have needed to design an adaptation board to solve the aforementioned disadvantages. The designed mote has two different parts: the processing and radio module, and the rest of interfaces, including the DALI interface and the power supply block. The reason for this division is to try different radio modules, so that both parts can later be integrated into just one board. In this way, we would

obtain a specific mote to control the ballasts at a low economic cost. The designed mote will be described in detail in Section 4.

The decision to design a specific mote is a usual process in many works related to wireless sensor networks. In works such as [10] motes were developed for specific applications to minimize total consumption. A wireless sensor mote was designed for precision horticulture applications. In other works, some motes were designed to search for possibilities not offered by commercial motes. For example, a mote was designed to support remote hardware modification. Lastly, regarding wireless nodes to control LED lamps in over-the-air lighting-control systems, motes using the XBee module of the Digi MaxStream radio frequency were designed. The Wasmote module was improved by connecting sensors to it and including control over LED lamps.



Smart City - Identify and test smart Lightening system and its components**Install, Test and troubleshooting of smart light****Objective:**

- to install, Test and troubleshooting of smart light.

Requirements:

- A Philips head screw driver
- A flat head screw driver
- A smart light switch that fits your wall switch and wiring setup (more on this later)
- Wire cutters
- Wire connectors
- A flashlight or battery-operated lantern
- Electrical tape

PROCEDURE:

The day when light fixtures were either incandescent or fluorescent is a thing of the past. Today the homeowner will also encounter LED -- light-emitting diode -- fixtures. Fortunately, the troubleshooting procedures are basically the same for all three types. The good news is that all you will need is a noncontact voltage tester and a digital Multimeter. If you need to buy a digital Multimeter, buy an auto-ranging meter instead of a manual-ranging meter.

The Continuity Test

The basic test used when troubleshooting any light fixture is the continuity test. A continuity test determines when an electrical circuit is good or bad, continuous or broken. The first test to be performed, before testing the fixture itself, is to determine whether or not there is voltage available at the lighting outlet box. A voltage check can be performed with either the AC voltage function on your digital Multimeter or with the noncontact voltage tester. The noncontact voltage tester is recommended for the homeowner because it does not require you to actually touch the probe to live, current-carrying wires; all you have to do is bring it near the light fixture when the light switch is on.

Troubleshooting Incandescent Fixtures

Turn off the circuit breaker for that branch circuit at the service panel. Check the circuit again with the voltage tester to make sure that you have turned off the right circuit breaker. Lower the light fixture from the outlet box, and then check to see that there is a good connection between the fixture wires and the branch circuit wires. Those connections are made using plastic, screw-on wire nuts and can slip out of them if installed improperly. Disconnect the fixture wires if you find the connections were good. You have to disconnect the wires so you can check the continuity.

With the meter set on the ohms function, check for continuity between the white fixture wire and the metal shell inside the light socket. If there is continuity, the meter will display "0.000" on its LCD. If the socket is bad, the LCD will display "O.L." Check for continuity between the black fixture wire and the brass button in the lamp socket. The LCD will either display "0.000" or "O.L." If the socket checks out OK but the light still does not work, the brass contact of the socket is not making contact with the base of the bulb and needs to be pried away from the base of the socket.

Checking Fluorescent Fixtures

When it comes to troubleshooting fluorescent lights, your senses of sight, hearing and will go a long way to helping you locate the problem. If there is voltage present but the light fails to come on, first check to see if the light bulbs are burned black at their ends. If they have blackened ends, replace the bulbs and your problem will most likely be solved. If the light flickers but does not come on, the problem again may just be bad bulbs. Usually the flickering will be combined with blackened ends. A buzzing sound and/or a hot, tarry smell is indicative of a bad ballast that requires replacement. Bad contact between the pins on the light bulbs and the brass strips in their sockets is not a common problem, but it can happen. With bad sockets, replace them; do not try to fix them.

Troubleshooting LED Lights

Troubleshooting LED lamps is the same as for incandescent lights unless they are LED rope lights. With rope lights, if there is voltage present and the circuit connections are good, the only thing you can do is replace them.

Bad Circuit Neutral

A broken neutral wire somewhere between the lighting outlet box and the neutral bar in the service panel will also keep the light from operating. The noncontact voltage tester will still indicate the presence of voltage even if the neutral is bad. To determine if you have a neutral problem, you will have to use your digital multimeter. Set the meter's function switch to AC volts, turn on the circuit breaker and touch the probes to the circuit wires.

Smart City - Identify, select, install and troubleshoot different module/ devices used in smart street light based on IoT and cloud Technology

Install and test solar street light

Objective:

- to install and test solar street light.

Requirement:

- Lamp
- Lamp pole
- Battery
- Micro controller

PROCEDURE:

SunMaster solar street lights are easy to install. You will simply need common hand tools to assemble the light. For bigger projects, however, we can provide onsite training and technical assistance. For these multiple installations, hiring technicians with professional certificates, specialising in street lamp installation is recommended.

What will you need to install a solar street light?

Before you start a street light installation, prepare the following tools:

- 20m start-up hoist
- Inner hexagon wrench
- Monkey wrench

What do you need to prepare before installation?

Before the start of any installation, make sure to do the following steps.

1. Surround the construction area with guard lines. Only installation specialists and personnel must be allowed in the construction area.
2. Read the installation manual and familiarize yourself with the installation procedures, the circuit diagrams, and the safety measures you must take before you proceed with the installation.
3. Make sure that you have the complete solar street lights kit. Check that you have the following parts:
 - Solar Panel
 - Kickstand (Solar Bracket)
 - Light Fixture
 - Lamp Pole
 - Access Panel
 - Flange
 - Basis Cage
 - Battery and Battery Box

- Controller

Installation Procedures for a Solar Street Lamp

1. Installation of the Foundation

To install the foundation of your solar street light, choose a level and flat ground, with no inclination. Screw and secure the Basis Cage to the ground using the four screws. One side of the Basis Cage should be parallel to the edge of the road. Secure the Basis Cage with concrete, and this will serve as the foundation of your street lamp.

Secure the Basis Cage with concrete, and this will serve as the foundation of your street lamp.

2. Installation of the Battery and Battery Box

Place the battery beside the Basis Cage, underground, at about 600 mm. Follow the electrical circuit diagram on how the battery will be connected.

3. Setup the Street Lamp

- A. Lift the lamp pole off the ground for 1 meter, with the use of a supporting bench.
- B. Install and secure the Solar Bracket at the top of the lamp pole.

Place the Solar Panel on the Solar Bracket and secure it with bolts. Solar Panels should be handled with care. Follow the electrical circuit diagrams on how the Solar Panels will be connected to the battery, the controller, and the street light.

- C. Install the Light Fixture on the lamp pole. Place the Light Fixture on the ground, with the back part facing upward. Press the red buttons on both sides of the Light Fixture. Turn the Light Fixture around, and screw the plastic black lamp holder. Place the LED light bulb in the holder. If electronic ballast is needed, place it inside the light fixture. Follow the circuit diagram when you connect the light fixture to the controller. Connect and secure the light fixture to the lamp pole.
- D. Check installation and connections to confirm that everything is in the right place. Make sure that wires and components are connected according to the electrical circuit diagram.

- E. Use a crane to erect the lamp pole. Ensure that the lamp pole is upright on the bottom plate. Fix and secure it with bolts.
- F. Place the controller inside the lamp pole, with the correct wires and leads properly connected to the

controller. Check that the erected street light is working and behaving as expected.

Congratulations! You have now successfully erected and installed Your Solar Street Light. Make sure to follow the instructions and safety guidelines to ensure a successful installation.



Smart City - Identify, select, install and troubleshoot different module/ devices used in smart street light based on IoT and cloud Technology**Execute testing of sensors used in street light like dusk to dawn, temperature sensor****Objective:**

- to execute testing of sensors used in street light like dusk to dawn, temperature sensor.

Requirements:

- LDR
- LED
- LCD
- Arduino UNO
- Internet connectivity
- Router

PROCEDURE:**Introduction**

The street light system is one of the major elements in every cities and small towns. There is nothing to explain the importance of street light, but the present system is having number of drawbacks. Primarily, most of the street light systems are manually operated which have to be activated by human being either locally or from a centralized point. At the same time, they do not have any feedback system through which it could be identify the functioning and nonfunctioning street light unit.

A system will be highly beneficial if the street light can be controlled from a centralized location remotely without wire connectivity. The wireless connectivity is again distinctly useful if the street lights are powered by battery back-up and solar energy. In case of street lights in many places conventional street lights are replaced by Automatic Street light which is based on light intensity sensor. This is a noble way to minimize the wastage of power, manpower as well as increase the life of the light unit (indirectly). But these systems are not having in built monitoring system i.e. whether the light is actually working or not. It is a very common phenomenon where we have found that street light is installed but many of the units are not in working condition, it is due to lack of proper maintenance and ignorance about the faulty system. A system with self-monitoring system and remote controlling can be improved a street light system up to an extent.

INTERNET OF THINGS (IoT)

The IoT allows objects to be sensed or controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention. When IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyberphysical systems, which also encompasses technologies such as smart grids, virtual power plants, smart homes, intelligent transportation and smart cities.

"Things", in the IoT sense, can refer to a wide variety of devices such as heart monitoring implants, biochip transponders on farm animals, cameras streaming live feeds of wild animals in coastal waters, automobiles with built-in sensors, DNA analysis devices for environmental/food/pathogen monitoring, or field operation devices that assist firefighters in search and rescue operations. Legal scholars suggest regarding "things" as an "inextricable mixture of hardware, software, data and service".

SMART STREET LIGHT

Smart street light refers to public street lighting that adapts to movement by pedestrians, cyclists and cars. Intelligent street lighting also referred to as adaptive street lighting, dims when no activity is detected, but brightens when movement is detected. This type of lighting is different from traditional, stationary illumination, or dimmable street lighting that dims at pre-determined times. Smart street light system tries to find solution for the faster depletion of energy resources due to the inefficient usage and wastage of these resources. Increasing electricity bill is something that can be witnessed by these practices. This paper helps to decrease the wastage of electricity by controlling the working of street light system that attributes to a good amount of electricity. Street lights can be made intelligent by placing cameras or other sensors on them, which enables them to detect. The other advantage of LED is that the intensity can be controlled easily.

Smart City - Identify, select, install and troubleshoot different module/ devices used in smart street light based on IoT and cloud Technology

Check solar battery management system

Objective:

- to check the solar battery management system.

Requirement:

- IV Sensor
- PV Panel
- Battery
- LCD
- Microcontroller
- Photovoltaic system

PROCEDURE:

System Description:

Photovoltaic System

Photovoltaic is the area of technology and research related to the use of solar cells in order to convert sun energy directly into electrical energy. Due to the increasing demand for clean energy sources, the production of solar cells and PV arrays has increased significantly in recent years. Solar cells generate dc current from light, where it is used to power devices to recharge a battery.

Photovoltaic Arrangements

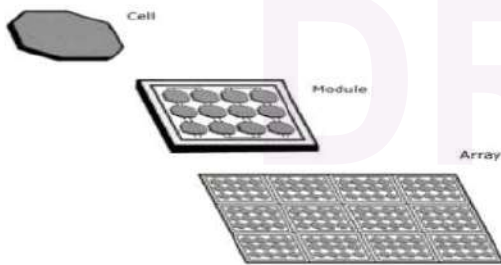


Fig: 1 Photovoltaic system

Fig.1 shows basic arrangements of Photovoltaic system such as cell, module, and array. The basic ingredients of PV cells are semiconductor materials, such as silicon. For solar cells, a thin semiconductor wafer creates an electric field, on one side positive and negative on the other. A PV cell can either be circular or square in construction. Because of the low voltage generation in a PV cell (around 0.5V), several PV cells are connected in series (for high voltage) and in parallel (for high current)

to form a PV module for desired output. A photovoltaic array is an interconnection of modules which is made up of many PV cells in series/parallel. The energy produced by single module is not enough to meet the requirements of commercial applications, so modules are connected to form array to supply load. In an array the connection of the modules is same in a module.

Solar Tracking System

A solar tracker is an electro-mechanical system used for a solar photovoltaic panel in the direction of the sun Light. it is also dependent on the angle of incidence of the sunlight providing power to the solar cell's surface. The solar cell applications require a high accuracy in order to concentrated sunlight precisely to the powered device. Solar trackers may be single axis or dual axis which use motors for responding to the solar direction.

Modelling of PV Panel

The photovoltaic system can generate direct current electricity without environmental impact when is exposed to sunlight. The basic building block of PV arrays is the solar cell, which is basically a p-n junction that directly converts light energy into electricity. The output characteristic of PV module depends on the cell temperature, solar irradiation, and output voltage of the module. The Fig.2 shows the equivalent circuit of a PV array with a load.

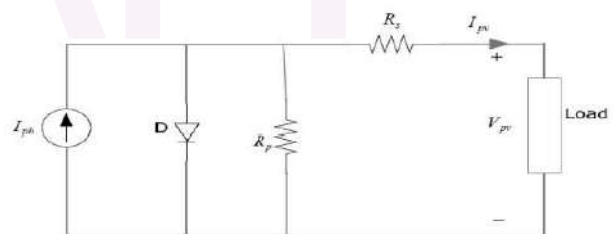


Fig: 2 Equivalent Circuit of PV panel

Usually the equivalent circuit of a general PV model consists of a photocurrent, a diode, a parallel resistor which expresses a leakage current, and a series resistor which describes an internal resistance to the current flow.

Smart City - Identify, select, install and troubleshoot different module/ devices used in smart street light based on IoT and cloud Technology**Check solar street light components****Objective:**

- to check solar street light components.

Requirements:

- LED
- LCD
- Controller
- Photovoltaic cell
- Solar cells

PROCEDURE:**Introduction**

Solar street lights are effective and efficient light sources in which power is fed with the help of Photo-voltaic Panels. They are generally mounted on the lighting structure. There is a Rechargeable battery, which is charged by photo voltaic panels. Then the charge of that battery is used to powers a fluorescent or Led Lamp during the night. There have sensors, through them solar panels turn on and turn off automatically by sensing outdoor light with the help of light source. They are designed to work at night.

Advantages

- The operation cost is not so high.
- This is a pollution free source of producing Electricity.
- NO external wires are used. so chances of occurrence of accidents are minimized.
- The parts of of this system are easy to carried. so remote access is applicable.
- They require less maintenance.

Disadvantages

- The starting setup cost is high.
- Risk of theft is also higher.
- Rechargeable batteries are required to be changed many times.
- Not works in cloudy and rainy days.
- Snow and moisture effects its working.

Uses

They are the used for lightning at nights.



Solar street Lights

Smart City - Identify, select, install and troubleshoot different module/ devices used in smart street light based on IoT and cloud Technology

Test LED used on solar street light

Objective:

- to test LED used on solar street light.

Requirements:

- Solar Panel
- Solar Charging controller
- LED Lamp
- Rechargeable battery
- Controller
- Pole
- Interconnecting cables

PROCEDURE:

- **Solar Panels:** Solar panel is known as Photovoltaic cell which uses light energy from the Sun to convert solar energy into electricity. This electricity can be used to run many applications.
- **LED light:** LED light which is used in solar street lights, emits brighter light with low energy consumption. A single LED will emit less light than compact fluorescent or incandescent. But a set of LEDs give brighter light than other lamps.
- **Rechargeable battery:** These batteries are used for storing electricity received by the solar panel during daytime, and it provides electricity at night.
- **Controller:** Controller is used to control when to switch on and switch off the street lights. It also controls battery from undercharging and overcharging. It consists of MCU, secondary power supply and battery charger, etc.
- **Pole:** For every street light, 12 Feet pole is required so that some components like batteries, fixtures, and panels can be mounted.
- **Interconnecting cables:** Battery box, LED light, and the solar panel are mounted on the top of pole and they are connected through interconnected cables. A PV module is connected with a controller and it is connected with battery and LED lamps. Size of the cable depends on current, how much it is used and the pole size.

Top 5 Applications of Street Lights and its Uses**1. Airport Lighting**

Many Airports are equipped with solar lights because they are easy to install with no conventional electricity cost. These LED lights are installed at the runway and

are portable. On the other hand, the maintenance is needed only once in 7 years. These lights are used for safe Aircraft operations at night time.

2. Highway and roadway lighting

The Solar LED street lights are also very easy to install. By the help of these lights, the risk of accidents is minimized as there are no external wires needed for electric connections. These lights can be installed easily in the remote roadway or highway where there is no electricity. The LED Streetlight provides clear visibility for the pedestrian and drivers. High-quality illumination LED light system beautifies urban area at night without polluting the environment.

3. Park and Playground lighting

Solar lighting systems are beneficial for park and playground lighting needs as it helps to light signs, trails, pavilions and parking lot. They are independent of the grid, so they can be installed anywhere in the park where there is a need for proper lighting. These lights provide safety for children who play after dark and also to the officials in order to secure the area during the night. These days, most of the cricket or football matches are held at night time due to several reasons. So Solar LED lights are installed in the stadium or playgrounds that can provide high power lighting inside the field. For any sport, Playground needs high power lighting, called floodlight which can be achieved by the solar panel at low cost and minimal maintenance.

4. Industrial lighting and commercial lighting

The Solar LED lighting system has exceptional storage system which allows 3 days continuous illumination without any recharge. Industrial areas are bigger in size; therefore, they take a large amount of electricity for outdoor lighting. While installing these solar lights, the cost for electric wires will reduce and maintenance cost will lower. Have a look at some of the Commercial Solar Lighting Services we offer.

5. Outdoor security lighting

Solar powered security lights are illuminated by solar energy which is converted into electric energy. As a result, these are a practical choice for security purpose, as without electricity they can operate at night because the sun provides needed energy during the day. These lights automatically turn on when it senses any motion. They also operate during the cloudy weather conditions and

provide reliable lighting anytime. Outdoor lights are very successful in rural areas and Intelizon's Zonstreet has been satisfyingly deployed it at Rural Maharashtra.

There are many areas such as parking lot, Hospital parking, and residential lighting where these solar-powered street lights are used. These Solar LED street

lights require less system for installation and also provide high power lighting when compared to conventional lighting. It also offers shelf-life and minimal maintenance. It provides portability facility and comes with 2 years of warranty. These Solar Street lights are Pollution free lights. Only the cost of the initial investment is high when compared to the conventional lights. And only dust, snow, and moisture can hinder the energy production. These lights are not a practical option for every application because the cost is on a higher side.



Solar street light with solar powered wifi camera is the newest developed edition of the solar street light system.

The solar street light with solar powered wifi camera has the following advantages:



Easiness to Install
Thanks to the wireless technology.



Smart Movement Detection
It detects the movement within the unit range.



Self-Optimized
It automatically regulates the brightness of the LED lamp depending on the motion detection.



High Definition Images
It captures high quality images & videos hence increases the reliability.



Durability
The HD Camera has an updated design fabricated from an aluminum extrusion which improves the LED thermal heat dissipation and longevity



Long Life
Thanks to the new generation Lithium Iron (LiFePO4) batteries, the system performance is increased. LiFePO4 batteries have a storage capacity three times more as compared to conventional lead-acid ones



Self-sufficient
The system does not require connection with the grid and can work anywhere, thanks to the solar panels.



Internet of things
Thanks to the **motion detection system, SMS control extension and live video & audio streaming**, the IoT facilitate connecting devices and applications to each other and enhancing their functionality.



Energy Savings
Provided with an **Infrared ray induction probe** so you can **save energy**.



Plug and play
The system does not need wiring or cable from the grid. It can be installed easily on any pole or wall.



Flexibility
Provided with an independent system for ensuring **night and day vision**. It gives the surveillance a whole new meaning.

We think about your **security**! Here are the features of the **solar-powered WI-FI Camera**:



DRAFT

Smart City - Identify, select, install and troubleshoot different module/ devices used in smart street light based on IoT and cloud Technology

Apply smart embedded system that controls the street light based on detection of sunlight

Objective:

- to apply smart embedded system that controls the street light based on detection of sunlight.

Requirements:

- PIR Sensor
- IR sensor
- LED
- LCD
- Microcontroller
- Raspberry pi

PROCEDURE:**Introduction**

We use intelligent instruments in every part of our lives. This project is on IoT Based Intelligent Street light Monitoring System. The street light is one of the huge expenses in a city. Currently a manual system is used where the light will be made to switched ON/OFF i.e., the light will made to be switched ON in the evening and switched OFF in the morning. Hence there is a lot of energy wastage between the ON/OFF. The main purpose of designing this project is to prevent loss of energy unnecessarily and to reduce maintenance costs.

Today, road lighting ordinarily utilizes high-force release lights, regularly HPS (High Pressure Sodium lamps). Globally 70% of all power is produced by consuming non-renewable energy sources, a wellspring of air contamination and ozone depleting substances, and furthermore comprehensively there are around 300 million road lights utilizing that electricity. Almost those lights are controlled manually. The taken a toll spent is huge to the point that all the sodium vapor lights devour more power.

The inspiration for this project was obtained from the shortage of power in an ever increasing power demand scenario in India. Relevant questions such as, "What can be done in order to reduce the deficit of power?" "Is setting up of more power plants the only solution for the problem?" "Do DSM and other power management techniques provide lasting results?" "Or is it sensible and feasible to set up an autonomous system that can manage its own power consumption?" cropped up. Though there are several power management techniques such as smart grid technology, home automation technology, that are in place to reduce wastage of power, it was intriguing to find out that there is no such system in place for street lights which are switched on for almost

12hrs or more daily. Though there is an automatic switching facility to switch on the street lights at dusk and switch off at dawn, it is more of a convenience than an effort to reduce power wastage.

The substantial scope of this new system provides dimming and on-site status check system is a perfect solution for total global electricity conservation. Intelligent Street light control system can switch lights reasonably, regulates voltage according to degrees of shine and runs in lower voltage in night .As a result, it saves electricity costs largely, prolongs the service life of street lamps and equipments, and saves labour costs of maintenance and materials significantly. Street light control and management system is made up of the network systems to report the status of the system.

METHODOLOGY

The reason for this task is to create road lights that will sparkle naturally when human and vehicle developments are detected. Road light are exchanged on relying upon the force of the daylight. It will shine naturally when there is activity. This framework controls the road lights utilizing light ward resistor, PIR sensor and IR sensors. Road lights are exchanged on relying upon the power of the Sun light on LDR. The controller checks top time amid which there is no movement and turn OFF the lights. At the point when there is any vehicle or human out and about, it is identified by the PIR sensor and IR sensors. At whatever point people or vehicles are detected by the PIR sensor inside its range it shows the Raspberry pi 3 to switch the comparing the road light. At that point lights are exchanged on for 2 to 3 minutes and turned off consequently. Additionally, at whatever point people or vehicles cut the infra-red sensors, it demonstrates the Raspberry pi 3 to switch the relating road light. At that point lights are exchanged on for 2 to 3 minutes and turned off automatically. The light will be switches on till the vehicle cuts the following IR sensors.

- Sensor block
- Control block
- Lighting block

Sensor block

Sensor block is a block consisting of a PIR motion sensor. A PIR-based motion detector is used to sense movement of people, animals, or other objects. They are commonly

used in burglar alarms and automatically-activated lighting systems. They are commonly called simply "PIR", or sometimes "PID", for "passive infra red detector".

Control block

The control block consists of Raspberry pi 3 unit. Raspberry pi 3 interprets the inputs from the sensor and sends out control signals to the LED driver in the Lighting Block.

Lighting block

Lighting piece comprises of a variety of LEDs for lighting, and a LED driver. A light-transmitting diode (LED) is a two-lead semiconductor light source. It is a p– n intersection diode, which discharges light when actuated. At the point when a reasonable voltage is connected to the leads, electrons can recombine with electron openings inside the gadget, discharging vitality as photons. This impact is called electro iridescence, and the shade of the light (comparing to the vitality of the photon) is controlled by the vitality band hole of the semiconductor.



- to configure and communicate 3 phase Modbus energy meter with IoT based smart streetlight controller.

- LED
- LCD
- Controller
- Street light controller

```

1  #include <cs50.h>
2  #include <ctype.h>
3  #include <stdio.h>
4  #include <string.h>
5  #include <stdlib.h>
6  #include <math.h>
7
8  int main(void)
9  {
10     // Prompt user for name
11     string name = get_string("Name: ");
12
13     // Compute length of name
14     int length = strlen(name);
15
16     // Compute number of vowels
17     int vowels = 0;
18
19     for (int i = 0; i < length; i++)
20     {
21         if (isalpha(name[i]) && (name[i] == 'a' || name[i] == 'e' || name[i] == 'i' || name[i] == 'o' || name[i] == 'u'))
22         {
23             vowels++;
24         }
25     }
26
27     // Print number of vowels
28     printf("Number of vowels: %i\n", vowels);
29 }

```

Step 4 Watch the expected result at cloud MQTT Server



The screenshot shows the AWS IoT Core console interface. The top navigation bar includes the AWS logo, a search bar, and several status icons. The main content area is titled 'Websocket' and displays a message received from the endpoint. The message content is: 'Frequency:10.00;Voltage:125.50;PowerFactor:1.00'. The console also shows a 'Send message' input field and a 'Received messages' list with the received message details.

Test Setup

[illegible]

Smart City - Identify, select, install and troubleshoot different module/ devices used in SMART Parking

Observe over voltage protection and over current protection

Objective:

- to observe over voltage protection and over current protection.

Requirement:

- Wires
- Nodes
- Ethernet
- Computer System
- Inverter

PROCEDURE:

Introduction

The purpose of this Technical Note is to describe proper protection of SolarEdge products in the field from overvoltage surges caused by lightning strikes, grid overvoltage events and ground faults. Properly installed surge protection can reduce the likelihood of permanent damage to inverter components, Control and Communication Gateways (CCGs), communication devices and interconnected meters.

Rapidly changing currents create electromagnetic pulses (EMPs) that radiate and create current and voltage “surge” when passing over conductive elements like electrical wires, communication lines, or metallic pipes. These surges may result in the destruction of delicate electronics and semiconductors. Devices known as surge protectors (SPD) or transient voltage surge

suppressors (TVSS) connected to these conductors can route these transient currents to the ground, protecting the equipment from damage.

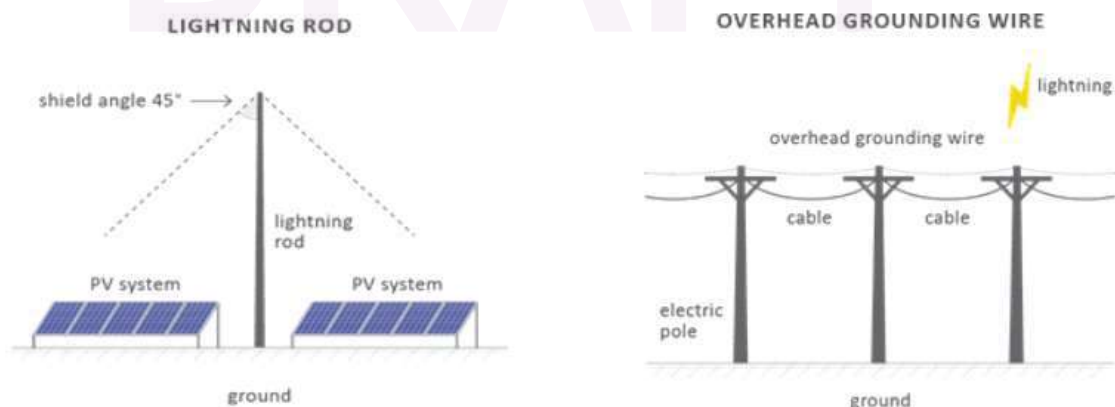
SolarEdge recommends that all three phase inverters be installed with external surge protection devices (SPDs) on the AC, RS485, and Ethernet lines to mitigate the effect of environmental factors which are beyond SolarEdge’s control and which fall outside of SolarEdge’s limited product warranty (<http://solaredge.com/groups/service/warranty>).

Lightning Strikes and Electromagnetic Pulses

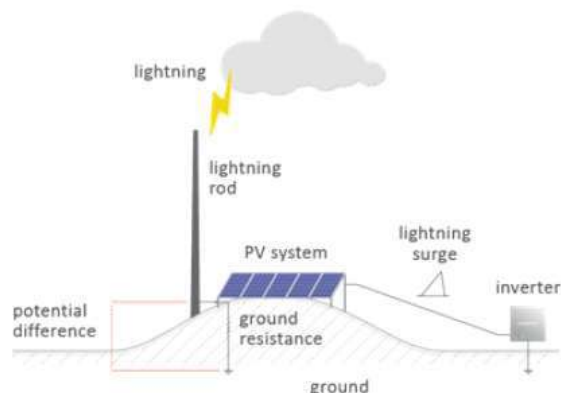
One of the common sources of voltage surge is lightning strikes. It is not necessary for lightning to strike the PV site to damage it; therefore, it is worthwhile to consider all the ways in which lightning can induce surge, including electrostatic and magnetic induction.

Direct Lightning Strike

Direct lightning strike energy is enormous, and a surge protector alone cannot protect the stricken instruments. External protection is required to attract the lightning and redirect it to the ground, during which SPDs only absorb residual energy. External protection equipment includes lightning rods, grounding wires, catching devices and conductors, as well as the accompanying ground system.

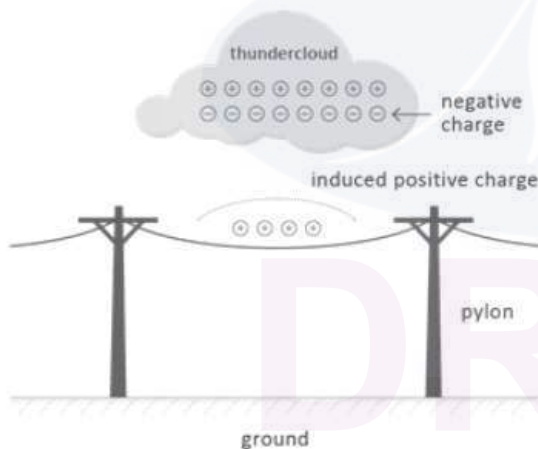


One effect of a direct lightning strike is increased ground potential – when the lightning strikes a building or a lightning rod, high current flows to the ground and the ground potential rises. This creates a potential difference between ground and external conductors, leading to surge behavior.

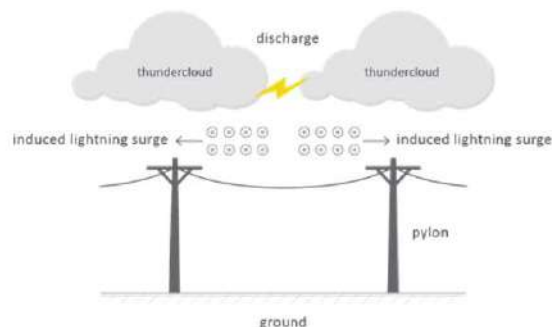


Electrostatic Induction

Thunderclouds contain negative charges in their lower sections. These high negative charges can induce high positive charges within nearby cables, especially power lines and communication cables.

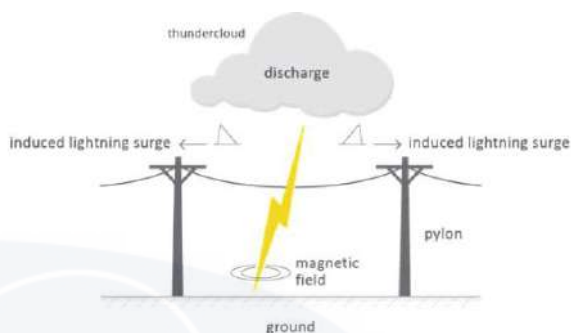


During discharge of the thundercloud (lightning), the positive charge accumulated in the cable is released, resulting in a surge in the cable in both directions.



Electromagnetic Induction (Indirect Lightning)

A discharge between clouds and the ground generates a surging magnetic wave. When the magnetic wave reaches AC lines or communication cables, it induces a voltage surge.



What is a Surge Protection Device?

In order to avoid high voltage damage to a PV system, voltage surges should have a path to ground to avoid high energy from passing through electronics. In order to provide this path, all of the wiring exiting and entering the system should be coupled to ground through a Surge Protection Device (SPD), and all conductive surfaces should be directly grounded. Examples of lines exiting and entering the system include the AC mains and communication lines, such as Ethernet cables, RS485 cabling and telephone lines. Note that SPDs on power lines provide protection that is different from circuit breakers; breakers protect equipment from overcurrent while SPDs protect equipment from over voltage. SPDs are usually a combination of Metal Oxide Varistors (MOVs), Gas Discharge Tubes (GDT) and/or Zener diodes, and current limiting devices that act to shunt charge to ground and to divert it from entering the protected system in the event of high voltage or current surges. Both MOVs and GDTs have a limited lifetime, and can handle a finite number of surge events.

Smart City - Identify, select, install and troubleshoot different module/ devices used in SMART Parking

Monitor smart streetlight management system having with map view based dash board and individual system details

Objective:

- to monitor smart streetlight management system having with map view based dash board and individual system details.

Requirements:

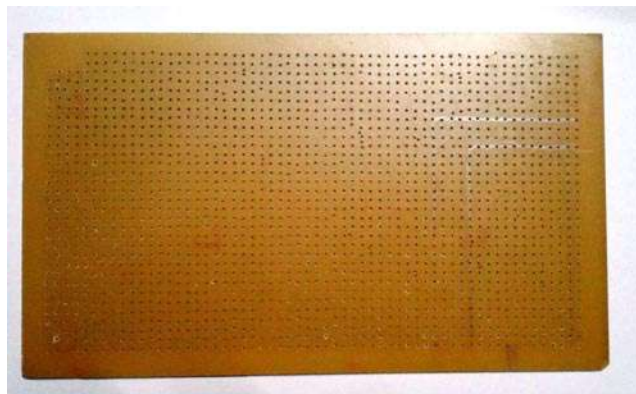
- Sensors and Actuators
- LED
- LCD
- Arduino
- Microcontrollers

PROCEDURE:

The home lighting system enhances the beauty of our home like never before. These systems are coming with LED's because they are Eco-friendly and power saving. The interesting thing is that many of the users are interested to have IoT kind of home lighting systems because they help-a-lot. The IoT my home lighting device designed with ESP8266 and Arduino with remote mobile app designed using Cayenne my devices will be helpful in various purposes. This device can fit and serve in any kind of place like for homes, shops, offices, schools, colleges, hospitals, function halls, meeting halls, etc. Cayenne app designing and controlling some actuators and reading data from sensors using arduino boards it is easy to access the IoT my home lighting devices remotely and it can be more helpful to every household and office purposes.

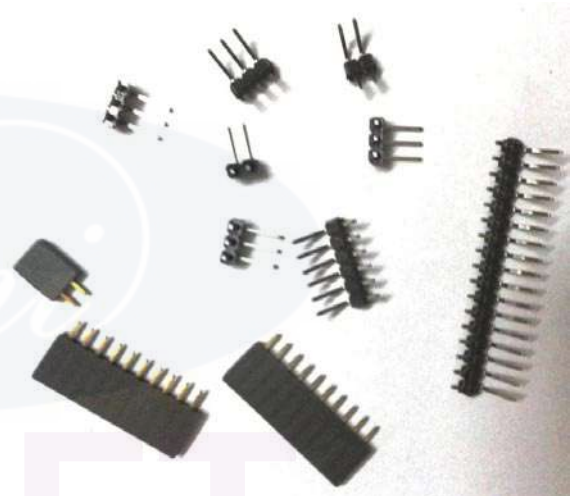
Printed Circuit Board (PCB) Selection

I am using Dot Printed Circuit Board (PCB) because it is inexpensive and to beginners it is easy to design for their home lighting device. I will cut a small portion on it marked with pencil at Right hand side corner shown in Fig 1.



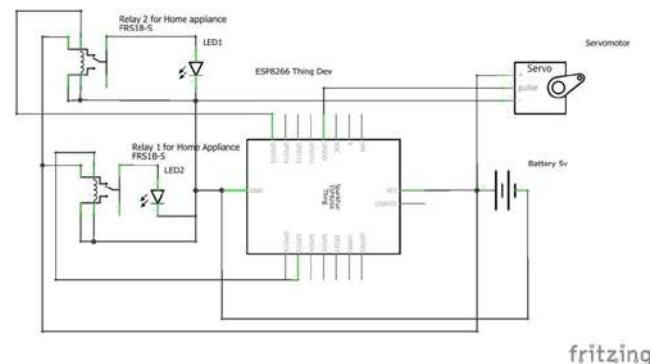
Male and Female Header Connector Pins

I have chosen the Male and Female Header connector pins to insert Sensors, Actuators and ESP8266 MCU for plug and replace which ever components are necessary in future. Fig. 2 show the male and female header connector pins use to interface relays, motors, solenoids, etc to read/write, enable/disable through port pins of MCU.



Schematic Circuit Diagram

PCB, bread board and schematic connection are shown in the following figure for IOT home lighting device.



Smart City - Identify, select, install and troubleshoot different module/ devices used in SMART Parking

Install LED display board

Objective:

- to install LED display board.

Requirements:

- LED
- LCD
- Screwdrivers, Philips and flat blade
- Wire cutters
- Wire strippers
- Soldering Iron & solder, preferably with 'No Clean' flux
- HDMI monitor
- mini HDMI adapter
- keyboard
- micro USB adapter
- 5V power supply
- SD Card Imager,
- Putty
- WinSCP

PROCEDURE:

RGB LED matrices come in a variety of pixel sizes, pixel densities and interface configurations. Pixel size refers to the number of rows and columns of RGB LEDs. Pixel density refers to how close the RGB LEDs are to each other. The pixel density of a display is stated by a 'Pn' designation. P4 refers to a display with 4mm spacing between the RGB LEDs. Lower 'P' numbers are higher density. The lower the 'P' number, the closer you can view the display. If you are too close then your eye will not blend the pixels properly. For this project you will need an RGB LED matrix with a Hub75 (the most common) interface. The Hub75 interface defines the connector used, the signals used and the wiring. I used a 32x64 pixel matrix with 4mm density (P4-64x32-16S). The matrix has 32 rows with 64 columns of RGB LEDs. This type of matrix can be purchased from Adafruit or on Ebay. The driver library used can handle longer displays (more columns or fewer columns) and can drive up to three displays. I chose to limit it to one display with a maximum of 32 rows.

The driver for the RGB LED Matrix drives two rows of RGB LEDs at a time. All of the rows are time multiplexed to yield the display. This means that at any given point in time, only two rows of LEDs are drawing current. Each

pixel consists of three LEDs. One red, one green and one blue. Each individual LED in a pixel draws approximately 20mA when lit up. So each pixel will draw around 60mA if all three LEDs are turned on. Multiply that by the number of pixels in a single row (64 in my case) and you get a maximum current per row of 3840 mA or 3.84 Amps. Multiply that by two because two rows are driven at the same time and the maximum current jumps to 7.68 Amps. Bottom line, if every LED on the display is being told to be on then it will draw in the neighborhood of eight Amps. This calculation does not take into consideration the power requirements of the Raspberry Pi used. An RPiW needs about 180mA when using WiFi, So an 8 Amp @ 5V power supply will suffice. A full sized Raspberry Pi 2 or 3 requires around 2.5Amps. You would need a 10Amp power supply for a full sized Raspberry Pi. Keep in mind that this represents a worst case scenario. For a scrolling sign most of the LEDs are turned off so you should be able to use a lower rated supply.

For the Adafruit Hub75 HAT, all you need to do is solder a few connectors in place. If you decide to build one of my boards then you will need to be skilled at hand soldering surface mount components. Or you can decide to use the schematic of my board to hand wire a Hub75 interface board. I highly recommend using someone's Hub75 interface board because the RGB matrices are 5V and the RPi is 3.3V. Adafruit's board and my board use level shifters to convert the RPi's 3.3V signals to 5V for the matrix.

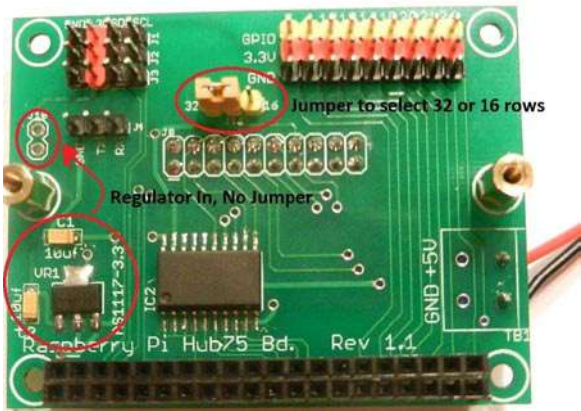
Configuring the RPi for WiFi, auto boot into the Python script, installing the matrix driver library and installing the Python libraries used can be a daunting task. I have laid out step by step instructions for doing this that should make it fairly easy to do. This project assumes that you are familiar with using Putty and WinSCP or equivalent programs and working with the Linux command line.

Step 1: What You Will Need to Build One:

- A Raspberry Pi Zero W.
- If you are using another Raspberry Pi then you will also need;
- A USB WiFi dongle compatible with the Raspberry Pi.
- if using a Raspberry Pi Zero, a micro USB to full size USB adapter.

- A micro SD card with a capacity of at least 4GB.
- A micro SD to full size SD card adapter or an SD card reader/writer that can accept a micro SD card.
- A USB cable with a micro USB connector on one end. This is used only during setup and development.
- A 32x64 RGB LED matrix display with a Hub75 interface.
- any RGB LED matrix that has 16 or 32 rows and a Hub75 interface.
- The power harness that came with the RGB LED matrix. The flat cable is not used.
- A 5V, 8 Amp minimum power brick.
- 5V, 10Amp power supply if using a full sized Raspberry Pi.
- Power jack that mates with your power supply.
- A Hub75 interface board that accepts a Raspberry Pi 40 pin connector.
- Adafruit sells one
- I designed my own for this project. Mine has the unused GPIO broken out to headers as well as headers for SPI, Serial and I2C. See BOM for my board or if you hand wire your own
- Four optional 2.5x11mm standoffs.
- Eight optional 2.5x6mm screws for the standoffs.
- Optional, for obtaining environmental information.
- BME280 sensor array board.
- Two, four pin DuPont shells.
- Eight DuPont female crimps.
- Four lengths of 22-28 awg stranded wire.
- Or you could use four female to female jumper wires





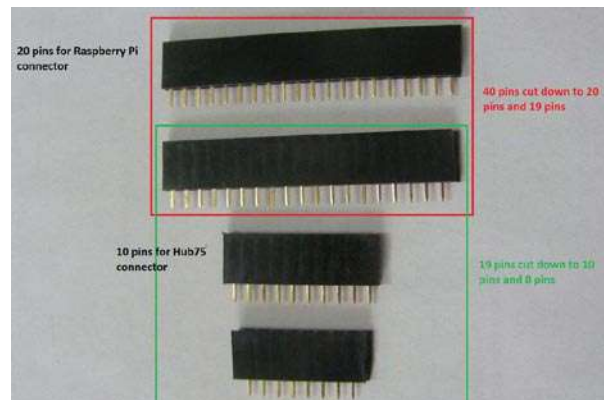
RGB LED Matrix displays are available on Ebay or from Adafruit. Just make sure it uses a Hub75 interface. You can buy a ready to run Hub75 interface board from Adafruit. They sell one for the Raspberry Pi and one for the Teensy. As this is a Raspberry Pi project, buy the one for the Raspberry Pi. I have included the Eagle files needed to have one of my custom Hub75 interface boards manufactured. I use AllPCB.com to have my boards fabricated. As of this writing, AllPCB.com is charging \$5.00 with free shipping to the USA for ten boards. The last order that I placed took about ten days from time of order to delivery to my door. Of course, if you are truly adventurous, you can hand wire your own

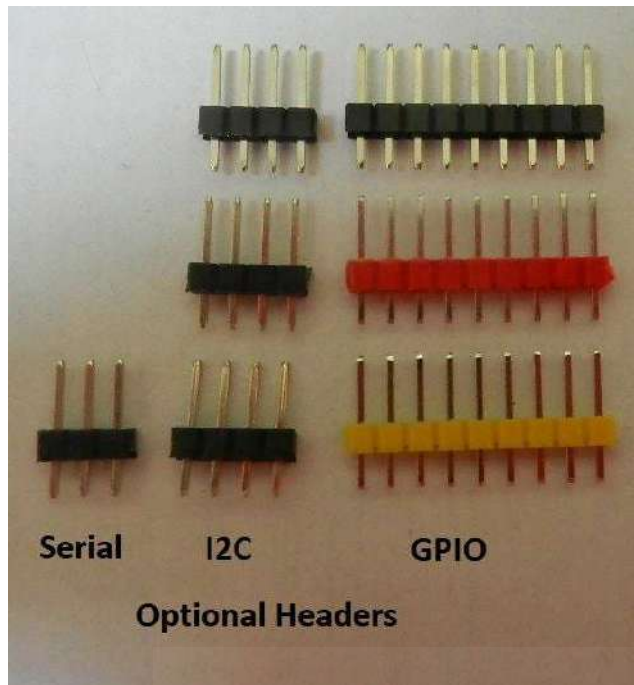
board from the PDF of the schematic. I also have some rev 1.1 boards left over from my order and some rev 1.0 boards. Contact me if you are interested in a blank board.

A 32x64 RGB LED matrix requires at least an 8 Amp power supply. If you decide to daisy chain in a second or third matrix to extend the length of the display, you will need an additional 8 Amps for each 32x64 panel that you add.

Matrix display panels are specified by the pixel (tri-color LED) density, number of columns and rows. A P4-64x32-16S display has a pixel spacing of 4mm. A P5-32x16-8S display has a pixel spacing of 5mm. Displays with a lower pixel density (higher P number) require a greater viewing distance for the human eye to blend the pixels. They are also usually a little less expensive. The display shown above is a P4-64x32-16S. The pixel density will also determine the actual physical dimensions of the display. The -16S indicates the number of rows that can be selected, 16S indicates 16 rows (four signals), 8S indicates 8 rows (three signals). It also indicates that the refresh is done 16 or 8 times for per panel. A panel has an upper and a lower half with one row from each half being refreshed simultaneously. So -16S is a 32 row panel and -8S is a 16 row panel.

Step 2: Hub75 Interface Board BOM:





You can ignore this if you purchased the Adafruit Hub75 Interface board. If you are hand wiring your own board then you may want to use through hole parts instead of the surface mount parts called for. In that case consider this more of a guideline. The BOM is detailed as two assemblies, a bare minimum and the ad-hoc optional assemblies. The first picture above shows the minimum parts required.

The BOM calls out for 74HC245DWR. Any part 74*245DWR will work. So 74LS245, 75ALS245 or

74AHC245 will all work. The important thing is that it is 74xxx series, 245 (indicates an octal bus transceiver) and that the wide surface mount package is used (DW indicates SOIC-20W package). CMOS (HC or AHC) is preferred over TTL (LS or ALS).

If you do the minimum build and leave out the AS1117-3.3 regulator then you need to solder a jumper wire to the two holes of J18. Doing this allows the 3.3V from the Raspberry Pi to connect with the 3.3V pins of the I2C headers and the 3.3V header associated with the unused GPIO. Leave the jumper out if you add the AS1117-3.3 regulator. When the jumper is left out the AS1117-3.3 handles the load for the added peripherals. It prevents overloading the Raspberry Pi's 3.3V regulator.

For the minimum build you should also place a jumper from the center pin to either of the outer pins of J11. This jumper selects 32 row or 16 row operation. Place the jumper to correspond with the RGB LED matrix that you are using. If you leave out the jumper then your display will look very weird. In place of a wire jumper, you could add a three pin post strip and then use a two pin shorting plug to select 32 or 16 rows.

Tip: You may find that you can save about a dollar by using two 2x10 sockets for J9 instead of two 1x20 socket strips.

Adafruit sells several 2x20 female headers. They vary in height (8.5mm or 13mm) and in the length of the solder pins. Stacking connectors have extra long pins so that you can stack another HAT on top. For this project, the standard 2x20 female header (Adafruit P/N 2222, \$1.50) is the best choice. If you buy the 13mm version then you will need to get longer standoffs (13mm instead of 11mm).

A 2x10 socket is used to mate with the display because it fits into the shroud that surrounds the 2x8 post strip on the display. The extra pins on each side prevent you from mis-aligning the connectors.

The rest of the parts are optional and add to the minimum build. You can add what you need for your project. I like to buy forty pin snap-apart post strips in black, red and yellow. I then break them apart into the sizes that I need. Of course you can use all black, but I find it very convenient to use black for ground, red for power and yellow or some other color for the signal.

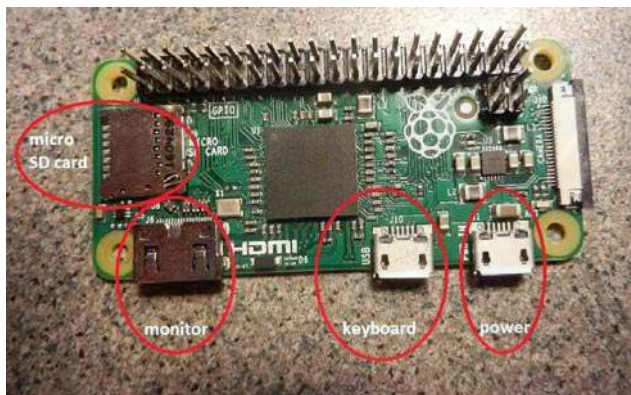
Optionally, you can use three row, right angle post strips for the I2C and GPIO headers. The advantage of using the right angle headers is that they will not interfere with the USB adapter and WiFi dongle if you are using a Raspberry Pi Zero. The right angle headers can be mounted on what ever side of the board is convenient for you.

Step 3: Tools Needed:

The SD card formatter is used to erase & format an SD card to give you a 'clean slate' to work with. The SD card imager is used to put a bootable image of the operating system (a .img file) onto your newly formatted SD card. Putty allows you to use SSH to talk to your Raspberry Pi

through its WiFi or Ethernet connection. With Putty and SSH you can operate and monitor your Raspberry Pi from a window on your PC. WinSCP allows you to copy files from your PC to your Raspberry Pi. WinSCP provides a drag and drop GUI for you to transfer files and view the directory structure of your Raspberry Pi and your PC.

Step 4: Setup the Raspberry Pi:



In addition to what was listed on the Raspberry Pi web site, you will also need the following;

- A full sized female HDMI to male mini HDMI adapter
- A full sized female USB to male micro USB adapter
- As we will be running headless, a mouse is not required
- If you are using a Raspberry Pi Zero, a USB WiFi dongle will be required
- To power your Raspberry Pi you will need either;
- To take power from your computer; a USB cable with a micro USB plug at one end
- To power from a wall socket; a 5V power pack with a micro USB on the end of the cable or an old 5V phone charger with a USB cable with a micro USB plug at one end.

The first picture above is of two styles of mini HDMI adapter. One is a simple plug and the other is a short cable. The second picture is of two styles of micro USB adapter. The third photo is a picture of a Raspberry Pi Zero. I have drawn red circles around where to plug in your power, keyboard, monitor and micro SD card. The micro USB plug adapter in the photo above works quite well, but is a little too big to plug in a micro USB cable

into the adjacent micro USB power jack (see fourth photo above). As a result, I filed down the side of the adapter until everything fit nicely (see fifth photo above). The cable style micro USB adapter does not have this problem.

Step 5: Configure the Raspberry Pi

Once you have your Raspberry Pi up and running the Raspbian Stretch Lite operating system it is time to configure your system. You need to enable the I2C interface, SSH, select your language and keyboard layout, select your timezone and configure WiFi to access your home network and set the WiFi for your country. Most of this is done using raspi-config. Setting up to work with your home network involves editing of a configuration file.

The easiest way to run raspi-config on your Raspberry Pi zero or Raspberry Pi Zero W is to use a keyboard and monitor. If you have a full sized Raspberry Pi you can put the SD card into that Raspberry Pi and then use a keyboard and monitor or plug in an Ethernet cable and SSH into the Raspberry Pi using Putty or connect a USB-to-Serial dongle to the console RX & TX pins and use a terminal program on your computer to run raspi-config.

When using raspi-config, Internationalization Options to change your Locale you must first deselect the default locale (en_GB.UTF-8 UTF-8) before you change to your locale. The selected locale is identified by an asterisk between square brackets ([*]). You deselect the locale by using the arrow keys to move the cursor to the currently selected locale and then press the space bar. The asterisk should disappear. You select a locale by using the arrow keys to move the cursor to your selection and then press the space bar. The asterisk should appear. You can select multiple locales. However, if you do then when you exit, you will find that the default locale is still selected. Until you deselect the default locale your selection of another locale will not 'stick'. If you live in the USA then select en_US.UTF-8 UTF-8.

Step 6: Configure WiFi for Your Home Network

If your WiFi network is not using some form of encryption (a very bad idea) you will not need a WiFi passkey. The WiFi passkey is not the same as your WiFi router's password. One allows you to use your WiFi network and the other allows access to your router to configure it.



Smart City - Identify, select, install and troubleshoot different module/ devices used in SMART Parking

Test Magnetic field for smart parking

Objective:

- to test Magnetic field for smart parking.

PROCEDURE:

Introduction

The main objective in designing the Magnopark algorithm is to leverage the low cost, efficient, accurate, and easy to use mobile internal sensors to help drivers finding a cheaper parking spot in the street comparing to the parking lots, in big crowded cities. In the Urban areas, finding a parking place is very time consuming, costly, and boring work. The main objective in designing the Magnopark is to design and develop a cost effective application for detecting the parking spots in the streets.

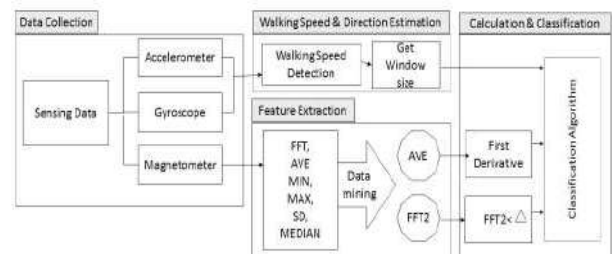
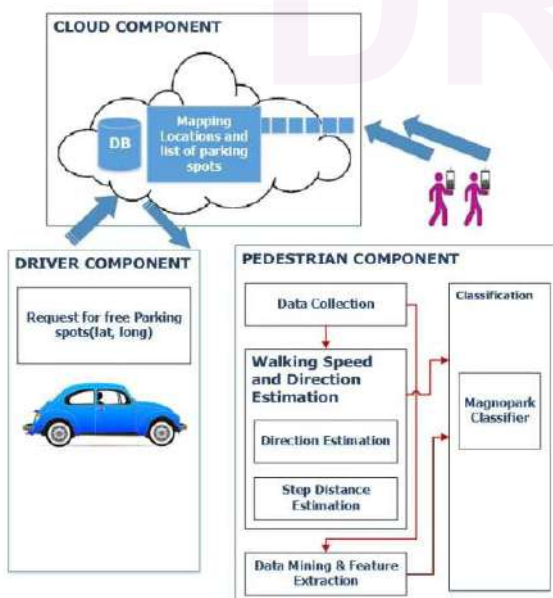
There are two parameters that we take care of: the cost of the application, and the accuracy. In most parking seeking applications, most of the calculations are performed on the user device, and also they use GPS for finding the parking spots which is a very expensive and non-accurate approach to be used in such a sensitive subject. In this thesis, the idea is to use a cost effective approach which helps people in urban crowded cities to use the negative point of always crowded streets as a positive issue. As almost all the people are carrying a cellphone in their bag or hand, we design an application that uses the pedestrians' cellphone sensors to detect the magnetometer changes while walking beside the vehicles parked beside the street.

MAGNOPARK COMPONENTS

Magnopark consists of three main components which will be described in details in the following sections.

DATA COLLECTION

We developed an android application to collect mobile internal sensors data. The system collects readings from accelerometer, gyroscope, magnetometer, and GPS. We resort to vector magnitude of 3-axis magnetometer sensor samples, which reads the variation of magnetic field in each of 3 coordinates, while the user is walking in the sidewalk in order to make Our algorithm independent of the orientation of the cellphone. Therefore, we can count on the data that is collected while the cellphone is being kept either in user's hand or pocket or bag. This is a good feature because it reduces the trouble of maintaining specific position or direction in the phone for the user. In other words, there is no need to hold the phone in hand, and in a specific direction to collect data. The application collects magnetometer data with frequency rate equals to 100 Hz. This rate is high enough to detect the slightest changes in the magnitude value. We also need gyroscope data to differentiate between 2 status of changing direction of the phone, and approaching a vehicle parked beside the street and accelerometer data is used to calculate the speed of walking. Also, using accelerometer data in company with magnetometer changes, we will be able to calculate the length of the cars parked beside the street. We need to calculate the speed of user walking to be able to decide about the window size that we need to detect a car and distinguishing its length. Also we would be able to tell the number of free spots by calculating the accelerometer rate. Moreover, GPS data is collected to use later for uploading the corresponding free parking spot latitude and longitude on the cloud to be mapped.



Magnopark system design

FIG. 3.1: Architecture overview of the different components of Magnopark system.

WALKING SPEED AND DIRECTION ESTIMATION

This module consists of 2 main sub-modules:

Distance Estimation Module

This module estimates the distance traversed by pedestrian at each step. We used accelerometer and gyroscope sensors of the user cellphone to detect and track user steps. It initiates with user moving and in order to detect the user moving, we utilize the changes in gyroscope sensor. Whenever the gyroscope sensor readings reaches above a certain threshold (in our case 0.3), we infer the user movement. Detecting the user walking triggers the application to start capturing accelerometer sensor data. Since the user steps length is, to some extent, proportional to his speed, we consider 2 parameters m_{max} and m_{min} to represent the maximum and minimum length of a person's step in terms of number of sample:

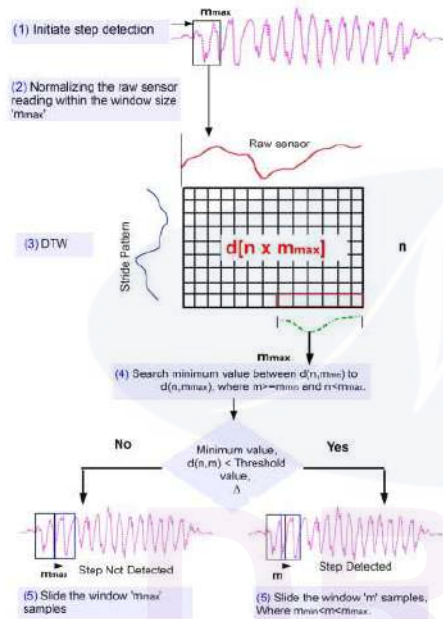
$$m_{max} = S_{max}/V_{max} * f_a \quad (1)$$

$$m_{min} = S_{min}/V_{min} * f_a \quad (2)$$

S is the length of a person's speed, V is the walking speed, and f_a is the frequency of the accelerometer samples in the smart phone. After collecting m_{max} rows of accelerometer samples, we calculate the magnitude of 3-axis accelerometer sample to be sure that it is independent of the orientation of the cellphone. Next, we apply the Finite Impulse Response low pass filter to remove the noise following by normalization. Then, we feed the samples to the Dynamic Time Wrapping (DTW) algorithm. DTW consider a window size of m_{max} and check if a predefined step pattern with size of n is within this specified window or not. If a step is detected, then we shift the window to the next window sample. Regardless of different step length in different users, the DTW could detect step. The whole overview of the algorithm is shown in figure.

Direction Estimation

We need to track the direction of each user walking in order to avoid the erroneous duplication parking spot detection, as well as to differentiate between the states in which the user is standing still beside a parking spot, and the status in which the user is walking beside multiple consecutive parking spots. In order to estimate the user direction, we have to align three different coordinate system which is shown in figure 4.3: cellphone coordinate system, users walking coordinate system, and global coordinate system. As the global coordination is fixed, we map the user coordinate and the cellphone coordinate to the global coordinate. After aligning these 3 systems on the globe, then the highest variation of the linear acceleration readings will show the users walking direction. We apply Principal Component Analysis (PCA) analysis to find out the direction of the user face which corresponds to one of its coordinate systems.



Adaptable step detection module in Spy Loc localization system

Smart City - Identify, select, install and troubleshoot different module/ devices used in SMART Parking

Execute installation of proximity sensor for boom barrier, IR sensor for presence

Objective:

- to execute installation of proximity sensor for boom barrier, IR sensor for presence.

Requirements:

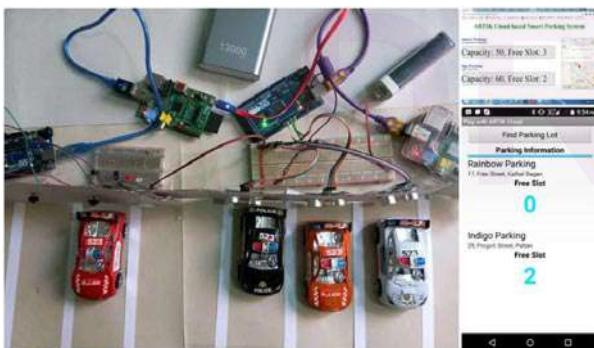
Hardware

1. Raspberry Pi 3 Model B (2pcs)
2. Arduino UNO (2pcs)
3. Intel Edison with Arduino Breakout
4. Grove starter kit plus for Intel Edison
5. Ultrasonic Ranger HC-SR04 (4pcs)
6. IR receiver (generic) (4pcs)
7. IR transmitter (generic) (4pcs)
8. Resistor 221 ohm (4pcs)
9. Resistor 10k ohm (4pcs)

Software & Online Services

1. Samsung IoT ARTIK Cloud for IoT
2. Google Developers Google Maps
3. Arduino IDE
4. Intel IoT XDK

PROCEDURE:



Steps involved in this project:

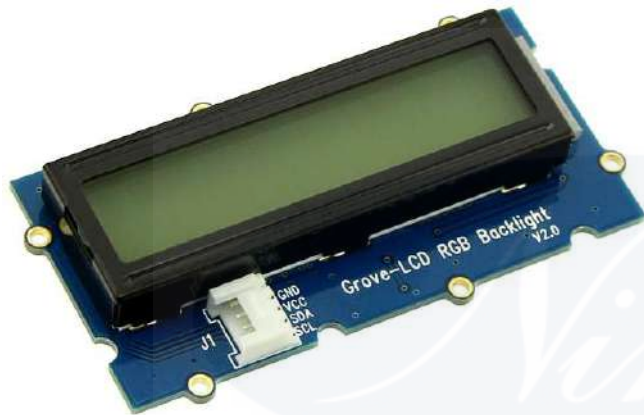
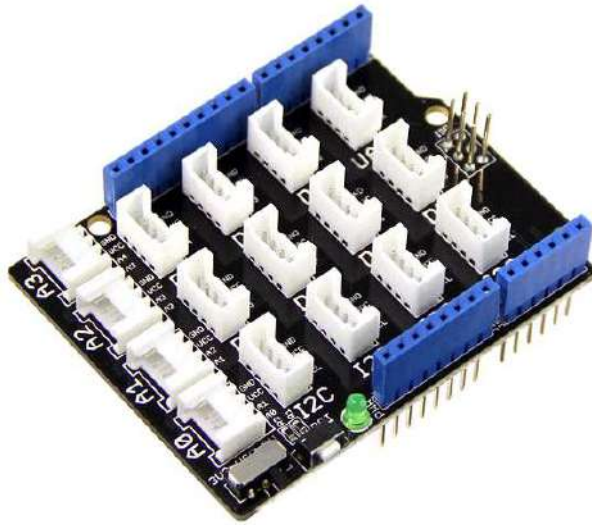
1. Making three devices in Artik Cloud Platform
2. Making one application in Artik Cloud
3. Making one rule in Artik Cloud
4. Preparing & Programming Arduino
5. Connecting the Sensors to Arduino
6. Preparing Raspberry Pi
7. Preparing Intel Edison
8. Developing Web Application

9. Developing Android Application

10. Completing the Project.

Step 1: Things Used in This Project

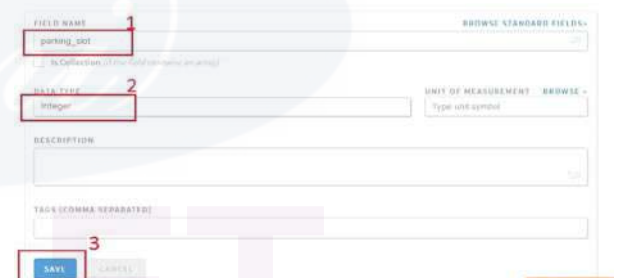
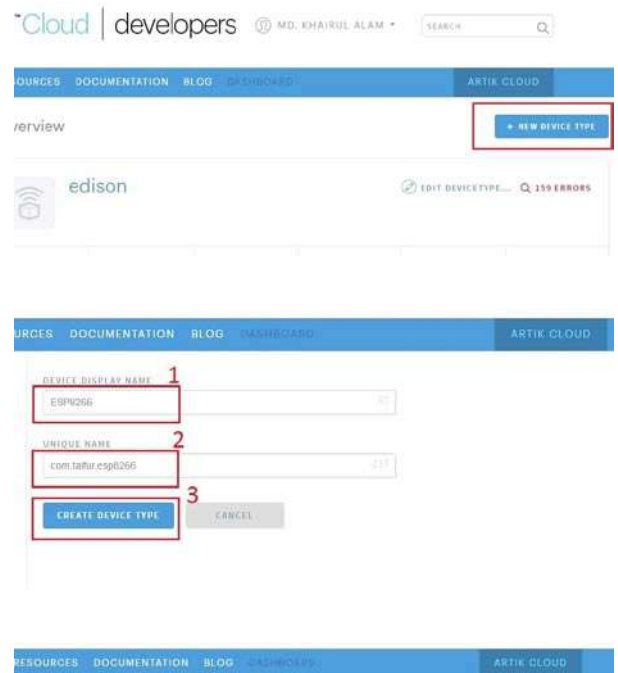




Programming Language

1. HTML
2. Java Script
3. Node.js
4. Python

Making a New Device in Artik Cloud Platform



[illegible]

The screenshot shows the BLOOD interface with a top navigation bar containing 'PRICING' and 'DEVELOPERS'. A green banner below the navigation bar states 'Your new device ESP8256 has been added'. The main content area displays a list of devices. The 'edison' device is highlighted with a red circle around its ID 'ESP8256' and a 'view your data' link. Below the device name, it shows 'Disconnected September 22, 2016' and a 'DELETE' button. Other devices visible include 'rainbow-parking' and 'raspberry-pi'.

Device Info

DEVICE TYPE
ESP8266

CONNECTED SINCE
October 3, 2016

LAST DATA TRANSFER
Never

DEVICE ID
9a9f502725b7-00000000000000000000000000000000

DEVICE TYPE ID
00750000000000000000000000000000

374047d3d5d6fcf0

NAME
ESP8266

GENERATE DEVICE TOKEN

SAVE CHANGES

DELETE


[OVERVIEW](#)
[WORKS WITH...](#)
[PRICING](#)
[MY ARTIK CLOUD](#)
[BLOG](#)

The screenshot shows the 'Device Info' page for a device named 'ESP8266'. The page is divided into two main sections: 'Device Info' and 'Device Actions'. The 'Device Info' section contains the following fields:

- Device ID:** a8f595b7... (highlighted with a red box)
- Device Type ID:** a8759339... (highlighted with a red box)
- Name:** ESP8266
- Device Token:** a1b1632a... (highlighted with a red box)
- Device Token:** a1b1632a... (highlighted with a red box)

At the bottom of the page, there is a 'DELETE' button and a 'SAVE CHANGES' button.



The screenshot shows the ARTIK Cloud website. The header includes the ARTIK Cloud logo, a user profile icon for MD. KHAIROL ALAM, and a search bar. The navigation bar has links for PRICING, MY ARTIK CLOUD, BLOG, and DEVELOPERS. The main heading is 'Connect another device'. Below it, a text line says 'ARTIK Cloud works with many smart device types - start typing to find yours.' A search bar contains the text 'es'. Below the search bar, a blue box highlights the search result 'ESP256 - com.taifun.esp0256'.

The screenshot shows the AWS IoT console interface. At the top right, there is a 'View your data' link with a magnifying glass icon. Below this, there are four device cards arranged in a 2x2 grid. Each card displays the device name, a settings gear icon, the connection status and date, and a 'DELETE' button. The 'ESP8266' device card is highlighted with a red rectangular box around its settings gear icon.

Device Name	Settings Icon	Connection Status	Action
edison		Connected September 23, 2016	DELETE
ESP8266		Connected October 5, 2016	DELETE
rainbow-parking			
raspberry-pi			

Smart City - Identify, select, install and troubleshoot different module/ devices used in SMART Parking

Apply full stack solution to deal with all aspects of parking including high level tools for management and analytics software down to street level occupation sensors and enforcing tools

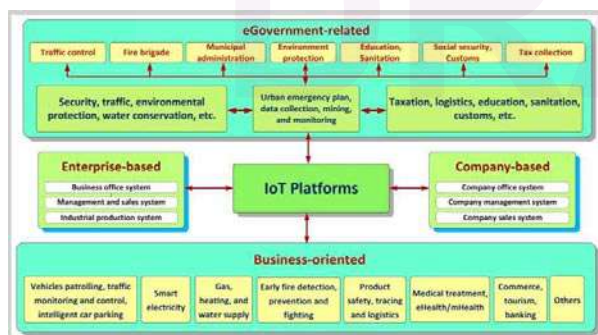
Objective:

- to apply full stack solution to deal with all aspects of parking.

PROCEDURE:

The generic concept of using cloud-based intelligent car parking services in smart cities as an important application of the Internet of Things (IoT) paradigm. This type of services will become an integral part of a generic IoT operational platform for smart cities due to its pure business-oriented features. A high-level view of the proposed middleware is outlined and the corresponding operational platform is illustrated. To demonstrate the provision of car parking services, based on the proposed middleware, a cloud-based intelligent car parking system for use within a university campus is described along with details of its design, implementation, and operation. A number of software solutions, including Kafka/Storm/Hbase clusters, OSGi web applications with distributed NoSQL, a rule engine, and mobile applications, are proposed to provide 'best' car parking service experience to mobile users, following the Always Best Connected and best Served (ABC&S) paradigm.

The IoT platforms could be categorized into four types: eGovernment-related, production/ enterprise-based, company-based, and pure business-oriented platforms. Figure 2 illustrates these four types of the IoT platforms, which are described in more detail below.



The eGovernment IoT is the supporting foundation that should facilitate the city's/region's/state's economic development and management. It is usually funded through a government's public welfare scheme, which gradually promotes the development of eGovernment towards the IoT. The eGovernment IoT platform needs information relevance to prevent the 'information island' effect and to ensure network- and information security.

The enterprise-based and company-based IoT platforms can improve competitiveness and service assurance of market-oriented enterprises and companies. These require independently funded IoT projects for more efficient production management, warehousing, distribution, transportation, logistics, marketing, and supply chain management.

The pure business-oriented IoT platforms may become an important facilitator as optimization and integration of information market resources require involving all end-users, network providers, and service providers to build a modern industry established on valuable IoT business models and platforms.

The CERP-IoT report predicted that the automotive industry will be using 'smart things' to monitor everything. Especially when using wireless technology for vehicle-to-infrastructure (V2I) communication, the real-time locating systems (RTLS) can enable tracking and tracing services, which will significantly advance the ITS applications. The intelligent car parking systems, such as the one described in this paper, constitute an important part of the ITS with a primary purpose to find, allocate, reserve, and provide the 'best' available car parking lot to each individual user who is driving a car in a particular area. Valuable add-on functionality could be the provision of navigation instructions to the driver for reaching the lot. These systems may serve as a foundation and a common business model for a generic IoT operational platform due to their properties of providing pure business-oriented services.

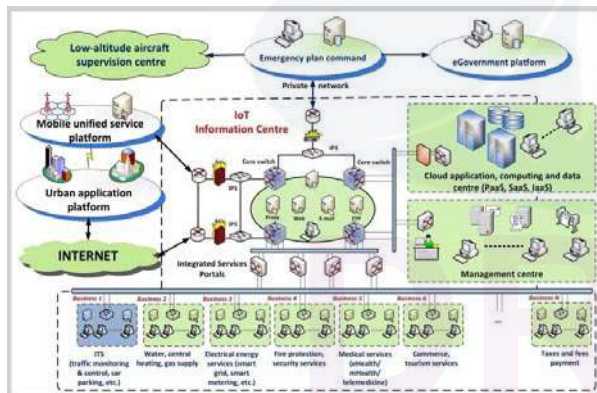
Researchers show that more than 66% of drivers do not mind paying for car parking facilities during their working hours. This directly adds value to the car parking business, which is a stimulus for the development of intelligent car parking services for smart cities.

The existing car parking systems are not very efficient as they do not provide the 'best' service, e.g., finding the nearest available car parking lot. At the sensor layer, most of researchers today focus on detecting the car-parking-lot occupancy. A car-parking-lot detection method is proposed based on an automatic threshold algorithm; as the image processing algorithms are expensive, a hardware solution is suggested. Sensors are used for intelligent autonomous parking. A laser scanners are used to retrieve the car-parking-lot position.

At the communication layer, an InfoStation-based multi-agent system facilitating a car parking locator service is proposed; users are provided with a personalized service based on their location and mobile device's capabilities. A wireless sensor network solution is proposed for car parking management along with a routing protocol for improving the transport reliability.

IoT Intelligent Car Parking System

ITS and other systems, such as electrical energy systems, water-, heating- and gas supply systems, city fire protection and security systems, eHealth/mHealth systems, etc., provide intelligent IoT services to make the city smarter. ITS locates in the business layer of IoT, communicates with the cloud-based information centre of the smart city, and delivers 'best' transport-related services to users, such as traffic monitoring & control, route planning, car parking service, etc. Figure 3 depicts a high-level view of a centralized IoT platform, which could serve as a generic architectural foundation for a 'smart city' establishment, operation, administration, and management. This top-level generic architectural design can unify the development of business applications as an efficient and economical process. For instance, if an ITS service provider wants to deploy a car parking service within a smart city's shared cloud infrastructure, it will only need to focus on the operating model to realize the smartness of the car parking.



Here we propose an intelligent car parking system for integration into a smart-city IoT architecture, which consists of three layers—A sensor layer, a communication layer, and an application layer (Figure 4).

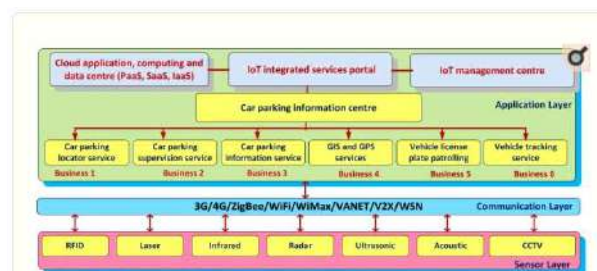


Figure 4

An IoT intelligent car parking system for a Smart City.

At the application layer, an information centre provides cloud-based services, i.e., Platform as a service (PaaS), Software as a service (SaaS), and Infrastructure as a service (IaaS); i.e., for allocating computing/storage resources for different car parking services. An IoT management centre administrates the smart city via an IoT integrated services portal. At the bottom, a number of business services explore a common interface to the communication layer. These include car parking locator, supervision, and information services, GIS/GPS services, vehicle license plate patrolling, car tracking services, etc.

At the communication layer, various wireless technologies provide connection between the application- and the sensor layer, based on the ABC&S communication paradigm. A 3-tier InfoStation-based network architecture could be integrated in this layer to enable 'anytime-anywhere-anyhow' communication functionality in smart cities.

Different sensing technologies could be utilized at the sensor layer for embedded parking solutions, such as the Radio Frequency Identification (RFID) for car parking access control; laser, passive infrared, microwave radar, ultrasonic, passive acoustic array sensors, or Closed-Circuit Television (CCTV) with video image processing for detecting the status of the car parking lots; license plates with installed 3G/4G communication module for cars' tracking and tracing; etc.

To enable the car parking system to work as an operational platform in a smart city, different car parking areas must be distinguished in providing 'best' car parking lots by executing different business roles and applications. Based on their properties, the car parking areas could be divided into four main categories (Figure 5): A transportation hub area; a residential/community area; a ground/street area; and a shopping mall/hotel/restaurant area. The relevant management and control entities, including a highway centre, emergency centre, traffic control centre, and police can get access to the information managed by the car parking information centre with high authority. The sensors deployed in the car parking area periodically send updated information as regards occupancy of the car parking lots to the car parking meters, which push this data to the information centre.



Figure 5

The intelligent car parking services' operational platform.

Users can interact with the system by installing the corresponding car parking application on their mobile devices. Facilitated by a personal assistant agent, each user can set up a personal profile which will be used by the application for finding, allocating, reserving, and paying for the 'best' parking lot in each particular scenario. Stored in the memory-based No-SQL database, the user profile will be dynamically updated to reflect changes in the user's context and behavior, which are analyzed by the system. With efficient car parking lot allocation algorithms/rules, the system is always able to provide the mobile user with the 'best' available car parking lot following the ABC&S paradigm.

Sample Car Parking System for University Campus:

Every big university has a number of different car parking areas, e.g., for visitors, students, staff members, etc. University car parking belongs to the category of a residential/community car parking. Every working day, students and staff members spend usually a lot of time just to find an available car parking lot. This is not only a time-consuming and energy-wasteful process but it may also cause car traffic jams. With the intelligent cloud-based car parking service proposed here, an efficient utilization of available car parking facilities could be achieved within a 'smart university' environment.

One way to achieve this is to have each car parking lot equipped with a sensor which is able to sense the presence of a car in it. An information station (InfoStation), operating in the car parking area, periodically collects and aggregates the car presence information from all sensors deployed in the area, e.g., by means of Wi-Fi, ZigBee, or other short-range wireless technology. In the case of paid car parks, optional parking meters could operate between the InfoStation and the sensors. When the occupation status of a car parking lot is changed, information about this is pushed by the InfoStation to the car parking Information Centre (InfoCentre) in the cloud via the university Intranet (Figure 6). Further in this paper, we are focusing only on the software implementation of this system.

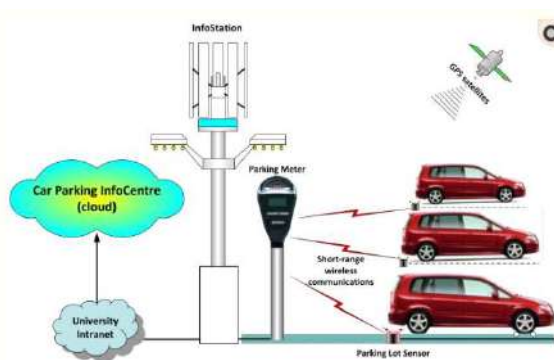


Figure 6
An infoStation-based university car parking system.

Design

From the high-level view of the IoT-based smart city and the layered intelligent car parking system (Figures 3 and 4), the car parking service's application layer, e.g., within a university campus, could be deployed with three tiers, as shown in Figure 7. In the cloud tier, web applications serialize data into the Hadoop centre; in the web servers tier, applications—developed as Bundles—dynamically register themselves on the Open Service Gateway initiative (OSGi) framework; and in the mobile apps tier, mobile devices access the web applications and provide 'best' car parking services to their users.

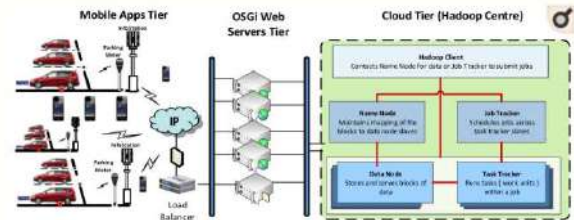


Figure 7
A high-level view of a cloud-based car parking system's application layer.

Cloud Tier

The cloud provides data storage and computing resources for the car parking service. It stores the 'big data' of available car parking lots, car parking area, cars' location, users' location and profiles, etc. The most recent data is usually stored in the Hadoop's Hbase database to support real-time queries, whereas the history data is serialized to Hive (a warehousing in Hadoop). For computing, a number of Map/Reduce algorithms are used, such as a recommendation algorithm for suggesting the 'best' car parking lots to users, a recommendation algorithm based on friends' car parking suggestions, a profile-updating algorithm based on users' parking history, etc. To build an efficient and scalable system, a rule engine Drools is used to make decisions, based on facts, quickly and reliably.

OSGi Web Servers Tier

This tier acts as a bridge between the mobile apps tier and the cloud tier. Considering the great number of web applications/services running in this tier, the deployment of a new/updated application should be possible without stopping/restarting the corresponding web container/server. The OSGi provides an environment to modularize web applications into bundles, which can dynamically register themselves in the bundle's execution context. To provide high-performance and on-demand car parking services for users, a key-value based NoSQL database—Redis—is used in this tier to provide scalable and distributed job queues. To optimize web resources' utilization, a load balancer distributes the user's requests across the cluster of web servers. A distributed system collects the web servers' log data and sends it to the cloud.

Mobile Applications Tier

The Allied Business Intelligence (ABI) Research [27] reported that the Android operating system played a dominant role in the smartphone market in 2013 (with 81% share). Thus the first version of the car parking mobile application is developed for Android mobile devices. When a user approaches the University campus, an automatic request is sent by the application (on behalf of the user) to an OSGi car parking web server asking for a car parking lot. The server finds the 'best' available car parking lot for this particular user, based on his/her preferences specified in the user profile, and (optionally) reserves it. Driving directions are then sent

to the user along with a detailed map, e.g., by utilizing the Google Map app with an Android API.

Figure 8 depicts the system's main components. The cloud tier includes a real-time stream computing part for user behavior real-time updating, parking fees charging, etc.; and a non-real-time modeling part for data mining and warehouse management. The distributed log data collection part acts as a high-speed data pipe in the system. The OSGi web servers tier in the centre acts as a bridge between the other two tiers. Different car parking services could be provided; these are described by their service descriptions (SD).

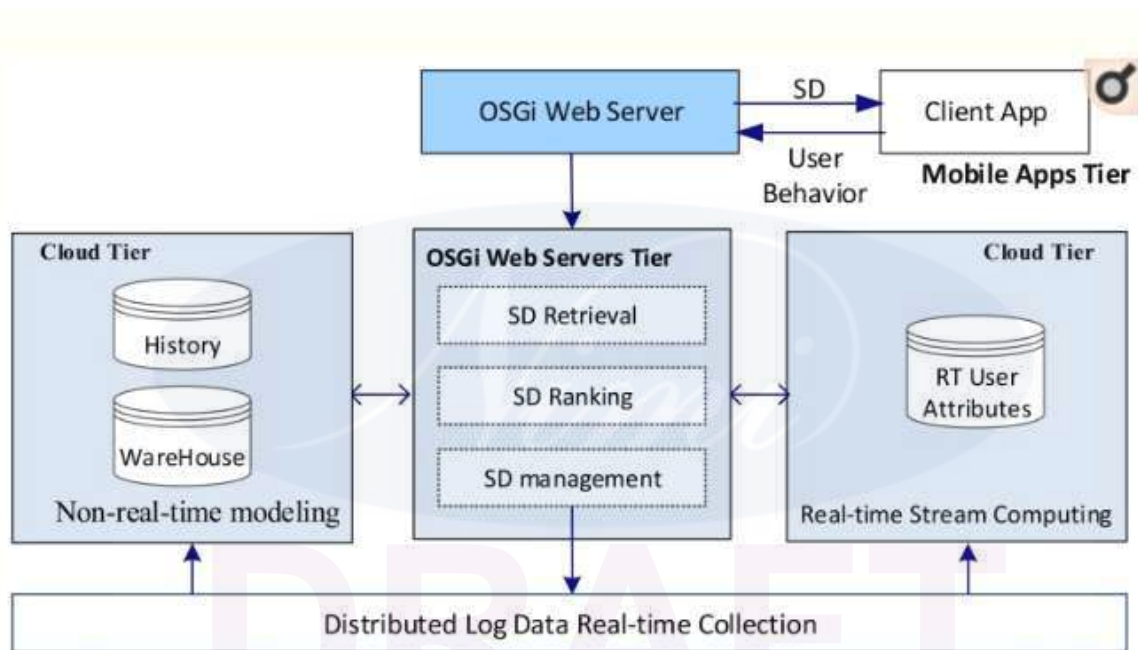


Figure 8.

The main components of the cloud-based car parking system.

Smart City - Identify, select, install and troubleshoot different module/ devices used in Smart Traffic

Use Location Sensors, GPS & GPS integrated circuits

Objective:

- to use location Sensors, GPS and GPS integrated circuits.

Requirements:

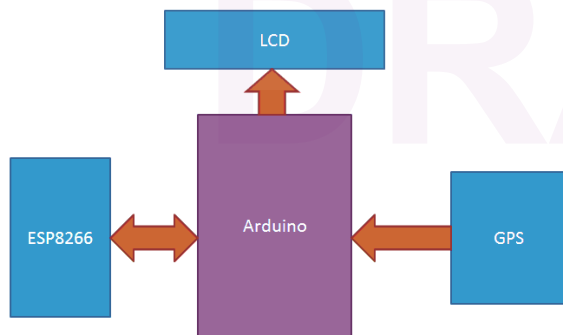
- LCD
- GPS
- Arduino
- Microcontroller

PROCEDURE:

About the Project

we are building more advanced version of Vehicle Tracking System in which you can Track your Vehicle on Google Maps. In this project, we will send the location coordinates to the Local Server and you just need to open a 'webpage' on your computer or mobile, where you will find a Link to Google Maps with your Vehicles Location Coordinates. When you click on that link, it takes you on Google Maps, showing your vehicles location. In this Vehicle Tracking System using Google Maps, GPS Module is used for getting the Location Coordinates, Wi-Fi module to keep send data to computer or mobile over Wi-Fi and Arduino is used to make GPS and Wi-Fi talk to each other.

How it Works



Overview of the diagram

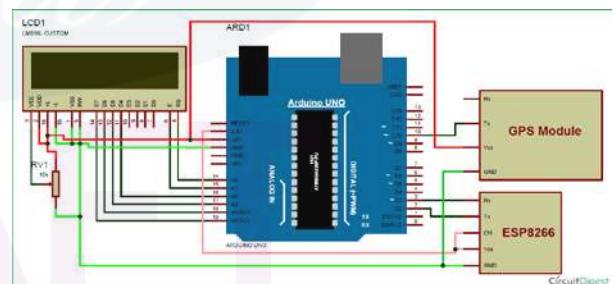
Before this, Arduino sends command to Wi-Fi Module ESP8266 for configuring & connecting to the router and getting the IP address. After it Arduino initialize GPS for getting coordinates and the LCD shows a 'Page Refresh message'. That means, user needs to refresh webpage. When user refreshes the webpage Arduino gets the GPS coordinates and sends the same to webpage (local server) over Wi-Fi, with some additional information and a Google maps link in it. Now by clicking this link user

redirects to Google Maps with the coordinate and then he/she will get the Vehicle Current Location at the Red spot on the Google Maps. The whole process is properly shown in the Video at the end.

Components Required

- Arduino UNO
- Wi-Fi Module ESP8266
- GPS module
- USB Cable
- Connecting wires
- Laptop
- Power supply
- 16x2 LCD
- Bread Board
- Wi-Fi router

Circuit Explanation



Circuit Design

Circuit for this 'Vehicle Tracking using Google Maps project' is very simple and we mainly need an Arduino UNO, GPS Module and ESP8266 Wi-Fi module. There is a 16x2 LCD optionally connected for displaying the status. This LCD is connected at 14-19 (A0-A5) Pins of Arduino.

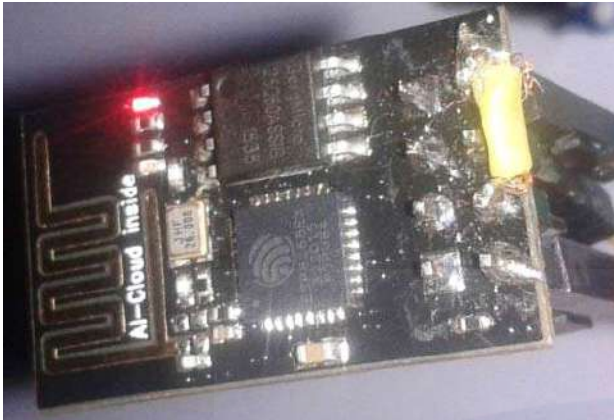
Here Tx pin of GPS module is directly connected to digital pin number 10 of Arduino. By using Software Serial Library here, we have allowed serial communication on pin 10 and 11, and made them Rx and Tx respectively and left the Rx pin of GPS Module open. By default Pin 0 and 1 of Arduino are used for serial communication but by using Software Serial library, we can allow serial communication on other digital pins of the Arduino. 12 Volt adaptor is used to power the GPS Module.

Wi-Fi module ESP8266's Vcc and GND pins are directly connected to 3.3V and GND of Arduino and CH_PD is also connected with 3.3V. Tx and Rx pins of ESP8266 are directly connected to pin 2 and 3 of Arduino. Software Serial Library is also used here to allow serial communication on pin 2 and 3 of Arduino. Wi-Fi module ESP8266's Vcc and GND pins are directly connected to 3.3V and GND of Arduino and CH_PD is also connected with 3.3V. Tx and Rx pins of ESP8266 are directly connected to pin 2 and 3 of Arduino. Software Serial Library is also used here to allow serial communication on pin 2 and 3 of Arduino.

ESP8266

ESP8266 has two LEDs, one is Red, for indicating Power and second is Blue that is Data Communication LED. Blue LED blinks when ESP sends some data via its Tx pin. Also, do not connect ESP to +5 volt supply otherwise your device may damage. Here in this project, we have selected 9600 baud rate for all the UART communications.

User can also see the communication between Wi-Fi module ESP8266 and Arduino, on the Serial Monitor, at the baud rate of 9600:



```
COM31
Send
AT
AT
OK
AT+CWJAP="1st floor","mudal884"
OK
AT+CWJAP="1st floor","mudal884"
OK
AT+RST
OK
Connecting Wifi...
AT+CWJAP="1st floor","mudal884"
AT+CWJAP="1st floor","mudal884"
AT+CWJAP="1st floor","mudal884"
OK
Wifi Connected
IP Address:"192.168.1.23"
Port:80
AT+CIPMUX=1
OK
AT+CIPSERVER=1,80
OK
Latitude:28.44
Longitude:77.31
System Ready..
Please Refresh Ur Page
Please Refresh Ur Page
Autoscroll Both NL & CR 9600 baud
```

Smart City - Identify, select, install and troubleshoot different module/ devices used in Smart Traffic

Apply solar panel, Antenna and Radio Technology

Objective:

- to apply solar panel, Antenna and Radio Technology.

Requirements:

- Solar panel
- Antenna
- Radio waves

PROCEDURE:

A compact UHF antenna has been presented in this paper for nanosatellite space mission. A square ground plane with slotted rectangular radiating element have been used. Coaxial probe feeding is used to excite. The rectangular slot of the radiating patch is responsible for resonating at lower UHF bands. One of the square faces of the nanosatellite structure works as the ground plane for the slotted radiating element. The fabricated prototype of the proposed antenna has achieved an impedance bandwidth ($S_{11} < -10\text{dB}$) of 7.0 MHz (398 MHz–405 MHz) with small size of 97 mm × 90 mm radiating element. The overall ground plane size is 100 mm × 100 mm × 0.5 mm. The proposed antenna has achieved a gain of 1.18 dB with total efficiency of 62.5%. The proposed antenna addresses two design challenges of nanosatellite antenna, (a) assurance of the placement of solar panel beneath the radiating element; (b) providing about 50% open space for solar irradiance to pass onto the solar panel, enabling the solar panel to achieve up to 93.95% of power under of normal conditions.

Antenna design

The design layout of the proposed antenna is illustrated in Fig.

The proposed antenna is designed considering the following issues.

- use satellite body as ground plane
- allow sufficient space for the placement of solar panels between the ground plane and radiating element of the proposed antenna.
- ensure more than fifty per cent

The antenna is designed to avoid deployment complexity and to mitigate power scarcity

considering the antenna design constraints of 1U, 2U nanosatellite structures. Initially, the following constraints were set to design the antenna.

1. The overall size of the proposed antenna is limited to 10×10×10 cm³.
2. The gap between ground plane and radiating structure should stay within 6 mm.

The main radiating element is designed in a way that it can provide maximum effective length to achieve resonance at UHF 401MHz. The small radiating element at the top left side is mostly responsible for tuning resonance. The UHF antenna is designed using 0.5mm brass sheet for radiating element and 1.50 mm thick hard anodized Aluminum 6061 sheet is used as ground plane. The considered brass material in simulation has 2.74×10^7 S/m electric conductivity, 109 W/K/m thermal conductivity and 0.38 kJ/K/kg heat capacity. The gap between ground plane and radiating structure optimized to 4.6 mm that provides sufficient space to the ground plane to be mounted on the CubeSat body and placement of solar panel, shown in Fig 2. The design parameters of the proposed antenna are enlisted in Table 1. The shorting wall technique is used to achieve resonance at lower frequency and better reflection coefficient.

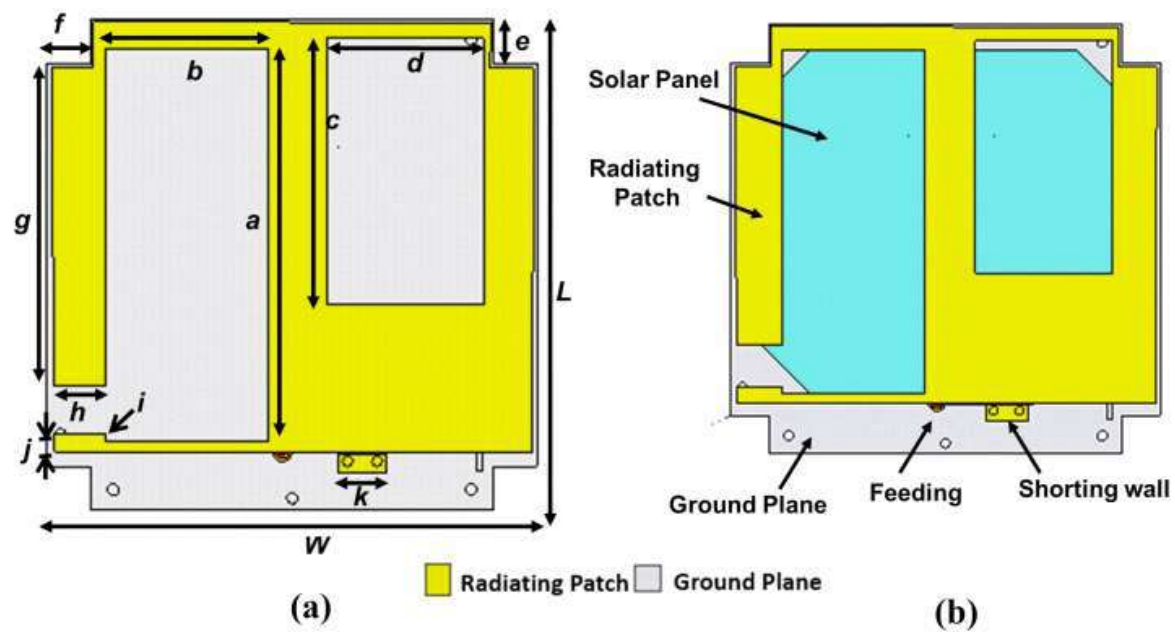


Fig 1. (a) Antenna Design Layout. (b). Antenna with Solar Panel.

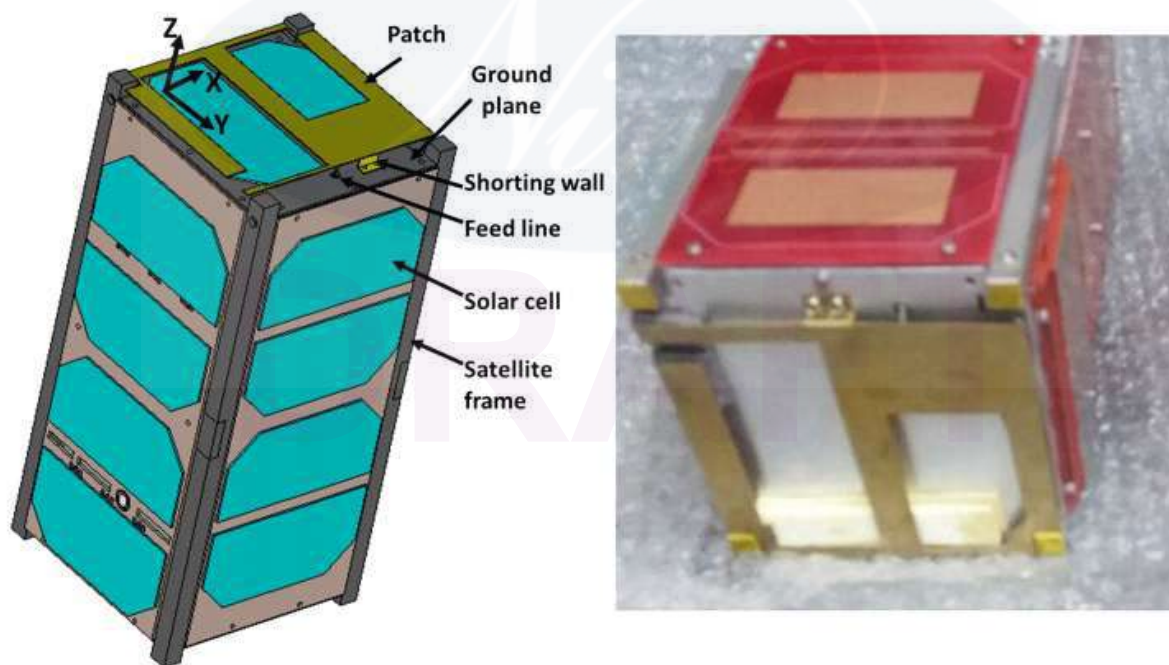


Fig 2. Antenna with 2U nanosatellite body.

Smart City - Identify, select, install and troubleshoot different module/ devices used in Smart Traffic

Use Scanner for real-time traffic and pedestrian estimation

Objective:

- to use Scanner for real-time traffic and pedestrian estimation.

PROCEDURE:

The period between the end of the green display on one phase and the start of the green display on the next phase gaining right-of-way is known as the inter green period. It comprises an amber display, red + amber display and may also contain a period when the red signals are shown to all approaches simultaneously. In some countries this inter green period is composed solely of an amber signal and an all red period. With a five second intergreen the amber and red + amber periods occur consecutively. Any period over five seconds will include a period where red signals are shown to all approaches simultaneously. (An all red period).

- To allow vehicles to clear the intersection when the distance across the junction is excessive.
- To improve safety on high speed roads.
- On roads where there are insufficient numbers of right-turning traffic to justify provision of a separate stage.

It should be noted that an intergreen period which is too short will be potentially dangerous but a period which is too long is equally unsatisfactory since it may lead to delay, frustration and lack of observation by drivers. A guide to determining the length of the intergreen period is illustrated in Table 4.

A vehicle which passes over the stop line at the start of the amber display must be clear of the potential collision point in relation to a vehicle starting at the onset of the green of the following stage, when travelling at the normal

speed for the intersection. The distances AF and BF should be determined and those distances which give the highest difference used. The recommended intergreen period can then be determined.

When East-West arms are losing right-of-way if $AF - BF$ is greater than $CH - DH$, then $x = AF - BF$ (or vice versa).

When the following stage is a pedestrian stage the distance 'x' should be determined from the position of the pedestrian crossing. Where pedestrians are losing right-of-way the start of the following stage should be delayed until the crossing area is clear.

MINIMUM GREEN PERIODS

Minimum Green Periods cannot be overridden by any demands, whether emanating from vehicles, manual control devices or received remotely from central computers or linked controllers. Such a period is built into signal controllers. The shortest minimum green period normally used for vehicle stages is six to eight seconds.

Site conditions may require a longer period where large numbers of heavy vehicles have difficulty in starting, or the approach is on a steep gradient.

Where pedestrians and traffic share the same stage, minimum green times may be governed by the time required by pedestrians to clear the crossing.

When North-South arms are losing right-of-way if $DE - AE$ is greater than $BG - CG$, then $x = DE - AE$ (or vice versa).

Smart City - Identify, select, install and troubleshoot different module/ devices used in Smart Traffic

Carry out Smartphone Detection (Bluetooth, Wifi, 3G/4G-GPRS etc.)

Objective:

- to carry out Smartphone Detection using Bluetooth, WiFi, 3G/4G- GPRS.

PROCEDURE:

Smartphones are playing an increasingly central role in our lives. They are ubiquitous, as we carry them nearly everywhere, and entrust them with sensitive and sometimes deeply personal information. We use them to carry out day-to-day tasks from communicating with family members and socialising on social media apps to tracking our health and taking care of our finances on banking apps.

But it is also a device with a camera, a GPS and a microphone that you have next to you at all times. Unfortunately, mobile phones were not designed for privacy and security. Turning this hardware into a surveillance tool is much easier and effective than you think. Not only they do a poor job of protecting your communications, they also expose you to new kinds of surveillance risks.

How it works

The mobile network/ SIM-card operators themselves have the ability to intercept and record all of the data about visited websites, who called or sent SMS to whom, when, and what they said.

Your Wi-Fi internet provider offers DNS as part of your service, which means your provider can also log your DNS traffic — in essence, recording your entire browsing history.

Any mobile network operator can also precisely calculate where a particular subscriber's phone is located whenever the phone is powered on and registered with the network. The ability to do this is called triangulation.

One way the operator can do this is to monitor the signal strength that different towers observe from a particular mobile phone, and then calculate where that phone must be located based on the observations. The accuracy with which the operator can figure out a subscriber's location varies, depending on many factors including the technology the operator uses (2G/3G/LTE) and cell tower numbers in the vicinity.

Normally only the mobile operator itself can perform this kind of tracking, though, these information might be

available to local or foreign governments through official or informal arrangements. In some cases, foreign governments have also hacked mobile operators' systems in order to get secret access to users' data. Also, Stingrays (explained below) can be used by someone physically nearby you to intercept communication packets.

Preventative Measures

There is no way to "hide" from this kind of tracking as long as your mobile phone is powered on and transmitting signals to an operator's network. The best way not to get detected? Don't connect it to the network or to any other computer, a practice known as air-gapping. Though, in a world where practically every machine connects to the internet, this is not easy.

However, for ultra-sensitive files and tasks — like storing Bitcoins or working with confidential blueprints — the inconvenience of working entirely offline can be justified, despite all the trouble. For these situations, the highly cautious rely on Faraday cages or bags. These are essentially metal-lined phone cases that block all radio frequencies. No signal can go in or out. These are easily available on Amazon for relatively cheap.

However, while the cage or bag might block your phone from revealing its location, it doesn't really prevent it from spying if it is already been hacked with a spyware — before it got air-gapped.

The safest practice is to assume that traditional calls and SMS text messages have not been secured against eavesdropping or recording.

The situation can be different when you are using secure communication apps to communicate (whether by voice or text), because these apps can apply end-to-end encryption (E2EE) to protect your conversations. Such apps with reliable E2EE implementation and strong anti-forensics capabilities can provide more meaningful protection.

The level of protection that you get from using such apps, depends significantly on which apps you use and how they work. One important question is whether there is

Smart City - Identify, select, install and troubleshoot different module/ devices used in Smart Traffic

Detect liquid presence over road by liquid presence sensor for smart security

Objective:

- to detect liquid presence over road by liquid presence sensor for smart city.

Requirements:

Computer system

Arduino UNO

Liquid presence sensor

PROCEDURE:

Many automobile industries are researching on the automation of the vehicles exploiting the safety, entertainment technologies, reliability, etc of the vehicle.

The automatic wiper systems are relatively uncommon in modern vehicles for a number of reasons. These wiper systems are too expensive and not suitable for new vehicles.

Wipers are designed to clean the water from a windshield. Up till now many attempts have been made to construct the cost effective and reliable automatic wiper control system for vehicles, so that the driver can concentrate on driving without any disturbance. The major drawback of most of the systems is a small area of windshield under observation. Also, the speed of wiper motor is constant when they are invented.

This forced automotive industries to the invention of different speed wiper motors increasing the visibility of driver. But again the major problem arises with this system is, driver has to change the speed of motor manually which will affect the speed of vehicle and also distracts the concentration of driver.

This newly proposed automatic wiper system will provides the solutions for all the above said problems. The system presented here uses two types of sensors (Water drop sensing plate and IR sensor) to detect rain drops or any liquid fallen on windshield. These sensors will activate the control unit which contents microcontroller and other circuitry. The AT89S51 Microcontroller is used in this system to make it cost effective and compact. According to the signals wiper motor is activated and the wiper speed is controlled as per the sensor activations.

METHODOLOGY

In every four wheeled vehicle, the wiper movements are done by the windshield wiper motor. By adjusting the speed of this motor, wiper speed can also be controlled. The system proposed here is able to detect and process the data received from sensors, then controls the speed of wiper.

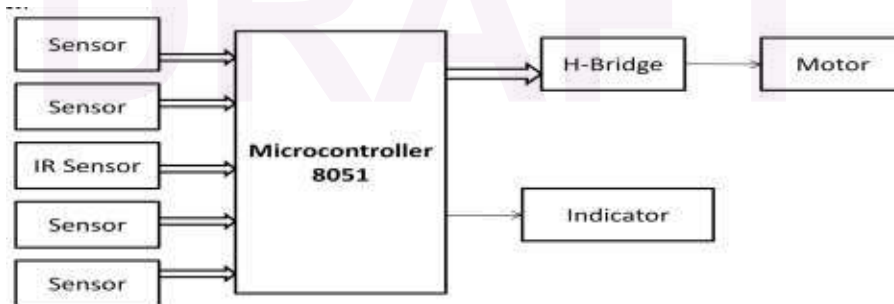


Fig. 1 Block Diagram of an automated wiper system

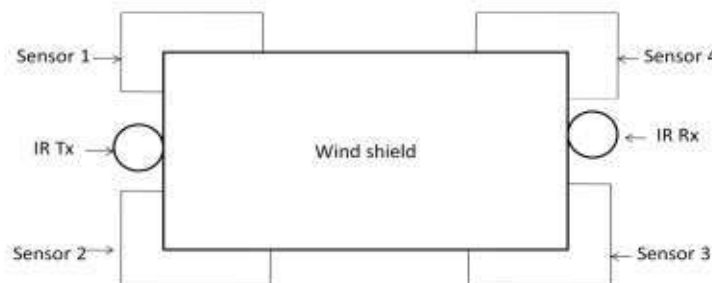


Fig. 2 Positions of rain sensors on the windshield

Smart City - Identify, select, install and troubleshoot different module/ devices used in Smart Traffic

Apply Linear Displacement sensor for structural crack monitoring

Objective:

- to apply Linear Displacement sensor for structural crack monitoring.

Requirements:

- Analog to Digital converter
- Digital to Analog converter
- Raspberry Pi
- Piezoelectric sensor

PROCEDURE:

Increase in the demand for reliable structural health information led to the development of Structural Health Monitoring (SHM). Prediction of upcoming accidents and estimation of useful life span of a structure are facilitated through SHM. While data sensing is the core of any SHM, tracking the data anytime anywhere is a prevailing challenge. With the advancement in information technology, the concept of Internet of Things (IoT) has made it possible to integrate SHM with Internet to track data anytime anywhere. In this paper, a SHM platform embedded with IoT is proposed to detect the size and location of damage in structures.

The proposed platform consists of a Wi-Fi module, a Raspberry Pi, an Analog to Digital Converter (ADC), a Digital to Analog Converter (DAC), a buffer, and piezoelectric (PZT) sensors. The piezoelectric sensors are mounted as a pair in the structure. Data collected from the piezoelectric sensors will be used to detect the size and location of damage using a proposed mathematical model.

Implemented on a Raspberry Pi, the proposed mathematical model will estimate the size and location of structural damage, if any, and upload the data to Internet. This data will be stored and can be checked remotely from any mobile device. The system has been validated using a real test bed in the lab.

SHM Damage Detection Techniques:

Damage detection in SHM systems is classified into two types:

local-based

global-based damage detection

Localbased damage detection is used for screening structures and global-based damage detection is used for vibrational characteristics. Various sensors are used to detect the size and location of the damage in SHM systems. Sensors like piezoelectric, optical fiber, ultrasonic, laser, image detection, and vibrational sensors are commonly used because of their leverage over other sensors.

Piezoelectric sensors are the most commonly used sensors; they can transmit and receive guided waves such as lamb waves in solids that can be used for damage detection. These lamb waves are much more cost effective as well as reliable as they are sensitive to change in structures. To identify the damage, the difference signal is acquired and compared with a damage-free signal. For structural damage localization, irrespective of the geometrical or imaging method used, the key to this process is the acquired time of flight and amplitude of the response to the signal. These factors directly determine the precision of damage localization. The time of flight is directly dependent on the properties of the material such as modulus of elasticity and modulus of rigidity. Using piezoelectric sensors, there are two basic active sensing techniques used to detect damage in structures, namely, pitch-catch and pulse-echo. In the pitch-catch technique, two PZT sensors are used: one as transmitter that located before the damage and the other as a receiver that located after the damage. The first PZT sends a signal, and the received signal by the second PZT is then used to determine the location and/or the extent of the damage.

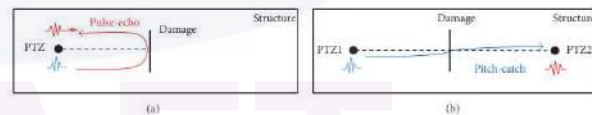
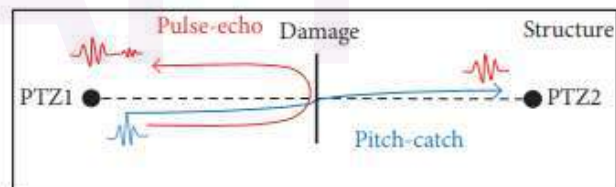


FIGURE 1: Guided waves damage detection techniques: (a) pulse-echo and (b) pitch-catch.



SHM Technique

On the other hand, the pulse-echo technique relies on the reflected wave from the damage, and one PTZ transducer is used for both transmitting and receiving the signal. Figure 1 shows both techniques.

SHM Technique

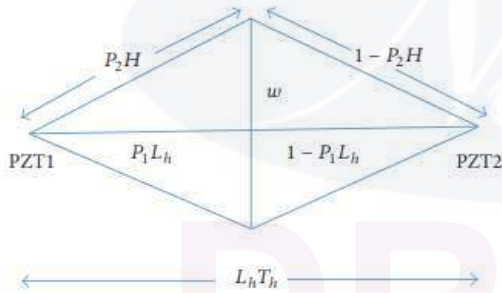
a combination of pulse-echo and pitch-catch techniques is proposed as shown in Figure. Pitch-catch technique is used to determine presence of damage. If any damage is found, the pulse-echo technique will be used to determine location of the damage. This method uses two PZT sensors. The first transducer (PZT1) excites

the signal towards the second transducer (PZT2) and picks up any waves that have been reflected back from damage that could not reach the second transducer, which uses the pulse-echo technique. The second, PTZ2, will receive wave feedback in a pitch catch manner. The proposed technique is good for any metal structure, for example, aluminum and steel.

Structure, for example, aluminum and steel. In order to detect if the structure is healthy or not, a cross correlation (CC) algorithm is used to check the similarity between excitation signal, E, at PTZ1 and received signal, R, at PTZ2. The CC value is used as a linear damage index and is defined as

$$CC = \frac{(1/N) \sum_{i=1}^N (E_i - \bar{E})(R_i - \bar{R})}{\sigma_E \sigma_R}, \quad (1)$$

where E and R are the mean values of the two sets of signal and σ_E and σ_R are standard deviations of the signature signal sets E and R, respectively. If the two signals are similar, $CC \sim 1$, which mean that the structure is healthy; otherwise, the structure has damage. In order to determine the size and location of the damage, a mathematical model is proposed based on the dimensional diagram shown in Figure.



Dimensional diagram for SHM technique.

First, the speed of the wave must be calculated using a healthy model as in

$$\frac{L_h}{T_h} = W_s, \quad (2)$$

where L_h is the length from exciter to sensor and T_h is the time between peak-peak. In order to find the damage location, L_c , we are using the following:

$$\frac{T_c}{2} W_s = L_c, \quad (3)$$

where T_c is the time it takes for a wave to travel to the damage and reflect back. After calculating the position, the next step would be determining the damage width. First, the proportion of the damage position to the total length, P_1 , could be determined by

$$\frac{L_c}{L_h} = P_1. \quad (4)$$

Due to the damage being present, the wave takes longer to travel from exciter to sensor, which implies that the wave travels a further distance. This distance, H , can be determined by

$$T_t W_s = H, \quad (5)$$

where T_t is the total time the wave takes to travel from exciter to the sensor with the presence of the damage. From these equations, a triangle can be made as seen in Figure 3. L_c is the base and $P_2 H$ is hypotenuse of this triangle. P_1 and P_2 are very close in value and only equal if P_1 is $1/2$. P_2 can be determined using the following:

$$\frac{(H^2 - L_c^2 + 2L_c^2 P_1)}{2H^2} = P_2. \quad (6)$$

After P_2 is calculated, the width of the damage is determined from the exciter/sensor axis using the Pythagorean Theorem:

$$\sqrt{(P_2 H)^2 - L_c^2} = W. \quad (7)$$

These equations are used if the damage has the same width above and below the axis. If the damage is not symmetrical, the two waves might reach the sensor at slightly different times resulting in two T_t s, two H s, and two P_2 s to calculate the upper and lower part of the damage. From the above mathematical model, the damage location can be detected using (5) and the damage width can be determined using (7). One unique advantage of the proposed mathematical model is that it does not require high computational processing unit, allowing for implementation almost anywhere, anytime, and also easier integration with an IoT platform. Figure shows the flow chart of the proposed SHM technique.

Select and install PH, Cupric (Cu²⁺), Silver (Ag⁺), Lithium (Li⁺), conductivity and temperature for maintenance of water quality

Objective:

- to select and install PH, Cupric (Cu²⁺), Silver (Ag⁺), Lithium (Li⁺), conductivity and temperature for maintenance of water quality.

Requirement:

- Arduino UNO
- PH sensor
- Conductivity sensor
- Temperature sensor

PROCEDURE:

NEED OF WATER QUALITY MONITORING IN SHRIMP CULTURE

Need of Water Quality in Shrimp Ponds

Water Quality (WQ) determines the ultimate Success or Failure of an aqua culture. The farmer must measure, record and manage WQ all through the growing season. Water Quality parameters affect respiration, feeding, metabolism and reproduction of shrimps.



Water Quality Parameters

- pH
- Temperature
- Dissolved Oxygen
- Hardness
- Ammonia, Nitrite & Nitrate

Water Quality Parameters

Water Parameter	Optimum level
Temperature	26-33C
Salinity	10-25 ppm
Dissolved oxygen	>3ppm
PH	7.5-8.5
Total Ammonia Nitrogen	<1.0ppm
Total Nitrate Nitrogen	<5.0ppm
Nitrite Nitrogen	<0.01ppm
Sulphide	<0.03ppm
Biological Oxygen Demand	<10ppm
Chemical Oxygen Demand	<70ppm
	(parts per million)

Formers take water samples from pond.

- Get results from Aqua Labs.
- Expensive.
- Difficult Process.
- Take More Time.

Install Smart dustbin

Objective of the project:

- keep our Environment clean
- aims at creating a clean as well as Green environment.

Requirements:

Materials Required:

- Ultrasonic Sensor
- Arduino
- Servo Motor S9
- Jumper Wire
- Switch
- Battery
- Battery Clipper

PROCEDURE:

In the recent decades, Urbanization has increased tremendously. At the same phase there is an increase in waste production. Waste management has been a crucial issue to be considered. This paper is a way to achieve this good cause.

In this paper, smart bin is built on a microcontroller based platform Arduino Uno board which is interfaced with GSM modem and Ultrasonic sensor. Ultrasonic sensor is placed at the top of the dustbin which will measure the stature of the dustbin. The threshold stature is set as 10cm. Arduino will be programmed in such a way that when the dustbin is being filled, the remaining height from the threshold height will be displayed.

Once the garbage reaches the threshold level ultrasonic sensor will trigger the GSM modem which will continuously alert the required authority until the garbage in the dustbin is squashed. Once the dustbin is squashed, people can reuse the dustbin. At regular intervals dustbin will be squashed.

Once these smart bins are implemented on a large scale, by replacing our traditional bins present today, waste can be managed efficiently as it avoids unnecessary lumping of wastes on roadside. Foul smell from these rotten wastes that remain untreated for a long time, due to negligence of authorities and carelessness of public may lead to long term problems.[1]

Breeding of insects and mosquitoes can create nuisance around promoting unclean environment. This may even cause dreadful diseases.

1. Problem Statement

To implement a smart bin built on a microcontroller based platform Arduino Uno board which is interfaced with GSM modem and Ultrasonic sensor which can give the status

of the waste present in the dustbin to the municipal authority.

2. System design:

For communication purpose Bluetooth technology can also be used in the transmitter section. Bluetooth is a wireless networking standard that is aimed at remote control and sensor applications which is suitable for operation in harsh radio environments and in isolated locations. But, the main disadvantages of Bluetooth is short range, low complexity, and low data speed.

Therefore, GSM is more advantages over Bluetooth for communication. Hence author use GSM modem. A GSM modem is a specialized type wireless modem that works with a GSM wireless network. It accepts a SIM card, and operates over a subscription to a mobile operator, just like a mobile phone.

A GSM modem can be an external device or a PC Card / PCMCIA Card. An external GSM modem is connected to a computer through a serial cable or a USB cable. When a GSM modem is connected to a computer, this allows the computer to communicate over the mobile network.

While these GSM modems are most frequently used to provide mobile 4 internet connectivity, many of them can also be used for sending and receiving SMS and MMS message. GSM Modem sends and receives data through radio waves. In this project GSM 900 modem is used to send the messages which is shown in figure 2.[2]

It consists of a GSM/GPRS modem with standard communication interfaces like RS-232 (Serial Port), USB, so that it can be easily connected to the other devices. The power supply circuit is also built in the module that can be turned ON by using a suitable adaptor.

Fig 1: GSM

The block diagram of transmitter section. Level detector consists of IR sensors which is used to detect the level of the garbage in the dustbin. The output of level detector is given to microcontroller.

Four IR sensors are used to indicate the different levels of the amount of the garbage collected in the dustbin which is placed in public area. When the dustbin is filled up to the highest level, the output of fourth IR receiver becomes active low.

This output is given to microcontroller to send the message to the Control room via GSM module as shown in below.

5 - Fig-2 Transmitter Part from Dustbin

The figure 4 shows the block diagram of receiver section. At receiver, control room is present where all the activities are managing. The number of the control room is depending on the dustbins present in the area.

The person sitting in the control room monitors the entire system. A GSM Module is connected to the computer of the control room through microcontroller. The entire system is monitor by the person sitting in the control room.

The same GSM Module is used to send the message to the contractor for cleaning the dustbin. GUI is developed using MATLAB software. This GUI will be displayed on the computer screen in the control room to display the status of the garbage level in the dust bin as shown in below.

Fig-3: Reception Part to Dustbin

6 2.1 Flow chat of Woking Principle:

Fig 4: Flowchart

The PIR sensor will observe a person nearby dustbin. If motion is detected the lid of dustbin is opened, the servo motor activates and as GSM connected it will send an alert message to user if dustbin is filled.

Dustbin placed in public place, people throw garbage in dustbin, place the ultrasonic sensor in top of the garbage bin. If dustbin reach in 75% then arduino send message through GSM module.

When dustbin level is reach threshold level buzzer will give alert sound for don't again put waste in dustbin. This all process updated in IOT GECKO platform for monitoring garbage bin.

7

Working of Line Follower Robot using Arduino
Working of line follower is very interesting. Line follower robot senses black line by using sensor and then sends the signal to Arduino. Then Arduino drives the motor according to sensors' output. Working of line follower Here in this project we are using two IR sensor modules namely left sensor and right sensor. When both left and right sensor senses white then robot move forward.

2.2 Literature survey:

The authors in [4] have made a quantitative analysis between existing dustbins and their serving population. The study first analyses the spatial distribution of dustbins in some areas of Dhaka city using average nearest neighbor functions of GIS. Remarkably, the spatial circulation of the current dustbins has appeared to be dominatingly in clustered pattern.

Next, an optimal number of additional dustbins were calculated. It is shown that the number of existing dustbins is insufficient in the study area. The extent of pollution caused by the existing dustbins was calculated

using spatial analyst functions of GIS. It is found that all the dustbins are burnt with wastes and causing pollution to the environment.

The results thus obtained would help to understand the present situation of the waste management of Research Article Volume 6 Issue No. 6 International Journal of Engineering Science and Computing, June 2016 7114 <http://ijesc.org/> Dhaka city and to optimally place the required number of dustbins to prevent further pollution to environment.

8

The authors in [5] have equipped the smart bins with ultrasonic sensors which measure the level of dustbin being filled up. The container is divided into three levels of garbage being collected in it. Every time the garbage crosses a level the sensors receives the data of the filled level.

This data is further sent to the garbage analyzer as instant message using GSM module. Placing three ultrasonic sensors at three different levels of the container may be a disadvantage as the cost of the dustbin increases due to the sensors and also the sensors can be damaged due to the rough action by the users.

An IoT-based smart garbage system (SGS) is proposed to reduce the amount of food waste. The authors in [6]. In an SGS, battery-based smart garbage bins (SGBs) exchange information with each other using wireless mesh networks, and a router and server collect and analyze the information for service provisioning. Furthermore, the SGS includes various IoT skills considering user convenience and increases the battery lifetime through two types of energy-efficient operations of the SGBs: stand-alone operation and cooperation based operation.

The proposed SGS had been functioned as a pilot project in Gangnam district, Seoul, Republic of Korea, for a one-year period. The test demonstrated that the normal measure of food waste could be decreased by 33%.

The authors in [7] has built a framework in which a Camera will be set at each garbage collection point alongside load cell sensor at base of the trash can. The camera will take continuous snapshots of the garbage can. A threshold level is set which compares the output of camera and load sensor. The comparison is done with help of microcontroller.

After analyzing the image an idea about level of garbage in the can and from the load cell sensor, weight of garbage can be known. Accordingly, information is processed that is controller checks if the threshold level is exceeded or not. This is convenient to use but economically not reliable.

3. Requirement Specification-

Smart bin is built on Arduino board platform. It is interfaced with a GSM modem (SIM 900A) and the bin is equipped with Ultrasonic sensor (HC-SR04) and PIR sensor.

3.1) Components Used-

1. Arduino
2. GSM
3. PIR Sensor
4. Ultrasonic Sensor
5. Servo Motor
6. Breadboard
7. Connecting wires

1. Arduino

Arduino is an open source, computer hardware and software company, project, and user community that designs and manufactures microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world.

The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself kits. Arduino board designs use a variety of microprocessors and controllers.

The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers.[3]

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.

It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE)

1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

The sensors will monitor activity—including bin fullness,

temperature and fill rates—in order to identify waste generation patterns. Collected data will help the city mitigate container overflow, reduce the number of overall collections, guide bin placement and optimize operational resources.

"Nordsense uses best-in-class laser technology that shoots multiple independent laser beams onto the waste surface and calculates the distance for each of those," says Manuel Maestrini, founder and chief technology officer for Nordsense. "This allows us to be extremely accurate as to when to dispatch someone to pick up the can and not being sensitive to false positives."

The city works with Recology, its private sector partner, to service the cans. Currently, the fill level, temperature location and position of the bins will be part of the data collected by the sensors.

"Our goal is to improve the cleanliness of San Francisco's public areas and to gain a better understanding, through real-time data, of the usage of our public trash receptacles," says Nuru. "If we can see how they're being used—how quickly they fill up—we can ensure that specific cans get prompt attention before they overflow, and we can determine whether more trash cans in a specific area are needed. At the same time, we can reduce the amount of servicing of public trash cans that get little use, providing more efficiency."

The contract with Nordsense is a direct result of the San Francisco's Startup in Residence (STIR) program that connects governments with startups to solve a broad range of civic challenges.

"San Francisco Public Works considered several proposals for smart can technology to help improve the cleanliness of San Francisco," says Nuru.

In 2018, as part of the program, Nordsense collaborated with San Francisco officials to complete a highly successful three-month pilot project.

"The pilot gave us the opportunity to see how the technology works and to make necessary adjustments," This included installing 48 sensors across the city to create public "smart" bins that collected real-time data about trash levels and patterns.

"The outcome of the pilot was beyond anyone's expectation—we saw an 80 percent decrease in overflowing cans, a 64 percent decrease in illegal dumping and a 66 percent decrease in street cleaning service requests," The project also generated data revealing how empty or full bins were—on average—when they were serviced. This information indicated increasing the number of bins overall would have less impact on the cleanliness of city streets than smarter distribution of bins based on measurable activity.

Using data collected from the 1,000 smart bins, San Francisco will be able to optimize bin placement and identify areas where more recycling options are necessary to reduce the amount of material transported to landfills.

“Down the road, this data will not only contribute to operational efficiencies and cleaner streets but also in identifying what's the optimal can's placement based on actual data instead of assumptions,”

(i) Power USB

Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection

(ii) Power (Barrel Jack)

Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack.

(iii) Voltage Regulator

The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

(iv) Crystal Oscillator

The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.

(v, xvii) Arduino Reset

You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).

(vi, vii, viii, ix) Pins (3.3, 5, GND, Vin)

3.3V (6) - Supply 3.3 output volt

5V (7) - Supply 5 output volt

Most of the components used with Arduino board works fine with 3.3 volt and 5 volt.

GND (8)(Ground) - There are several GND pins on the Arduino, any of which can be used to ground your circuit.

Vin (9) - This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.

(x) Analog pins

The Arduino UNO board has five analog input pins A0 through A5.

These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.

(xi) Main microcontroller

Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board

has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.

(xii) ICSP pin

Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.

(xiii) Power LED indicator

This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.

(xiv) TX and RX LEDs

On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

(xv) Digital I/O

The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labelled can be used to generate PWM.

(xvi) AREF

AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Microcontroller- ATmega2560

Operating Voltage -5V

Input Voltage (recommended)- 7-12V

Input Voltage (limit)- 6-20V

Digital I/O Pins 54 (of which 15 provide PWM output)

Analog Input 16

2. Gsm (Global System for Mobile Communication)

GSM Modem can accept any GSM network operator SIM card and act just like a mobile phone with its own unique phone number. Advantage of using this modem will be that you can use its RS232 port to communicate and develop embedded applications.

Applications like SMS Control, data transfer, remote control and logging can be developed easily using gsm. The modem can either be connected to PC serial port directly or to any microcontroller through

MAX232. It can be used to send and receive SMS or make/receive voice calls. It can also be used in GPRS mode to connect to internet and do many applications for data logging and control. In GPRS mode you can also connect to any remote FTP server and upload files for data logging.

This GSM modem is a highly flexible plug and play quad band SIM900A GSM modem for direct and easy integration to RS232 applications. Supports features like Voice, SMS, Data/Fax, GPRS and integrated TCP/IP stack.

A GSM network comprises of many functional units. These functions and interfaces are explained in this chapter. The GSM network can be broadly divided into:

- The Mobile Station (MS)
- The Base Station Subsystem (BSS)
- The Network Switching Subsystem (NSS)
- The Operation Support Subsystem (OSS)

Given below is a simple pictorial view of the GSM architecture.

Fig 6: Architecture of GSM

The additional components of the GSM architecture comprise of databases and messaging systems functions:

- Home Location Register (HLR)
- Visitor Location Register (VLR)
- Equipment Identity Register (EIR)
- Authentication Centre (AuC)
- SMS Serving Centre (SMS SC)
- Gateway MSC (GMSC)
- Chargeback Centre (CBC)
- Transcoder and Adaptation Unit (TRAU)

The following diagram shows the GSM network along with the added elements:

Fig 7: GSM Network architecture

The MS and the BSS communicate across the Um interface. It is also known as the air interface or the radio link. The BSS communicates with the Network Service Switching (NSS) centre across the A interface.

GSM network areas In a GSM network, the following areas are defined:

Cell: Cell is the basic service area; one BTS covers one cell. Each cell is given a Cell Global Identity (CGI), a number that uniquely identifies the cell.

Location Area: A group of cells form a Location Area (LA). This is the area that is paged when a subscriber gets an incoming call. Each LA is assigned a Location Area Identity (LAI). Each LA is served by one or more BSCs. MSC/VLR Service Area: The area covered by one MSC is called the MSC/VLR service area.

PLMN: The area covered by one network operator is called the Public Land Mobile Network (PLMN). A PLMN can contain one or more MSCs.

3. PIR Sensor

PIR sensors allow you to sense motion. They are used to detect whether a human has moved in or out of the sensor's range. They are commonly found in appliances and gadgets used at home or for businesses. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors.

Following are the advantages of PIR Sensors ?

- Small in size
- Wide lens range
- Easy to interface
- Inexpensive
- Low-power
- Easy to use
- Do not wear out

Fig 8: PIR sensor

PIRs are made of pyroelectric sensors, a round metal can with a rectangular crystal in the centre, which can detect levels of infrared radiation. Everything emits low-level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is split in two halves.[4] This is to detect motion (change) and not average IR levels.

The two halves are connected so that they cancel out each other. If one-half sees more or less IR radiation than the other, the output will swing high or low. PIRs have adjustable settings and have a header installed in the 3-pin ground/out/power pads.

Fig 9: PIR Sensor Diagram

4. ULTRASONIC SENSOR

The HC-SR04 ultrasonic sensor uses SONAR to determine the distance of an object just like the bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package from 2 cm to 400 cm or 1" to 13 feet.

The operation is not affected by sunlight or black material, although acoustically, soft materials like cloth can be difficult to detect. It comes complete with ultrasonic transmitter and receiver module.

Fig 10: Ultra sonic sensor

Technical Specifications

Power Supply - +5V DC

Quiescent Current - <2mA

Working Current - 15mA

Effectual Angle - <15°

Ranging Distance ? 2cm – 400 cm/1" – 13ft

Resolution - 0.3 cm

Measuring Angle - 30 degree

5. Servo Motor

A Servo Motor is a small device that has an output shaft. This shaft can be positioned to specific angular positions by sending the servo a coded signal. As long as the coded signal exists on the input line, the servo will maintain the angular position of the shaft.

If the coded signal changes, the angular position of the shaft changes. [8] In practice, servos are used in radio-controlled airplanes to position control surfaces like the elevators and rudders. They are also used in radio-controlled cars, puppets, and of course, robots.

Fig 11: Servo Motor

Servos are extremely useful in robotics. The motors are small, have built-in control circuitry, and are extremely powerful for their size. A standard servo such as the Futaba S-148 has 42 oz/inches of torque, which is strong for its size. It also draws power proportional to the mechanical load. A lightly loaded servo, therefore, does not consume much energy.

The guts of a servo motor is shown in the following picture. You can see the control circuitry, the motor, a set of gears, and the case. You can also see the 3 wires that connect to the outside world. One is for power (+5volts), ground, and the white wire is the control wire.

Fig 12: Servo motor parts

Working of a Servo Motor

The servo motor has some control circuits and a potentiometer (a variable resistor, aka pot) connected to the output shaft. In the picture above, the pot can be seen on the right side of the circuit board. This pot allows the control circuitry to monitor the current angle of the servo motor.

If the shaft is at the correct angle, then the motor shuts off. If the circuit finds that the angle is not correct, it will turn the motor until it is at a desired angle. The output shaft of the servo is capable of traveling somewhere around 180 degrees. Usually, it is somewhere in the 210-degree range, however, it varies depending on the manufacturer.

A normal servo is used to control an angular motion of 0 to 180 degrees. It is mechanically not capable of turning any farther due to a mechanical stop built on to the main output gear.

The power applied to the motor is proportional to the distance it needs to travel. So, if the shaft needs to turn a large distance, the motor will run at full speed. If it needs to turn only a small amount, the motor will run at a slower speed. This is called proportional control.

How Do You Communicate the Angle at Which the Servo Should Turn?

The control wire is used to communicate the angle. The angle is determined by the duration of a pulse that is applied to the control wire. This is called Pulse Coded Modulation. The servo expects to see a pulse every 20 milliseconds (.02 seconds).

The length of the pulse will determine how far the motor turns. A 1.5 millisecond pulse, for example, will make the motor turn to the 90-degree position (often called as the neutral position). If the pulse is shorter than 1.5 milliseconds, then the motor will turn the shaft closer to 0 degrees. If the pulse is longer than 1.5 milliseconds, the shaft turns closer to 180 degrees.

6. Breadboard

A breadboard is a construction base for prototyping of electronics. Originally it was literally a bread board, a polished piece of wood used for slicing bread. Because the solderless breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solderless breadboards are also extremely popular with students and in technological education. Older breadboard types did not have this property.

A stripboard (Vero board) and similar prototyping printed circuit boards, which are used to build semi-permanent soldered prototypes or one-offs, cannot easily be reused. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units (CPUs).

Install GPS based tracking system for smart bin

Objective:

- for making smart cities Municipal corporations needs a smart solution also.

Requirement:**Hardware Materials Required:**

- Grover Base Shield v2
- Arduino 101
- 9v Battery these batteries will power the Arduino board
- Plastic Container
- Ultrasonic Sensor
- Jumper Wires
- Arduino MKR1000
- Electric Drill
- Glue

Software Required:

- Arduino IDE
- BLYNK Android App that allows communication with WiFi compatible micro-controllers.

PROCEDURE:**Introduction to the project**

The Internet of Things (IoT) shall be able to connect to a large number of different systems, while providing data for millions of people to use and capitalize. Building a general architecture for the IoT is hence a very complex task, mainly because of the extremely large variety of devices, link layer technologies, and services that may be involved in such a system.

What does GPS Tracking of Trashcan can do?

It gives a real time indicator of the garbage level in a trashcan at any given time. Using that data we can then optimize waste collection routes and ultimately reduce fuel consumption. It allows trash collectors to plan their daily/weekly pick up schedule.

Construct the Model

Look for an old small plastic container and make sure your components fit.



Fig 1

Now remove the lid and trace the two "eyes" of the ultrasonic sensor. this will be the side facing the bottom of the bin.



Fig 2

Take your biggest drill bit, mine was 10mm and drill out the holes. If they still are a bit small lightly file them till the ultrasonic sensor Fits in.



Fig 3

Fig 4



Fig 5

Components in the box

Maintain dry waste and wet waste separately

Objective:

- to maintain dry waste and wet waste separately.

Requirements:

Components needed

- Arduino MKR Fox 1200
- Ultrasonic Sensor - HC-SR04 (Generic)
- DHT11 Temperature & Humidity Sensor (4 pins)
- Pycom LoRa / Sigfox Antenna
- SparkFun Mini Microswitch
- KY-026 Keyes - Flame Sensor Module
- Resistor 100 ohm
- General Purpose Transistor PNP
- Garbage container
- Raspberry Pi 3 Model B

Necessary Tools and Machines

- Soldering iron (generic)

Apps and Online Services

- Sigfox
- Arduino IDE
- Autodesk Fusion 360

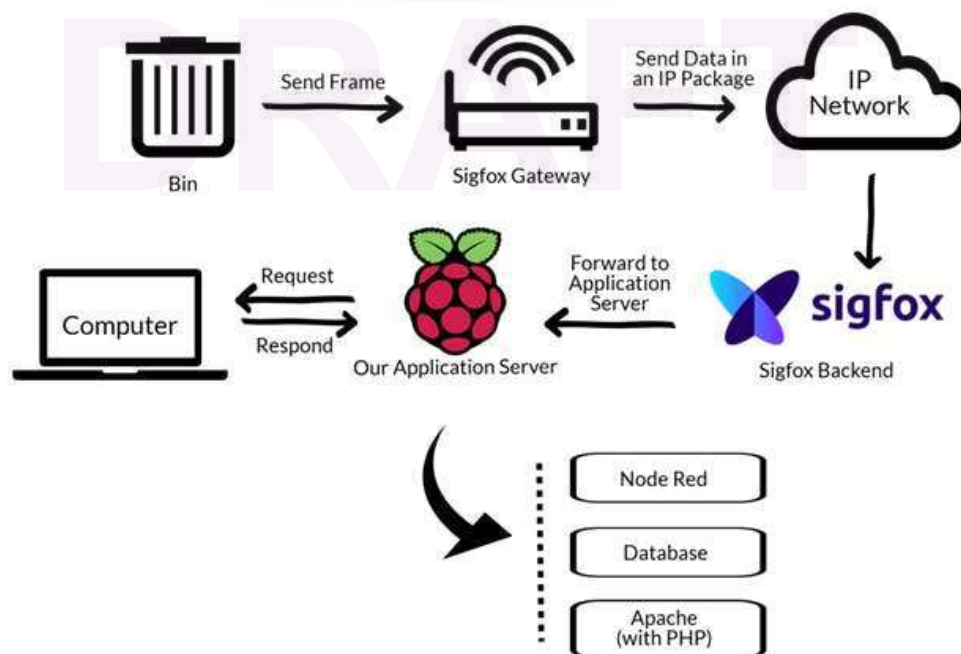
PROCEDURE:

What is waste segregation?

Waste segregation refers to the separation of wet waste and dry waste, the purpose is to recycle dry waste easily and to use wet waste as compost.

Why should we segregate waste?

When we segregate waste, there is reduction of waste that gets landfilled and occupies space, air and water pollution rates are considerably lowered. Segregating waste also makes it easier to apply different processes - composting, recycling and incineration can be applied to different kinds of waste.



Install, test and apply different components like ultrasonic sensor, wifi module and Thing speak (IoT Platform)

Objective:

- to install, test and apply different components like ultrasonic sensor, Wi-Fi module and Thing speak (IoT Platform).

Requirements:

- Thing speak
- Arduino
- Microcontroller
- Internet connectivity
- Computer system

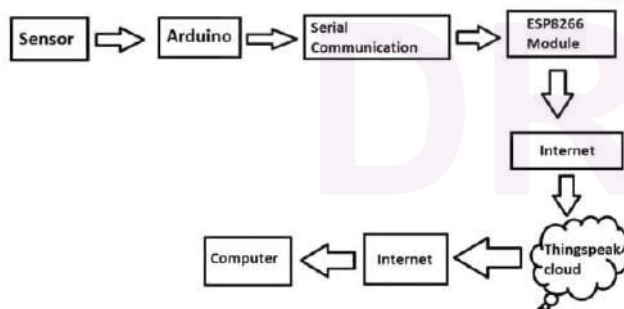
PROCEDURE:

We are going to learn how to upload ultrasonic sensor (HC-SR04) data to Thingspeak using Arduino and generic ESP8266 Wi-Fi module

Sending Measured Data to Thingspeak Using Arduino and ESP8266

Ultrasonic sensor need no introduction it is one of the first sensors you might have tried. Now you are learning about IoT and you came across Thingspeak and you want to send some data to Thingspeak server and you chose ultrasonic sensor.

Block Diagram



Block Diagram for Ultrasonic sensor Data to Thingspeak

Here is the block diagram on how we are going to upload ultrasonic sensor data to Thingspeak. Firstly the Arduino will trigger the ultrasonic sensor and collects the data,

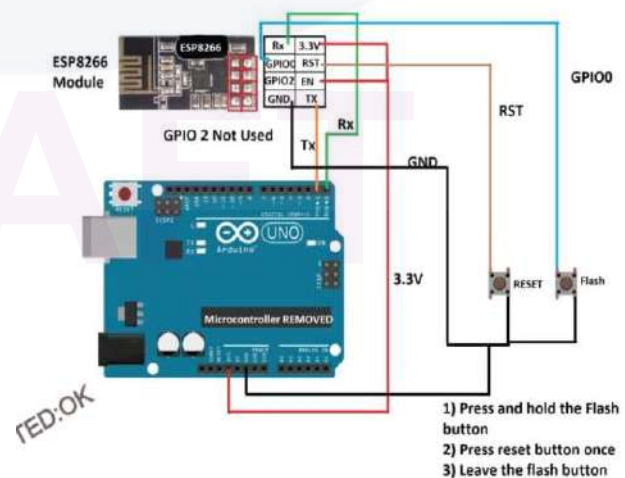
after calculating the distance arduino will convey this data to ESP8266 module via UART / serial communication. The serial data consists of distance data in cm with 3 to 5 decimal places. The ESP8266 will send this data to your Thingspeak account via internet.

Since we are using a generic ESP8266 module we have to upload an appropriate program code that makes ESP8266 to accept serial data and send it to Thingspeak. We also need to upload another program code to Arduino board, so that it will convert the measured ultrasonic sensor data in serial data.

Now let's see how to upload the program code to ESP8266 module.

Circuit Diagram

You need to wire up the circuit as illustrated above and remember that ESP8266 works on 3.3V and 5V will kill your module and make sure that ATmega328P is removed from Arduino board.



Uploading code to ESP8266