UNIT II

Prototyping IoT Objects using Microprocessor/Microcontroller

Working principles of sensors and actuators, setting up the board – Programming for IoT, Reading from Sensors, Communication: communication through Bluetooth, Wi-Fi. IoT Sensor Working and Its Applications.

Explanation:

What Is Prototyping in IoT?

IoT prototyping is the action of experimenting and implementing design ideas into preliminary versions of a finished product.

Essentially, it involves trying out and testing different ways to bring something from the planning phase to reality.

In the world of IoT, a prototype could be:

- A user interface (UI)
- A hardware device
- Backend software
- Connectivity of a system

I. Working principles of sensors and actuators:

The term IoT stands for Internet of Things and it is the most significant as well as promising technology nowadays. Some of the researchers in the market estimated that there are a billion devices are connected with sensors like wearables, smartphones, etc.

Currently, every sensor plays an essential role in the Internet of Things.

These sensors are mainly used for detecting or monitoring the quality of air, health status, home security, etc.

Similarly, these sensors are used in IoT for monitoring the process of production, so named as IoT sensor. Because of these reasons, one has to know about its importance, working, and its usage to obtain information.

What is an IoT Sensor?

Sensors are devices that detect and respond to changes in an environment.
Or
IoT sensors are pieces of hardware that detect changes in an environment and collect data.

There are different types of sensors available in the market which is used for different applications like to collect the data from the environment.

In an IoT ecosystem, there are two main things we have to consider like the internet & the physical devices such as actuators & sensors.

The sensor and network connectivity in the IoT mainly located in the bottom layer.
The main function of this is to collect the information.
This bottom layer in the IoT is a very important part, and it includes connectivity of network to next layer like the gateway & network layer.

Fig: internet-of-things

The main function of these sensors is to gather information from the surroundings. The connection of these to IoT can be done directly otherwise indirectly once the conversion of signal & processing is done. All the sensors are not similar because different IoT applications need different kinds of sensors. For example, the interfacing of digital sensors with a microcontroller with the help of SPI bus (Serial Peripheral Interface). Although for analog sensors, either ADC otherwise Sigma-Delta modulator can be applicable for changing the data into SPI o/p.

Types of IoT Sensors:

The different types of IoT sensors available in the market. Here is the IoT sensors list with its working is discussed below.

Temperature Sensor

The temperature sensor is used to detect the heat energy which is produced from an object otherwise nearby area. These sensors are applicable for the Internet of Things (IoT), which includes from manufacturing to farming. The main role of these sensors in manufacturing is for temperature monitoring of machines. Similarly, in the agriculture field, these sensors are used to monitor the temperature of plants, soil, and water.

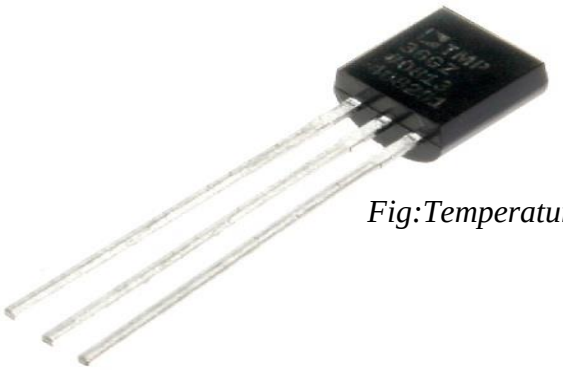


Fig:Temperature Sensor

Temperature sensors are thermistors, thermocouples, ICs (integrated circuits), and RTDs (resistor temperature detectors). The applications of temperature sensors mainly include refrigerators, ACs, etc.

Smoke Sensor

Smoke sensors have been using in various applications like homes, industries, etc. These sensors are very convenient as well as easy to use by the arrival of the Internet of Things. Also, by adding a wireless connection to smoke detectors, the additional features can be enabled to increase security & ease.

Fig: smoke-sensor

Motion Sensor

The motion sensor is used for security reasons however these are also used in hand dryers, energy management systems, automatic parking systems, automatic door controls, automated toilet flushers, automated sinks, etc. These sensors are also applicable for the Internet of Things to check them with the help of computer otherwise smartphone.

Fig:motion-sensor

Humidity Sensors

Humidity sensors are used to monitor the level of humidity in the amount of vapor of water within the air. Otherwise, it will influence the comfort of human & several industrialized processes. The units for measurement humidity is RH (relative humidity), D/F PT (/frost point) & PPM (parts per million).

Fig: humidity-sensors

Pressure Sensor

The pressure sensors are used in IoT for monitoring devices and systems which are determined by force signals. As the range of pressure is outside the threshold stage, then the device gives an alert to the user regarding the issues that must be fixed. The best example of a pressure sensor is BMP180, which can be used in PDAs, mobile phones; external device, GPS navigation devices, etc. These sensors are also applicable in aircraft and smart vehicles to decide altitude & force correspondingly. In a motor vehicle, TMPS (tire pressure monitoring system) can also be used for giving an alert to the driver while tire pressure is extremely less & it could make unsafe driving situations.

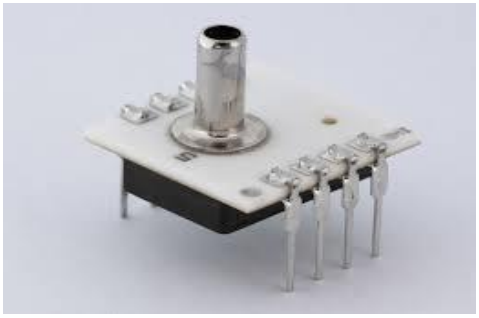


Fig: pressure-sensors

Gas Sensor

Gas sensors are mainly used for detecting toxic gases. The most frequently used technologies are photo-ionization, semiconductor, and electrochemical. There are different types of gas sensors are available based on technical specifications & advancements to expand the connectivity of wired & wireless arranged within IoT applications.

Fi: gas-sensor

IR Sensors

Infrared sensors are mainly used to measure the heat which is produced by objects. These sensors are used in the various applications of IoT like healthcare for monitoring the flow of blood, BP, etc. These sensors are used in smartphones for controlling, wearable devices for detecting the amount of light, detection of blind-spot within vehicles, etc.

Accelerometer Sensor

Accelerometer sensors are utilized in aircrafts vehicles, smartphones. Similarly, these are used in different applications to identify the direction of an object, tilt, tap, shake, positioning, and motion, vibration, or shock. Types of accelerometers are like capacitive, Hall-effect & piezoelectric.

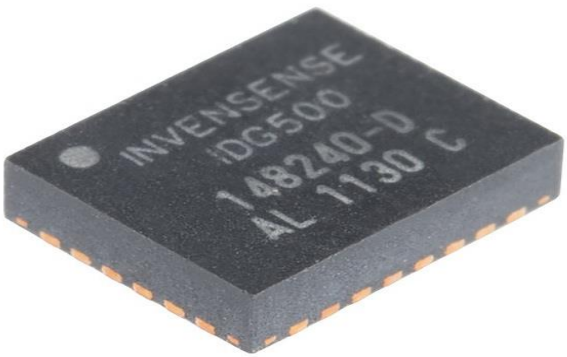


Fig:accelerometer-sensor

Image Sensor

Image sensors are applicable in medical imaging systems, media house, thermal imaging devices, digital cameras, night-vision equipment, sonars, radars, & biometric systems. These sensors are used in the retail industry for monitoring the visiting count of the customers in the store with the help of network like IoT. The applications of image sensors mainly include offices, corporate buildings for monitoring the employees.

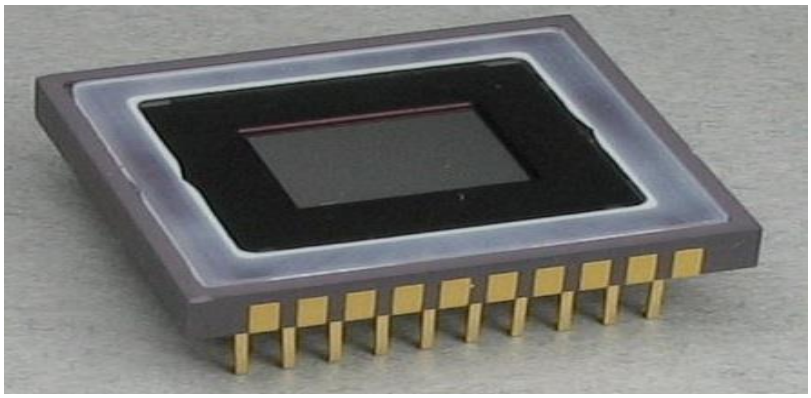


Fig: image-sensor

Proximity Sensors

Proximity sensors are used to detect the existence or nonexistence of a near object with no physical contact. These sensors are classified into different types like capacitive, inductive, ultrasonic, magnetic, and photoelectric. These sensors are frequently used for process monitoring, control, and object counters.

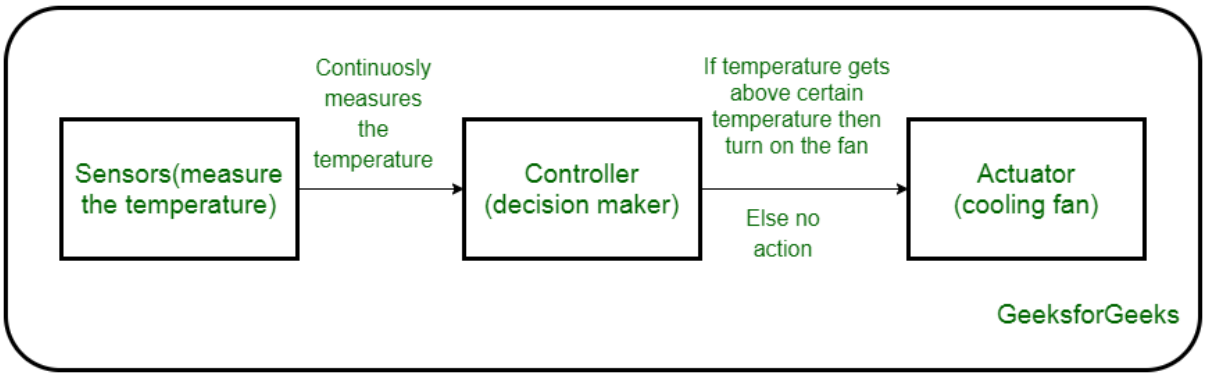
Fig:proximity-sensors

This is all about an overview of IoT sensors. These sensors play a key role in our daily life. These are used to check your health status, home security, air quality, and are extensively used within the IIoT (Industrial Internet of Things) to observe the process in production. Internet of Things allows detection, heightened surveillance, and monitoring to combine for improving health & increasing safety.

ACTUATORS:

An IoT device is made up of a Physical object ("thing") + Controller ("brain") + Sensors + Actuators + Networks (Internet). An actuator is a machine component or system that moves or controls the mechanism of 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.

ıctuators

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.

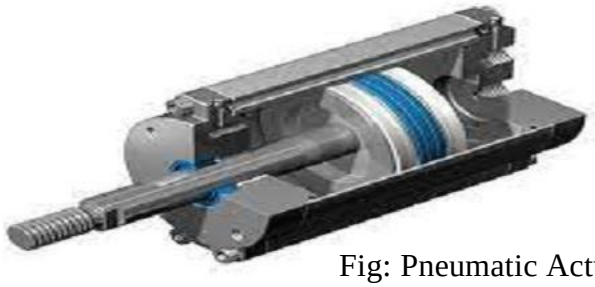


Fig: Pneumatic Actuators

Advantages :

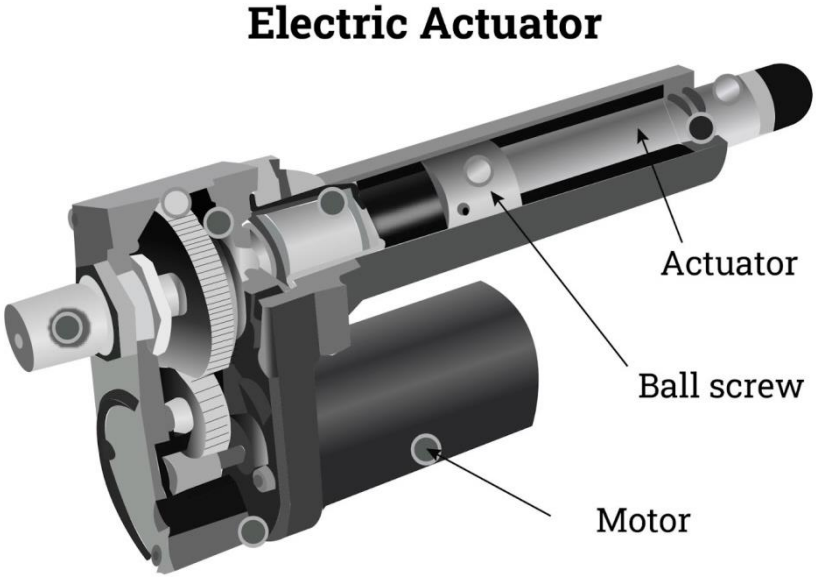
- They are a low-cost option and are used at extreme temperatures where using air is a safer option than chemicals.
- They need low maintenance, are durable, and have a long operational life.
- It is very quick in starting and stopping the motion.

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.

II. Setting up the board:

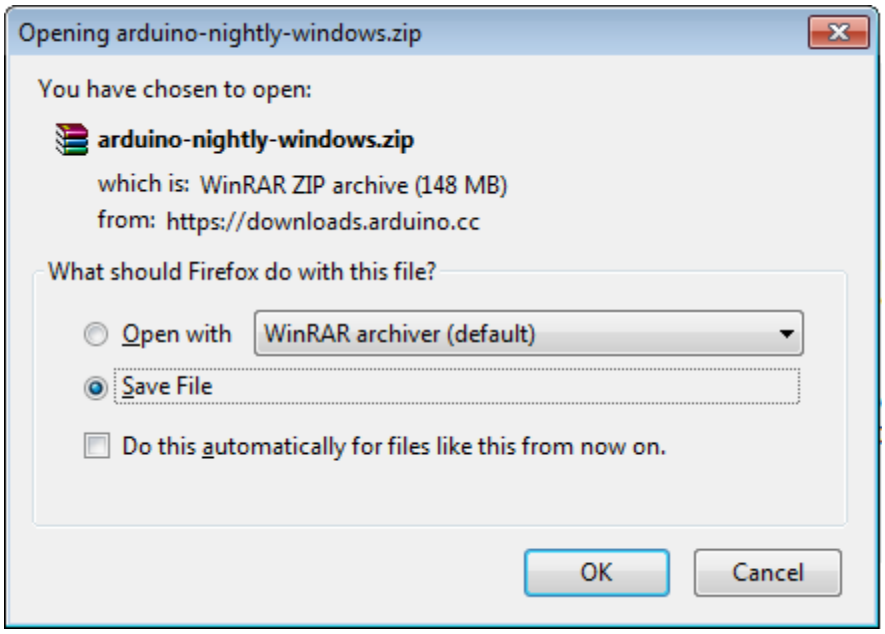
how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.



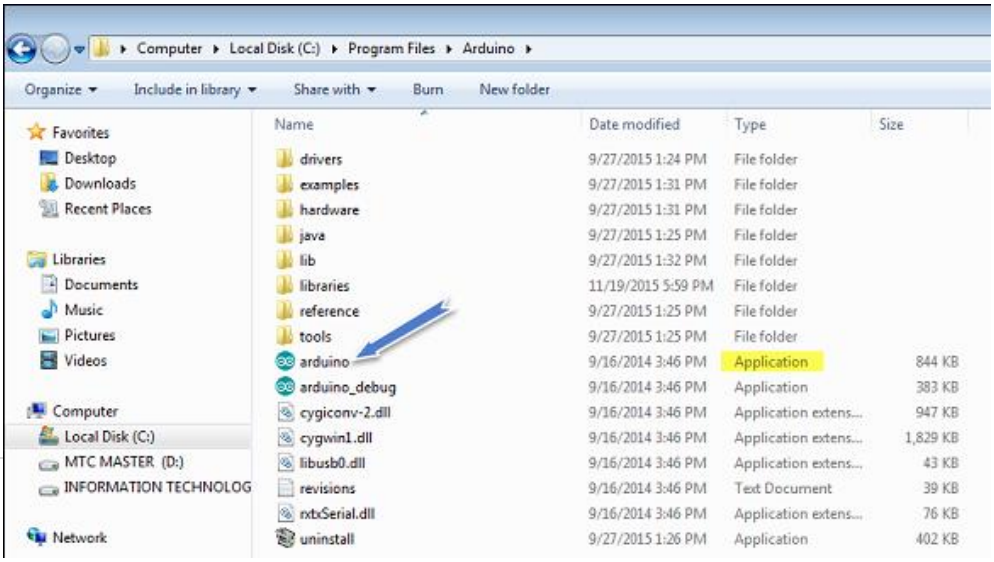
Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

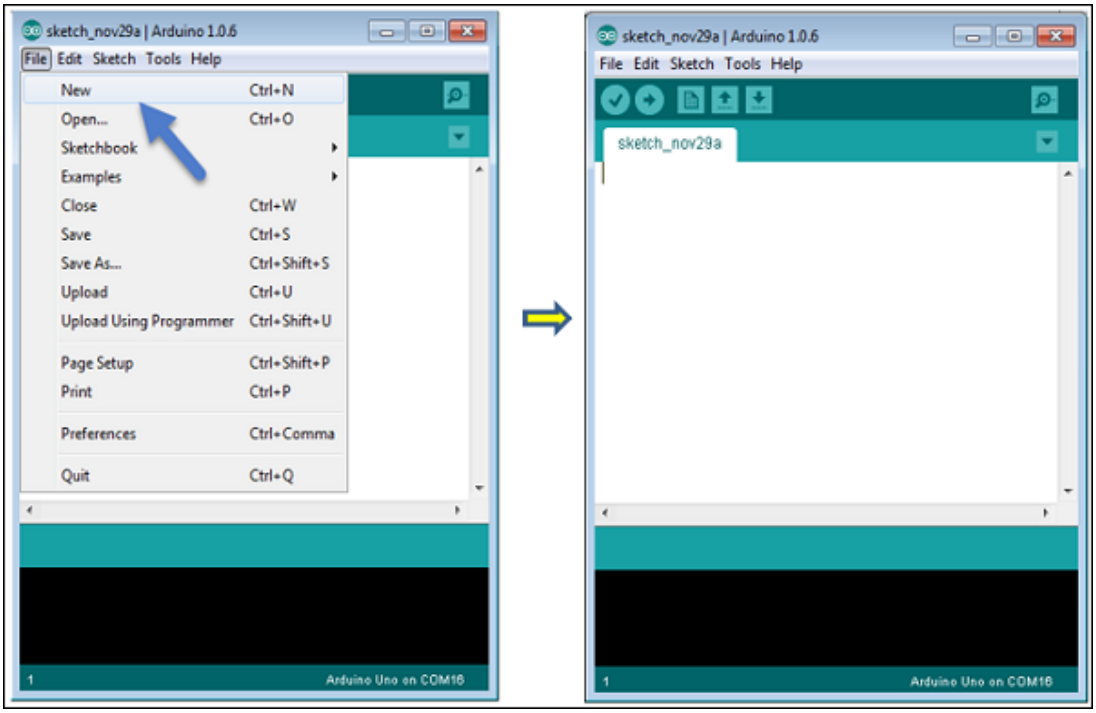


Step 5 − Open your first project.

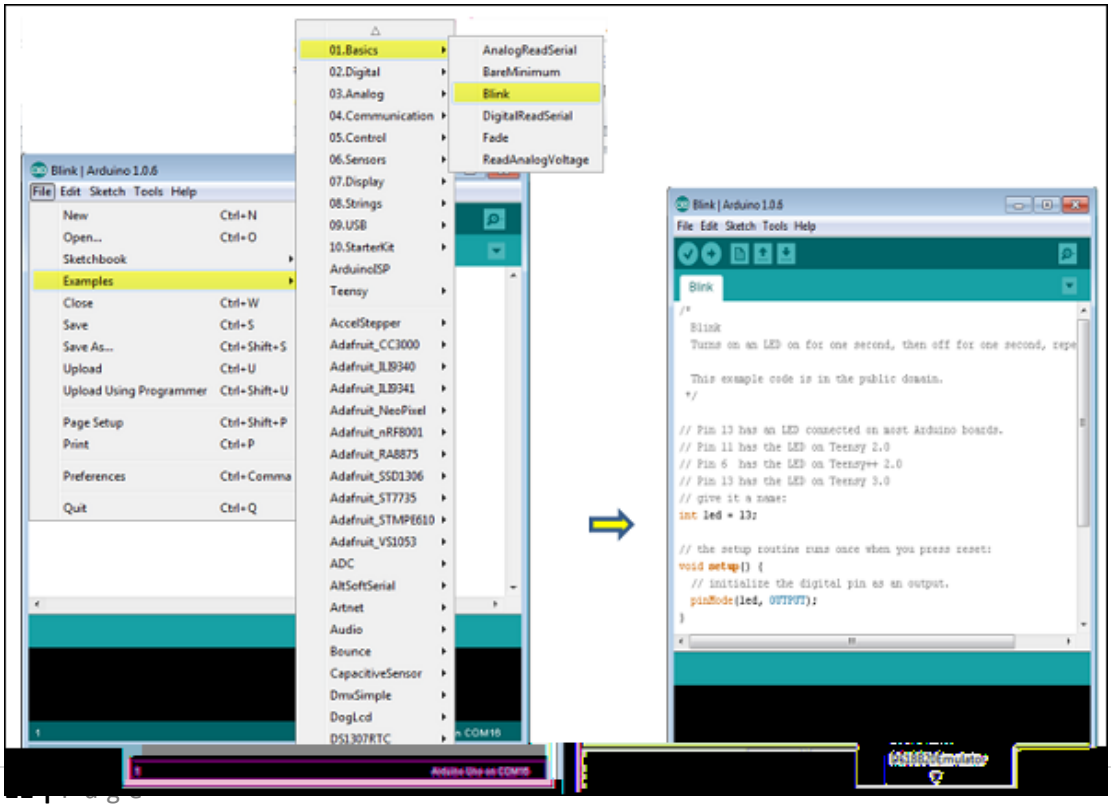
Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.



To open an existing project example, select File → Example → Basics → Blink.

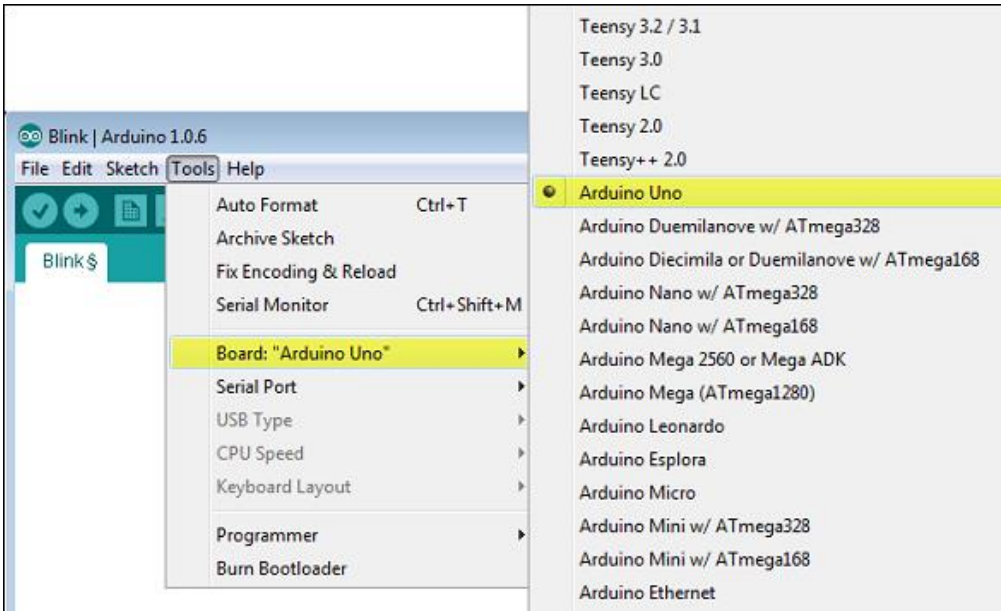


Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

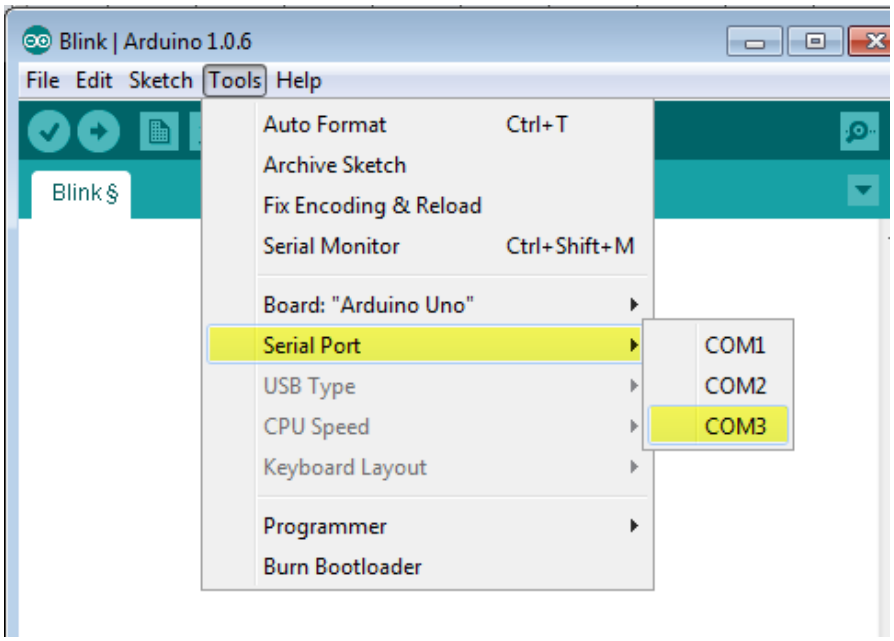
Go to Tools → Board and select your board.



Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

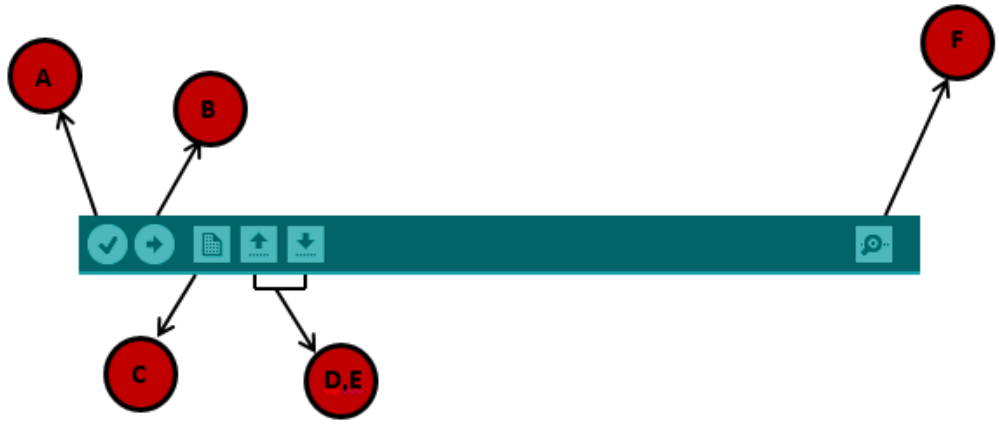
Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.



Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.



A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

III. Programming for IoT:

The most popular languages in IoT are Java, C, C++, Python, Javascript, PHP, C#, Lua, R, Go, Ruby, Node.js, and Assembler

Finally, the choice of programming language completely depends on the developers and requirements. All of the top programming languages have their benefits and use cases. But we widely consider that for devices there are C and C++; for gateway and IoT applications, Java and python; for Cloud, Java, JavaScript, and PHP are the best choices.

Example programs:

1. IoT program for Displaying humidity and temperature values on LCD

Code:

```
//Interface the DHT11 Temp & Humidity sensor and display humidity and   temperature

//in Celsius on a 16x2 character LCD
```

```
#include <dht.h>
#include   <LiquidCrystal.h>

dht DHT;
const int RS = 2, EN = 3, D4 = 4, D5 = 5, D6   = 6, D7 = 7;
LiquidCrystal lcd(RS,EN,D4,D5,D6,D7);   //set Uno pins that are   connected to LCD, 4-bit mode

void setup() {
  lcd.begin(16,2);    //set   16 columns and 2 rows of 16x2 LCD

}

void loop() {
  int readDHT   = DHT.read11(8);    //grab 40-bit data packet from DHT sensor
  lcd.setCursor(0,0);
  lcd.print("Temp: ");
  lcd.print(DHT.temperature);
  //lcd.print((char)223);         //used to display degree symbol on display
  //lcd.write(0xdf);              //another   way to display degree symbol
  lcd.print("C");
  lcd.setCursor(0,1);
   lcd.print("Humidity: ");
  lcd.print(DHT.humidity);
  lcd.print("%");
   delay(3000);
}
```

2. IoT program to calculate the distance to an object with the help of an ultrasonic sensor and display it on anLCD.

Code:

```
const int trigPin = 5;

const int echoPin = 18;


//define sound speed in cm/uS

#define SOUND_SPEED 0.034

#define CM_TO_INCH 0.393701


long duration;

float distanceCm;

float distanceInch;


void setup()

{

  Serial.begin(115200); // Starts the serial communication

  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

  pinMode(echoPin, INPUT); // Sets the echoPin as an Input

}


void loop() {

  // Clears the trigPin

  digitalWrite(trigPin, LOW);

  delayMicroseconds(2);

  // Sets the trigPin on HIGH state for 10 micro seconds

  digitalWrite(trigPin, HIGH);

  delayMicroseconds(10);

  digitalWrite(trigPin, LOW);


  // Reads the echoPin, returns the sound wave travel time in microseconds

  duration = pulseIn(echoPin, HIGH);
```

```
  // Calculate the distance
  distanceCm = duration * SOUND_SPEED/2;

  // Convert to inches
  distanceInch = distanceCm * CM_TO_INCH;

  // Prints the distance in the Serial Monitor
  Serial.print("Distance (cm): ");
  Serial.println(distanceCm);
  Serial.print("Distance (inch): ");
  Serial.println(distanceInch);

  delay(1000);
}
```

IV. Reading from Sensors:

Reading a Sensor with Python

In this section, you will learn how to read a sensor connected to an external piece of hardware (an Arduino) with Python. To accomplish this, the following hardware is required:

- A computer running Python
- An Arduino
- A potentiometer (the sensor)
- wires, a resistor, an LED, and a breadboard to connect the sensor to the Arduino
- A USB cable to connect the Arduino to the computer

You will also need to download the Arduino IDE (the Arduino Integrated Development Environment) using the following link as shown in the previous section:

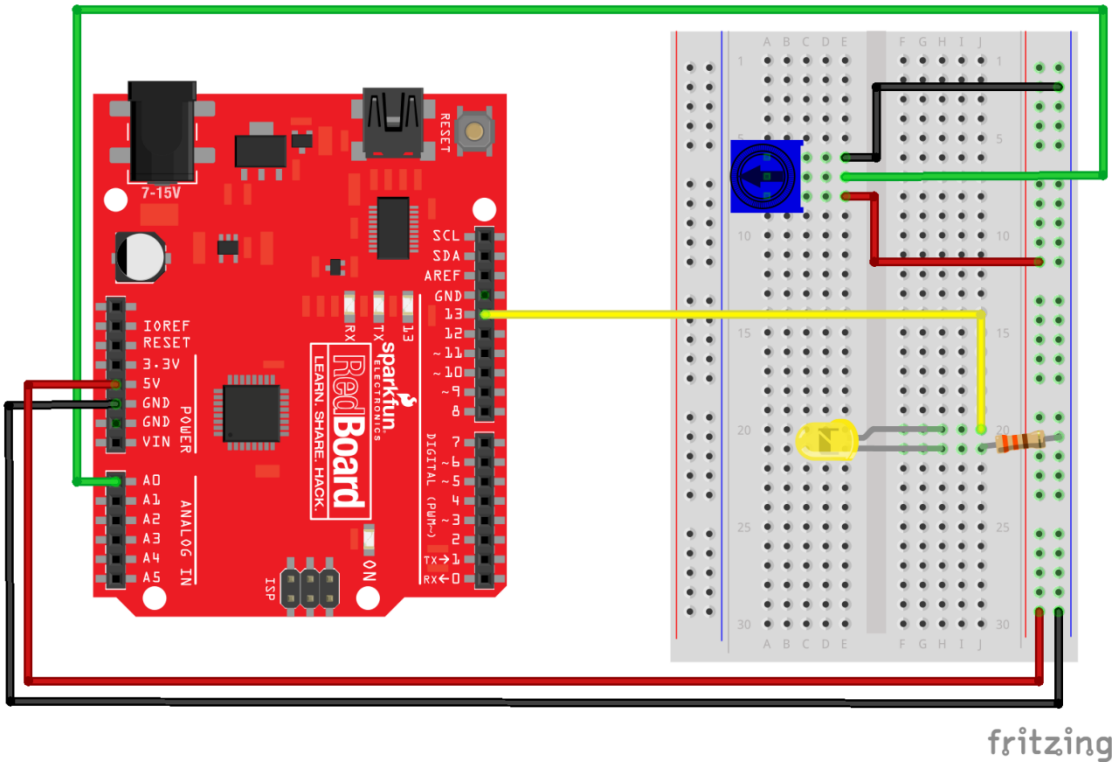
https://www.arduino.cc/en/Main/Software

Wire the sensor to the Arduino

Connect the potentiometer sensor to the Arduino using a resistor, wires and a breadboard. The middle pin of the potentiometer is connected to pin A0 on the Arduino (green wire). Connect the LED to the Arduino. Note the long lead of the LED is connected to PIN13 on the Arduino (yellow wire) and the short lead of the LED is connected through a resistor to ground (black wire). If the LED is wired backward, the LED will not turn on.

In the hardware schematic, the blue square with an arrow on it is a potentiometer. The user knows when the sensor signal changes because the user manually turns the potentiometer dial.

The hardware schematic describes how to the Arduino, LED, resistor, and potentiometer are connected with jumper wires and a
breadboard.



Upload code to the Arduino

Once the LED and potentiometer are hooked up the Arduino, upload the following code to the Arduino using the Arduino IDE. Note that Arduinos don't use the Python programming language. The programming language used by Arduinos is a variant of the C programming language.

The Arduino sketch below (an Arduino program is called a sketch) accomplishes a couple things. First, the Arduino reads the potentiometer sensor value and stores the sensor value in the variable sensorValue. Next, the Arduino sends the sensor value over the serial line (as a byte string). Finally, sensorValue is compared to 500. If sensorValue is less than 500, the LED stays off. If sensorValue is greater than 500, the LED turn on. The read-send-compare process repeats in a loop.

```
// potentiometer_read.ino
// reads a potentiometer and sends value over serial
int sensorPin = A0;  // The potentiometer on pin 0
int ledPin = 13;     // The LED is connected on pin 13
int sensorValue;     // variable to stores data

void setup() // runs once when the sketch starts
{
  // make the LED pin (pin 13) an output pin
```

```
  pinMode(ledPin, OUTPUT);

  // initialize serial communication
  Serial.begin(9600);
}

void loop() // runs repeatedly after setup() finishes
{
  sensorValue = analogRead(sensorPin);  // read pin A0
  Serial.println(sensorValue);         // send data to serial

  if (sensorValue < 500) {            // less than 500?
    digitalWrite(ledPin, LOW); }      // Turn the LED off

  else {                              // greater than 500?
    digitalWrite(ledPin, HIGH); }     // Keep the LED on

  delay(100);             // Pause 100 milliseconds
}
```

V. IOT Communication :

IoT is connection of devices over internet, where these smart devices communicate with each other , exchange data , perform some tasks without any human involvement. These devices are embedded with electronics, software, network and sensors which help in communication. Communication between smart devices is very important in IOT as it enables these devices to gather, exchange data which contribute in success of that IOT product/project.

Types of Communications in IOT :

The following are some communication types in IoT:-

1. Human to Machine (H2M):

In this human gives input to IOT device i.e as speech/text/image etc. IOT device (Machine) like sensors and actuators then understands input, analyses it and responds back to human by means of text or Visual Display. This is very useful as these machines assist humans in every everyday tasks. It is a combo of software and hardware that includes human interaction with a machine to perform a task.

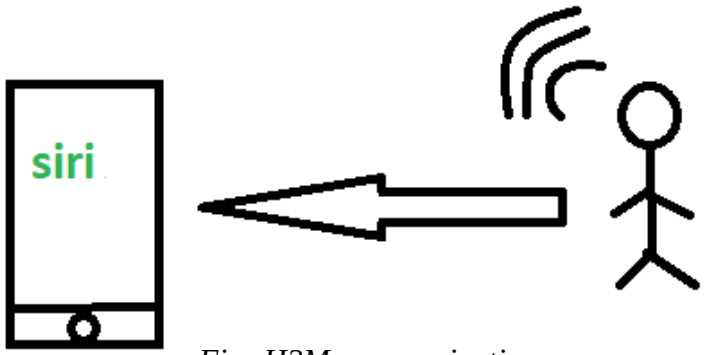


Fig: H2M communication

Merits: This H2M has a user-friendly interface that can be quickly accessed by following the instructions. It responds more quickly to any fault or failure. Its features and functions can be customized.

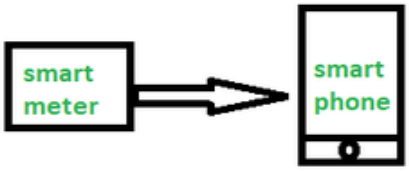
Examples:

- Facial recognition.
- Bio-metric Attendance system.
- Speech or voice recognition.

2. Machine to Machine (M2M) :

In this the interaction or communication takes place between machines by automating data/programs. In this machine level instructions are required for communication. Here communication takes place without human interaction. The machines may be either connected through wires or by wireless connection. An M2M connection is a point-to-point connection between two network devices that helps in transmitting information using public networking technologies like Ethernet and cellular networks. IoT uses the basic concepts of M2M and expands by creating large "cloud" networks of devices that communicate with one another through cloud networking platforms.

M2M communication



Advantages –

This M2M can operate over cellular networks and is simple to manage. It can be used both indoors and outdoors and aids in the communication of smart objects without the need for human interaction. The M2M contact facility is used to address security and privacy problems in IoT networks. Large-scale data collection, processing, and security are all feasible.

Disadvantages –

However, in M2M, use of cloud computing restricts versatility and creativity. Data security and ownership are major concerns here. The challenge of achieving interoperability between cloud/M2M IoT systems is daunting. M2M connectivity necessitates the existence of a reliable internet connection.

Examples:

- Smart Washing machine sends alerts to the owners' smart devices after completion of washing or drying of clothes.
- Smart meters tracks amount of energy used in household or in companies and automatically alert the owner.

3. Machine to Human (M2H) :

In this machine interacts with Humans. Machine triggers information (text messages/images/voice/signals) respective / irrespective of any human presence. This type of communication is most commonly used where machines guide humans in their daily life. It is way of interaction in which humans co-work with smart systems and other machines by using tools or devices to finish a task.

Examples:

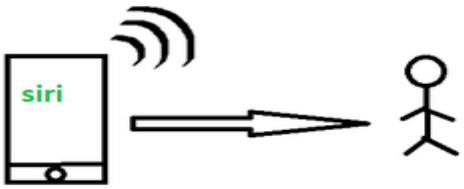
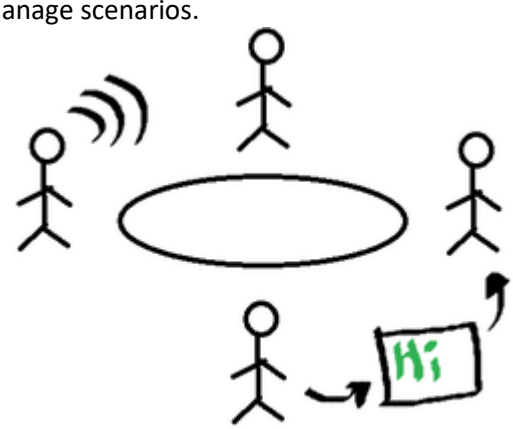


Fig: M2H communication

- Fire Alarms
- Traffic Light
- Fitness bands
- Health monitoring devices

4. Human to Human (H2H) :

This is generally how humans communicate with each other to exchange information by speech, writing, drawing, facial expressions, body language etc. Without H2H, M2M applications cannot produce the expected benefits unless humans can immediately fix issues, solve challenges, and manage scenarios.



H2H communication

For, communication of IoT devices many protocols are used. These IoT protocols are modes of communication which give security to the data being exchanged between IoT connected devices. Example bluetooth, wifi, zigbee etc.

CONNECTION THROUGH BLUETOOTH

Bluetooth is an universal for short range wireless voice and data communication.It is a Wireless Personal Area Network (WPAN) technology and is used for exchanging data over smaller distances. This technology was invented by Ericson in 1994. It operates in the unlicensed, industrial, scientific and medical (ISM) band from 2.4 GHz to 2.485 GHz. Maximum devices that can be connected at the same time are 7. Bluetooth ranges up to 10 meters. It provides data rates up to 1 Mbps or 3 Mbps depending upon the version. The spreading technique that it uses is FHSS (Frequency-hopping spread spectrum). A Bluetooth network is called a **piconet** and a collection of interconnected piconets is called **scatternet**.

What is bluetooth.

Bluetooth Transmission capacity 720 kbps.

Bluetooth is Wireless.

Bluetooth is Low cost short distance radio communications standard .

Bluetooth is robust and flexible .

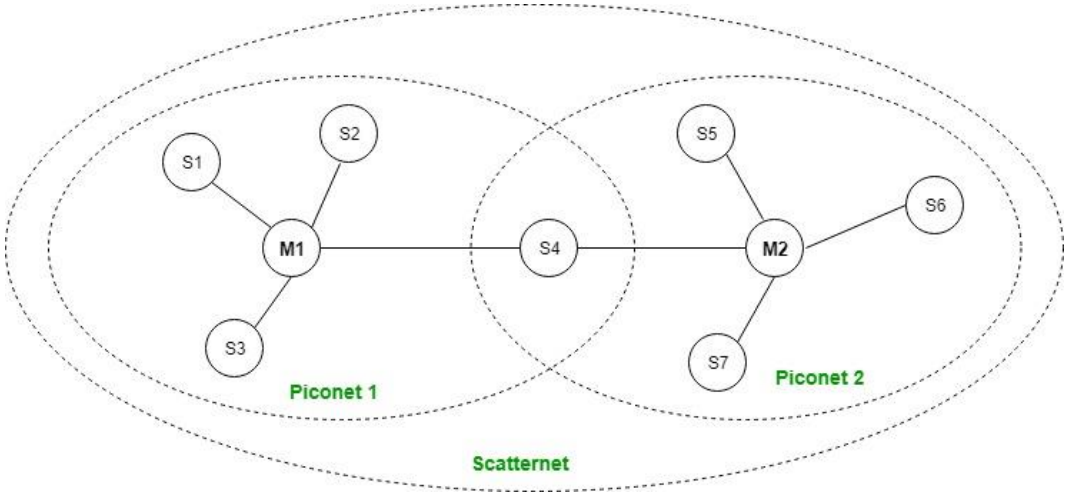
Bluetooth is cable replacement technology that can be used to connect almost any device to any other device.

The basic architecture unit of a bluetooth is a piconet.

Bluetooth Architecture:

The architecture of Bluetooth defines two types of networks:

1. Piconet
2. Scatternet



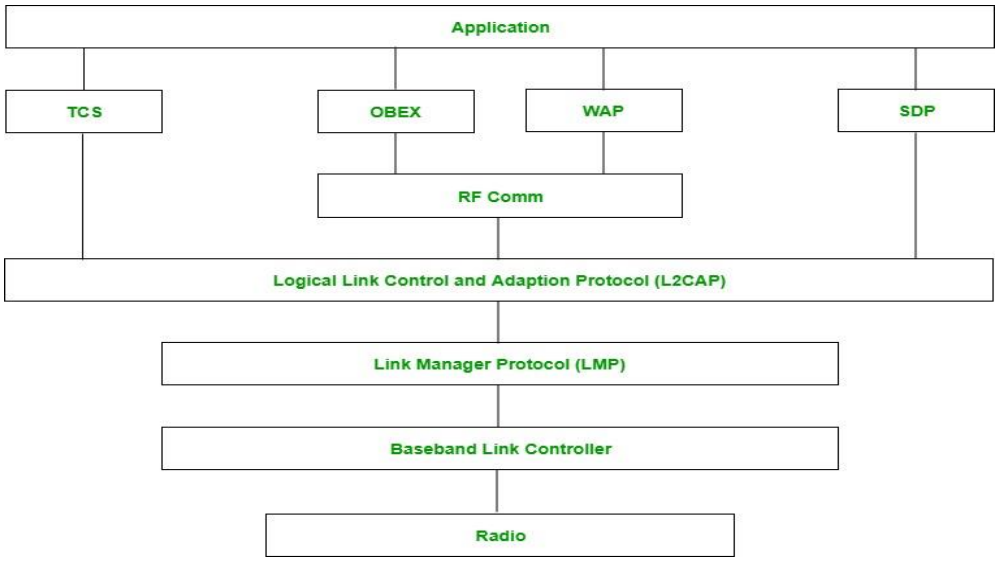
Piconet:

Piconet is a type of Bluetooth network that contains **one primary node** called the master node and **seven active secondary nodes** called slave nodes. Thus, we can say that there is a total of 8 active nodes which are present at a distance of 10 meters. The communication between the primary and secondary nodes can be one-to-one or one-to-many. Possible communication is only between the master and slave; Slave-slave communication is not possible. It also has **255 parked nodes**, these are secondary nodes and cannot take participation in communication unless it gets converted to the active state.

Scatternet:

It is formed by using **various piconets**. A slave that is present in one piconet can act as master or we can say primary in another piconet. This kind of node can receive a message from a master in one piconet and deliver the message to its slave in the other piconet where it is acting as a slave. This type of node is referred to as a bridge node. A station cannot be mastered in two piconets.

Bluetooth protocol stack:



1. **Radio (RF) layer:** It specifies the details of the air interface, including frequency, the use of frequency hopping and transmit power. It performs modulation/demodulation of the data into RF signals. It defines the physical characteristics of Bluetooth transceivers. It defines two types of physical links: connection-less and connection-oriented.

2. **Baseband Link layer:** The baseband is the digital engine of a Bluetooth system and is equivalent to the MAC sublayer in LANs. It performs the connection establishment within a piconet, addressing, packet format, timing and power control.

3. **Link Manager protocol layer:** It performs the management of the already established links which includes authentication and encryption processes. It is responsible for creating the links, monitoring their health, and terminating them gracefully upon or failure.

4. **Logical Link Control and Adaption (L2CAP) Protocol layer:** It is also known as the heart of the Bluetooth protocol stack. It allows the communication between upper and lower layers of the Bluetooth protocol stack. It packages the data packets received from upper layers into the form expected by lower layers. It also performs segmentation and multiplexing.

5. **Service Discovery Protocol (SDP) layer:** It is short for Service Discovery Protocol. It allows discovering the services available on another Bluetooth-enabled device.

6. **RF comm layer:** It is a cabal replacement protocol. It is short for Radio Frontend Component. It provides a serial interface with WAP and OBEX. It also provides emulation of serial ports over the logical link control and adaption protocol(L2CAP). The protocol is based on the ETSI standard TS 07.10.

7. OBEX: It is short for Object Exchange. It is a communication protocol to exchange objects 2 devices.

8. WAP: It is short for Wireless Access Protocol. It is used for internet access.

9. TCS: It is short for Telephony Control Protocol. It provides telephony service. The basic function of this layer is call control (setup & release) and group management for gateway serving multiple devices.

10. Application layer: It enables the user to interact with the application.

Advantage:

- Low cost.
- Easy to use.
- It can also penetrate through walls.
- It creates an Ad-hoc connection immediately without any wires.
- It is used for voice and data transfer.

Disadvantages:

- It can be hacked and hence, less secure.
- It has a slow data transfer rate: of 3 Mbps.
- It has a small range: 10 meters.
- Bluetooth communication does not support routing.
- The issues of handoffs have not been addressed.

Applications:

- Used in laptops, and in wireless PCs.
- In printers.
- In wireless headsets.
- Connecting digital camera wirelessly to a mobile phone.
- Data transfer from one cell phone to other cell phone or computer.
- Medical health care
- sports and fitness
- military
- security
- Consumer ,
- games,
- professional.
- Services.
- Industry

VI. IoT Sensor Working and Its Applications.

The Internet of Things (IoT) ecosystem comprises web-enabled smart devices that use embedded systems, such as processors, sensors, and communication devices, to gather, send, and act on the data they get from their surroundings.

By connecting to an IoT gateway or other edge device, which either sends data to the cloud for analysis or analyzes it locally, IoT devices exchange the sensor data they collect. These gadgets occasionally converse with other similar devices, acting on the data they exchange.

Although individuals can engage with the devices to set them up, give them instructions, or retrieve the data, the hardware accomplishes the majority of the job without their help.

The following table provides an overview of the types of sensors and application examples for various measurement targets.

Measurement target	**Sensor type**	**Application example**
People/objects	Human sensor	Detecting workers
	Photoelectric sensor	Factory product count
Distance	Distance sensor	Distance to object, fluid level
Temperature	Thermocouple	Surface temperature / fluid temperature (high temperature)
	Resistance thermometer	Surface temperature / fluid temperature (low temperature)
	Infrared thermometer	Surface temperature (non-contact)
Temperature and humidity	Temperature/humidity sensor	Ambient temperature and humidity
Pressure	Pressure sensor	Hydraulic/pneumatic pressure
Position	GPS	Satellite position information
	Gyro sensor	Object posture (tilt, angle)
	Encoder	Rotation position
	Potentiometer	Rotation angle measurement
Acceleration	Acceleration sensor	Object movement and impact detection
Vibration	Vibration sensor	Motor vibration measurement
	Accelerometer	Object movement and vibration measurement
Power (alternating current)	Current transformer	Facility power monitoring
Information reading	Barcode reader	Product information reading