IOT LECTURER NOTE 6th SEM CSE DIPLOMA

DETAIL CONTENTS:

1. Introduction to Internet of Things

1.1 Introduction
1.2 Characteristics of IoT
1.3 Applications of IoT
1.4 IoT Categories

1.5 IoT Enablers and connectivity layers
1.6 Baseline Technologies
1.1 Sensor
1.2 Actuator
1.3 IoT components and implementation
1.4 Challenges for IoT

2. **IOT Networking**

2.1 Terminologies
2.2 Gateway Prefix allotment
2.3 Impact of mobility on Addressing
2.4 Multihoming
2.5 Deviation from regular Web
2.6 IoT identification and Data protocols

3. **Connectivity Technologies**

3.1 Introduction
3.2 IEEE 802.
3.3 ZigBee, 6LoWPAN
3.4 RFID, HART and wireless HART
3.5 NFC, Bluetooth, Z wave, ISA100.11.A

4. Wireless Sensor Networks

4.1 Introduction
4.2 Components of a sensor node
4.3 Modes of Detection
4.4 Challenges in WSN
4.5 Sensor Web
4.6 Cooperation and Behaviour of Nodes in WSN
4.7 Self Management of WSN
4.8 Social sensing WSN
4.9 Application of WSN
4.10 Wireless Multimedia sensor network
4.11 Wireless Nanosensor Networks
4.12 Underwater acoustic sensor networks
4.13 WSN Coverage
4.14 Stationary WSN, Mobile WSN

5. **M2M Communication**

5.1 M2M communication
5.2 M2M Ecosystem
5.3 M2M service Platform
5.4 Interoperability

6. **Programming with Arduino**

6.1 Features of Arduino
6.2 Components of Arduino Board
6.3 Arduino IDE
6.4 Case Studies

7. Programming with Raspberry Pi

7.1 Architecture and Pin Configuration
7.2 Case studies
7.3 Implementation of IoT with Raspberry Pi

8. Software defined Networking

8.1 Limitation of current network
8.2 Origin of SDN

8.3 SDN Architecture
8.4 Rule Placement, Open flow Protocol
8.5 Controller placement
8.6 Security in SDN
8.7 Integrating SDN in IoT

9. Smart Homes

9.1 Origin and example of Smart Home Technologies
9.2 Smart Home Implementation

9.3 Home Area Networks(HAN)
9.4 Smart Home benefits and issues

10. Smart Cities

10.1 Characteristics of Smart Cities

10.2 Smart city Frameworks
10.3 Challenges in Smart cities
10.4 Data Fusion
10.5 Smart Parking
10.6 Energy Management in Smart cities

11. Industrial IoT

11.1 IIoT requirements
11.2 Design considerations
11.3 Applications of IIoT
11.4 Benefits of IIoT
11.5 Challenges of IIoT

DETAILS OF LECTURER NOTE:

1. Introduction to Internet of Things

“The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.”

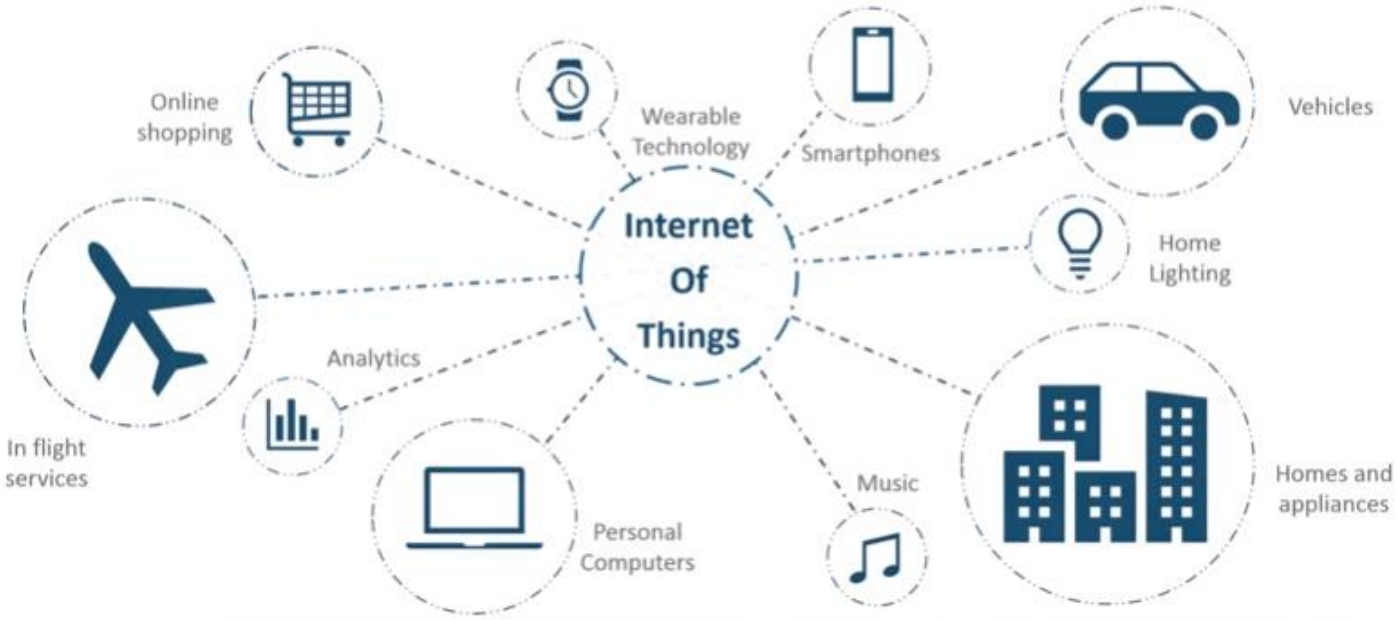
IoT stands for Internet of Things, which means accessing and controlling daily usable equipments and devices using Internet.

1.1 Introduction

Let's us look closely at our mobile device which contains GPS Tracking, Mobile Gyroscope, Adaptive brightness, Voice detection, Face detection etc. These components have their own individual features, but what about if these all communicate with each other to provide a better environment? For example, the phone brightness is adjusted based on my GPS location or my direction.

Connecting everyday things embedded with electronics, software, and sensors to internet enabling to collect and exchange data without human interaction called as the Internet of Things (IoT).

The term "Things" in the Internet of Things refers to anything and everything in day to day life which is accessed or connected through the internet.



1.2 Characteristics of IoT

Characteristics of the Internet of Things :

There are the following characteristics of IoT as follows. Let's discuss it one by one.

1. **Connectivity –**
   Connectivity is an important requirement of the IoT infrastructure. Things of IoT should be connected to the IoT infrastructure. Anyone, anywhere, anytime can connect, this should be guaranteed at all times. For example, connection between people through internet devices like mobile phones ,and other gadgets, also connection between Internet devices such as routers, gateways, sensors, etc.
2. **Intelligence and Identity –**
   The extraction of knowledge from the generated data is very important. For example, a sensor generates data, but that data will only be useful if it is interpreted properly. Each IoT device has a unique identity. This identification is helpful in tracking the equipment and at times for querying its status.

3. **Scalability –**
   The number of elements connected to the IoT zone is increasing day by day. Hence, an IoT setup should be capable of handling the massive expansion. The data generated as an outcome is enormous, and it should be handled appropriately.

4. **Dynamic and Self-Adapting (Complexity) –**
   IoT devices should dynamically adapt themselves to the changing contexts and scenarios. Assume a camera meant for the surveillance. It should be adaptable to work in different conditions and different light situations (morning, afternoon, night).

5. **Architecture –**
   IoT architecture cannot be homogeneous in nature. It should be hybrid, supporting different manufacturers ' products to function in the IoT network. IoT is not owned by anyone engineering branch. IoT is a reality when multiple domains come together.

6. **Safety –**
   There is a danger of the sensitive personal details of the users getting compromised when all his/her devices are connected to the internet. This can cause a loss to the user. Hence, data security is the major challenge. Besides, the equipment involved is huge. IoT networks may also be at the risk. Therefore, equipment safety is also critical.

1.3 Applications of IoT

1. Smart Agriculture

Food is an integral part of life without which we cannot survive. However, it is an unfortunate fact that a lot of food is wasted in developed countries like America while people starve in poorer countries like Chad, Sudan, etc. One way to feed everyone is better agricultural practices which can be enhanced using IoT. This can be done by first collecting data for a farm such as soil quality, sunlight levels, seed type, rainfall

density from various sources like farm sensors, satellites, local weather stations, etc. and then using this data with Machine Learning and IoT to create custom recommendations for each farm that will optimize the planting procedure, irrigation levels required, fertilizer amount, etc. All this will result in better yield or crops with a focus on reducing world hunger in the future. This is done very efficiently by SunCulture, which is an initiative by Microsoft AI for Earth.

2. Smart Vehicles

Smart vehicles or self-driving cars as they can be called are pretty dependent on IoT. These cars have a lot of features that are integrated with each other and need to communicate such as the sensors that handle navigation, various antennas, controls for speeding or slowing down, etc. Here the Internet of Things technology is critical especially in the sense that self-driving cars need to be extremely accurate and all the parts need to communicate with each other in milliseconds on the road. Tesla Cars are quite popular and working on their self-driving cars. Tesla Motors' cars use the latest advancements in Artificial Intelligence and the Internet of Things. And they are quite popular as well!!! Tesla Model 3 was the most sold plug-in electric car in the U.S. in 2018 with a total yearly sales of around 140,000 cars.

3. Smart Home

Maybe the most famous application of IoT is in Smart Homes. After all, who hasn't heard about connecting all the home applications like lighting, air conditioners, locks, thermostat, etc. into a single system that can be controlled from your smartphone! These IoT devices are becoming more and more popular these days because they allow you complete freedom to personalize your home as you want. In fact, these IoT devices are so popular that every second there are 127 new devices connected to the internet. Some popular ones that you might have heard have, or even have in your home, include Google Home, Amazon Echo Plus, Philips Hue Lighting System, etc. There are also all sorts of other inventions that you can install in your home including Nest Smoke Alarm and Thermostat, Foobot Air Quality Monitor, August Smart Lock, etc.

4. Smart Pollution Control

Pollution is one of the biggest problems in most of the cities in the world. Sometimes it's not clear if we are inhaling oxygen or smog! In such a situation, IoT can be a big help in controlling the pollution levels to more breathable standards. This can be done by collecting the data related to city pollution like emissions from vehicles, pollen levels, airflow direction, weather, traffic levels, etc using various sensors in combination with IoT. Using this data, Machine Learning algorithms can calculate pollution forecasts in different areas of the city that inform city officials beforehand where the problems are going to occur. Then they can try to control the pollution levels till it's much safer. An example of this is the Green Horizons project created by IBM's China Research Lab.

5. Smart Healthcare

There are many applications of IoT in the Healthcare Industry where doctors can monitor patients remotely through a web of interconnected devices and machines without needing to be in direct contact with them. This is very useful if the patients don't have any serious problems or if they have any infectious diseases like COVID-19 these days. One of the most common uses of IoT in healthcare is using robots. These include surgical robots that can help doctors in performing surgeries more efficiently with higher precision and control. There are also disinfectant robots that can clean surfaces quickly and thoroughly using high-intensity ultraviolet light (which is pretty useful these days!) Other types of robots also include nursing robots that can handle the monotonous tasks that nurses have to perform for many patients day in and day out where there is little risk to the patients.

6. Smart Cities

Cities can be made more efficient so that they require fewer resources and are more energy-efficient. This can be done with a combination of sensors in different capacities all over the city that can be used for various tasks ranging from managing the traffic, controlling handling waste management, creating smart buildings, optimizing streetlights, etc. There are many cities in the world that are working on incorporating IoT and becoming smarter such as Singapore, Geneva, Zurich, Oslo, etc. One example of creating smart cities is the Smart Nation Sensor Platform used by Singapore which is believed to be the smartest city in

the world. This platform integrates various facets of transportation, streetlights, public safety, urban planning, etc. using sensors in conjugation with IoT.

7. Smart Retail

There is a way to make shopping even more exciting for customers and that's to use the latest tech like IoT of course! Retail stores can make use of IoT in a wide range of operations to make shopping a much smoother experience for customers and also easier for the employees. IoT can be used to handle the inventory, improve store operations, reduce shoplifting and theft, and prevent long queues at the cashiers. A prime example of this is the Amazon Go stores which provide an IoT enabled shopping experience. These stores monitor all their products using IoT so that customers can pick up any products and just walk out of the store without stopping at the cashier's queue. The total bill amount is automatically deducted from the card associated with the customer's Amazon account after they leave the store.

1.4 IoT Categories

IoT devices can be classified into three main groups: **consumer, enterprise, and industrial**.

1.5 IoT Enablers and connectivity layers

Internet of Things (IoT) is the networking of physical objects that contain electronics embedded within their architecture in order to communicate and sense interactions amongst each other or with respect to the external environment. In the upcoming years, IoT-based technology will offer advanced levels of services and practically change the way people lead their daily lives. Advancements in medicine, power, gene therapies, agriculture, smart cities, and smart homes are just a very few of the categorical examples where IoT is strongly established.

IoT is network of interconnected computing devices which are embedded in everyday objects, enabling them to send and receive data.

Over 9 billion 'Things' (physical objects) are currently connected to the Internet, as of now. In the near future, this number is expected to rise to a whopping 20 billion.

Main components used in IoT:

- **Low-power embedded systems:** Less battery consumption, high performance are the inverse factors that play a significant role during the design of electronic systems.
- **Sensors :** Sensors are the major part of any IoT applications. It is a physical device that measures and detect certain physical quantity and convert it into signal which can be provide as an input to processing or control unit for analysis purpose.

1. Different types of Sensors :
2. Temperature Sensors
3. Image Sensors
4. Gyro Sensors
5. Obstacle Sensors
6. RF Sensor
7. IR Sensor
8. MQ-02/05 Gas Sensor
9. LDR Sensor
10. Ultrasonic Distance Sensor

- **Control Units :** It is a unit of small computer on a single integrated circuit containing microprocessor or processing core, memory and programmable input/output devices/peripherals. It is responsible for major processing work of IoT devices and all logical operations are carried out here.
- **Cloud computing:** Data collected through IoT devices is massive and this data has to be stored on a reliable storage server. This is where cloud computing comes into play. The data is processed and learned, giving more room for us to discover where things like electrical faults/errors are within the system.
- **Availability of big data:** We know that IoT relies heavily on sensors, especially in real-time. As these electronic devices spread throughout every field, their usage is going to trigger a massive flux of big data.
- **Networking connection:** In order to communicate, internet connectivity is a must where each physical object is represented by an IP address. However, there are only a limited number of addresses available according to the IP naming. Due to the growing number of devices, this naming system will not be feasible anymore. Therefore, researchers are looking for another alternative naming system to represent each physical object.

There are two ways of building IoT:

1. Form a separate internetwork including only physical objects.

2. Make the Internet ever more expansive, but this requires hard-core technologies such as rigorous cloud computing and rapid big data storage (expensive).

In the near future, IoT will become broader and more complex in terms of scope. It will change the world in terms of

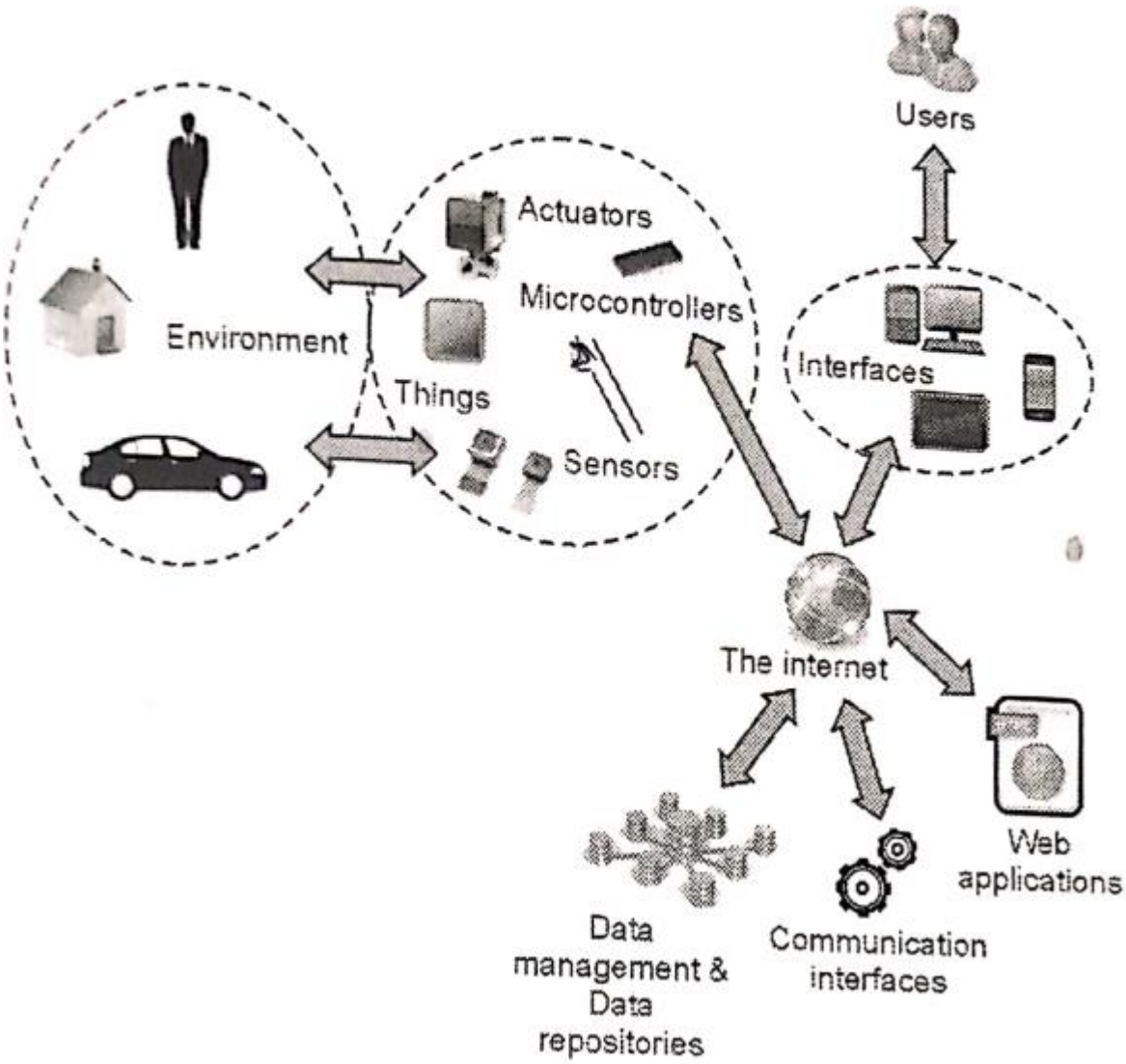
"anytime, anyplace, anything in connectivity."

IoT Enablers:

- **RFIDs:** uses radio waves in order to electronically track the tags attached to each physical object.
- **Sensors:** devices that are able to detect changes in an environment (ex: motion detectors).
- **Nanotechnology:** as the name suggests, these are extremely small devices with dimensions usually less than a hundred nanometers.
- **Smart networks:** (ex: mesh topology).

Working with IoT Devices :

- *Collect and Transmit Data :* For this purpose sensors are widely used they are used as per requirements in different application areas.
- *Actuate device based on triggers produced by sensors or processing devices :* If certain condition is satisfied or according to user's requirements if certain trigger is activated then which action to performed that is shown by Actuator devices.
- *Receive Information :* From network devices user or device can take certain information also for their analysis and processing purposes.
- *Communication Assistance :* Communication assistance is the phenomena of communication between 2 network or communication between 2 or more IoT devices of same or different Networks. This can be achieved by different communication protocols like : MQTT , Constrained Application Protocol, ZigBee, FTP, HTTP etc.



Working of IoT

Characteristics of IoT:

- Massively scalable and efficient
- IP-based addressing will no longer be suitable in the upcoming future.
- An abundance of physical objects is present that do not use IP, so IoT is made possible.
- Devices typically consume less power. When not in use, they should be automatically programmed to sleep.
- A device that is connected to another device right now may not be connected in another instant of time.
- Intermittent connectivity – IoT devices aren't always connected. In order to save bandwidth and battery consumption, devices will be powered off periodically when not in use. Otherwise, connections might turn unreliable and thus prove to be inefficient.
- ***Desired Quality of any IoT Application :***
- ***Interconnectivity***

It is the basic first requirement in any IoT infrastructure. Connectivity should be guaranteed from any devices on any network then only devices in a network can communicate with each other.

- ***Heterogeneity***

There can be diversity in IoT enabled devices like different hardware and software configuration or different network topologies or connections but they should connect and interact with each other despite of so much heterogeneity.

- ***Dynamic in nature***

IoT devices should dynamically adapt themselves to the changing surroundings like different situation and different prefaces.

- ***Self adapting and self configuring technology***

For example surveillance camera. It should be flexible to work in different weather conditions and different light situations (morning, afternoon, or night).

- ***Intelligence***

Just data collection is not enough in IoT, extraction of knowledge from the generated data is very important. For example, sensors generate data, but that data will only be useful if it is interpreted properly. So intelligence is one of the key characteristics in IoT. Because data interpretation is the major part in any IoT application because without data processing we can't make any insights from data . Hence big data is also one of the most enabling technology in IoT field.

- ***Scalability***

The number of elements (devices) connected to IoT zone is increasing day by day. Therefore, an IoT setup should be capable of handling the expansion. It can be either expand capability in terms of processing power, Storage, etc. as vertical scaling or horizontal scaling by multiplying with easy cloning

- ***Identity***

Each IoT device has a unique identity (e.g., an IP address). This identity is helpful in communication, tracking and to know status of the things. If there is no identification then it will directly effect security and safety of any system because without discrimination we can't identify with whom one network is connected or with whom we have to communicate. So there should be clear and appropriate discrimination technology available between IoT networks and devices.

- ***Safety***

Sensitive personal details of a user might be compromised when the devices are connected to the Internet. So data security is a major challenge. This could cause a loss to the user. Equipment in the huge IoT network may also be at risk. Therefore, equipment safety is also critical.

- ***Architecture***

It should be hybrid, supporting different manufacturer's products to function in the IoT network.

As a quick note, IoT incorporates trillions of sensors, billions of smart systems, and millions of applications.

Application Domains: IoT is currently found in four different popular domains:
1) Manufacturing/Industrial business - 40.2%

2) Healthcare - 30.3%

3) Security - 7.7%

4) Retail - 8.3%

Modern Applications:

1. Smart Grids and energy saving
2. Smart cities
3. Smart homes/Home automation
4. Healthcare
5. Earthquake detection
6. Radiation detection/hazardous gas detection
7. Smartphone detection
8. Water flow monitoring
9. Traffic monitoring
10. Wearables
11. Smart door lock protection system
12. Robots and Drones
13. Healthcare and Hospitals, Telemedicine applications
14. Security
15. Biochip Transponders(For animals in farms)
16. Heart monitoring implants(Example Pacemaker, ECG real time tracking)

1.6 Baseline Technologies

IoT primarily exploits standard protocols and networking technologies. However, the major enabling technologies and protocols of IoT are RFID, NFC, low-energy Bluetooth, low-energy wireless, low-energy radio protocols, LTE-A, and WiFi-Direct. These technologies support the specific networking functionality needed in an IoT system in contrast to a standard uniform network of common systems.

NFC and RFID

RFID (radio-frequency identification) and NFC (near-field communication) provide simple, lowenergy, and versatile options for identity and access tokens, connection bootstrapping, and payments.

- RFID technology employs 2-way radio transmitter-receivers to identify and track tags associated with objects.
- NFC consists of communication protocols for electronic devices, typically a mobile device and a standard device.

Low-Energy Bluetooth

This technology supports the low-power, long-use need of IoT function while exploiting a standard technology with native support across systems.

Low-Energy Wireless

This technology replaces the most power hungry aspect of an IoT system. Though sensors and other elements can power down over long periods, communication links (i.e., wireless) must remain in listening mode. Low-energy wireless not only reduces consumption, but also extends the life of the device through less use.

Radio Protocols

ZigBee, Z-Wave, and Thread are radio protocols for creating low-rate private area networks. These technologies are low-power, but offer high throughput unlike many similar options. This increases the power of small local device networks without the typical costs.

LTE-A

LTE-A, or LTE Advanced, delivers an important upgrade to LTE technology by increasing not only its coverage, but also reducing its latency and raising its throughput. It gives IoT a tremendous power through expanding its range, with its most significant applications being vehicle, UAV, and similar communication.

WiFi-Direct

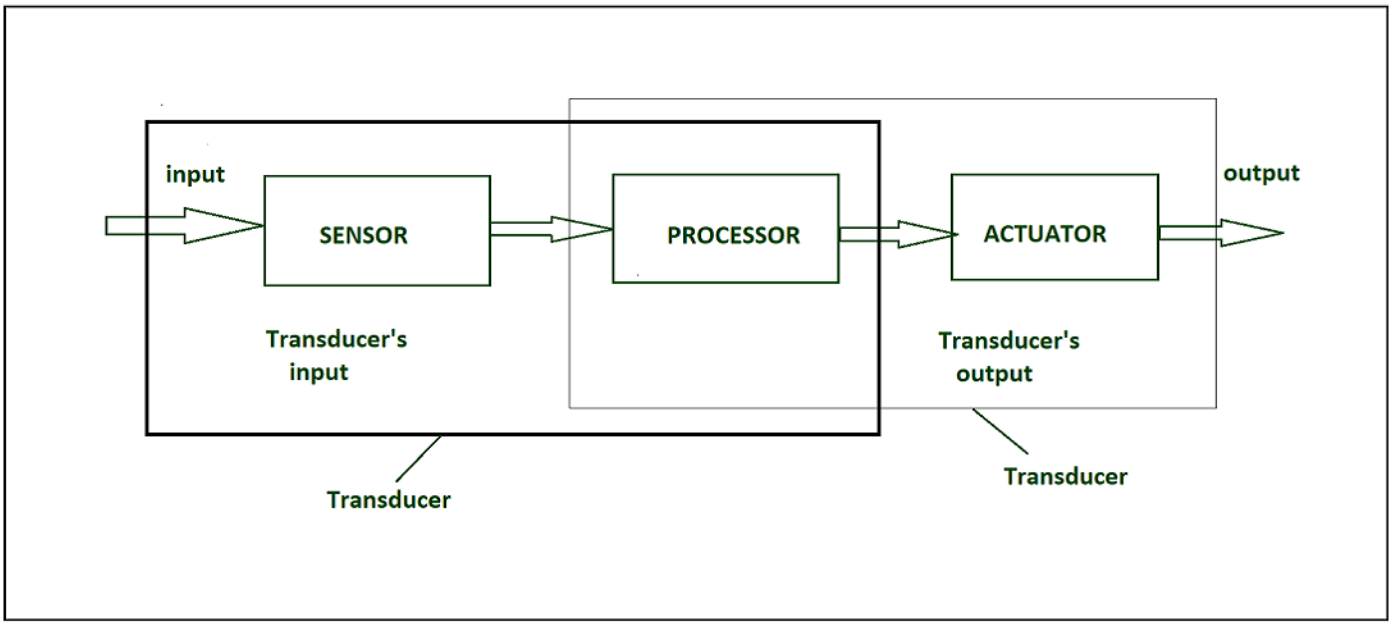
WiFi-Direct eliminates the need for an access point. It allows P2P (peer-to-peer) connections with the speed of WiFi, but with lower latency. WiFi-Direct eliminates an element of a network that often bogs it down, and it does not compromise on speed or throughput.

1.1 Sensor

Generally, sensors are used in the architecture of IOT devices. Sensors are used for sensing things and devices etc. **A device that provides a usable output in response to a specified measurement**.

a device that responds to a physical stimulus (such as heat, light, sound, pressure, magnetism, or a particular motion) and transmits a resulting impulse (as for measurement or operating a control)

Generally, sensors are used in the architecture of IOT devices.
Sensors are used for sensing things and devices etc.
A device that provides a usable output in response to a specified measurement.
The sensor attains a physical parameter and converts it into a signal suitable for processing (e.g. electrical, mechanical, optical) the characteristics of any device or material to detect the presence of a particular physical quantity.
The output of the sensor is a signal which is converted to a human-readable form like changes in characteristics, changes in resistance, capacitance, impedance etc.



IOT HARDWARE

Transducer :

- A transducer converts a signal from one physical structure to another.
- It converts one type of energy into another type.
- It might be used as actuators in various systems.

Sensors characteristics :

1. Static
2. Dynamic

1. Static characteristics :

It is about how the output of a sensor changes in response to an input change after steady state condition.

- **Accuracy –**
  Accuracy is the capability of measuring instruments to give a result close to the true value of the measured quantity. It measures errors. It is measured by absolute and relative errors. Express the correctness of the output compared to a higher prior system. Absolute error = Measured value – True value
  Relative error = Measured value/True value
- **Range –**
  Gives the highest and the lowest value of the physical quantity within which the sensor can actually sense. Beyond these values, there is no sense or no kind of response.
  e.g. RTD for measurement of temperature has a range of -200`c to 800`c.
- **Resolution –**
  Resolution is an important specification towards selection of sensors. The higher the resolution, better the precision. When the accretion is zero to, it is called threshold.
  Provide the smallest changes in the input that a sensor is able to sense.
- **Precision –**
  It is the capacity of a measuring instrument to give the same reading when repetitively measuring the same quantity under the same prescribed conditions.
  It implies agreement between successive readings, NOT closeness to the true value.
  It is related to the variance of a set of measurements.
  It is a necessary but not sufficient condition for accuracy.
- **Sensitivity –**
  Sensitivity indicates the ratio of incremental change in the response of the system with respect to incremental change in input parameters. It can be found from the slope of the output characteristics curve of a sensor. It is the smallest amount of difference in quantity that will change the instrument's reading.
- **Linearity –**
  The deviation of the sensor value curve from a particular straight line. Linearity is determined by the calibration curve. The static calibration curve plots the output amplitude versus the input amplitude under static conditions.
  A curve's slope resemblance to a straight line describes the linearity.
- **Drift –**
  The difference in the measurement of the sensor from a specific reading when kept at that value for a long period of time.

- **Repeatability –**
  The deviation between measurements in a sequence under the same conditions. The measurements have to be made under a short enough time duration so as not to allow significant long-term drift.

Dynamic Characteristics :

Properties of the systems

- **Zero-order system –**
  The output shows a response to the input signal with no delay. It does not include energy-storing elements.
  Ex. potentiometer measure, linear and rotary displacements.
- **First-order system –**
  When the output approaches its final value gradually.
  Consists of an energy storage and dissipation element.
- **Second-order system –**
  Complex output response. The output response of the sensor oscillates before steady state.

Sensor Classification :

- Passive & Active
- Analog & digital
- Scalar & vector

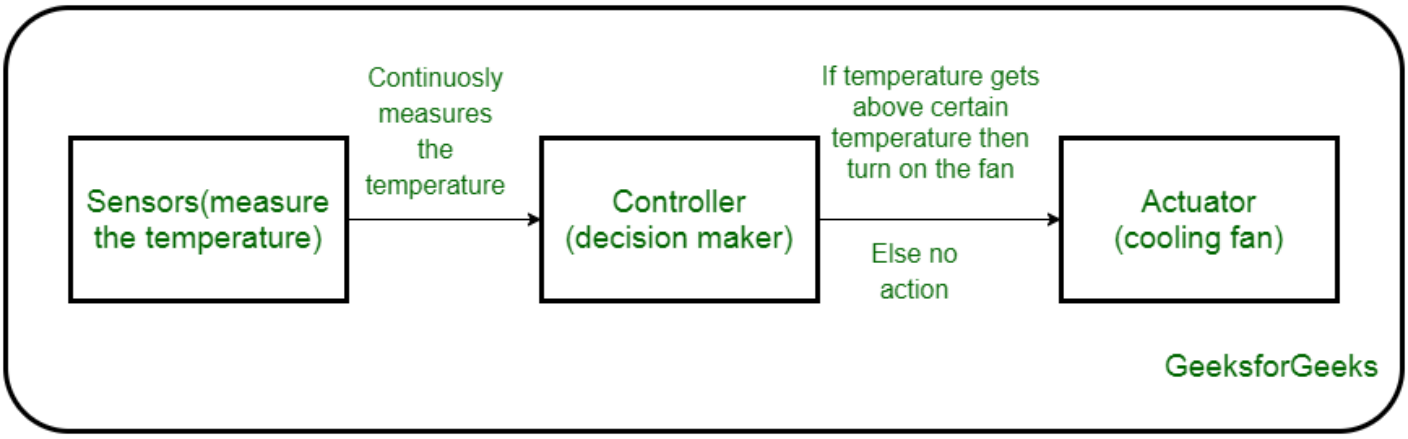
1. **Passive Sensor –**
   Can not independently sense the input. Ex- Accelerometer, soil moisture, water level and temperature sensors.
2. **Active Sensor –**
   Independently sense the input. Example- Radar, sounder and laser altimeter sensors.
3. **Analog Sensor –**
   The response or output of the sensor is some continuous function of its input parameter. Ex- Temperature sensor, LDR, analog pressure sensor and analog hall effect.
4. **Digital sensor –**
   Response in binary nature. Design to overcome the disadvantages of analog sensors. Along with the analog sensor, it also comprises extra electronics for bit conversion. Example – Passive infrared (PIR) sensor and digital temperature sensor(DS1620).
5. **Scalar sensor –**
   Detects the input parameter only based on its magnitude. The answer for the sensor is a function of magnitude of some input parameter. Not affected by the direction of input parameters.
   Example – temperature, gas, strain, color and smoke sensor.
6. **Vector sensor –**
   The response of the sensor depends on the magnitude of the direction and orientation of input parameter. Example – Accelerometer, gyroscope, magnetic field and motion detector sensors.

1.2 Actuator

An actuator is a machine component or system that moves or controls the mechanism or the system. Sensors in the device sense the environment, then control signals are generated for the actuators according to the actions needed to perform.

A servo motor is an example of an actuator. They are linear or rotatory actuators, can move to a given specified angular or linear position. We can use servo motors for IoT applications and make the motor rotate to 90 degrees, 180 degrees, etc., as per our need.

The following diagram shows what actuators do, the controller directs the actuator based on the sensor data to do the work.



Working of IoT devices and use of Actuators

The control system acts upon an environment through the actuator. It requires a source of energy and a control signal. When it receives a control signal, it converts the source of energy to a mechanical operation. On this basis, on which form of energy it uses, it has different types given below.

Types of Actuators :

1. Hydraulic Actuators –

A hydraulic actuator uses hydraulic power to perform a mechanical operation. They are actuated by a cylinder or fluid motor. The mechanical motion is converted to rotary, linear, or oscillatory motion, according to the need of the IoT device. Ex- construction equipment uses hydraulic actuators because hydraulic actuators can generate a large amount of force.

Advantages :

- Hydraulic actuators can produce a large magnitude of force and high speed.
- Used in welding, clamping, etc.
- Used for lowering or raising the vehicles in car transport carriers.

Disadvantages :

- Hydraulic fluid leaks can cause efficiency loss and issues of cleaning.
- It is expensive.
- It requires noise reduction equipment, heat exchangers, and high maintenance systems.

2. Pneumatic Actuators –

A pneumatic actuator uses energy formed by vacuum or compressed air at high pressure to convert into either linear or rotary motion. Example- Used in robotics, use sensors that work like human fingers by using compressed air.

Advantages :

- They are a low-cost option and are used at extreme temperatures where using air is a safer option than chemicals.
- They need low maintenance, are durable, and have a long operational life.
- It is very quick in starting and stopping the motion.

Disadvantages :

- Loss of pressure can make it less efficient.
- The air compressor should be running continuously.
- Air can be polluted, and it needs maintenance.

3. Electrical Actuators –

An electric actuator uses electrical energy, is usually actuated by a motor that converts electrical energy into mechanical torque. An example of an electric actuator is a solenoid based electric bell.

Advantages :

- It has many applications in various industries as it can automate industrial valves.
- It produces less noise and is safe to use since there are no fluid leakages.
- It can be re-programmed and it provides the highest control precision positioning.

Disadvantages :

- It is expensive.
- It depends a lot on environmental conditions.

Other actuators are –

- **Thermal/Magnetic Actuators –**
  These are actuated by thermal or mechanical energy. Shape Memory Alloys (SMAs) or Magnetic Shape-Memory Alloys (MSMAs) are used by these actuators. An example of a thermal/magnetic actuator can be a piezo motor using SMA.

- **Mechanical Actuators –**
  A mechanical actuator executes movement by converting rotary motion into linear motion. It involves pulleys, chains, gears, rails, and other devices to operate. Example – A crankshaft.
- Soft Actuators
- Shape Memory Polymers
- Light Activated Polymers
- With the expanding world of IoT, sensors and actuators will find more usage in commercial and domestic applications along with the pre-existing use in industry.

1.3 IoT components and implementation

The **Internet of Things** (**IoT**) is the network of physical objectsdevices, vehicles, buildings and other items which are embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data. Implementing this concept is not an easy task by any measure for many reasons including the complex nature of the different components of the ecosystem of IoT. To understand the gravity of this task, we will explain all the five components of IoT Implementation.

Components of IoT implementation

- Sensors
- Networks
- Standards
- Intelligent Analysis
- Intelligent Actions

Sensors

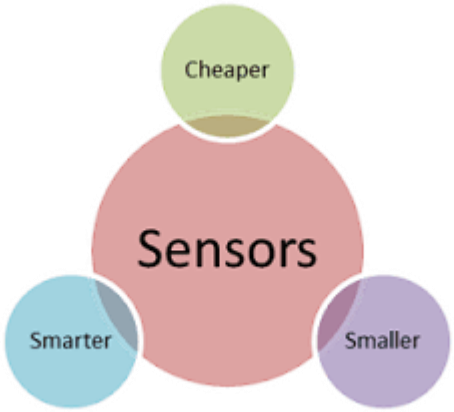
According to (IEEE) sensors can be defined as: An electronic device that produces electrical, optical, or digital data derived from a physical condition or event. Data produced from sensors is then electronically transformed, by another device, into information (output) that is useful in decision making done by intelligent devices or individuals (people).

Types of Sensors: Active Sensors & Passive Sensors

The selection of sensors greatly impacted by many factors, including:

- Purpose (Temperature, Motion, Bioetc.)
- Accuracy
- Reliability
- Range
- Resolution
- Level of Intelligence (dealing with noise and interference)

The driving forces for using sensors in IoT today are new trends in technology that made sensors *cheaper, smarter and smaller*.



1.4 Challenges for IoT

The Internet of Things (IoT) has fast grown to be a large part of how human beings live, communicate and do business. All across the world, web-enabled devices are turning our global rights into a greater switched-on area to live in.

There are various types of challenges in front of IoT.

Security challenges in IoT :

1. **Lack of encryption –**
   Although encryption is a great way to prevent hackers from accessing data, it is also one of the leading IoT security challenges.
   These drives like the storage and processing capabilities that would be found on a traditional computer.
   The result is an increase in attacks where hackers can easily manipulate the algorithms that were designed for protection.
2. **Insufficient testing and updating –**
   With the increase in the number of IoT(internet of things) devices, IoT manufacturers are more eager to produce and deliver their device as fast as they can without giving security too much of although.
   Most of these devices and IoT products do not get enough testing and updates and are prone to hackers and other security issues.
3. **Brute forcing and the risk of default passwords –**
   Weak credentials and login details leave nearly all IoT devices vulnerable to password hacking and brute force.
   Any company that uses factory default credentials on their devices is placing both their business and its assets and the customer and their valuable information at risk of being susceptible to a brute force attack.
4. **IoT Malware and ransomware –**
   Increases with increase in devices.
   Ransomware uses encryption to effectively lock out users from various devices and platforms and still use a user's valuable data and info.
   Example –
   A hacker can hijack a computer camera and take pictures.
   By using malware access points, the hackers can demand ransom to unlock the device and return the data.
5. **IoT botnet aiming at cryptocurrency –**
   IoT botnet workers can manipulate data privacy, which could be massive risks for an open Crypto market. The exact value and creation of cryptocurrencies code face danger from mal-intentioned hackers.
   The blockchain companies are trying to boost security. Blockchain technology itself is not particularly vulnerable, but the app development process is.

Design challenge in IoT :

1. **Battery life is a limitation –**
   Issues in packaging and integration of small-sized chip with low weight and less power consumption. If you've been following the mobile space, you've likely see how every yr it looks like there's no restriction in terms of display screen size. Take the upward thrust of 'phablets', for instance, which can be telephones nearly as huge as tablets. Although helpful, the bigger monitors aren't always only for convenience, rather, instead, display screen sizes are growing to accommodate larger batteries. Computers have getting slimmer, but battery energy stays the same.

2. **Increased cost and time to market –**
   Embedded systems are lightly constrained by cost.
   The need originates to drive better approaches when designing the IoT devices in order to handle the cost modelling or cost optimally with digital electronic components.
   Designers also need to solve the design time problem and bring the embedded device at the right time to the market.
3. **Security of the system –**
   Systems have to be designed and implemented to be robust and reliable and have to be secure with cryptographic algorithms and security procedures.
   It involves different approaches to secure all the components of embedded systems from prototype to deployment.

Deployment challenges in IoT :

1. **Connectivity –**
   It is the foremost concern while connecting devices, applications and cloud platforms.
   Connected devices that provide useful front and information are extremely valuable. But poor connectivity becomes a challenge where IoT sensors are required to monitor process data and supply information.
2. **Cross platform capability –**
   IoT applications must be developed, keeping in mind the technological changes of the future.
   Its development requires a balance of hardware and software functions.
   It is a challenge for IoT application developers to ensure that the device and IoT platform drivers the best performance despite heavy device rates and fixings.
3. **Data collection and processing –**
   In IoT development, data plays an important role. What is more critical here is the processing or usefulness of stored data.
   Along with security and privacy, development teams need to ensure that they plan well for the way data is collected, stored or processed within an environment.
4. **Lack of skill set –**
   All of the development challenges above can only be handled if there is a proper skilled resource working on the IoT application development.
   The right talent will always get you past the major challenges and will be an important IoT application development asset.

2. **IOT Networking**

The Internet of Things (IoT) describes the network of physical objects—“things”—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet.

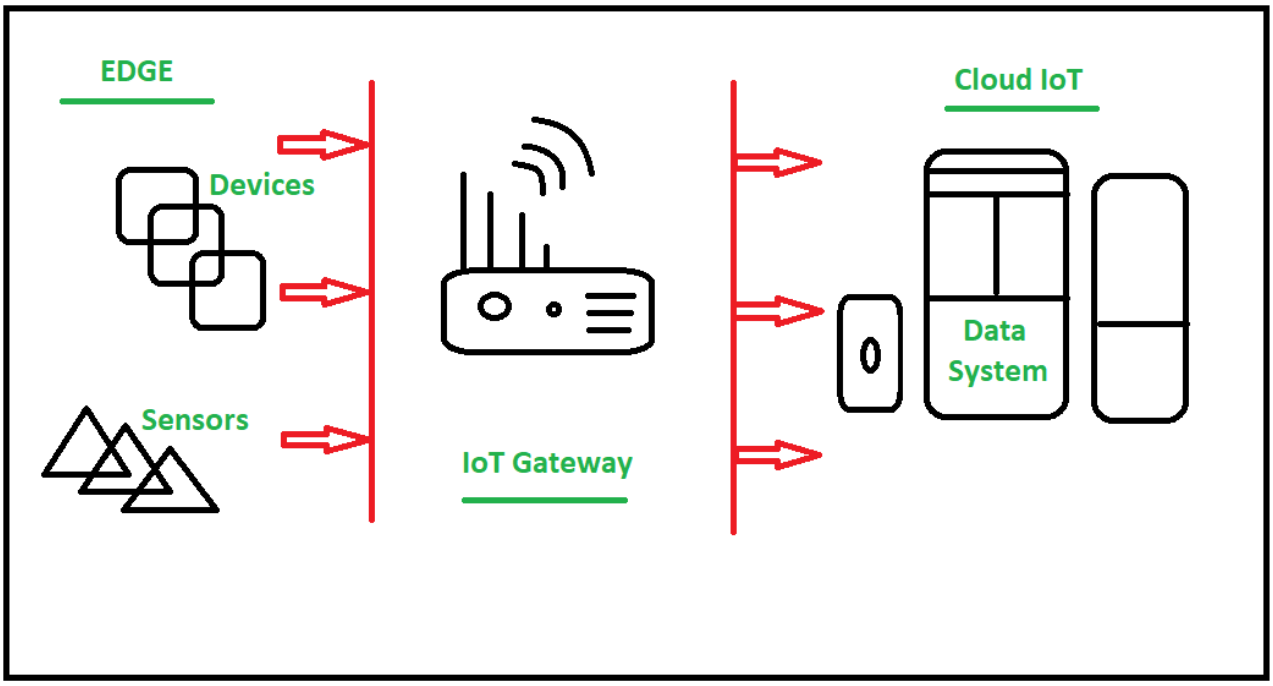
2.1 Terminologies

- IoT Cloud Platform
- Edge Computing
- Mobile IoT (MIoT)
- Bluetooth Low Energy (BLE)
- IoT Protocol
- Global Navigation Satellite System (GNSS)
- Narrowband IoT (NB-IoT)
- Quality of Service (QoS)

2.2 Gateway Prefix allotment

Gateway provides bridge between different communication technologies which means we can say that a Gateway acts as a medium to open up connection between cloud and controller(sensors / devices) in Internet of Things (IoT). By the help of gateways it is possible to establish device to device or device to cloud communication. A gateway can be a typical hardware device or software program.
It enables a connection between sensor network and Internet along with enabling IoT communication, it also performs many other tasks such as this IoT gateway performs protocol translation, aggregating all data, local processing and filtering of data before sending it to cloud, locally storing data and autonomously controlling devices based on some inputted data, providing additional device security.

The below figure shows how IoT Gateways establish communication between sensors and cloud (Data System) :



As IoT devices work with low power consumption(Battery power) in other words they are energy constrained so if they will directly communicate to cloud/internet it won't be effective in terms of power. So they communicate with Gateway first using short range wireless transmission modes/network like ZigBee, Bluetooth, etc as they consume less power or they can also be connected using long range like Cellular and WiFi etc.

Then Gateway links them to Internet/ cloud by converting data into a standard protocol like MQTT. using ethernet, WiFi/cellular or satellite connection. And in mostly Gateway is Mains powered unlike sensor nodes which are battery powered. In practice there are multiple Gateway devices.

Let's think about a simple IoT gateway, then our smartphone comes into picture as it can also work as a basic IoT gateway when we use multiple radio technologies like WiFi, Bluetooth, Cellular network of smart phone to work on any IoT project in sending and receiving data at that time this also acts as a basic IoT Gateway.

Key functionalities of IoT Gateway :

- Establishing communication bridge
- Provides additional security.
- Performs data aggregation.
- Pre processing and filtering of data.
- Provides local storage as a cache/ buffer.
- Data computing at edge level.
- Ability to manage entire device.
- Device diagnostics.
- Adding more functional capability.
- Verifying protocols.

Working of IoT Gateway :

1. Receives data from sensor network.
2. Performs Pre processing, filtering and cleaning on unfiltered data.
3. Transports into standard protocols for communication.
4. Sends data to cloud.

IoT Gateways are key element of IoT infrastructure as Gateways establish connection for communication and also performs other task as described above. So IoT Gateway is one of most essential thing when we start think about an IoT ecosystem.

2.3 Impact of mobility on Addressing

Mobility facilitates employee productivity and increased profit as a result, fitting right in with the increased connectivity of IoT. Additionally, IoT is enabled by enterprise mobility because companies rely on the connected devices and big data analytics that mobility provides.

Mobility is **the cell phone you carry and the computer processor in your car**. Mobility is driven by two fundamental technology facts: 1) the power of computing chips doubles every 18 months (Moore's Law); and, 2) the value of a network increases exponentially with the number of connected devices (Metcalf's Law)

2.4 Multihoming

Multihoming is a mechanism used to configure one computer with more than one network interface and multiple IP addresses. It provides enhanced and reliable Internet connectivity without compromising efficient performance. The multihoming computer is known as the host and is directly or indirectly connected to more than one network.

Multihoming provides many benefits, including the following:

- The multiple simultaneous Internet connections make system failure less likely than with a system with a single Internet connection.
- Users interact with the Internet through multiple doorways. During failover, only one door closes, while the other doors continue working.
- In Web management, multihoming helps load balancing and allows a network to work with the lowest downtime.
- The system is maintained during disaster and recovery.

The two main types of multihoming are:

- IPv4 multihoming: A multihomed public IP address must be configured with two or more Internet service provider (ISP) connections. When any link or route fails, network traffic is automatically rerouted. IPv4's major drawback is its central connection point (shared transmission line and/or edge router) for two ISPs, which can result in failure of the entire network if the central point fails. The Border Gateway Protocol (BGP) is used for multihoming purposes.
- IPv6 multihoming: Multihoming is on the rise with IPv6 computer systems, which provide more efficient support for it. Many communication devices are shifting to IPv6, and multihoming allows easy data transfer. However, IPv6 multihoming is not yet standardized.

2.5 Deviation from regular Web

So to summarise, an IoT network is a closed-loop mesh of autonomously operating devices which can communicate with each other when required, while regular networks rely on readiness to connect with an external system.

2.6 IoT identification and Data protocols

- Bluetooth. Good for high-speed data transfer, Bluetooth sends both voice and data signals up to 10 meters.
- Z-Wave. A mesh network using low-energy radio waves to communicate from appliance to appliance
- 4G LTE IoT
- Cat-1
- Narrowband or NB-IoT/Cat-M2
- Application layer
- Transport layer
- Network layer

3. **Connectivity Technologies**

Connectivity is a critical component of the Internet of Things. **IoT devices rely on networks to communicate with gateways, applications, servers, routers, and other IoT devices**. This communication—transmitting and receiving data—enables IoT devices to perform the functions they were designed for.

3.1 Introduction

Traditionally, the IoT landscape or rather the machine-to-machine (M2M) communication has been dominated by radio technologies such as ZigBee, Bluetooth and Wi-Fi for short range local area networks,

and traditional cellular such as 2G/3G/4G for wide area networks, with 5G having recently been added to the latter.

3.2 IEEE 802.

IEEE 802 is a collection of networking standards that cover the physical and data-link layer specifications for technologies such as Ethernet and wireless. These specifications apply to local area networks (LAN) and metropolitan area networks (MAN). IEEE 802 also aids in ensuring multi-vendor interoperability by promoting standards for vendors to follow.

Essentially, the IEEE 802 standards help make sure internet services and technologies follow a set of recommended practices so network devices can all work together smoothly.

IEEE 802 is divided into 22 parts that cover the physical and data-link aspects of networking. The family of standards is developed and maintained by the IEEE 802 LAN/MAN Standards Committee, also called the LMSC. IEEE stands for Institute of Electrical and Electronics Engineers.

The set of standards started in 1979 with a "local network for computer interconnection" standard, which was approved a year later. The LMSC has made more than 70 standards for IEEE 802.

Some commonly used standards include those for Ethernet, bridging and virtual bridged LANs, wireless LAN, wireless PAN, MAN and radio access networks as well as media independent handover services. The better-known specifications include 802.3 Ethernet, 802.11 Wi-Fi and 802.15 Bluetooth/ZigBee. However, some of these standards have been labeled as disbanded or hibernating and are either superseded by newer standards or are being reworked. Using an open process, the LMSC advocates for these standards globally.

Individual "working groups" are decided on and assigned to each area in order to provide each area with an acceptable amount of focus. IEEE 802 specifications also split the data link layer into two different layers -- an LLC layer and a MAC layer.

Standards can be found in a PDF provided by the LMSC for up to six months after they have been published. All standards stay in place until they are replaced with another document or withdrawn.

Why IEEE 802 standards are important

LMSC was formed in 1980 in order to standardize network protocols and provide a path to make compatible devices across numerous industries.

Without these standards, equipment suppliers could manufacture network hardware that would only connect to certain computers. It would be much more difficult to connect to systems not using the same set of networking equipment. Standardizing protocols help ensure that multiple types of devices can connect to multiple network types. It also helps make sure network management isn't the challenge it could be if it wasn't in place.

IEEE 802 will also coordinate with other international standards, such as ISO, to help maintain international standards.

In addition, the "802" in IEEE 802 does not stand for anything with high significance. 802 was just the next numbered project.

Examples of IEEE 802 uses

The IEEE 802 specifications can be used by commercial organizations to ensure their products maintain any newly specified standards. So, for example, the 802.11 specification that applies to Wi-Fi could be used to make sure Wi-Fi devices work together under one standard. In the same way, IEE 802 can help maintain local area network standards.

These specifications can also define what connectivity infrastructure will be used for -- individual networks, or those at a larger organizational scale.

The IEEE 802 specifications apply to hardware and software products. So, to ensure manufacturers don't have any input on the standards, there is a voting protocol in place. This makes sure that one organization does not influence the standards too much.

Working groups

The working groups are the different areas of focus within the 802 specifications. They are numbered from 802.1 onward.

3.3 ZigBee, 6LoWPAN
ZigBee

There are three ZigBee Network topologies: **star, Cluster tree and Mesh**. A star network consists of a coordinator and any number of end devices. These devices are then connected to the coordinator but isolated from each other. In Cluster Tree topology, end devices connect the coordinator via Router.

ZigBee is a wireless technology standard that provides a set of communication protocols for short-range communications. It is an open-source global standard developed by Zigbee Alliance to address the needs of low-cost, low power wireless IoT networks. The protocol is used in low data rate, short to medium range wireless networking devices like senors and control networks.
ZigBee provides flexibility for developers and end-users and delivers great interoperability. Because of its important feature of being low-cost, low-power consumption and having faster wireless connectivity, the protocol has many applications. For instance, its a popular technology for smart home, because it outstands other wireless technologies with some distinct features. Firstly, communication is two -way which makes ZigBee devices reliable. Secondly, it caters to all sectors like lightings, security, appliances and home access. Thirdly, and most importantly, this technology requires very little power mainly due to its low latency and low duty cycle. In addition, it uses mesh network and thus reduce chances of failure at nodes.

source elprocus.com

ZigBee offers a wireless range of 70m indoors and 400m outdoors. It supports multiple networks like point to point, point to multipoint mesh- networks. Notably, it uses AES 128 encryption thus protecting your information while on air transfers. In addition, its easy installation and maintenance makes it more affordable.

Here's an interesting fact on the origin of the name ZigBee! The name is closely in relation with Bees because they do 'waggle dance' to communicate with each other when they return to their hives. This zigzag dance is where ZigBee got its name.

ZigBee Frequency

Zigbee uses physical and MAC layers of the IEEE 802.15.4 specification. In addition, it operates in an unlicensed 2.4 GHz ISM band. Although 2.4 GHz is pervasive worldwide, there are devices that use frequency bands like 915 MHz, 868 MHz, and 784 MHz in USA, Europe, and China respectively. Also, it transfers data at a rate of 250 kbps.

ZigBee vs Wi-Fi

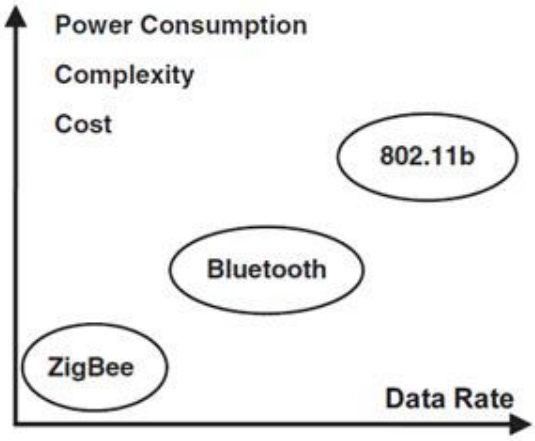
Wondering why ZigBee when you already have other communication standards like Wi-Fi, Bluetooth?

As Wi-Fi and Bluetooth, ZigBee is used for short-range communications or to establish a personal area network and works on the same band as Wi-Fi and Bluetooth. It is specially built for control and sensor networks. It is mainly used to monitor and control devices of which both Wi-Fi and Bluetooth are not quite suitable for this specific application of wireless communication.

Mainly, ZigBee is based on IEEE 802.15.4 whereas Wi-Fi is dependent on IEEE 802.11 series. Both are different technologies because ZigBee uses WPAN while Wi-Fi is WLAN based. Wi-Fi covers a distance of 30-100 meters but the other is used to get a range of 10-30meters. Also, it provides a data rate of 250 Kbps. On the other hand, devices operating with high power and high data rate uses Wi-Fi technology for communication as it works at a rate of 54 Mbps. Zigbee's best quality is its low power consumption rate and sustained battery life of the devices.

ZigBee vs Bluetooth

ZigBee when compared with Bluetooth, uses a transmit power of the only 100mW which makes it efficient in case of power consumption. In addition, it is designed to support hundreds of devices which is a vantage over Bluetooth that supports a maximum of 7 devices. Now you know the substantive use of ZigBee! Here's a table highlighting the differences.



	Data Rate	Typical Range	Application Examples
ZigBee	20 to 250 Kbps	10–100 m	Wireless Sensor Networks
Bluetooth	1 to 3 Mbps	2–10 m	Wireless Headset Wireless Mouse
IEEE 802.11b	1 to 11 Mbps	30–100 m	Wireless Internet Connection

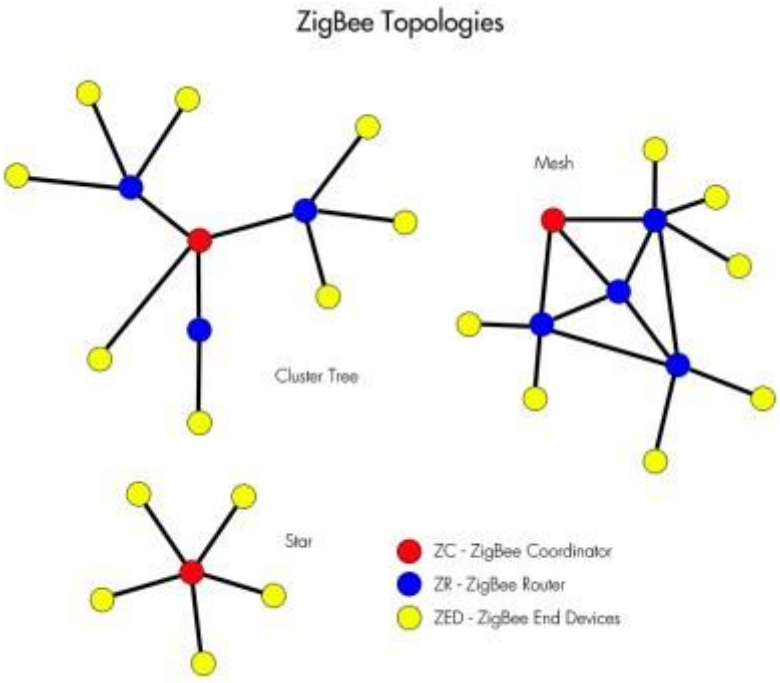
source electronicshub.com

System Structure

ZigBee system consists of ZigBee Co-ordinator, Router and End device.

1. **ZigBee Coordinator (ZC):** A network consists of at least one central unit. ZC is the most important device as it coordinates and acts as a bridge of network. It is the device responsible for the start of the network. Most importantly, this unit is responsible for the transmission and reception of data. It also facilitates the handling and storing of information.
2. ZigBee Router (ZR): A router is an intermediate unit. It allows data to pass through them to and fro to other devices.
3. ZigBee End-Device (ZED): A ZED interfaces to a sensor and executes the control operation. The end device contains just enough functionality to talk to either the coordinator or the router. This causes the node to stay asleep for a long time thereby increasing battery life to a great extent.
   A **ZED** device requires less energy as compared to the **ZC or ZR.**

Topologies



source: assured-systems.com

The number of coordinators, routers, devices in the network depends on the topology used. There are three ZigBee Network topologies: star, Cluster tree and Mesh.

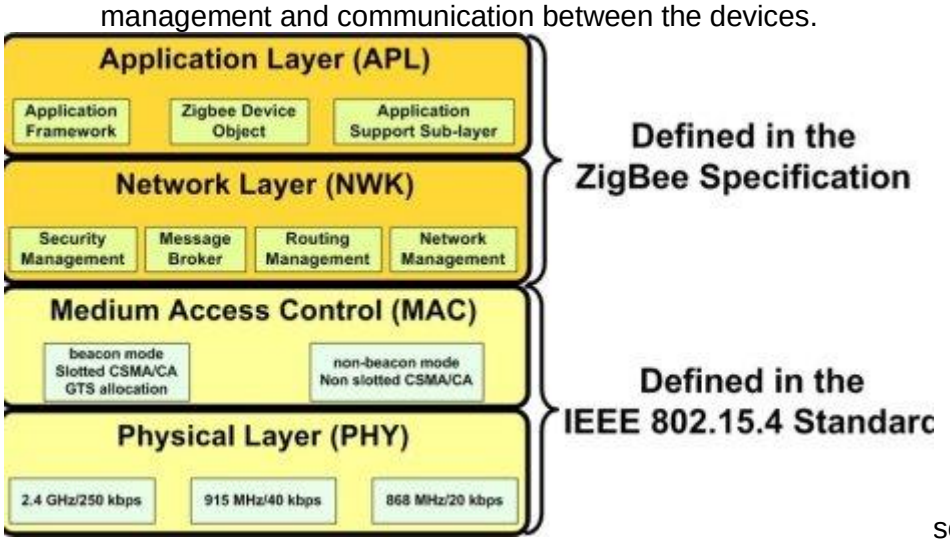
- A **star network** consists of a coordinator and any number of end devices. These devices are then connected to the coordinator but isolated from each other.

- In **Cluster Tree topology**, end devices connect the coordinator via Router.
- In the **mesh networks**, the nodes are interconnected with other nodes so that there exist multiple pathways connecting each node. The connection between nodes is updated using built-in routing methodologies. Thus, it provides good stability in changing conditions or failure at any node.

ZigBee Architecture

There are four layers in ZigBee network architecture. ZigBee protocol architecture consists of a stack of various layers of which physical and **MAC** layers are as defined in IEEE 802.15.4 and the other two layers belong to Zigbee specification.

- The **Physical layer** performs modulation and demodulation to the signals send and received.
- **MAC layer** transfer data using CSMA/CA. Moreover, the MAC layer synchronizes the communication between the devices.
- The **network layer** is responsible for setting up a network, connecting to the devices, routing data, etc.
- Lastly, the **application layer** allows the device to interface with a network layer for data management and communication between the devices.



source slideshare.net

What devices use ZigBee?

Now you know what ZigBee is, it's important to know the devices that work with it. The ZigBee Alliance industry consists of companies using this protocol. Currently, there are more than 400 members registered in the Alliance and over 2,500 devices in use. Here are some users of this standard:

- Amazon Echo Plus
- Samsung SmartThings
- Apple
- Comcast
- Honeywell
- Philips
- Bosch
- Nokia

You can get a full list of members of the ZigBee alliance.

ZigBee Applications

ZigBee enables a wide usage in wireless networks with low- cost, low-power solutions. Most importantly, it provides the ability to run for years on inexpensive batteries to monitor and control applications. Generally,

home automation, Healthcare, Material tracking are some of the areas where ZigBee is making its advancements. Some of its applications include:

- Home Automation including security systems, meter reading systems, Light control systems
- consumer electronics including Gaming consoles, wireless remote controls.
- Industrial Automation systems as in asset management, personnel tracking, livestock management.
- Healthcare, the patient wears a ZigBee device, which periodically collects patients metrics and sends to the hospital for analysis.
- Building's structural health monitoring. It is useful in earthquake-prone areas. Sensors installed throughout the building sends data to detect signs of damage.
- Grid monitoring involving temperature monitoring, power management, fault locating etc.
- Smart metering to prevent theft, pricing support etc

6LoWPAN

6LoWPAN specification contains **packet compression and other optimization mechanisms to enable the efficient transmission of IPv6 packets on a network with limited power resources and reliability**, which makes efficient IPv6 communication over low-power wireless networks possible.

6LoWPAN stands for IPv6 over Low-power Wireless Personal Area Networks. It is a standard protocol for realizing IPv6 communication on wireless networks composed of low-power wireless modules. 6LoWPAN specification contains packet compression and other optimization mechanisms to enable the efficient transmission of IPv6 packets on a network with limited power resources and reliability, which makes efficient IPv6 communication over low-power wireless networks possible.

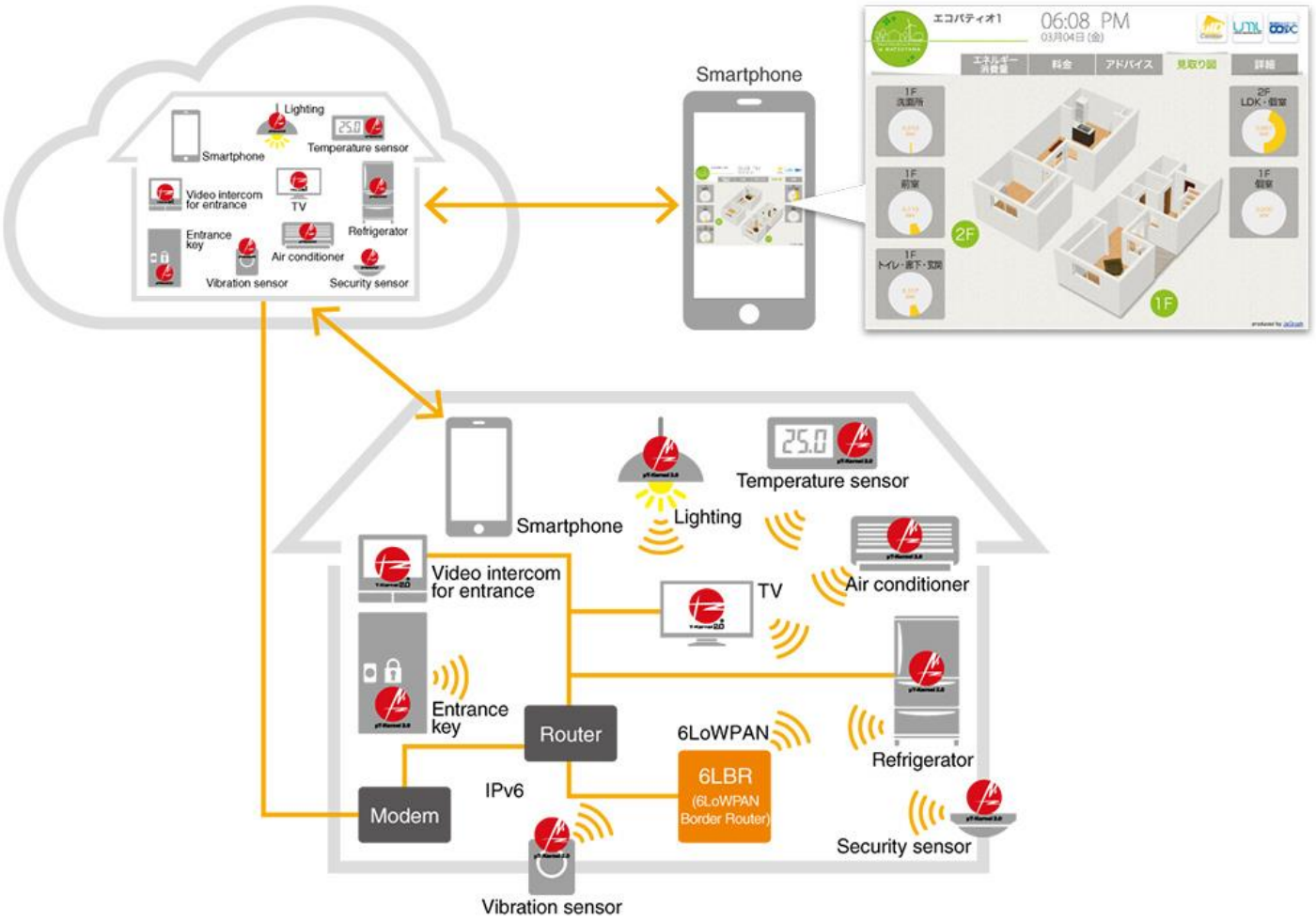
Various low-power wireless networks have been proposed and implemented before 6LoWPAN, but currently, 6LoWPAN is regarded as one of the preferred protocols to realize the Internet of Things (IoT).

This is because 6LoWPAN communication based on IPv6 allows once closed low-power wireless networks to interface with the global network, the Internet, and implement more advanced intelligent services that were not possible in the past.

Let us take the future smart house architecture that UC Technology aims to realize, for example (the figure below).
Optimized control of a smart house requires the information to be collected from various sensors. The control system consolidates the information to recognize the real-world context then gives appropriate optimized feedback to the real-world environment using actuators.
Management of a smart house by executing control logic on the reliable cloud server infrastructure is cost-effective from the viewpoint of high scalability, usability, and low maintenance cost. To realize such a control framework, 6LoWPAN's conformance to the Internet protocol plays an important role.

Currently, 6LoWPAN is published as a series of standard specifications, including RFC6282. Related specifications also have been discussed actively and are being standardized one by one.

UC Technology provides development tools including 6LoWPAN.
For more information, please visit "**6LoWPAN Development Tool**" and "**IoT-Engine Development Kit**."



Usage of 6LoWPAN for Smart House

Based on the status of IoT devices inside the house, a virtual house reflecting the real house environment will be created on the cloud.

Using the status of the virtual house, the IoT application framework will control the equipment inside the real house.

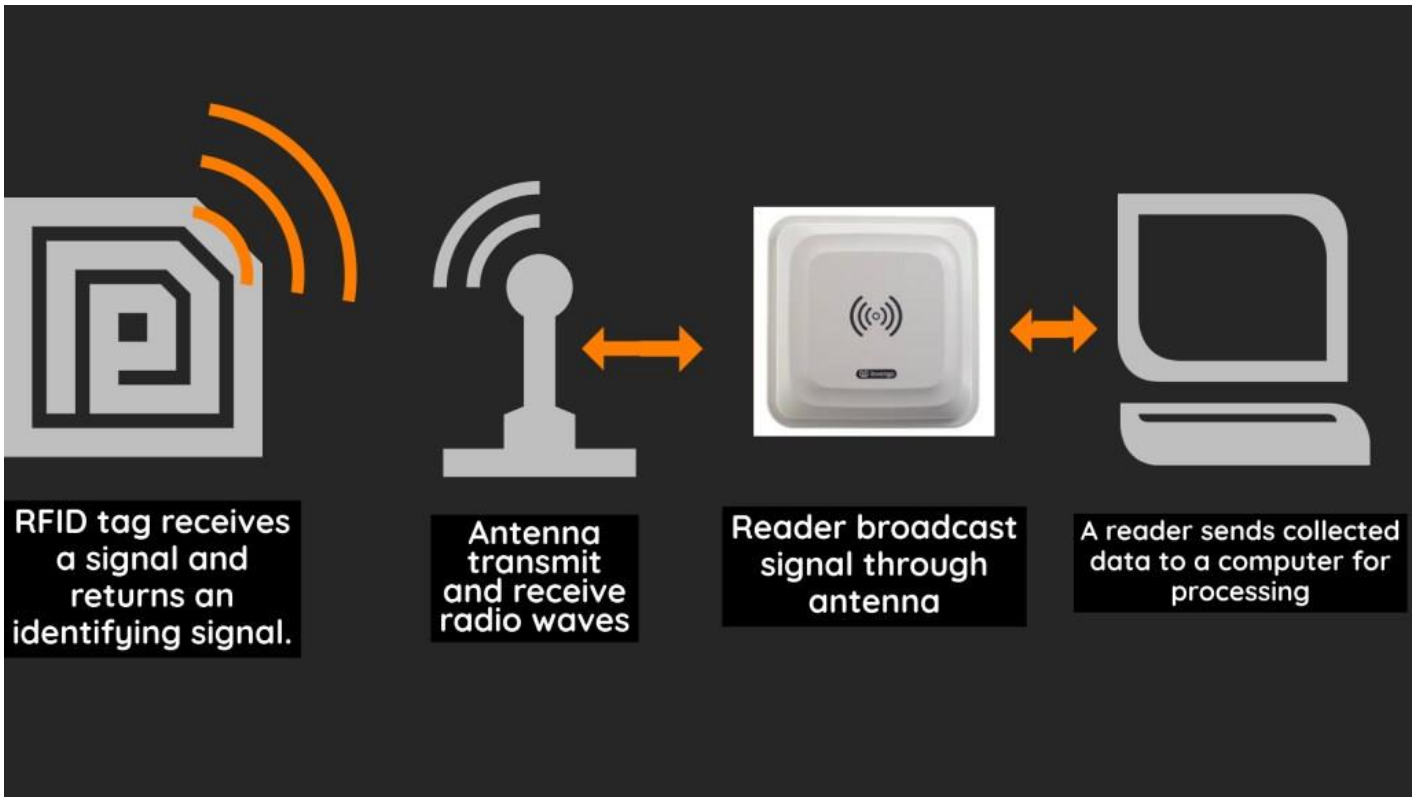
The status of the virtual house can be monitored from smartphones, etc.

3.4 RFID, HART and wireless HART
RFID

RFID technology in IoT **connects up the things into a network and makes them create and send data**. Unlike simple RFID tags that don't actively broadcast signals. So, sensor-enabled tags are used. It can generate and send metrics and other data in real-time.

Radio-frequency identification (RFID) is a technology that **enables communication and data transmission via radio waves.** It automatically identifies and tracks tags attached to the object.

How RFID works?



RFID Working

RFID is a wireless system referred to as Automatic Identification and Data Capture (AIDC). The purpose of AIDC is identifying, tracking, recording, storing, communicating essential data. The system consists of an RFID reader, RFID tags, and an antenna. An RFID tag consists of a transponder, a radio receiver, and a transmitter. Firstly, tags transmit digital data; it emits a unique identification code. Secondly, the primary responsibility of the antenna is to transmit and receive radio waves for communication. Thirdly, the Reader communicates with any tags in its read range. After that, it sends tags' data to an application that can use the data. The data collected from the tag can then be sent either directly to a host computer or stored in a portable reader and uploaded later to the host computer.

Operating frequency

The system is mainly used in three frequency bands.

1) Low-frequency band(LF):

General Frequency Range: 30 – 300 kHz

Primary Frequency Range: 125 – 134 kHz

Read Range: Contact – 10 Centimeters

2) High-frequency band (HF):

Primary Frequency Range: 13.56 MHz

Read Range: Near Contact – 30 Centimeters

3) Ultra-high frequency band:

General Frequency Range: 300 – 3000 MHz

Primary Frequency Ranges: 433 MHz, 860 – 960 MHz

RFID tags

There are three different kinds of tags.

1) **Active tags**: Active tag has its power source for internal circuitry and sends the Reader's response.

Primary frequency range: 433 MHz, (Can use 2.45 GHz – under the Extremely High-Frequency Range)

Read range: 30 – 100+ Meters.

2) **Passive tags**: They used to get their power from the Reader's incoming radio waves.

Primary frequency range:860 – 960 MHz

Read Range: Near Contact – 25 Meters

3) **Semi-passive tags**: They have a power supply for internal circuitry. But, it relies on the radio waves received from the Reader for sending the response.

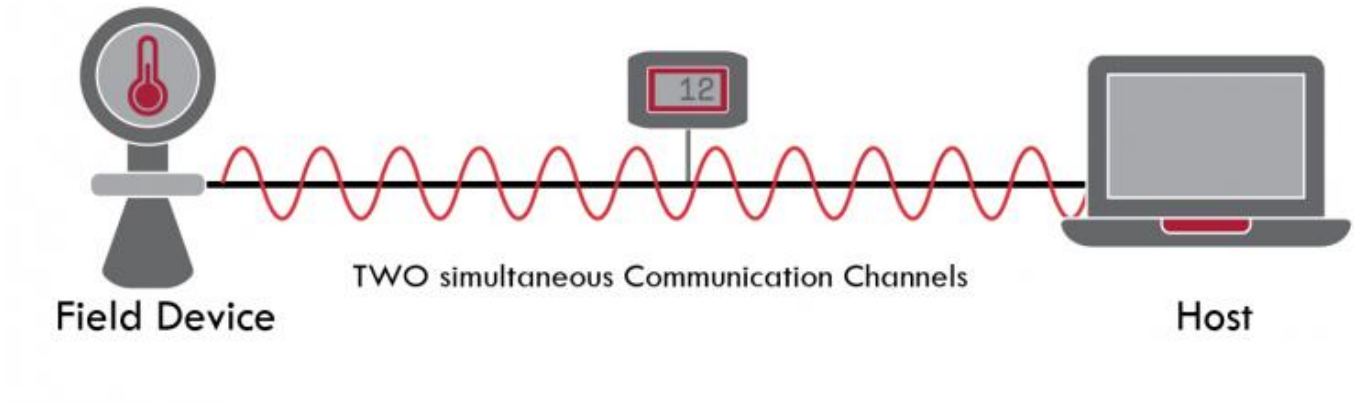
RFID Technology in IoT

The evolution of IoT is coinciding with that of RFID and sensor technologies. This technology is a vital and fundamental groundwork for IoT. RFID tags can make **everyday physical objects communicate, and the central hub and reports their status**. For instance, Attached RFID tags to finished items, products, and assembled goods can speed up service operation. Tracking of tagged persons or objects and sending real-time data to a model. It **creates a connected device system.** That continuously transmit data about their location, conditions, amount, etc. In short, These capabilities make the foundation for building an IoT system.

Applications tracking assets can make a wide range of business activities more efficient. RFID technology in IoT connects up the things into a network and makes them create and send data. Unlike simple RFID tags that don't actively broadcast signals. So, sensor-enabled tags are used. It can g**enerate and send metrics and other data in real-time.** It expands the connected devices' capabilities and makes them truly smart. As a result, RFID is one of the key technologies that the Internet of Things depends on.

HART

HART is a bi-directional communication protocol that provides data access between intelligent field instruments and host systems. A host can be any software application from technician's hand-held device or laptop to a plant's process control, asset management, safety or other system using any control platform. Communication occurs between two HART-enabled devices, typically a smart field device and a control or monitoring system. Instrumentation grade wiring and standard termination practices assure reliable communication.



HART provides two simultaneous communication channels, one analog, the other digital: A 4-20mA signal communicates the primary measured value (PV) as an analog value of current using the wiring that provides power to the instrument. The host system then converts the current value to a physical value according to parameters defined by HART Software. For example, 7 mA = 80 degrees F.

Digital device information is communicated by encoding a digital signal, generally using a technique known as Frequency Shift Keying on the same 4-20mA wiring used for analog communications. The digital signal contains information from the device including PV, device status, diagnostics, and additional measured or calculated values, etc.

Together, the two communication channels provide a complete field communications solution that is easy to design, simple to use, low cost and extremely reliable.

How HART Works

“HART” is an acronym for Highway Addressable Remote Transducer. The HART Protocol makes uses Frequency Shift Keying (FSK) standard to superimpose digital communication signals at a low level on top of the 4-20mA. This enables two-way field communication to take place and makes it possible for additional information beyond just the normal process variable to be communicated to/from a smart field instrument.

The HART Protocol communicates at 1200 bps without interrupting the 4-20mA signal and allows a host application (master) to get two or more digital updates per second from a smart field device. As the digital FSK signal is phase continuous, there is no interference with the 4-20mA signal. The HART Protocol provides two simultaneous communication channels: the 4-20mA analog signal and a digital signal. The 4-20mA signal communicates the primary measured value (in the case of a field instrument) using the 4-20mA current loop - the fastest and most reliable industry standard. Additional device information is communicated using a digital signal that is superimposed on the analog signal.

The digital signal contains information from the device including device status, diagnostics, additional measured or calculated values, etc. Together, the two communication channels provide a low-cost and very robust complete field communication solution that is easy to use and configure.

wireless HART

WirelessHART

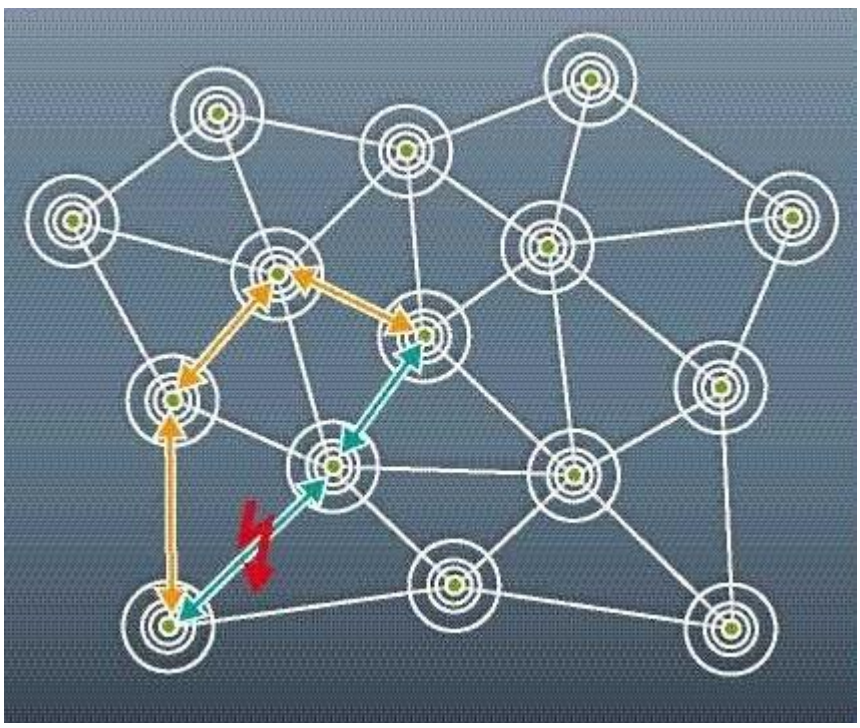
WirelessHART uses a **2.4 GHz band**—license-free and used worldwide—as a transfer medium for several radio technologies, including WLAN, Bluetooth, and ZigBee. But, **WirelessHART** is much more than a WLAN variant.

WirelessHART uses a **flat mesh network** where all radio stations (field devices) form a network. Every participating station serves simultaneously as a **signal source** and a **repeater**. The original transmitter sends a message to its nearest neighbor, which passes the message on until the message reaches the base station and the actual receiver. In addition, **alternative routes** are set up in the initialization phase. If the message cannot be transmitted on a particular path, due to an obstacle or a defective receiver, the message is automatically passed to an alternative route. So, in addition to extending the range of the network, the **flat mesh network** provides redundant communication routes to increase reliability.

The communication in the **Wireless Network** is coordinated with TDMA (Time Division Multiple Access), which synchronizes the network participants in 10 ms timeframes. This enables a very reliable (collision-free) network, and reduces the lead and lag times during which a station must be active.

To avoid jamming, **WirelessHART** uses also FHSS (Frequency Hopping Spread Spectrum). All 15 channels as defined in IEEE802.15.4 are used in parallel; **WirelessHART** uses FHSS to “hop” across these channels. Channels that are already in use are blacked out to avoid collisions with other wireless communication systems.

The combination of 10s synchronization and 15 channels allows 1500 communications per second.



3.5 NFC, Bluetooth, Z wave, ISA100.11.A

NFC

Near Field Communications (NFC) is a very short-range wireless technology that is based on and is similar to RFID technology. This very short range (a few inches) allows for some unique use experiences – you almost have to touch two devices, bringing them very close to each other to initiate a transaction.

Bluetooth

Bluetooth: **A short range wireless communication technology for exchanging data using short-wavelength UHF radio waves (2.4 to 2.485 GHz) and build personal area networks (PANs)**.

Bluetooth has been in the tech market as a wireless channel of connection between devices since Ericsson invented it in 1994. Since then, Bluetooth technology has evolved and has become the go-to wireless connectivity solution for wearables, gadgets, and other devices. Nowadays, you will find Bluetooth everywhere; cars, speakers, wearables, medical devices, wireless headphones, shoes, etc. If you own any modern device, it is safe to assume that you have encountered and used Bluetooth technology at one point or the other. In other words, Bluetooth is a short-range wireless technology medium used for exchanging data between two electronic devices (usually mobile) over a short distance. This process completely eliminates the primitive use of cables for connectivity. A typical example is how you can listen to music with a headset on the go without having to plug it into the headset jack of your mobile device.

Bluetooth exchange works using UHF radio waves, otherwise known as short wave radio, with radio bands ranging from 2.402 GHz to 2.480 GHz and building a Personal Area Network (PAN). Typically, a master Bluetooth device can connect to a maximum of seven devices at a go. Still, some Bluetooth devices do not have the capacity to connect up to this number of devices. However, this kind of connection is called a piconet, an ad hoc computer network created at that moment using Bluetooth technology. And in this technology system, connected operate in a master-slave relationship. For example, suppose you initiate a connection between a phone and a wireless headset through a headset; in that case, the headset becomes the master (the initiator), and the phone is the slave. Subsequently, both devices can switch roles and have the phone operate as the master, while the headset becomes the slave. Ultimately, in a Bluetooth piconet, it is possible for a master to have seven slaves; and for a slave to have more than one master.

Evolution of Bluetooth

Bluetooth technology has evolved from the classic Bluetooth stage to the smart Bluetooth stage obtainable today, including the latest version, Bluetooth 5, which has four times the range, double the speed, and 800% more data broadcasting frequency, compared to earlier versions. These additional features will increase the number of Bluetooth IoT devices and ultimately make Bluetooth a smart choice for firms with extensive infrastructure because of the 100% uptime and cost-effective options available by deploying Bluetooth 5 IoT devices.

Bluetooth 5.0 also brought with it a mode that allows for the correction of errors called Forward Error Correction (FEC). FEC allows data lost from errors that occur due to noise and interference to be collected by the receiver when needed.

Bluetooth Classic and Bluetooth Low Energy (BLE)

There are two Bluetooth variants of the Bluetooth technology; hence all Bluetooth devices can be classified into two categories – Bluetooth Classic and Bluetooth Low Energy (BLE). On the one hand, the Bluetooth Classic is usually used in wireless speakers, headsets, and car infotainment systems. On the other hand, Bluetooth Low Energy (just as the name implies) is more prominent in applications that are keen on power consumption and transfer small amounts of data less often. In other words, BLE is commonly found in battery-powered devices like mobile phones, sensor devices, etc. As opposed to the Bluetooth Classic that consumes high energy, Bluetooth Low Energy thrives on reduced power consumption and cost, even while maintaining a similar communication range as Bluetooth Classic.

It is important to note that these two kinds of Bluetooth devices are inharmonious even when they share the same brand and specification document. That is to say, a Bluetooth Classic cannot work together with Bluetooth Low Energy. So, it is not farfetched why some devices like smartphones integrate both Bluetooth variants to communicate with and connect to either type of Bluetooth present in other devices.

Z wave

The concept of Z-Wave technology is that it uses a low-power RF radio circuitry which is embedded into home electronics devices and systems.

Z-Wave technology is aimed at a number of wireless home automation areas including lighting, residential access control, entertainment systems and all forms of household appliances. Z-Wave can be used within a network (Home Area Network, HAN), and can therefore be used to set up all areas of home automation, possibly controlled by a single controller.

With many more home devices becoming remotely controlled, Z-Wave technology is seen as having a large market opportunity, especially with the talk about the Internet of Things, IoT becoming more widespread.

Z-Wave modules are available from a variety of sources relatively cheaply and therefore they provide an excellent format for home automation.

The International Telecommunications Union, ITU has included the Z-Wave PHY and MAC layers as an option in its G.9959 standard. This defines a set of guidelines for sub-1-GHz narrowband wireless devices.

Z-Wave Alliance

To support and promote Z-Wave technology, and organisation known as the Z-Wave Alliance was founded.

This is a consortium of manufacturers who have products int his sector. By having a common standard, the market share is increased as users are able to select products from different manufacturers to more exactly suit their needs.

The Alliance also provides certification of products, thereby enabling standards to be maintained and user to select products they know will operate alongside each other.

Z-Wave technology basics

Z-Wave uses a mesh network topology and accordingly any non battery powered device acts as a signal repeater, enabling reliable connections from one node to the next. Battery powered devices do not act as repeaters as this would result in high levels of battery drain.

The mesh network approach means that, the more devices in the network, the more resilient it becomes.

the frequencies used for Z-Wave are below that of the normal 2.4 GHz Wi-Fi band and this enables better penetration of walls and other items found in all homes, but in addition to this, the mesh network means that data to be transferred can intelligently routed by the network to get around obstacles and thereby obtaining robust whole-home coverage.

Z-Wave typically has a range of about 100 metres or 328 feet in open air. However walls and other items in the home will considerably reduce this and therefore it is recommended that the maximum device spacing Z-Wave network is around 10 metres of 30 feet. Anything closer will provide better communications.

The Z-Wave signal can hop roughly 600 feet, and Z-Wave networks can be linked together for even larger deployments. Each Z-Wave network can support up to 232 Z-Wave devices allowing the flexibility to provide sufficient devices for a complete automated home.

Z-Wave RF interface

The Z-Wave technology uses a simple RF interface to ensure that encode and decode functions are able to be achieved with a minimum level of processing, and hence power consumption. It also ensures that the RF signal can be transmitted with the maximum efficiency.

Z-Wave Network layer

The Z-Wave network layer is the area of the protocol stack that controls the data exchange between the different devices, sending data over the RF or radio layer.

The network layer consists of three layers:

- ***Media Access Layer:*** Referred to as the MAC, this layer controls the basic usage of the wireless hardware. It does this in a manner that is not visible to the end user.
- ***Transport Layer:*** The transport layer within the Z-Wave technology protocol stack controls message transfer between two wireless nodes and ensures error free transmission.
- ***Routing Layer:*** The routing layer manages the Z-Wave wireless mesh capabilities. It enables the various nodes to link together and route messages from one node to another if one node is out of range of another..

Z-Wave devices

In order to have a hierarchy within a wireless network, various types of Z-Wave device are specified:

- ***Controller:*** As the name implies, these devices are those that control other Z-Wave devices. Controller devices are factory programmed with what is termed a Home ID. This cannot be changed by the user.
- ***Slave:*** Slave devices are those that are controlled by controllers. Slave devices do not have a pre-programmed Home ID, but instead they take the Home ID assigned to them by the Z-Wave network controller.
- ***Routing slave:*** This form of Z-Wave slave is one that knows its neighbours and has partial knowledge of routing table. It can reply to the node from which it has received the message. It can also send unsolicited messages to a number of predefined nodes to which it has routes.

ISA100.11.A

ISA100.11a is a IoT protocol used by many IoT devices and is most likely to be found in industrial settings, like petroleum refineries and manufacturing plants. As you might understand from its name, ISA100.11a was developed and promoted by the Industrial Society of Automation (ISA). It provides a platform that can be built upon to supplement some controls. It also presents a reliable method of gathering data that is less expensive and maintenance intensive than traditional wired communications.

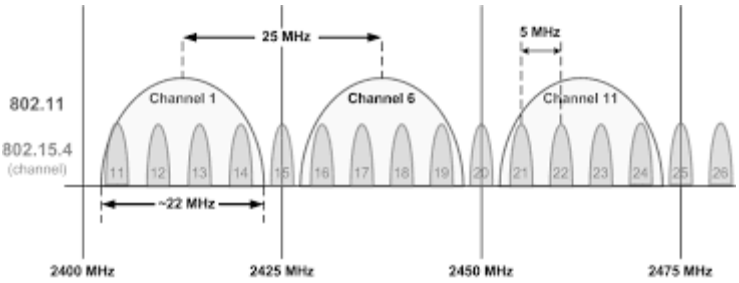
IoT is supposed to make our lives easier, or more accessible; If you use Alexa or Google Home devices, you have interacted with IoT devices. Chances are, even if you don't have one of these hubs, you have some manner of IoT devices in your home — even if it's just your utility meters. Let's cover some basic IoT protocols.

How Does ISA100.1A Work?

ISA100.11a is based on a PHY (Physical Layer) and MAC layer (1 and 2 of the OSI model) called 802.15.4. You're likely at least passingly familiar with this standard, as it also governs Bluetooth and BLE. 802.15.4 covers a lot of spectrum, including some of the 900MHz band, but is most famously in the 2.4 GHz band, where it crosses over WiFi.

When your phone is connected to your headphones, or speaker, you're using this PHY. 802.15.4 is considered a LR-PAN or Low Rate Personal Area Network type of PHY. What makes ISA100.11a so useful is that it takes advantage of the lower rate structure to allow it to have a longer reach. ISA100.11a, however, doesn’t use the same channel structure as some of you may be familiar with for WiFi. It has 16 channels in

the same space, and uses a smaller channel width as well. You can see the channels commonly used in this band below.



Key Features of 802.15.4 PHY

We're going to take a deep dive into the unique and important parts of the protocol now. This isn't an exhaustive list, of course. We don't have space here to go into all of it, and it gets deep pretty quickly.

TDMA

ISA100.11a uses a version of TDMA (Time Division Multiple Access), a standard method of contention free, or more deterministic access to the medium. In TDMA, all the clocks on the devices that make up the network are rigorously synchronized, and the timing is used to provide slots for the individual devices to transmit without getting in each other's way through collisions.

For example, slots are allocated to device A- Device A is set to report to the system every 60 seconds. It would be allocated the number of slots (10 ms duration) that it needs to send its data on a rotating basis. These slots are "reserved" from the total amount of slots available to be used, but only as many times as it needs. In contrast, a device that needs to send twice the data, or reports twice as often would generally be allocated twice the slots. Time slots can also be allocated dynamically by a network manager or analogous module.

It should be pointed out that this is also used in conjunction in both with frequency hopping, and the two differ by the domain that they vary in. TDMA varies the time domain, when the transmission is sent, whereas DSSS varies the frequency space that the transmission is sent on.

Superframes

A superframe is a unique feature to these types of networks: a collection of timeslots repeating on a cyclic schedule. More plainly, superframs are sets of timeslots that repeat according to a pattern. The number of timeslots included in a particular superframe determines its length, and therefore, how often it repeats, and logically extended, how often a device that uses this superframe may communicate.

Superframes can be conceptualized as a method of organizing communication through "links". Each time slot in a given superframe is dedicated to a specific link, such as device A communicating with Device B. Conversely, from Device B's point of view it is scheduled to listen for device A in the same time period. Multiple superframes can overlap in a network in the same time period. Remember, each device has to be viewed as its own entity, so this does not mean that a device is listening and talking at the same time.

A device can also participate in multiple superframes, and participation in a superframe is not considered compulsory. As mentioned before, the period of a superframe is dependent on the length of the superframe and is how often that particular superframe repeats. How often the superframe repeats is inversely proportional to the length of the superframe. (EG If the Superframe is Length 2, it will repeat at rate 1. If it is length 1, it will repeat at rate 2, twice as often, because it is half as long.) Readers who are familiar with traditional 802.11 WLANs will notice that this is a stark contrast to the pseudo-random access that WLANs enjoy through the CSMA/CA in the order that is imposed on the network. One should recall that the focus of these networks is not arbitrary user access, but rather instrumentation and controls responding (often) in cycles as long as once a day.

A key learning for operators of these networks is this: shorter superframes mean lower latency and higher bandwidth, but are more battery intensive to the devices themselves and the entire superframe need not be filled with transmissions. Superframes can be filled out with null timeslots to artificially lengthen the superframe and increase the amount of time between individual transmission cycles.

Network Layer Peculiarities

The network layer in ISA100.11a uses the 6LoWPAN model. At the network level, most of the

the unique character of ISA100.11a drops off, and gives way to 6LoWPAN. Addresses assigned to devices are IPv6 analogues, and allows up to 128-Bit addresses to be used. At the high end, ISA100.11a addressing means that the network is nearly infinite in scalability and eliminates the possibility of duplicate addressing, even in hyper-dense environments. This also means that ISA100.11a allows for connectivity between different devices, even across a linked routing backbone. Backbone routing complicates the process for ISA100.11a and provides a different Schema for different situations.

The network layer in ISA100.11a is also notable for having an intrinsic power-saving mechanism built in.

Using fully 128-bit addresses isn't always practical in a small network. Small here is considered to be less than 1000 devices, compared with a WAN network or similar. As transmitting 128-bits for every address is power consuming and not an efficient use of airtime, the standard allows for 16-bit aliases to be used over the local subnet. This can be conceptualized as the way we use names in the social world — within your family you might use a shorter version of a name, which is generally understood.

Your uncle Rich, for example, within your family is known simply as Rich. In the wider world he is known by his full name which would be cumbersome and unwieldy to use in common conversation but may be required to identify him among a much larger group like the general public. In this example, Lord Rich Richington Pendleton the Third, or similar. Imagine having to use full names every time you address someone instead of pronouns! The system manager assigns 16-bit aliases and 128-bit full addresses in an ISA100.11a network. Each device maintains an address table with these name associations, which is

constructed during the initial join process. When a transmission traverses the network backbone, the network layer of the backbone or the device performing the "Gate" function of the subnet handles this translation from 16 to 128 bit and the inverse.

Routing

Routing in a ISA100.11a network is handled at the data link layer in most cases, but can be

handled by the network layer in some cases as mentioned previously. Inside a discrete subnet, between devices, the traffic is routed at Layer 2 through the data link layer. However, when the traffic passes a backbone router, or crosses into another subnet, it's both translated into the 128-bit address and handled by the network layer by necessity.

It's notable that this is not considered a problem because this is no longer a WIRELESS medium but instead being transmitted over wireline. Power conservation on wireline is not considered a priority. A third schema is present when the data is moved over a gateway, the formal boundary of the WSN (wireless sensor network). The entire data payload must be de- and re-encapsulated as it's being actively reinterpreted to fit within the plant network and being made ready to be used with the controls or sensor interpretation networks. This combination of Layer2 and Layer 3 routing is referred to as Mesh-Under and Routing-Over.

Final Thoughts

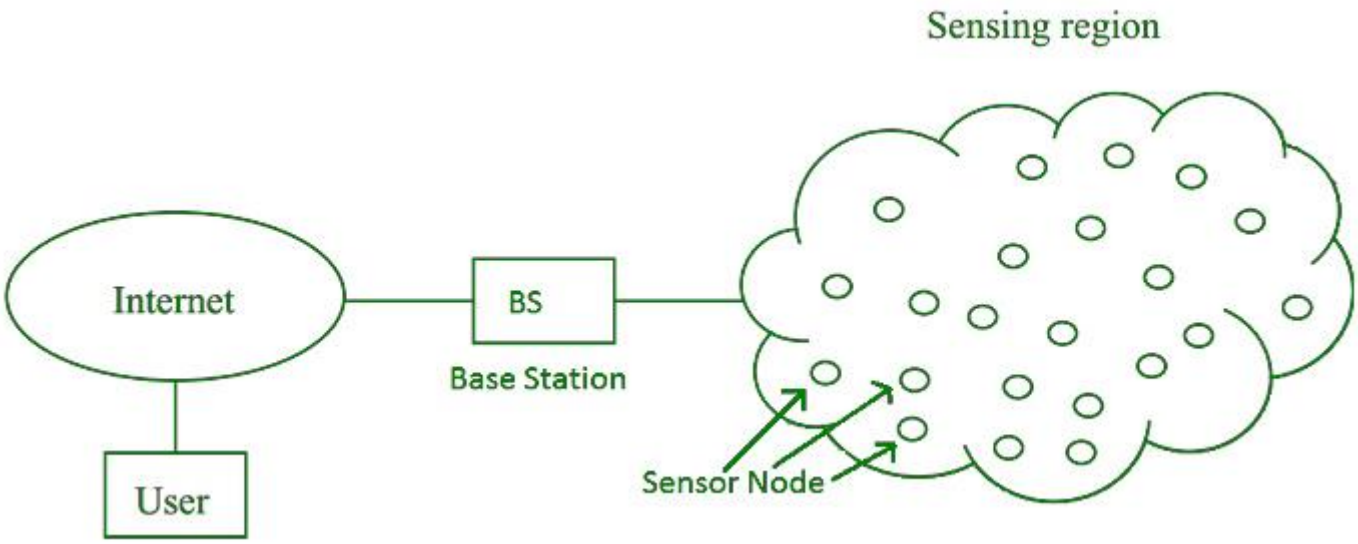
That's a short but deep preview of ISA100.11a. It's much more than this, but anything less than a week class may not be fair to the protocol. That said, the above highlights some major differences between it and other protocols you might run into more often like WiFI. It is the author's strong conviction that IoT will only grow and that network engineers must be prepared for this growth, or be left behind.

4. Wireless Sensor Networks

Wireless Sensor Network (WSN) is an infrastructure-less wireless network that is deployed in a large number of wireless sensors in an ad-hoc manner that is used to monitor the system, physical or environmental conditions.

Sensor nodes are used in WSN with the onboard processor that manages and monitors the environment in a particular area. They are connected to the Base Station which acts as a processing unit in the WSN System.

Base Station in a WSN System is connected through the Internet to share data.



WSN can be used for processing, analysis, storage, and mining of the data.

Applications of WSN:

1. Internet of Things (IOT)
2. Surveillance and Monitoring for security, threat detection
3. Environmental temperature, humidity, and air pressure
4. Noise Level of the surrounding
5. Medical applications like patient monitoring
6. Agriculture
7. Landslide Detection

Challenges of WSN:

1. Quality of Service
2. Security Issue
3. Energy Efficiency
4. Network Throughput
5. Performance
6. Ability to cope with node failure
7. Cross layer optimisation
8. Scalability to large scale of deployment

Components of WSN:

1. **Sensors:**
   Sensors in WSN are used to capture the environmental variables and which is used for data acquisition. Sensor signals are converted into electrical signals.
2. **Radio Nodes:**
   It is used to receive the data produced by the Sensors and sends it to the WLAN access point. It consists of a microcontroller, transceiver, external memory, and power source.
3. **WLAN Access Point:**
   It receives the data which is sent by the Radio nodes wirelessly, generally through the internet.
4. **Evaluation Software:**
   The data received by the WLAN Access Point is processed by a software called as Evaluation Software for presenting the report to the users for further processing of the data which can be used for processing, analysis, storage, and mining of the data

4.1 Introduction

Wireless Sensor Network (WSN)is **deployed in a large area with a large number of wireless sensors nodes in an ad-hoc manner that is used to monitor the system, physical or environmental conditions**. The data is then sent to gateway for processing or edge computing.

4.2 Components of a sensor node

Sensor nodes are used in WSN with the onboard processor that manages and monitors the environment in a particular area. They are connected to the Base Station which acts as a processing unit in the WSN System.

...

Components of WSN:

- Sensors: ...
- Radio Nodes: ...
- WLAN Access Point: ...
- Evaluation Software:

4.3 Modes of Detection

Surveillance and Monitoring for security, threat detection. Environmental temperature, humidity, and air pressure. Noise Level of the surrounding. Medical applications like patient monitoring.

4.4 Challenges in WSN

- Sensor networks do not fit into any regular topology, because while deploying the sensor nodes they are scattered
- Very limited resources

  WSN PROTOCOL STACK

  - Limited memory,
  - Limited computation
  - Limited power
- It comes under fewer infrastructures and also maintenance is very difficult.
- Unreliable communication
  - Unreliable data transfer
  - Conflicts and latency
- Sensor node relies only on battery and it cannot be recharged or replaced. Hardware design for sensor node should also be considered.
- Achieving synchronization between nodes is also another issue.
- Node failure, topology changes and adding of nodes and deletion of nodes is another challenging issue.

4.5 Sensor Web

An IoT (Internet of Things) Wireless Sensor Network (WSN) refers to a group of spatially dispersed and dedicated sensors for monitoring, and recording the physical conditions of the environment, and collectively pass on such data through a wireless network to a internet-based location.

4.6 Cooperation and Behaviour of Nodes in WSN

We present a game-theoretic self-organizing approach for scheduling the radio activity of wireless sensor nodes. Our approach makes each node play a win-stay lose-shift (WSLS) strategy to choose when to schedule radio transmission, reception and sleeping periods. The proposed strategy relies only on local interactions with neighboring nodes, and is thus fully decentralized. This behavior results in shorter communication schedules, allowing to not only reduce energy consumption by reducing the wake-up cycles of sensor nodes, but also to decrease the data retrieval latency. We implement this WSLS approach in the OMNET++ sensor network simulator where nodes are organised in three topologies: line, grid and random. We compare the performance of our approach to two state-of-the-art scheduling protocols, namely S-MAC and D-MAC, and show that the WSLS strat- egy brings signifficant gains in terms of energy savings, while at the same time reduces communication delays. In addi- tion, we show that our approach performs particularly well in large, random topologies.

Introducing cooperation into a wireless sensor network (WSN) has gained much attention in the recent few years mainly because of the significant effect it has on optimizing energy consumption and on enhancing the lifetime and the overall performance of the network. Cooperation can be exploited at different levels, ranging from a collection of nodes collaborating to forward the data they gathered from the environment towards the base station through efficient data aggregation and clustering techniques, to nodes collaborating to report events occurrences, track targets or control the topology. Motivated by a large variety of attractive wireless sensor applications, such as environmental monitoring, smart environments and healthcare applications, we survey mechanisms that take advantage of cooperation among sensor nodes in the network for the purpose of delivering information reliably and efficiently to nodes of the network that are interested in receiving it. We provide detailed overviews and highlight the importance of cooperation from different perspectives.

4.7 Self Management of WSN

A Wireless Sensor Network (WSN) is deployed primarily with the interest of collecting information about an area of interest or some events of interest occurring in that area. A WSN transmits the sensed information to the end users in the form of data-packets. The number of packets is directly proportional to the transmission rate of the nodes in the WSN. In this paper, a distributed self-management scheme for WSNs, named as Information Theoretic Self-Management (InTSeM), for WSNs, is proposed. The scheme helps the nodes in the network to adapt themselves with the changes in their environment and to select an appropriate transmission rate using an information theoretic metric called Symmetric Kullback-Leibler Distance. This dynamic selection of transmission rate conserves the energy of nodes while maintaining the quality of data. Extensive simulation results show that InTSeM performs better than other schemes with fixed transmission rate.

4.8 Social sensing WSN

Social sensing broadly refers to **a set of sensing and data collection paradigms where data are collected from humans or devices on their behalf**.

4.9 Application of WSN

Applications of WSN:

Surveillance and Monitoring for security, threat detection. Environmental temperature, humidity, and air pressure. Noise Level of the surrounding. Medical applications like patient monitoring.

4.10 Wireless Multimedia sensor network

Wireless Multimedia Sensor Networks (WMSNs) have emerged and shifted the focus from the typical scalar wireless sensor networks to networks with multimedia devices that are capable to retrieve video, audio, images, as well as scalar sensor data. WMSNs are able to deliver multimedia content due to the availability of inexpensive CMOS cameras and microphones coupled with the significant progress in distributed signal processing and multimedia source coding techniques. In this paper, we outline the design challenges of WMSNs, give a comprehensive discussion of the proposed architectures, algorithms and protocols for the different layers of the communication protocol stack for WMSNs, and evaluate the existing WMSN hardware and testbeds.

4.11 Wireless Nanosensor Networks

Wireless Nanosensor Network Model. **The nanosensors can be deployed arbitrarily at organs of the human body and they may be moved by body fluid**. We assume that the nanosensors are distributed in 3-dimensional space in the nanonetworks according to a homogeneous spatial Poisson process.

4.12 Underwater acoustic sensor networks

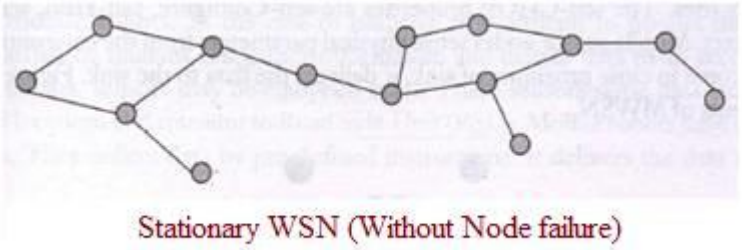
An underwater acoustic communication network is **con- sidered which consists of a large number of nodes operating in a shallow-water environment at a depth of approximately 50–100 m**. The nodes are mounted on the bottom and separated by distances of up to 10 km.

4.13 WSN Coverage

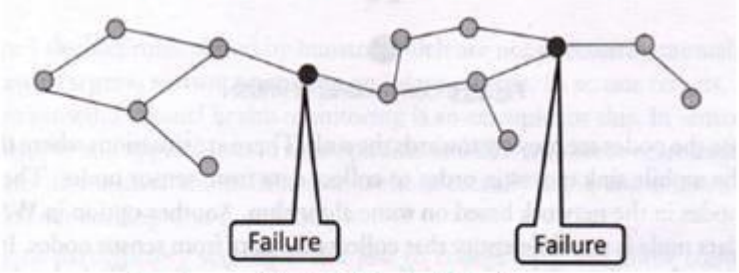
In accordance to the subject of interest, three types of coverage in WSNs may be identified: **area coverage, point coverage, and barrier coverage**. Area (or regional) coverage expresses the ability of the network to monitor an area of interest, meaning that all points within this area are always monitored.

4.14 Stationary WSN, Mobile WSN

Stationary WSN



Stationary WSN (Without Node failure)



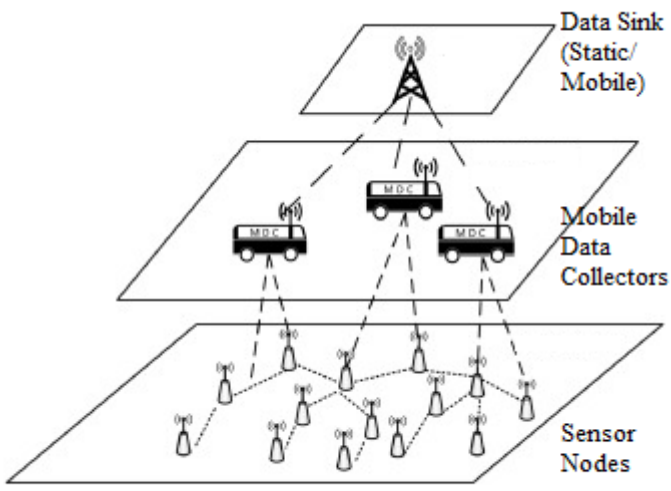
Stationary WSN (Node failure)

The wireless sensor network (WSN) in which sensor nodes are static is called as stationary WSN. This is shown in the figure-2 above.

Advantages : The stationary WSNs are easy to deploy and easier to maintain. The sensor nodes can be installed at optimized place and hence more number of nodes can be placed in lesser space.

Disadvantages : The topology of this WSN type can be changed automatically. If any node fails, it results into network partition.

Mobile WSN



Mobile WSN Architecture

The WSN in which nodes are mobile is known as mobile WSN. It is also called as MANET (Mobile Ad hoc network). It is infrastructure less network of mobile devices or nodes which has following properties i.e. self-configure, self-heal, self-optimize and self-protect.

The mobile sensor nodes senses physical parameters and when they reach near to sink, it delivers the data collected. There are situations where sink itself is mobile in nature and collects the data from stationary sensor nodes.

The figure depicts mobile WSN architecture in which sensor nodes and sink are immobile. Hence mobile data collectors are used which collects the data from sensor nodes and delivers to the data sink. These mobile data collectors are called data mules and the process of data collection/delivery is referred as data muling technique.

The mobile WSN is used in following domains.

Underwater Mobile WSN: In this type of MWSN, sensor nodes sense various parameters under the sea or water levels.

Terrestrial Mobile WSN: In this type of MWSN, sensor nodes are deployed over the land surface for various applications which include wildlife monitoring, surveillance, object tracking etc.

Aerial Mobile WSN: In this type of mobile WSN, nodes fly on air and sense data.

5. **M2M Communication**

M2M systems use point-to-point communications between machines, sensors and hardware over cellular or wired networks, while IoT systems **rely on IP-based networks to send data collected from IoT-connected devices to gateways, the cloud or middleware platforms**.

M2M stands for **Machine to Machine** communication. It is a direct communication system between the devices using wired or wireless communications channels without any human interaction. It collects the data and shares it with other connected devices. It is a technology that allows devices without the use of the internet to connect between devices. Various applications, such as defense, monitoring and tracking, production and facility management, are provided by M2M communications.

M2M technology may be present in offices, shopping malls, houses, and many other places. A common example of a machine to machine is controlling electrical devices like fans and bulbs using Bluetooth from the smartphone. Here, the smartphone and electrical devices are the two interacting devices with each other.

5.1 M2M communication

Machine to Machine : This is commonly known as Machine to machine communication. It is a concept where two or more than two machines communicate with each other without human interaction using a wired or wireless mechanism. M2M is an technology that helps the devices to connect between devices without using internet. M2M communications offer several applications such as security, tracking and tracing, manufacturing and facility management.

5.2 M2M Ecosystem

M2M (Machine-to-Machine) refers to **the flow of data between physical objects, without the need for human interaction**. M2M connectivity has opened a multi-billion dollar revenue opportunity for mobile operators, MVNOs and service aggregators, addressing the application needs of several verticals markets.

5.3 M2M service Platform

An M2M platform can have a wide range of features and functionalities, but overall they manage data transmitted by devices, the backend systems that process the data, the provisioning of software updates to devices, and general device lifecycle administration. **M2M platforms can support IoT ecosystems**.

5.4 Interoperability

IoT interoperability is **the capacity for multiple components within an IoT deployment to effectively communicate, share data and perform together to achieve a shared outcome**. Organizations must be able to transmit and understand data throughout all the connections from devices to the cloud.

6. **Programming with Arduino**

1. Download & install the Arduino environment (IDE)

2. Launch the Arduino IDE
3. If needed, install the drivers
4. Connect the board to your computer via the USB cable
5. Select your board
6. Select your serial port
7. Open the blink example
8. Upload the program.

6.1 Features of Arduino

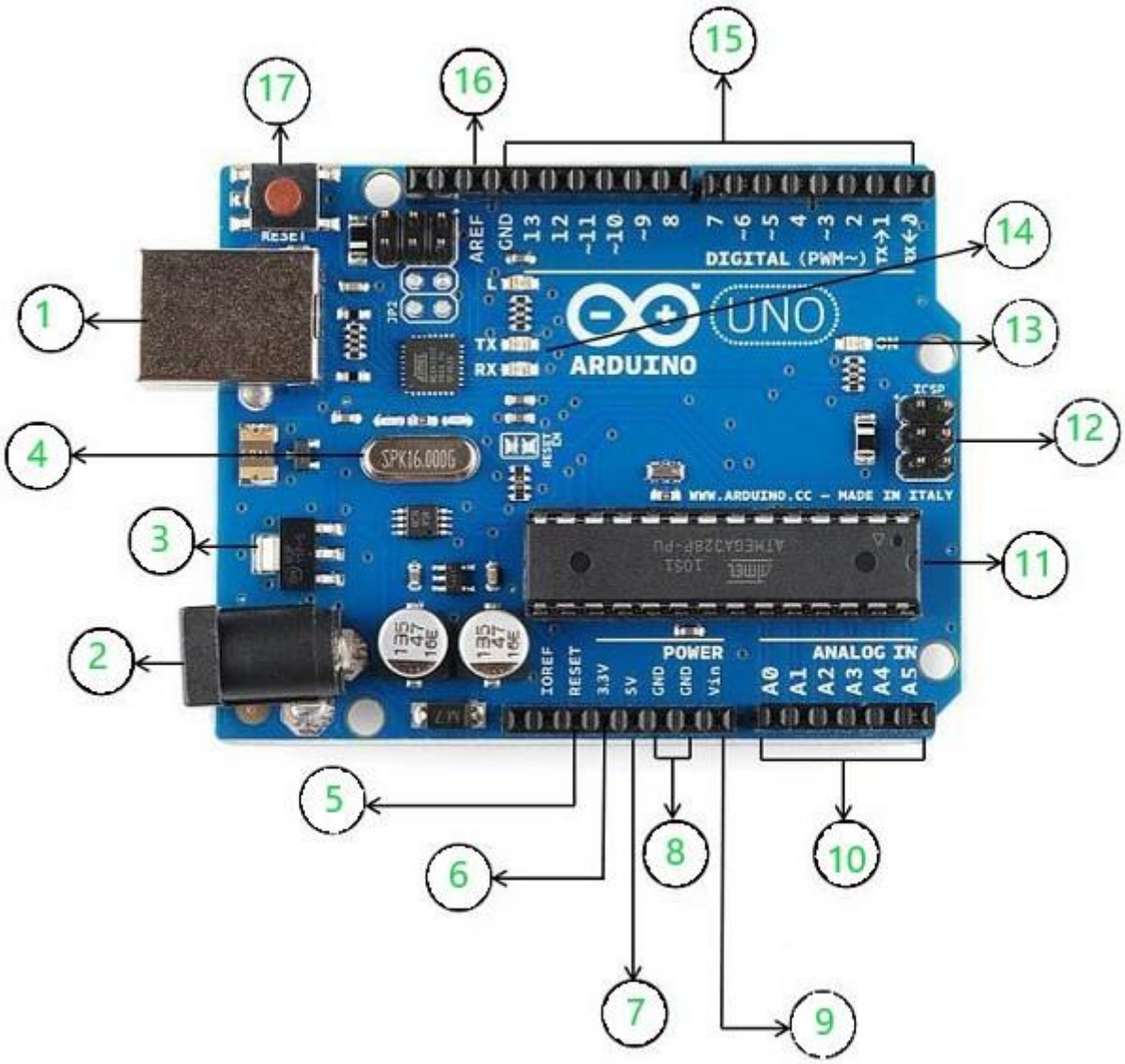
Arduino was a project started at Interaction Design Institute Ivrea (IDII) in Ivrea, Italy, with its primary goal being creating affordable and straightforward tools for non-engineers to use and create digital projects. During its infancy, the project consisted of just three members- Hernando Barragán, Massimo Banzi, and Casey Reas. Hernando Barragán worked under the guidance of Massimo Banzi and Casey Reas and created a development platform called Wiring as his masters' thesis project at IDII. The development platform consisted of the ATMega168 microcontroller as its brains and used an IDE based on Processing, which was co-created by Casy Reas. Later, Massimo Banzi, along with two other students from IDII, namely- David Mellis and David Cuartielles, added support for the cheaper ATMega8 microcontroller. The three, instead of working on developing and improving Wiring, they forked it and renamed the project to Arduino. The initial core Arduino team consisted of Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis, but Barragán was not included.

The Hardware

Now that you know the origin of Arduino, it is essential to get yourself acquainted with the hardware that Arduino as a company offers. One of the main reasons for Arduino being so accessible and affordable across the globe is because all of the Arduino hardware is open-source. Being open-source has a plethora of advantages- anyone can access the design and build of the device and make improvements; anyone can use the same hardware design to create their product lineup. Since Arduino is open-source, it has its own devoted community that strives to help the core company develop and improve its hardware products. Another significant advantage of being open-source, especially in the case of hardware, is that local companies can create replicas of the products, making it more accessible and affordable to the local consumers as it avoids hefty customs and shipping charges. All of these advantages contribute to Arduino being so widespread, affordable and ever-improving.
It is necessary to know that Arduino doesn't necessarily offer just one piece of hardware, it provides a range of boards, each of which caters to a different level of expertise and have different use-cases altogether. Arduino Uno is one of the most basic and popular boards that Arduino offers. This is because it features an ATMega328 microcontroller that is both cheap and powerful enough for most basic beginner-level projects. Once you're familiar with Arduino IDE, you can move up to boards with more powerful and sophisticated chipsets like the MKR range which is concerned with IoT applications and inter compatibility, or the Nano range which as the name suggests is designed to keep the form factor as small as possible while packing most of the features and power of the full-sized boards.

Understanding the Hardware

Note: Since this guide is aimed at absolute beginners, this article is limited to getting started with Arduino Uno.
So you got yourself an Arduino Uno, and you're ready to jump into the world of electronics and join the community of makers from around the world, but before you begin with programming and external circuitry through breadboards and whatnot, it is necessary to understand the layout and circuitry of your Arduino Uno.

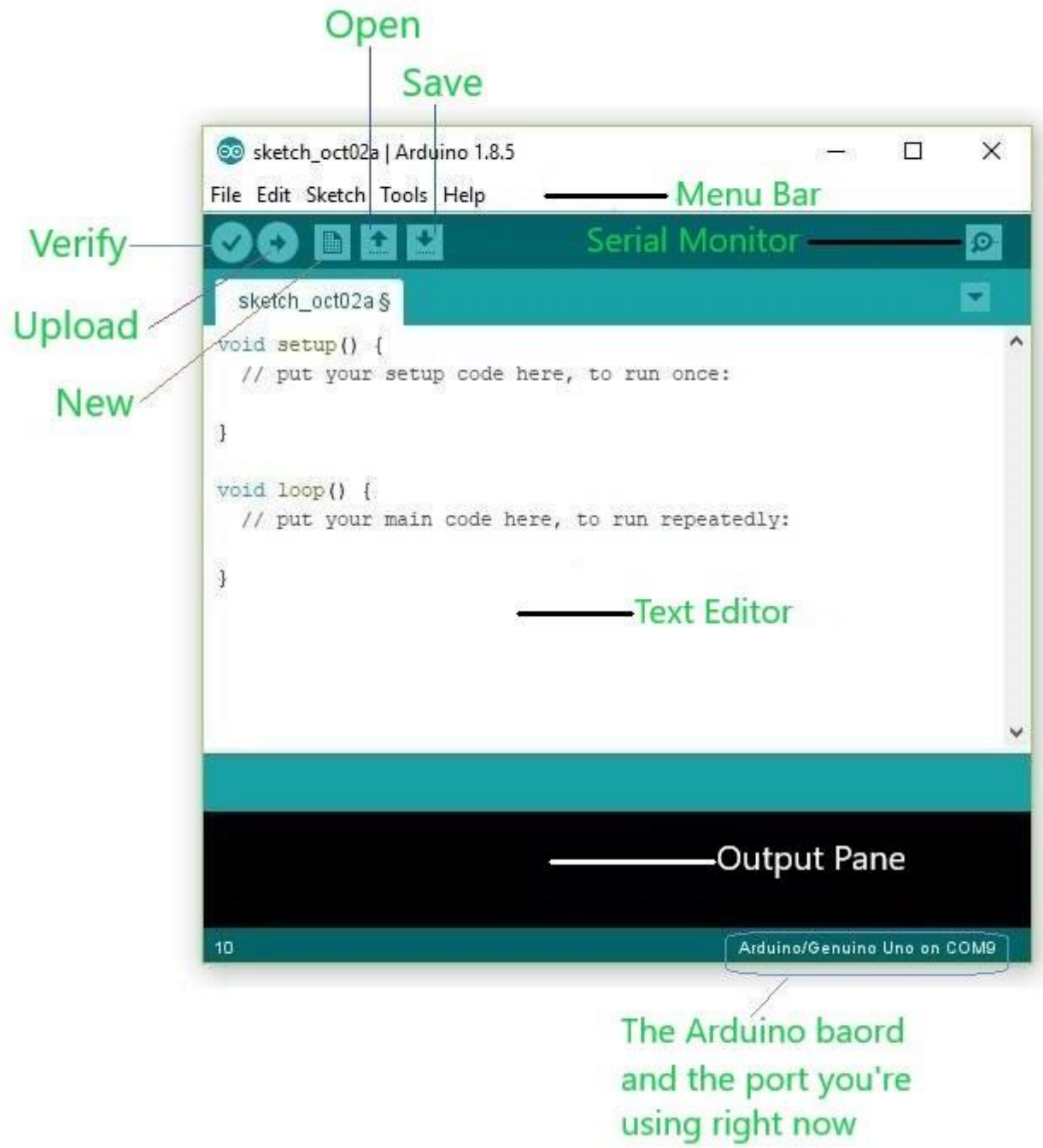


Using the above image as a reference, the labeled components of the board respectively are-

1. **USB:** can be used for both power and communication with the IDE
2. **Barrel Jack:** used for power supply
3. **Voltage Regulator:** regulates and stabilizes the input and output voltages
4. **Crystal Oscillator:** keeps track of time and regulates processor frequency
5. **Reset Pin:** can be used to reset the Arduino Uno
6. **3.3V pin:** can be used as a 3.3V output
7. **5V pin:** can be used as a 5V output
8. **GND pin:** can be used to ground the circuit
9. **Vin pin:** can be used to supply power to the board
10. **Analog pins(A0-A5):** can be used to read analog signals to the board
11. **Microcontroller(ATMega328):** the processing and logical unit of the board
12. **ICSP pin:** a programming header on the board also called SPI
13. **Power indicator LED:** indicates the power status of the board
14. **RX and TX LEDs:** receive(RX) and transmit(TX) LEDs, blink when sending or receiving serial data respectively
15. **Digital I/O pins:** 14 pins capable of reading and outputting digital signals; 6 of these pins are also capable of PWM
16. **AREF pins:** can be used to set an external reference voltage as the upper limit for the analog pins
17. **Reset button:** can be used to reset the board

Getting started with the Arduino IDE

Now that you're familiar with the hardware, its time to learn about the development environment using which you're going to program your Uno. The Arduino IDE is the best place to start your journey in programming your Uno. To get started, visit this page and download the latest build of the Arduino IDE for your Mac or PC. Go ahead and install the IDE on your PC or Mac and open it.



As you open the IDE, you'll be greeted by a window similar to the one shown in the above image. The text editor is where you'll be writing your code; you'll use the verify button to compile and debug the written program, the save button to save the program and the upload button to upload the program to the board. Before you click on the upload button, it is necessary to select your board, Uno in this case, from the tools menu in the Menu Bar. After you choose your appropriate board, make sure you specify the correct port on your PC or Mac that you've connected your Uno to, in the IDE.

Uploading your first program

In this example program, we'll be blinking the inbuilt L LED located right above the RX and TX LEDs. The Arduino IDE includes many basic programs to help you get started with your Uno. For this example, we'll be using the inbuilt 'Blink' program. To open this program, go to the Files menu in the Menu Bar; click on Examples; click on 01.Basics; select Blink. Now that you've opened the example program, its time to upload the program, to do this, click on the upload button and wait for the process to complete. If your Output Pane header turns amber and shows an error which reads "Serial Port COM'x' not found", you've not connected your board correctly or that you've not specified the correct port that your board is connected to in the IDE. When you advance and start writing your own programs, you might run into errors while compiling and uploading; this can be because of a syntax error in the program. After you've

corrected the errors and uploaded the program, you'll see that the inbuilt LED blinks, alternating between the ON and OFF state every second.

Congrats on uploading and executing your first piece of code on your Arduino Uno. You can now tinker with the program you just uploaded by changing the values of delay. This will change the pattern and the rate of blinking. Do keep in mind that the default unit of time in the Arduino IDE is milliseconds; also remember that you've to upload the program to the board after you've made changes in the values of delay to notice the changes in the rate and pattern of the blinking.

Moving Ahead

Now that you're familiar with the IDE and the hardware on the board, you can move up to programs that require external actuators and sensors using the inbuilt example programs as a reference. After you've gained some expertise with the board, you can move on to create projects that inculcate your innovative and innovative ideas. Soon in your journey through electronics, you'll realize that the Uno is not powerful enough or does not pack the features you require for your expert level programs, that is when you'll have to consider upgrading your board to something from the MKR line or to more powerful lines like the Yun.

Features of the Arduino UNO:

- Microcontroller: ATmega328.
- Operating Voltage: 5V.
- Input Voltage (recommended): 7-12V.
- Input Voltage (limits): 6-20V.
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6.
- DC Current per I/O Pin: 40 mA.
- DC Current for 3.3V Pin: 50 mA.

6.2 Components of Arduino Board

- Power (USB / Barrel Jack)
- Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)
- Reset Button
- Power LED Indicator
- TX RX LEDs
- Main IC
- Voltage Regulator.

6.3 Arduino IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - **contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus**. It connects to the Arduino hardware to upload programs and communicate with them.

6.4 Case Studies

Introduction:

Arduino is a single board microcontroller that can be of handy to develop multidisciplinary projects and it is an open-source electronics prototyping platform based on flexiblity, easy-to-use hardware and software. Arduino is a small computer that can be programmed with your instructions to interact with various forms of input and output devices. Arduino board allows you to create devices that can interact with world around you; it can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language and the Arduino development environment.

Hardware:

The hardware consists of an 8-bit Atmel AVR microcontroller or a 32-bit atmel ARM with complementary components to facilitate programming and incorporate into other circuits. An important aspect of the Arduino is the standard way that connectors are exposed, allowing the CPU board to be connected to a variety of interchangeable add-on modules known as shields. Some shields communicate with the Arduino board directly over various pins, but many shields are individually addressable via an I²C serial bus, allowing many shields to be stacked and used in parallel. Official Arduino's have used the mega AVR series of chips,

specifically the ATmega8, ATmega168, ATmega328, ATmega1280, and ATmega2560. A handful of other processors have been used by Arduino compatibles.

Most of boards include a 5 volt linear regulator and a 16 MHz crystal oscillator (or ceramic resonator in some variants), although some designs such as the LilyPad run at 8 MHz and dispense with the on board voltage regulator due to specific form-factor restrictions.

An Arduino microcontroller is also pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory, compared with other devices that typically need an external programmer. At a conceptual level, when using the Arduino software stack, all boards are programmed over RS-232 serial connection, but the way this is implemented varies by hardware version. Serial Arduino boards contain a simple inveter circuit to convert between RS-232-level and TTL-level signals. Current Arduino boards are programmed via USB, implemented using USB-to-serial adapter chips such as the FTDI FT232. Some variants, such as the Arduino Mini and the unofficial Board uno, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. (When used with traditional microcontroller tools instead of the Arduino IDE, standard AVR ISP programming is used.)

Software:
The Arduino IDE is a cross-platform application written in Java, and is derived from the IDE for the Processing programming language and Wiring project. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. There is typically no need to edit makefiles or run programs on a command-line interface. The Arduino IDE comes with a C/C++ library called "Wiring", which makes many common input/output operations much easier. Arduino programs are written in C/C++, although users only need to define two functions to make a run-able program:

- setup() – a function run once at the start of a program that can initialize settings.
- loop() – a function called repeatedly until the board powers off or reset.

The open source Arduino environment (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS, and Linux.

It is a feature of most Arduino boards that they have an LED and load resistor connected between pin 13 and ground, a convenient feature for many simple tests. The Arduino IDE uses the GNU tool chain and AVR libc to compile programs, and uses avr dude to upload programs to the board. As the Arduino platform uses Atmel microcontroller development environment, AVR Studio or the newer Atmel Studio, may also be used to develop software for the Arduino.

The Arduino boards can be built by hand or purchased. The Arduino hardware reference designs (CAD files) are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino Web site. Layout and production files for some versions of the Arduino hardware are also available. The software can be downloaded for free and the source code for the IDE and the on-board library are available and released under the GPLv2 license. In our case we are using Arduino Uno board.

Arduino Uno:
The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, 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 AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards and it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode. Revision 3 of the board has the following new features:

- 1.0 pin out: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that uses the AVR, which operate with 5V and with the Arduino Due, which operate with 3.3V. The second one is a not connected pin that is reserved for future purposes.
- Stronger RESET circuit.

- Atmega 16U2 replace the 8U2.

Characteristics:

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40mA
DC Current for 3.3V Pin	50mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by boot loader
EEPROM	1 KB (ATmega328)
SRAM	2 KB (ATmega328)
Clock Speed	16 MHz

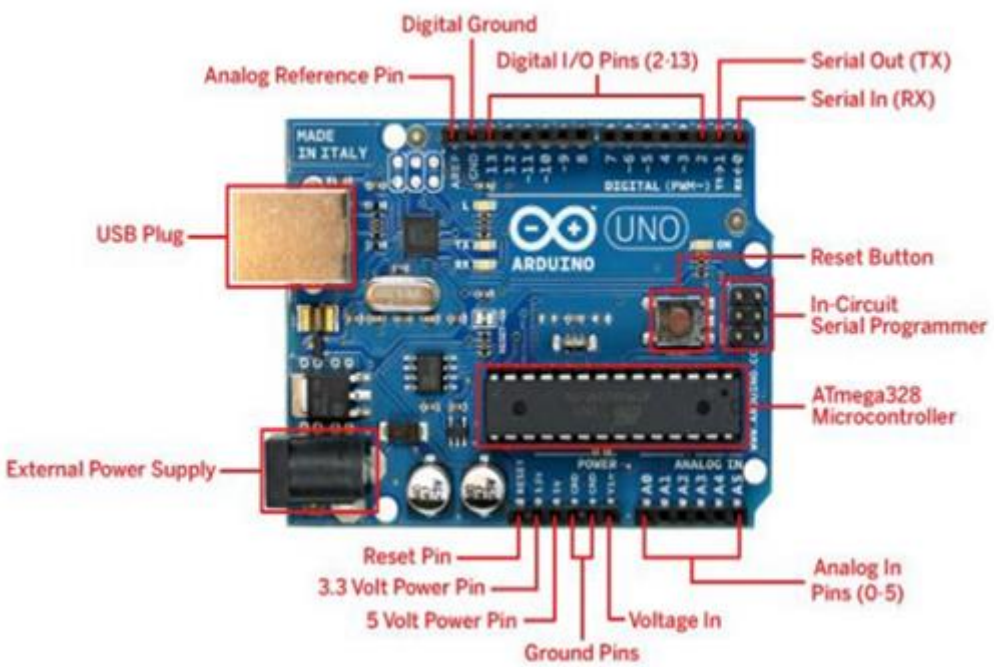


Figure1. Arduino Uno

Pin Description:
Power:
The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External power can come either from an AC-to-DC adapter or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from

the battery can be inserted in the GND and VIN pin headers of the power connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN -** The input voltage to the Arduino board when it is using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin or supplying voltage pin via the power jack.
- **5V -** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3.3v -** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND -** Ground pins.
- **IOREF -** This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.
- **Memory -** The ATmega328 has 32 KB (with 512Bytes used for the boot loader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).
- **Input and Output -** Each of the 14 digital pins on the board can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead()functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms.

In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** 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:** 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
- **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:
TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

There are a of other couple pins on the board:
AREF: Reference voltage for the analog inputs. Used with analogReference().
Reset: Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication:
The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 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 16U2 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. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but 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 ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. For SPI communication, use the SPI library.

Installing Arduino IDE for Windows:
The Arduino software (IDE) is available for windows, Linux and MAC operating systems. The installation process is different for all three platforms. In this lecture we are going work on how to install in windows PC, for other platforms they mentioned in the Arduino website how to install and use it. Get started by visiting the following website arduino.cc/en/Main/Software

Download the Arduino software for windows from the website as shown in the figure2, there is only one version of software for windows XP and windows7.

After downloading “arduino.1.6.0-windows.exe” file, by double clicking on the file will install the Arduino software and create a folder with name arduino in C folder.

Figure2: Arduino Software website

The arduino folder contains example programs and also drivers that allow the arduino to be connected to your computer by a USB cable. Before we launch the Arduino software we have to be install USB drivers (Sometimes it will automatically installs the required driver software through online, in that case no need to install the drivers manually as we going to see in next few steps).

Plug one end of USB cable to Laptop/Desktop PC and the other end to the Arduino board. The power light (LED) on the board will glow and you may get a new message in your PC showing “Found New Hardware”. Ignore this message and cancel any attempts that windows makes to try install drivers automatically for you and try to install USB drivers manually through Device manager. Select my computer and right click on that and then select “Manage” it will open computer management pop up window as shown in figure3.

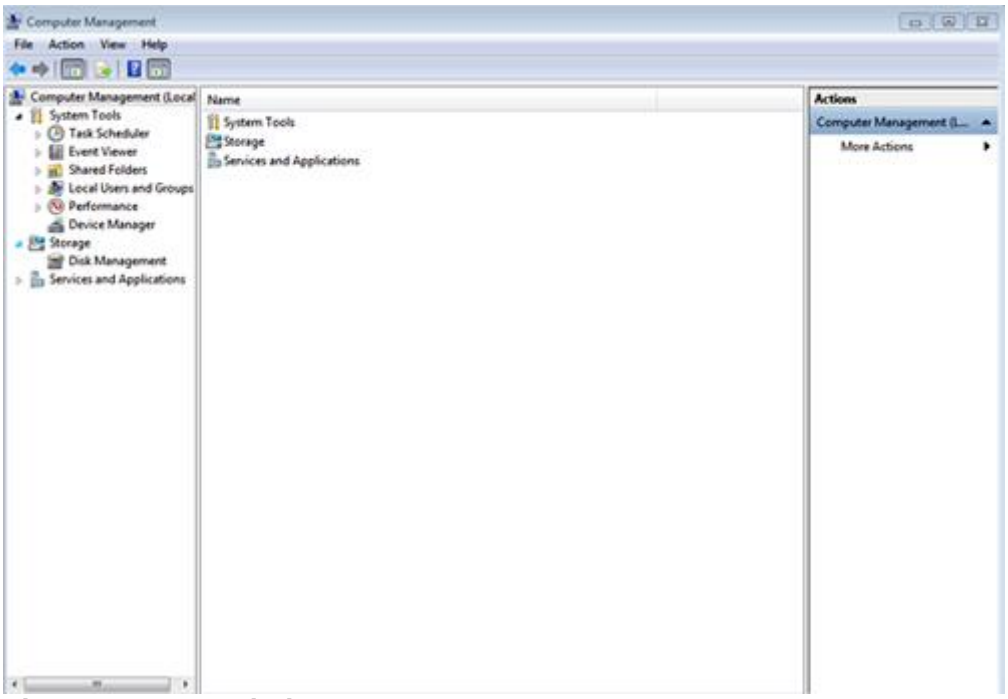


Figure3: Popup window

In this window select “Device Manager” in the list, under this select “Other Devices” shown in the list where you can see an icon “Unknown Device” with a yellow warning triangle on it as shown figure4.

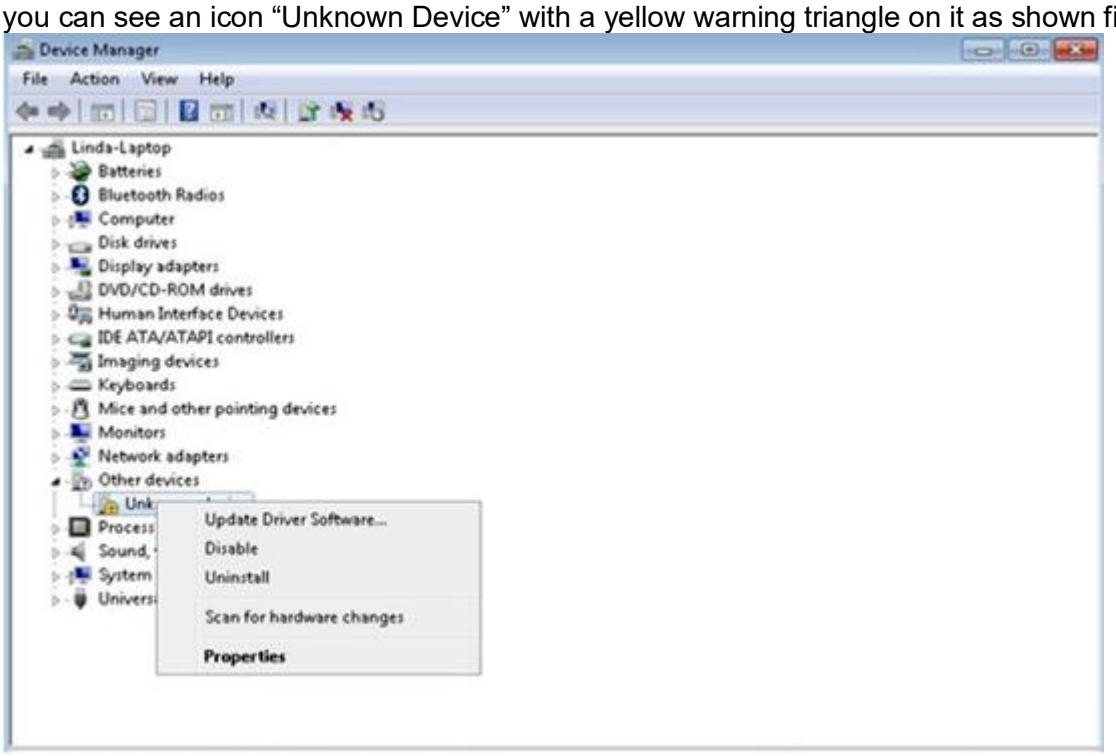
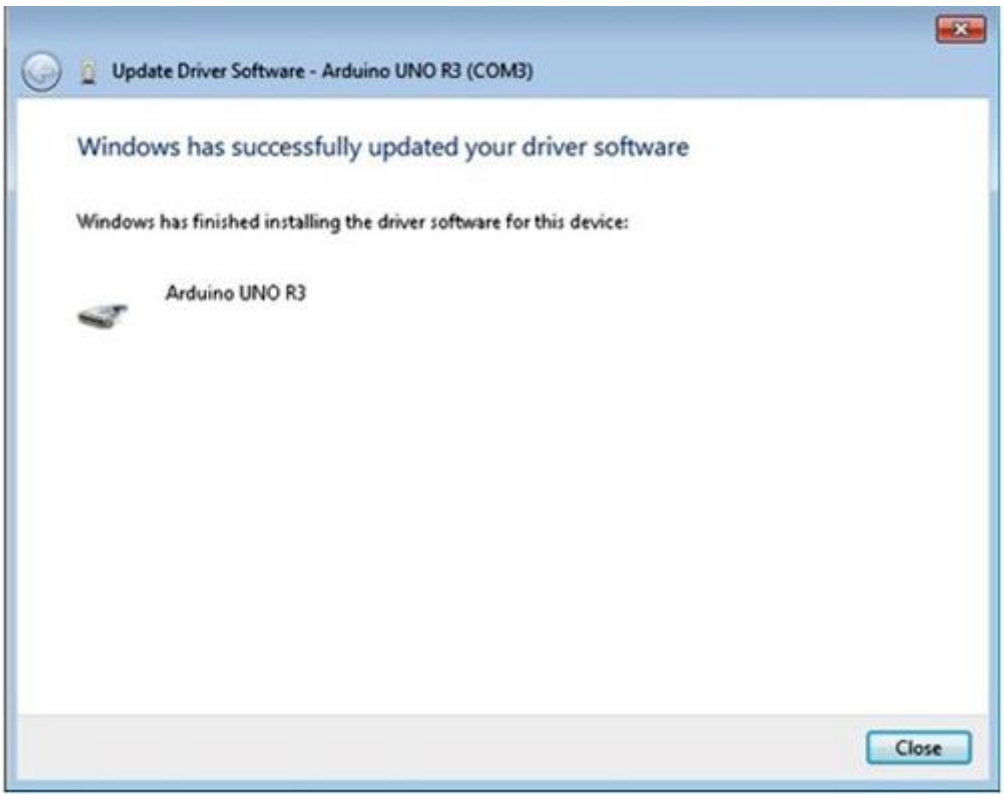


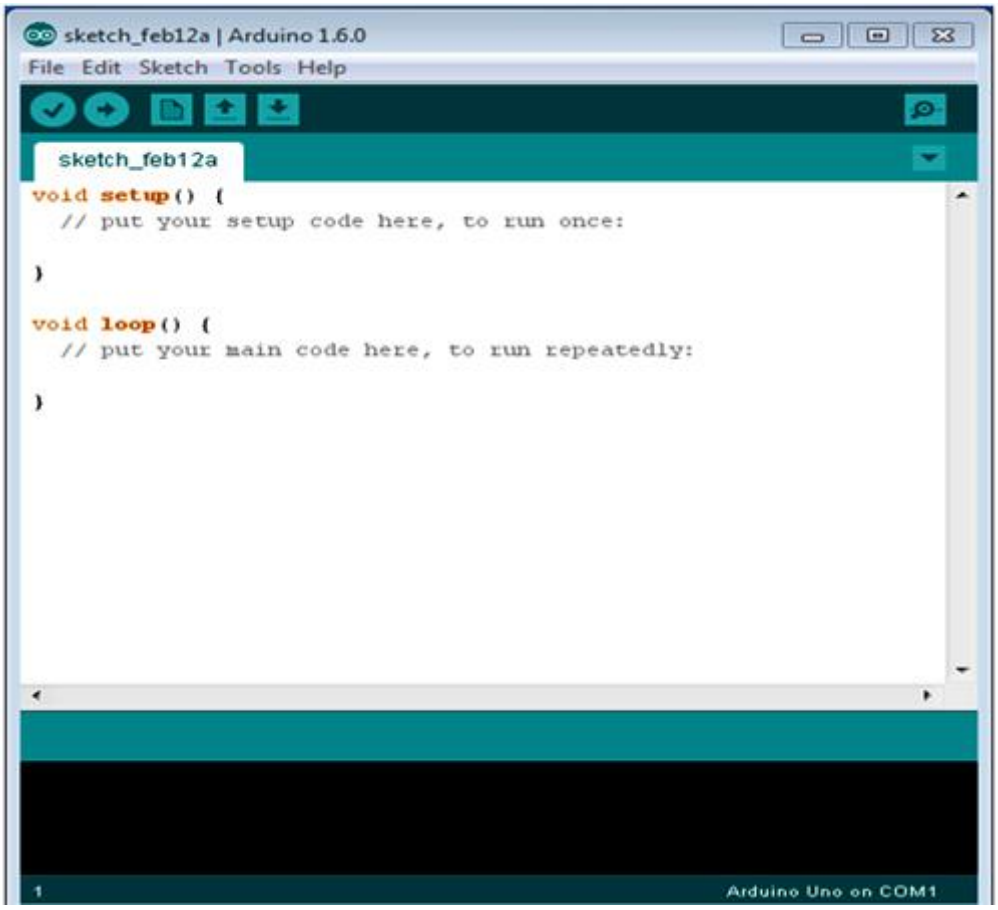
Figure4: Popup window

Right click on that and select the “Update Driver Software” then it will open up a pop up window shows to select either “Search Automatically for updated driver software” or “Browse for driver software on my computer”. Select the option to browse for driver software on my computer and navigate to the “C:\ProgramFiles\Arduino\drivers”.

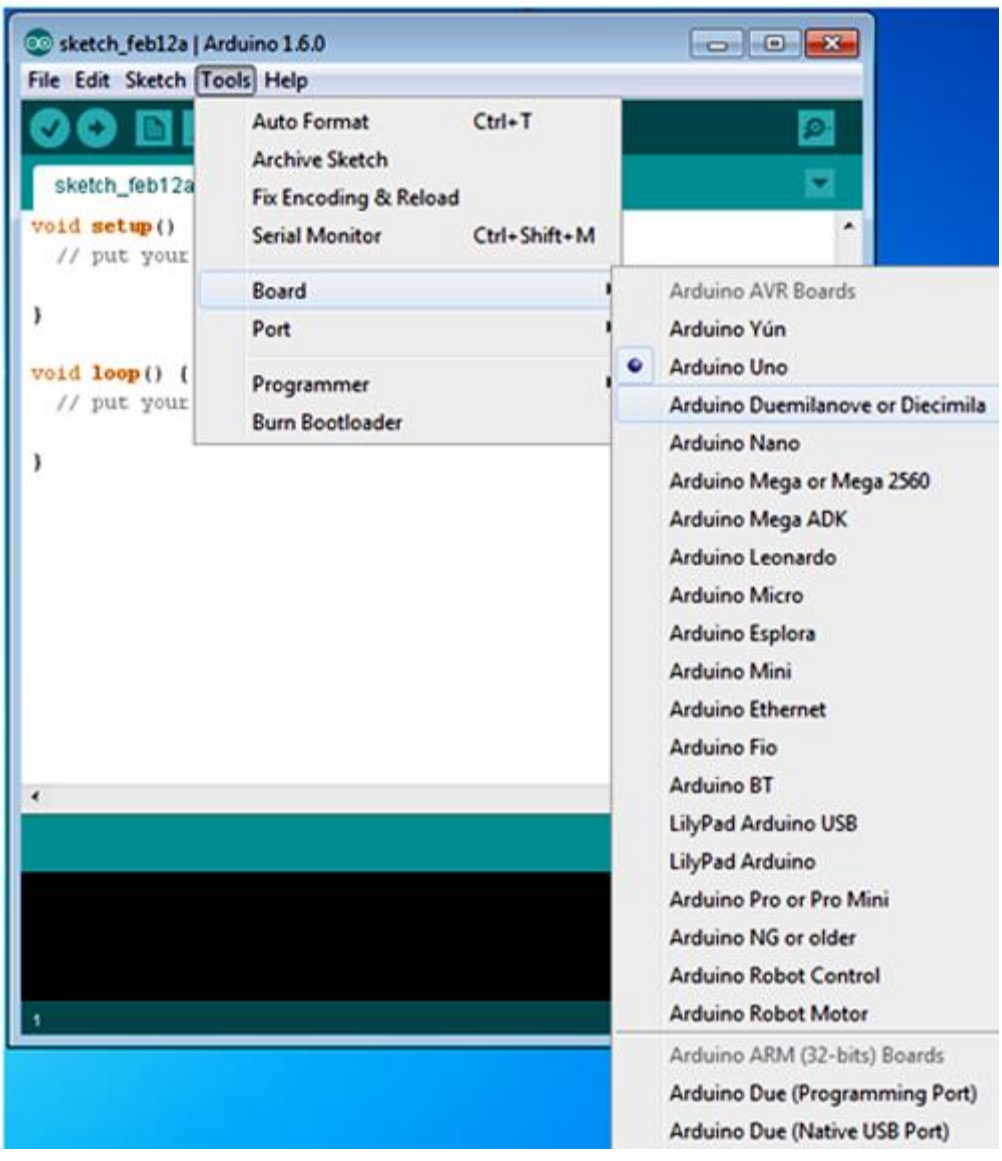
Click “Next” and if you get any security warnings, allow software to be installed. Once software installed successfully you will get a confirmation message that “Windows has successfully updated your driver software” and it will show the Arduino board name.



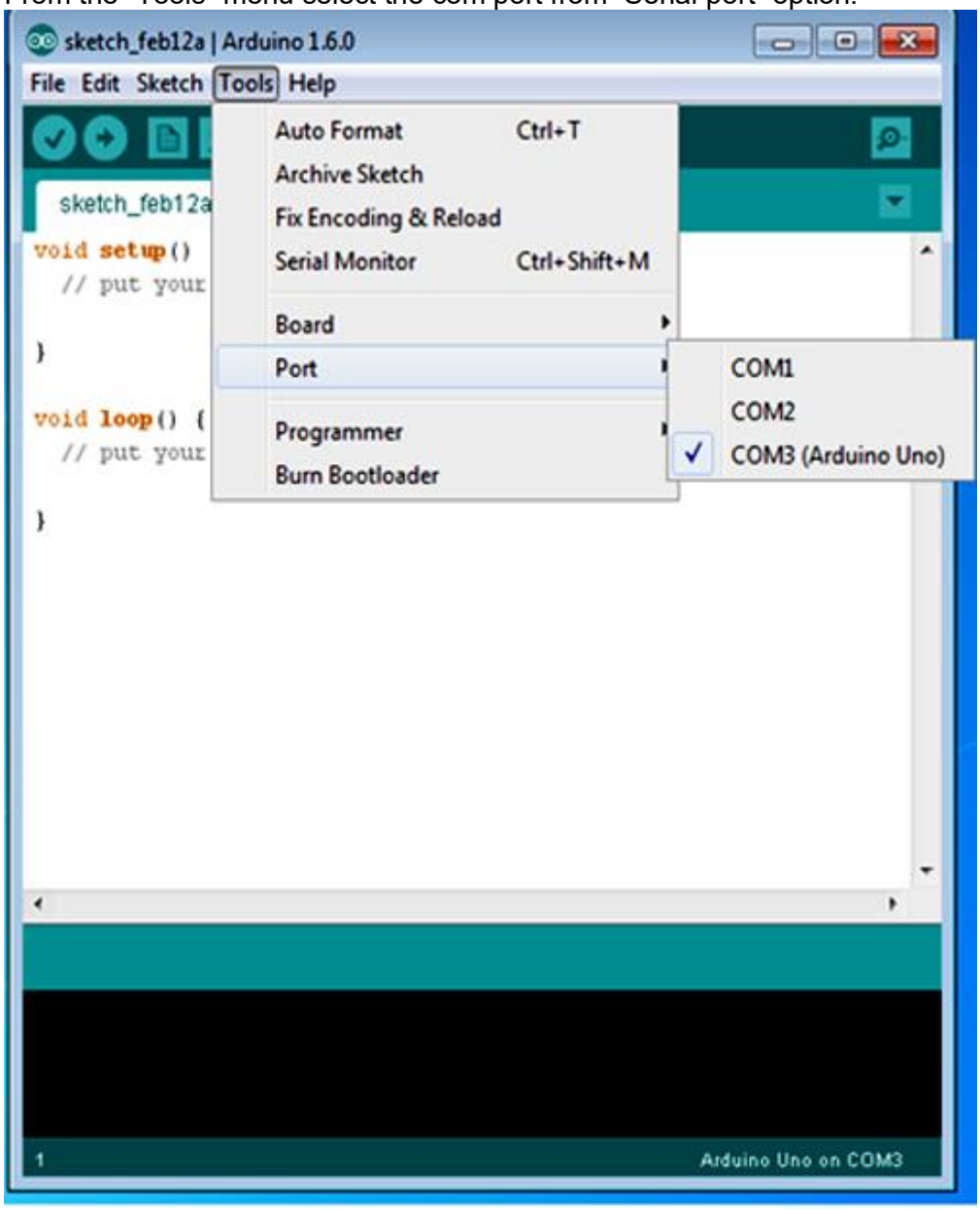
Now the Arduino software is ready and you can start by clicking “Arduino application”, this will start the Arduino IDE.



Before starting programming you have to tell the Arduino software which type of Arduino board you are using and the com port for which the Arduino board is connected.
From the “Tools” menu select the board that is “Arduino Uno”



From the “Tools” menu select the com port from “Serial port” option.



7. Programming with Raspberry Pi
Open IDLE by selecting the Raspberry Pi logo in the top-left, and click Programming > Python 3 (IDLE). You should be presented with the Python interactive interpreter. To write a program, go to File > New File. Enter in your code.

7.1 Architecture and Pin Configuration

A Raspberry Pi 3 board has 40 pins on it. Among these pins, we have four power pins on the Raspberry Pi, two of which are 5v pins and another two are 3.3v pins. The 5v power pins are connected directly to the Raspberry Pi's power input and we can use these pins to run low power applications.

Then there are the ground pins. There are eight ground pins and all of these are connected to each other; you can use any of these ground pins for your projects.

That leaves us with 28 GPIO pins, labeled starting from GPIO 0 and going up to GPIO 27.

The GPIO pins, as indicated by their full form, can be programmed to be output pins or input pins. So we can set values of output pins and we can even read values of input pins. The GPIO pins can be digitally programmed so that they can be turned ON or OFF. The output of any GPIO pin is 3.3v and can be used to control output components like an LED or a motor. These ON/OFF conditions can also be interpreted as a Boolean True/False, 1/0 or HIGH/LOW.

These are the common types of pins on a Raspberry Pi 3 board. Some of these pins also have a dual function. For example, pin 3 or GPIO 2 also acts like an I2C pin.

7.2 Case studies

What is a Raspberry pi?

It is a computer which only has one board. It was developed by the Raspberry Pi Foundation, which has its home base in the United Kingdom of Great Britain. As with any organization with the name foundation, the Raspberry Pi foundation is also in the market to not make a profit.

It was formed with the sole purpose of providing young and passionate struggling to make ends meet with an affordable computing solution. Through this new "solution," they would be able to learn the fabled art of programming. Since then, Raspberry Pi has only gained its fan following.

The most notable community which uses Raspberry Pi is the DIY community because of its small footprint, capacity to run an entire Linux distro, and the fact that it contains GPIO (which is often known as General Purpose Input Output pins.)

There are many ways in which Raspberry pi ships. You always find one specifically designed for your use case. But for this raspberry pi python projects, we are using the Raspberry pi 4. There are many components to a Raspberry Pi 4 system. Let us discuss them one by one below:

1. GPIO or General Purpose Input and Output Pins: The pins which are GPIO in nature are used to connect the board of the Raspberry pi to other electronical components.
2. Ethernet Port: You might not expect this to be present in such a small board but a distribution of Raspberry Pi has an ethernet port baked into it. You can use this port to hook your board to the internet using the wired internet technology (also known as LAN). You will find that your board also has WiFi and Bluetooth technologies also present inside. In case you are not able to furnish a wired internet connection you can always use WiFi.
3. There are two USB 3.0 ports and two USB 2.0 port: There are a grand total of about four USB ports in your credit card size computing device. Two of them are the high-speed USB 3.0 ports which are based on the new USB technology. To be able to make use of it you would be needing a USB 3.0 (blue color USB) compatible device. There are two USB 2.0 also made available. You can use them for your external mouse and keyboards.
4. AV jack: AV or Audio Visual Jack can be used to plug in your audio devices like speakers or headphones. Yes, you can connect an external speaker unit to your Raspberry pi.
5. Port to houses a Camera Module: This port is used to connect the official camera module for raspberry pi.
6. HDMI Port: Yes, you also get an option to hook up a full-fledged monitor to your raspberry pi. Since there happen to be two micro sized HDMI ports you can use a maximum of two monitors with your raspberry Pi.
7. Power port: There is one USB power port that is needed to power your raspberry pi. Raspberry Pi version four and above would be using the newer USB Type C tech while the older models use the outdated micro USB for charging.
8. A port used for External Displays: You can even attach a touch based input system on your Raspberry Pi. Plug in the official seven inches touch panel for the raspberry pi into this port to get the job done.
9. Micro SD card slot: Yes, you can even plug in a micro SD card into the slot which you can find on the underside of your raspberry pi board.

Also Read: Raspberry Commands

Raspberry Pi Projects

1. Creating a Media Center

With pandemic shutting down the conventional means of entertainment, everyone is shifting towards their trusted TV for their fill. You can create a media controller for your TV with the help of your raspberry pi, all the while staying shut indoors to help flatten the curve.

To set this up, you would need your Raspberry PI unit. Ensure that the one you have comes in loaded with a GPU (to render things on the big screen.) You can then use Kodi (formerly known as XMBC), which should help you playback what you desire on your television. You even would be able to play YouTube if you install the plugin.

You can design the look and feel of your media center yourself if you have the time and technical knowledge, or you can easily switch up to an open-source version that has a base of Kodi. Before you begin laying the foundations of this project, you should decide on the raspberry pi model you would want to use because certain features only would work on the newer boards.

2. Creating a close circuit television

If you happen to have a pet or a small child or you would like a home security system, then you will find this project useful. You can set up a system through which you would be able to take photos, capture videos, and even stream real-time footage, basically create a closed-circuit television (or CCTV) with your raspberry pi board.

You would need the raspberry pi camera module, mainly because it is straightforward to use and change according to our will and need. However, the best choice considering that we would have to do monitoring of

a place would be an infrared camera. The best part about using the infrared camera with the raspberry pi is that the infrared LED is programmable.
So, with the help of raspberry pi, you would be able to adjust the brightness of the takes and even see in the dark. After the acquisition of all the needed hardware, there are two ways in which you could move ahead with the project. We have listed all of them below:
Taking still images regularly
In case you want to have a general awareness of the surroundings or the situation does not demand strict real-time footage, you could set up your CCTV camera so that it would take still images after a predefined interval.
Setting this up in python should not be an arduous task and would be done with a relatively short script. You can even switch out python entirely for a more CLI (command-line interface) based approach using rapistil, and for scheduling the process, use Cron. If you would like to save all your stills, you would also need to attach dropbox or other online storage solutions.
Recording Video
If you think that recording a video of the happenings is the way to go, then with the help of MJPEG, you could do that in a heartbeat. You would also be able to stream it in real-time. To see your stream, you must visit pi's IP address on port 8000 (You can configure the ports through code.)
Also, if you want to take things up a notch, you could also use the pistreaming module. You should see a significant jump in performance as well as the complexity post switch. If you know your way around your raspberry pi, you should not face any significant issues while getting either to work. However, in the latter, you would have to write more code as you would have to open two ports in order to get it to work.
3. Ad blocker
With the help of Adafruit's onion PI, you would be able to create a VPN mask to abstract your IP address, thus allowing you to be truly anonymous while browsing the internet. So, if you were to route your network through your raspberry pi first, you would be able to effortlessly create this mask.

You also have the power to block any and every piece of advertisement. However, you would need to install Pi-hole software into your pi to be able to create an ad blocker.
Must Read: Raspberry Pi IoT Project Ideas
Conclusion
We hope that among these raspberry pi python projects and raspberry pi python project ideas, you could find something you like. Raspberry pi is a powerful board. We have only managed to scratch the surface of what raspberry pi enables its users to do.
If you feel you lack the needed python knowledge to complete these projects, we would suggest you take a look at the free python course we offer. For a more comprehensive deep dive, you can see our various diploma courses. With that being said, we will you all the best for your raspberry pi journey.
Also, If you're interested to learn more about Machine learning, check out IIIT-B & upGrad's Executive PG Programme in Machine Learning & AI which is designed for working professionals and offers 450+ hours of rigorous training, 30+ case studies & assignments, IIIT-B Alumni status, 5+ practical hands-on capstone projects & job assistance with top firms.

7.3 Implementation of IoT with Raspberry Pi
The Raspberry Pi is a series of low-cost, programmable computers that include a set of GPIO, or 'General Purpose Input Output', pins that can be used to connect and control external electronic devices, and to create Internet of Things (IoT) solutions.

The Raspberry Pi is a series of low-cost, programmable computers that include a set of GPIO, or 'General Purpose Input Output', pins that can be used to connect and control external electronic devices, and to create Internet of Things (IoT) solutions.

The exact number and role of the pins changes between individual models but is generally divided into power, ground, and general-purpose pins. The power and ground pins are not programmable. The power pin supplies a constant 3.3V/5V power to the circuit while the ground pin is used to connect to the cathode of the circuit.

The general-purpose pins are fully programmable and can be used in either an output or input mode. When set to output mode, the pins provide a constant 3.3V power that can be turned on or off. When set to input mode, the pin reads the current supplied by the circuit and returns a Boolean value indicating if it receives 3.3V power or not.
Of course, these capabilities aren't new and have been widely available to developers through microcontrollers such as Arduino or NodeMCU. However, these devices generally came with limited memory and computing power and required the use of particular programming languages.

The Raspberry Pi, on the other hand, supports a more robust CPU that is capable of running Linux and supports NodeJS, allowing JavaScript developers to use their existing skillset and build sophisticated devices with relative ease.
To interact with the GPIO pins, we use a NodeJS module called onoff that provides simple access to the individual pins.

The equivalent of a 'hello world' demo in the world of microcontrollers is a blinking led. Most of the code in the example below should already be familiar to JavaScript developers.

```
const Gpio = require('../onoff').Gpio;
const led = new Gpio(17, 'out');
```

After requiring the module, we define the roles of the pins we wish to interact with. The number identifies the pin on the board, followed by determining if the pin is used to read ('in') or write ('out').

In this example, we defined a pin called led, assigned it to physical pin 17 and set it to write mode ('out')

```
const blinkInterval = setInterval(blinkLED, 500);

function blinkLED() {
  if (led.readSync() === 0) {
    led.writeSync(1);
  } else {
    led.writeSync(0);
  }
}
```

Now all that's left is to make our led blink by creating a 500ms interval and turning the led on/off based on its current state.

```
setTimeout(() => {
  clearInterval(blinkInterval);
  led.writeSync(0);
  led.unexport();
}, 5000);
```

Assuming we do not wish to continue blinking the LED indefinitely, we would also need to clean up at the end. In this example, we will wait 5 seconds before clearing our interval, turning off the led, and releasing its resource.
Of course, the Raspberry Pi can do more than just blinking LEDs. Developers have built anything from drones to Raspberry Pi skateboards. While not specifically written for JavaScript, you can find many ideas for exciting projects of all levels at the Raspberry Pi website

8. Software defined Networking

Software-Defined Networking (SDN) is an approach to networking that uses software-based controllers or application programming interfaces (APIs) to communicate with underlying hardware infrastructure and

direct traffic on a network.This model differs from that of traditional networks, which use dedicated hardware devices (i.e., routers and switches) to control network traffic. SDN can create and control a virtual network – or control a traditional hardware – via software.

8.1 Limitation of current network

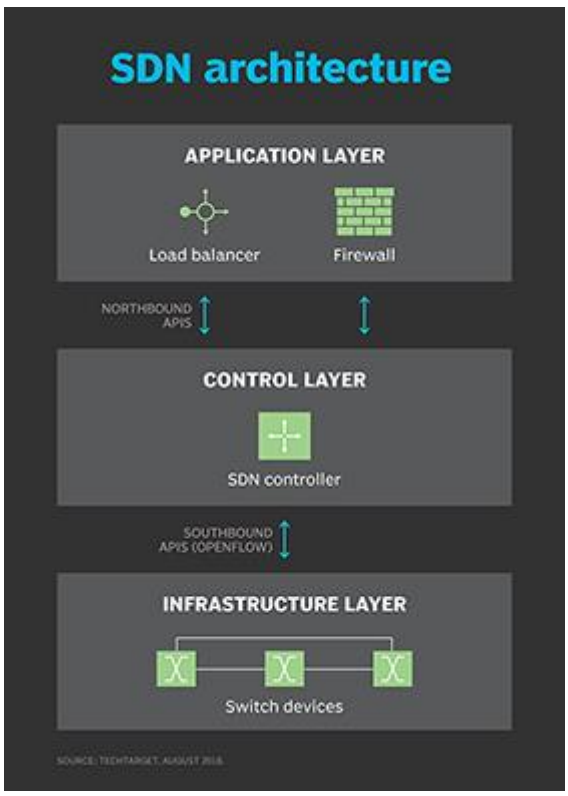
- 1). Security issues. Data & information held on a network is open to many people across the world, and can easily be accessed illegally
- 2). High initial cost
- 3). Moral and cultural effects
- 4). Spread of terrorism and drug trafficking
- 5). Over-reliance on networks.

8.2 Origin of SDN

The history of SDN principles can be traced back to **the separation of the control and data plane first used in the public switched telephone network as a way to simplify provisioning and management** well before this architecture began to be used in data networks.

8.3 SDN Architecture

Software-defined networking (SDN) is **an architecture that abstracts different, distinguishable layers of a network to make networks agile and flexible**. The goal of SDN is to improve network control by enabling enterprises and service providers to respond quickly to changing business requirements.



8.4 Rule Placement, Open flow Protocol

Software-Defined Networking (SDN) abstracts low- level network functionalities to simplify network management and reduce costs. The OpenFlow protocol implements the SDN concept by abstracting network communications as flows to be processed by network elements. In OpenFlow, the high-level policies are translated into network primitives called rules that are distributed over the network. While the abstraction offered by OpenFlow allows to potentially implement any policy, it raises the new question of how to define the rules and where to place them in the network while respecting all technical and administrative requirements. In this paper, we propose a comprehensive study of the so-called OpenFlow

rules placement problem with a survey of the various proposals intending to solve it. Our study is multi-fold. First, we define the problem and its challenges. Second, we overview the large number of solutions proposed, with a clear distinction between solutions focusing on memory management and those proposing to reduce signaling traffic to ensure scalability. Finally, we discuss potential research directions around the OpenFlow rules placement problem.

8.5 Controller placement

In a distributed SDN controller architecture, multiple controllers are deployed to minimize communication latency between the switch and the control plane. **The controllers are strategically placed to optimize network performance**, the so-called controller placement problem (CPP

8.6 Security in SDN

1. Network segmentation

One of the tenants – and intrinsic benefits – of software-defined network security is easier network segmentation. Network segmentation involves creating subnetworks inside of a larger network. Segmentation can help **compartmentalize and organize** your organization's network traffic. For instance, it may restrict your sales department's machines (physical or virtual) from communicating with your financial team's machines.

This allows for more efficient bandwidth use by reducing the size of broadcast domains and reducing unnecessary traffic on the network. From a security perspective, it helps reduce an organization's attack surface and thus restricts the area of data security breaches. Therefore, when one machine or application is infected, segmentation blocks it from affecting separate devices and applications.

2. Easier centralized remote management

Virtualized software-based network security is easier to manage from a single centralized dashboard. This means network and security administrators can access it and view it remotely, so if there is a breach, the relevant parties can be notified instantly.

The Covid-19 pandemic saw companies migrating their workforces offsite. Network **security for remote and hybrid work environments** must be more flexible. With software-defined security, your organization's network security experts can track the security of all employees, no matter where they are. You can ensure that network security is consistent for onsite and offsite employees.

3. Automation

Virtualizing network functions such as firewalls facilitates a greater potential for automation. A good example of this is firewall architecture. Current firewall architectures do not scale well, and this may interfere with your business's agility. Virtual network firewalls allow you to benefit from the same features as physical firewalls, but they add more agility, flexibility, and scalability.

Traditionally, to deploy and virtualize your network, you were required to script it manually. Today, companies can implement a turnkey solution to automate network firewall virtualization.

Furthermore, because functions like these are virtualized, they can be updated automatically – from their licensing to their policies. This makes it easier to keep up with the latest security trends. If you **plan to migrate your databases** or core network infrastructure, a virtualized infrastructure can offer a smoother transition,

4. Scalability

A huge advantage of the virtualized and software-defined network is scalability. It's far easier to **scale virtualized processes** and network components because they don't require the purchase of new hardware. You don't have to add more RAM and processing power to machines or buy new ones – especially if your virtual functions are running on a cloud server.

Most cloud vendors offer automated scaling. If your security requires more system resources, your vendor can provision new instances or services to it. And as your company continues to expand along with its network, its security requirements will also change. Security tools such as virtual firewalls can be deployed nearly at will, which allows for seamless growth in your operations.

5. Smaller physical footprint

Now that the physical network infrastructure doesn't handle your security, it leaves a smaller physical footprint. Software-defined security is hosted on virtual machines. Multiple instances can be run from a single server, which may be located on the cloud. Virtualized functions can be scaled up or down depending

on your company's requirements at any given time which allows you to cut costs on infrastructure and service fees.

8.7 Integrating SDN in IoT
IoT architecture with SDN: a high-level view The SDN-IoT integration brings several significant benefits for IoT traffic: (1) **Intelligent traffic routing and better network resources use**. (2) Simplified information acquisition facilitating information analysis, decision making and network configuration actions.

9. Smart Homes
Overview :

- Home automation is constructing automation for a domestic, mentioned as a sensible home or smart house. In the IoT home automation ecosystem, you can control your devices like light, fan, TV, etc.
- A domestic automation system can monitor and/or manage home attributes adore lighting, climate, enjoyment systems, and appliances. It is very helpful to control your home devices.
- It's going to in addition incorporates domestic security such as access management and alarm systems. Once it coupled with the internet, domestic gadgets are a very important constituent of the Internet of Things.
- A domestic automation system usually connects controlled devices to a central hub or gateway.
- The program for control of the system makes use of both wall-mounted terminals, tablet or desktop computers, a smartphone application, or an online interface that may even be approachable off-site through the Internet.

Smart Home Components :
Here, you will see the smart home components like smart lighting, smart appliances, intrusion detection, smoke/gas detector, etc. So, let's discuss it.
Component-1 :
Smart Lighting –

- Smart lighting for home helps in saving energy by adapting the life to the ambient condition and switching on/off or dimming the light when needed.
- Smart lighting solutions for homes achieve energy saving by sensing the human movements and their environments and controlling the lights accordingly.

Component-2 :
Smart Appliances –

- Smart appliances with the management are here and also provide status information to the users remotely.
- Smart washer/dryer can be controlled remotely and notify when the washing and drying are complete.
- Smart refrigerators can keep track of the item store and send updates to the users when an item is low on stock.

Component-3 :
Intrusion Detection –

- Home intrusion detection systems use security cameras and sensors to detect intrusion and raise alerts.
- Alert can we inform of an SMS or an email sent to the user.
- Advanced systems can even send detailed alerts such as an image shoot or short video clips.

Component-4 :
Smoke/gas detectors –

- Smoke detectors are installed in homes and buildings to detect smoke that is typically an early sign of Fire.
- It uses optical detection, ionization for Air sampling techniques to detect smoke.
- Gas detectors can detect the presence of harmful gases such as CO, LPG, etc.
- It can raise alerts in the human voice describing where the problem is.

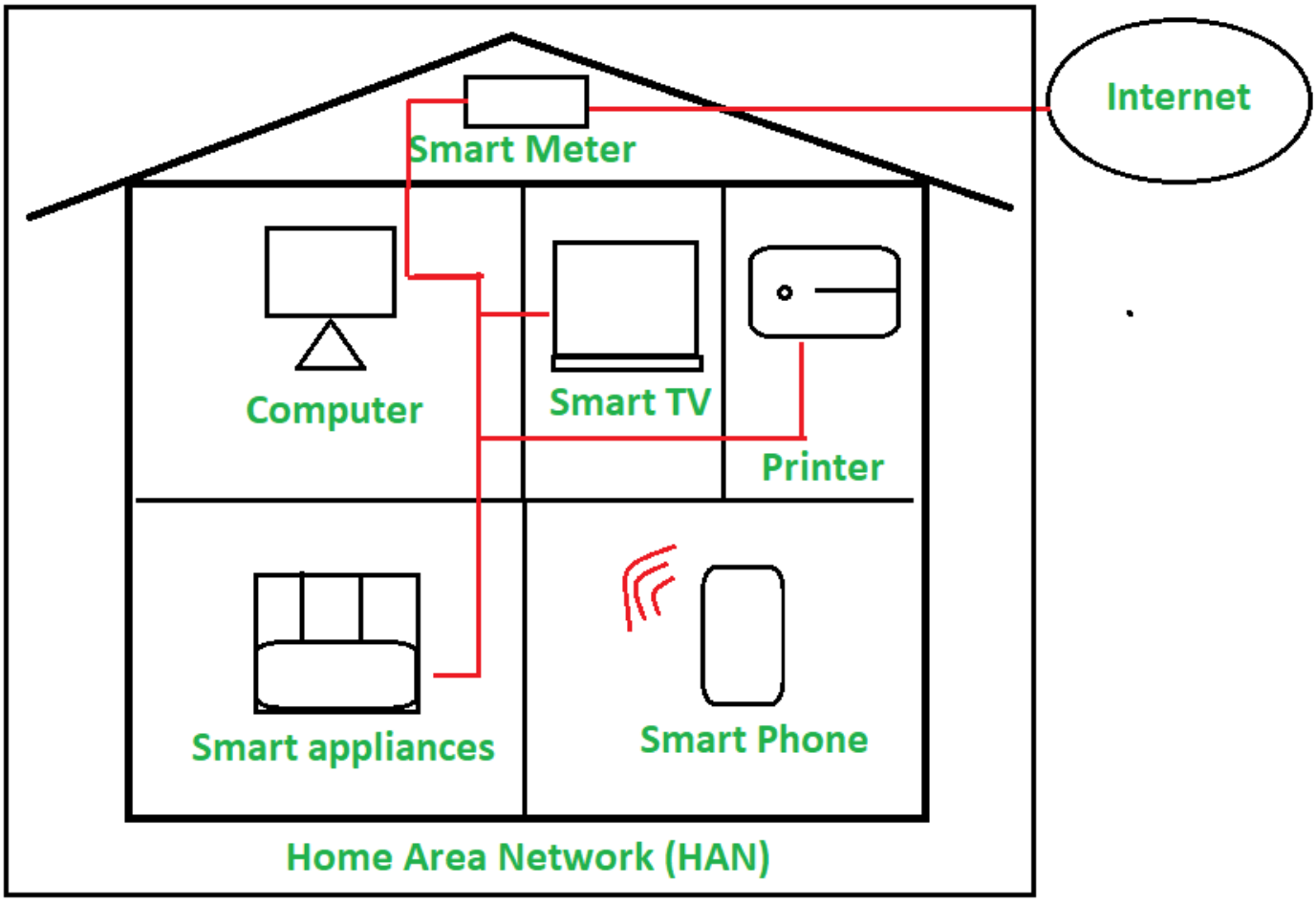
A **smart home system** can be something that makes our life quite easy. Starting from energy management where the power controls system in the AC appliances where we use the thermostat, all this is managed to cut down the power consumption that's taking place. A door management system, security management system, water management system are the part of this as well. Still, these are vital things that stand out in the smart home system. The limitation of IoT in smart home application stops where our imagination stops. Anything that we wish to automate or want to make our life easier can be a part of smart home, a smartphone system as well.

9.1 Origin and example of Smart Home Technologies
Examples of smart home hubs include **Amazon Echo, Google Home, Insteon Hub Pro, Samsung SmartThings and Wink Hub**. Some smart home systems can be created from scratch, for example, using a Raspberry Pi or other prototyping board.

9.2 Smart Home Implementation
IoT home automation is **the ability to control domestic appliances by electronically controlled, internet-connected systems**. It may include setting complex heating and lighting systems in advance and setting alarms and home security controls, all connected by a central hub and remote-controlled by a mobile app.

9.3 Home Area Networks(HAN)

Home Area Network (HAN) is a network in a user's home where all the laptops, computers, smartphones, and other smart appliances and digital devices are connected into a network. This facilitates communication among the digital devices within a home which are connected to the Home network. Home Area Network may be wired or wireless. Mostly wireless network is used for HAN. One centralized device is there for the function of Network Address Translation (NAT). This Home Area Network enables communication and sharing of resources between the smart devices over a network connection.
Example – Think about a home where computers, printers, game systems and tablets, smartphones, other smart appliances are connected to each other through wired or wireless over a network is an example of Home Area Network.



Infrastructure of HAN :

- A modem is used which is provided by an ISP to expose Ethernet to WAN. In homes they comes in DSL modem or cable modem.
- A router is used to manage connection between Home Area Network (HAN) and Wide Area Network (WAN).
- A wireless access point is used for connecting wireless digital devices to the network.
- Smart Devices/ Digital Devices are used to connect to the Home Area Network.

Devices connected in HAN –

- Laptop and Computers
- Smart Phones
- Network Printers
- Network Attached Storage (NAS) devices
- Security Alarms
- Smart TV & Bulbs
- Smart speakers
- Garage door and gate openers
- Media Players or Streaming Devices etc

Advantages of Home Area Network (HAN) :

1. **Accessibility –** Home area Network gives better accessibility for the devices in the network for accessing internet connection.
2. **Resources sharing –** Resources on the network can be shared over the network. For example, if you want to share a video file from your computer to smart television that's very using HAN.
3. **Security –** Home Area Network provides better security as it is enabled with security software, passwords etc which protects it from unauthorized access.
4. **Management –** All the devices/appliances connected to the home network are easy to manage and can be controlled the user's little effort.
5. **Maintenance –** Once a home network is set up, it does not require frequent maintenance with a little care and monitoring it works well.
6. **Multiuser –** Home Area Network allows to work multiple users in that home without any problem. All members can work simultaneously as per their requirements.

7. **Comfort Life –** This network connects all the devices to a single network, and with the addition of IoT technology, everything becomes automatic where it provides a better comfort lifestyle to the human being.

Disadvantages of Home Area Network (HAN) :

1. **Expensive –** Set up of HAN is a little bit expensive because it requires smart devices and appliances to work in the network. For example, it requires Laptops, Smart Television, Smart Washing machines, smartphones, etc.
2. **Slow Connectivity –** When all the users of the home use shared Home Area Network, they may face slow internet speed. For example, when anyone is downloading a high volume file by taking a high amount of internet during that others may feel slow down in internet speed.
3. **High Security –** It requires high security otherwise if an attacker targets any device and enters the Home network then they may steal important data from the laptop also as all the devices are connected to each other and work on a shared network.

9.4 Smart Home benefits and issues

Smart homes allow you to have **greater control of your energy use**, all while automating things like adjusting temperature, turning on and off lights, opening and closing window treatments, and adjusting irrigation based on the weather.

- Security Issues: As with all computing devices, security will become a greater issue as more people use smart home devices.
- Cost: Extremely expensive.
- Greater acceptance.

10. Smart Cities

he smart city is an evolution of a smart home. Here, it is not just the sensors of a single home that is connected, here its correlation or a network or a connection between various organizations, various domains as well as multiple segments of that city as a whole. In the smart city, the life of every single dependent becomes more comfortable and in tune really help to develop that city to greater extends as such. Now, the key factor for a smart city is government support as well, and if the governments are willing to take this step, then we hope we would see a smart city completely build on the Internet of Things.

10.1 Characteristics of Smart Cities

A city's smartness is determined using a set of characteristics, including:

- An infrastructure based around technology.

- Environmental initiatives.
- Effective and highly functional public transportation.
- Confident and progressive city plans.
- People able to live and work within the city, using its resources.

10.2 Smart city Frameworks

IoT-Enabled Smart City Framework. Two barriers currently exist to effective and powerful smart city solutions. First, **many current smart city ICT deployments are based on custom systems that are not interoperable, portable across cities, extensible, or cost-effective**.

A Smart City Framework will enable cities to establish a standard “catalog” system for recording, measuring, and collating city data, and for making it easily accessible for efficient, effective implementation and management of Smart City solutions for economic, social, and environmental gain.

10.3 Challenges in Smart cities

- 1: Overcoming city vendor lock-in
- 2: Overcoming developer city lock-in
- 3: Meeting real citizen needs
- 4: Sharing IoT infrastructure for new business cases
- 5: Quantifying social and economic benefits
- 6: Sharing more than just open data.

10.4 Data Fusion

Multi sensor data fusion technology is to use multiple sensors to collect data from the same target, analyze and synthesize the collected data using computer technology, and form data with high accuracy and low redundancy to support the decision-making process.

Data fusion is **the joint analysis of multiple inter-related datasets that provide complementary views of the same phenomenon**. The process of correlating and fusing information from multiple sources generally allows more accurate inferences than those that the analysis of a single dataset can yield.

10.5 Smart Parking

Smart Parking is a parking strategy that combines technology and human innovation in an effort to use as few resources as possible—such as fuel, time and space—to achieve faster, easier and denser parking of vehicles for the majority of time they remain idle.

An IoT based smart parking system, also known as a connected parking system, is a centralized management system that allows drivers to use a smartphone app to search for and reserve a parking spot.

The system’s hardware features sensors that detect available parking slots and communicate this information to all drivers in the area. This data is updated in real-time, which means drivers never have to worry about not finding an available space.

In addition to helping drivers find a spot, the system also sends alerts about peak times and peak prices. The goal of these alerts is to help save drivers money while also reducing congestion.

How Does Iot Based Parking System Work?

Parking systems are installed on the outside of buildings or inside of buildings. When a vehicle enters the space, sensors detect its presence and calculate available parking slots. This information is then sent to the driver’s phone via an app.

The smart parking system also has real-time data on occupancy rates, which can be found on the app. This data is collected from each sensor and is updated every five minutes.

One major drawback of automated parking system is that it has increased competition for parking spaces in urban areas with limited space nearby. However, even though these systems are helpful for drivers, there are some drawbacks to this type of initiative.

Drivers who rely on public transportation may not have the ability to use this app because they don’t own a car or drive their own vehicle. These systems also require a lot of maintenance because many sensors need to be replaced often due to wear and tear or vandalism.

Why IoT Based Smart Parking System Is Needed?
The demand for parking is ever-growing, and with cars becoming more fuel-efficient, the number of vehicles will continue to increase. As a result, there are fewer available parking spaces.

Smart parking systems solve the problem of finding an available spot by providing drivers with information about available spots near them. Drivers can also use the system remotely via their smartphone to find a space before arriving at the parking lot.

These systems help drivers to find an available spot faster and more easily than traditional methods like circling around or waiting for someone to leave. By leveraging this technology, drivers can avoid wasting time looking for a space.

How To Use IoT For Smart Parking Solution Development
The average parking space in the U.S. is 25 feet long and costs about $1,000 per year to rent. This means that it costs the average driver about $250 each month for parking alone. In some large cities, a monthly parking bill can cost upwards of $500!

Traditional methods involve driving around looking for available spaces on foot, which is not only time-consuming but can be frustrating when you have to circle a block or two just to find a spot.

Luckily, smart parking systems allow drivers to use their smartphones to locate a close and available spot from a distance instead of wasting gas and time hunting for one on foot. These systems also provide information about the availability rates so customers know what they might have to pay before arriving at the location.

These IoT-based solutions help reduce traffic congestion by allowing drivers to park remotely without having to worry about finding the right space on the street themselves. It’s also more environmentally friendly because people don’t have to waste gas driving around looking for an elusive parking space.

Products for IoT Based Smart Parking System
The parking system is comprised of three components: an interactive map of the area, a sensor that provides data on parking availability, and a payment module for convenience. The interactive map is designed to show drivers’ available slots and information about their location, such as how much time they’ll be available after purchase.

The sensor detects empty spaces and communicates this information to the map and to the driver's phone. This allows drivers to find the best parking space for their needs and keep track of it remotely. The third component, the payment module, allows drivers to pay for their parking slot without having to walk back to their car.

When you equip your business with an IoT-based smart parking system, you will ensure that your customers can easily find a spot on your property at any given time. It also offers convenience for those who don't want to walk or worry about traffic congestion when they're on their way out.

CHINT products for Parking System
IoT-based smart parking systems are offered by **CHINT**. These systems provide users with advanced information about available parking slots and other useful data like current prices and availability rates. The **Rear-view System** is tailored to the user's preferences and conveniently reserving a space before driving to the location. If drivers prefer not to walk or would rather avoid potential traffic congestion, they can reserve a space remotely via their smartphones and drive to the location.
CHINT offers multiple types of parking services, including:

- Parking management: This service includes managing car parks for events, conferences, and community and residential complexes or developments.
- Parking reservation: This service allows drivers to reserve a space in advance before reaching their destination. It also enables drivers to see which spaces are available at any time before arriving.
- RFP (Request for proposal): This service assists in identifying potential vendors who can meet your requirements for car parks, such as size requirements, specific requirements such as temperature control, and more.

Conclusion
IoT has many different applications, but one of the most exciting is its use in smart parking. IoT-based parking systems are able to better track the availability of parking spots on a given lot, making it easier to find an available parking spot.

It is important to note that not all IoT-based parking systems are the same. For example, some use QR codes to identify available parking spots, while others use sensors to detect when a car leaves a parking spot. The benefits of an IoT-based smart parking system are that it is more creative, efficient, and convenient for both drivers and owners of the parking lot.

10.6 Energy Management in Smart cities

A smart city is a sustainable and efficient urban centre that provides a high quality of life to its inhabitants through optimal management of its resources. Energy management is **one of the most demanding issues within such urban centres owing to the complexity of the energy systems and their vital role**.

A process that includes planning & management of your energy consumption patterns in commercial & industrial sectors. Our Energy management Solution takes complete control of your energy data at a fundamental & granular level while reducing your energy costs.

11. Industrial IoT

Industrial IoT (IIoT) is **the use of network-connected sensors and other monitoring devices to improve the manufacturing and quality of an organization's products and product parts**. IIoT devices are used primarily for insights on machine health, causes for defective parts, and general data collection.

Industrial IoT (IIoT) is the use of network-connected sensors and other monitoring devices to improve the manufacturing and quality of an organization’s products and product parts.

IIoT devices are used primarily for insights on machine health, causes for defective parts, and general data collection.

While having data can lead to improvements throughout the process, the amount of data — in addition to the time and resources it takes to process and analyze it — is considerable. IIoT is still an emerging field with certain manufacturers hoping it gains greater adoption in the coming years. The delays are due in part to the longevity of non-network enabled manufacturing machines, and the cost of replacing them.

Industrial IoT (IIoT) brings machines, cloud computing, analytics, and people together to improve the performance and productivity of industrial processes. With IIoT, industrial companies can digitize processes, transform business models, and improve performance and productivity, while decreasing waste. These asset intensive companies operating in a range of industries such as manufacturing, energy, agriculture, transportation and utilities, are working on IoT projects that connect billions of devices and deliver value across a variety of use cases including predictive quality and maintenance analytics, asset condition monitoring, and process optimization.

A typical industrial facility has thousands of sensors generating data. With IIoT, manufacturers, for example, can combine machine data from a single line, factory, or a network of sites, such as manufacturing plants, assembly facilities, and refineries, to proactively improve performance by identifying potential bottlenecks, failures, gaps in production processes, and quality issues before they happen. Combining data from a network of sites can also result in a more efficient control of material flow, early detection and identification and elimination of production or supply bottlenecks, and the optimized operation of machinery and equipment in all facilities.

11.1 IIoT requirements

Requirements for the Industrial IoT are a superset of the main IoT requirements. Just like the total IoT market, the Industrial IoT market needs **inexpensive nodes that work on easy to install links like wireless, power line, and simple twisted pair**.

11.2 Design considerations

- Front-end Edge Devices. Sensor data is most of the IIoT, therefore the hardware used to gather and collect it is a critical component of the system. ...
- Connectivity Technology. ...
- Industrial IoT Platforms For Data Analytics.

11.3 Applications of IIoT

1. Remote access of machines

With remote access to industrial machines, the service engineers and other stakeholders can conveniently access the machine from their current locations, check their log files on the PLCs and change settings if required. It will take only a few minutes to access the machine and find problem, which will save a time-consuming trip to the manufacturer’s site.

2. Update new functionalities on HMIs

New functionalities are added to the machines to make the job more efficient and fast. While the programmer implements this functionality in the control panel of the machine, the HMI software needs to be updated, and tested in order to launch the new functionality. In that case, HMI software updates can be applied remotely through secure network access over the internet. With the web-based virtual network connection, you are able to view and check the HMI functionality anytime on that IIoT platform.

3. Predictive analysis for machine maintenance

As with all hardware, even the IIoT enabled machines undergo wear and tear before finally replaced with new equipments. In such scenarios, active and regular maintenance is crucial to prevent downtime and decreased production output. Using cloud to collect, store and access information on the machine parts, maintenance engineers can keep track of the remaining useful life (RUL) for every asset. Automatic notifications can be sent to the right person if an asset reaches its maintenance limit. By analysing the potential problems via remote access and online diagnostics tools, you are likely to get the right spare parts.

4. Analyse and optimize industrial robot actions

Industrial robots can make repetitive work easy. IIoT features with remote access can change the robot program actions and get better insights of the log files. Video analysis can also help in improving the actions of certain robots. Access to live stream and IP camera recordings can make improvements far more easy and fast. A VPN connection can be set up easily for full network access to any device that is connected to the robot.

5. Manage building automation data from multiple locations

IIoT can be used to monitor and control the heating, lighting, energy consumption, fire protection, employee safety and many other systems for multiple buildings from a central location. The real-time machine data can be transferred to a central cloud application, using industrial communication networks.

- Process monitoring – mining: In the classic game “Rock, Paper, Scissors” rock can be beaten by paper. ...
- Equipment maintenance – elevators: An elevator company wants to ensure that the customers who install their product receive the highest quality service.

11.4 Benefits of IIoT

- Increase productivity and uptime.
- Improve process efficiencies.
- Accelerate innovation.
- Reduce asset downtime.
- Enhance operational efficiency.
- Create end-to-end operational visibility.
- Improve product quality.
- Reduce operating costs.
- Optimize production scheduling.
- Improve overall equipment effectiveness (OEE).

11.5 Challenges of IIoT

- Connectivity Outage Challenge. There is a constant need for uninterrupted connectivity if an enterprise is planning to go IIoT
- Delivering Value to The Customer
- Data Storage
- Security
- Analytics Challenges
- Conclusion.