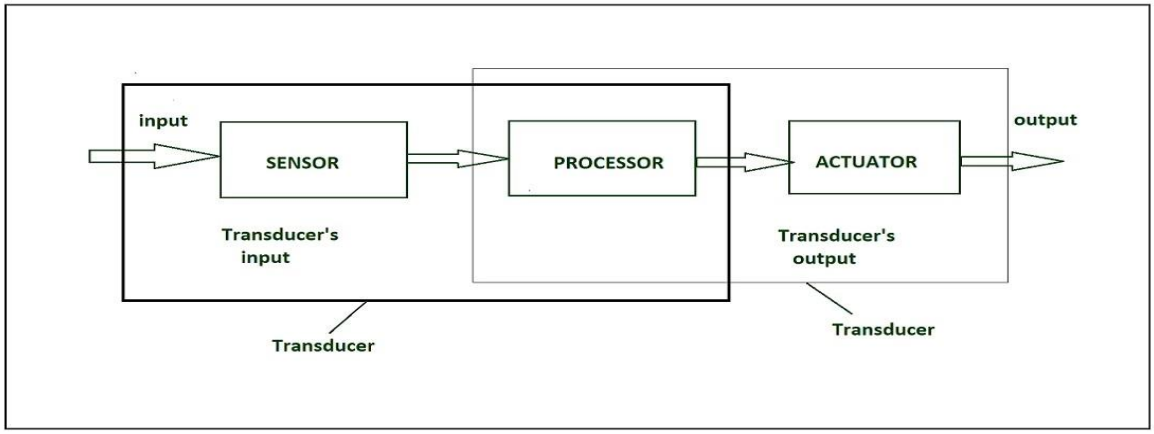
MODULE 4

DEVELOPMENT OF IOT APPLICATIONS WITH EMBEDDED COMPUTING BOARDS

Sensors in Internet of Things(IoT)

- 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



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

Type of sensors

1.Electrical sensor :

- Electrical proximity sensors may be contact or non contact.
- Simple contact sensors operate by making the sensor and the component complete an electrical circuit.
- Non- contact electrical proximity sensors rely on the electrical principles of either induction for detecting metals or capacitance for detecting non metals as well.

2.Light sensor:

- Light sensor is also known as photo sensors and one of the important sensor.
- Light dependent resistor or LDR is a simple light sensor available today.
- The property of LDR is that its resistance is inversely proportional to the intensity of the ambient light i.e when the intensity of light increases, it's resistance decreases and vise versa.

3.Touch sensor:

- Detection of something like a touch of finger or a stylus is known as touch sensor.
- It's name suggests that detection of something.
- They are classified into two types:

4.Mechanical sensor:

- Any suitable mechanical / electrical switch may be adopted but because a certain amount of force is required to operate a mechanical switch it is common to use micro-switches.

5.Pneumatic sensor:

- These proximity sensors operate by breaking or disturbing an air flow.
- The pneumatic proximity sensor is an example of a contact type sensor. These cannot be used where light components may be blown away.

6.Optical sensor:

- In there simplest form, optical proximity sensors operate by breaking a light beam which falls onto a light sensitive device such as a photocell.
- These are examples of non contact sensors. Care must be exercised with the lighting environment of these sensors for example optical sensors can be blinded by flashes from arc welding processes, airborne dust and smoke clouds may impede light transmission etc.

7.Speed Sensor:

- Sensor used for detecting the speed of any object or vehicle which is in motion is known as speed sensor .For example – Wind Speed Sensors, Speedometer ,UDAR ,Ground Speed Radar .

8.Temperature Sensor:

- Devices which monitors and tracks the temperature and gives temperature's measurement as an electrical signal are termed as temperature sensors .
- These electrical signals will be in the form of voltage and is directly proportional to the temperature measurement .

9.PIR Sensor:

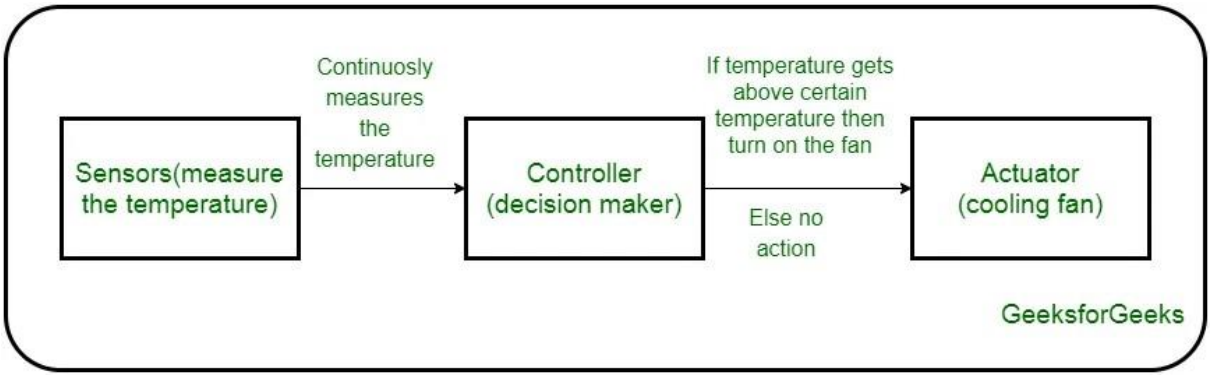
- PIR stands for passive infrared sensor and it is an electronic sensor that is used for the tracking and measurement of infrared (IR) light radiating from objects in its field of view and is also known as Pyroelectric sensor .
- It is mainly used for detecting human motion and movement detection .

10.Ultrasonic Sensor:

The principle of ultrasonic sensor is similar to the working principle of SONAR or RADAR in which the interpretation of echoes from radio or sound waves to evaluate the attributes of a target by generating the high frequency sound waves .

Actuators in IoT

- 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



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.

What is an Embedded System?

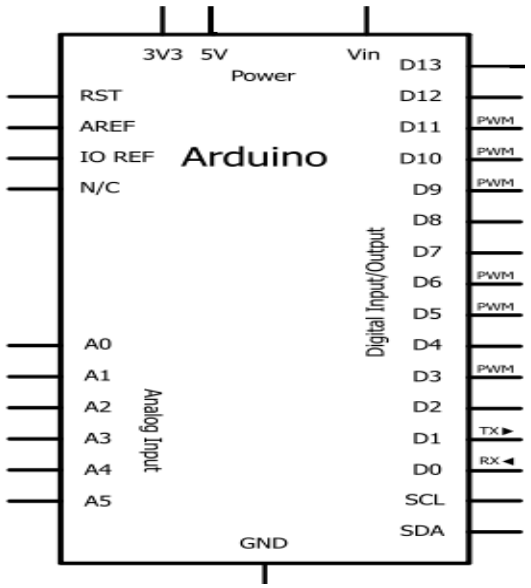
• An embedded system is a combination of hardware and software elements which are designed for a specific purpose. It can either be an independent system or a part of a large system.

Single Board Computer

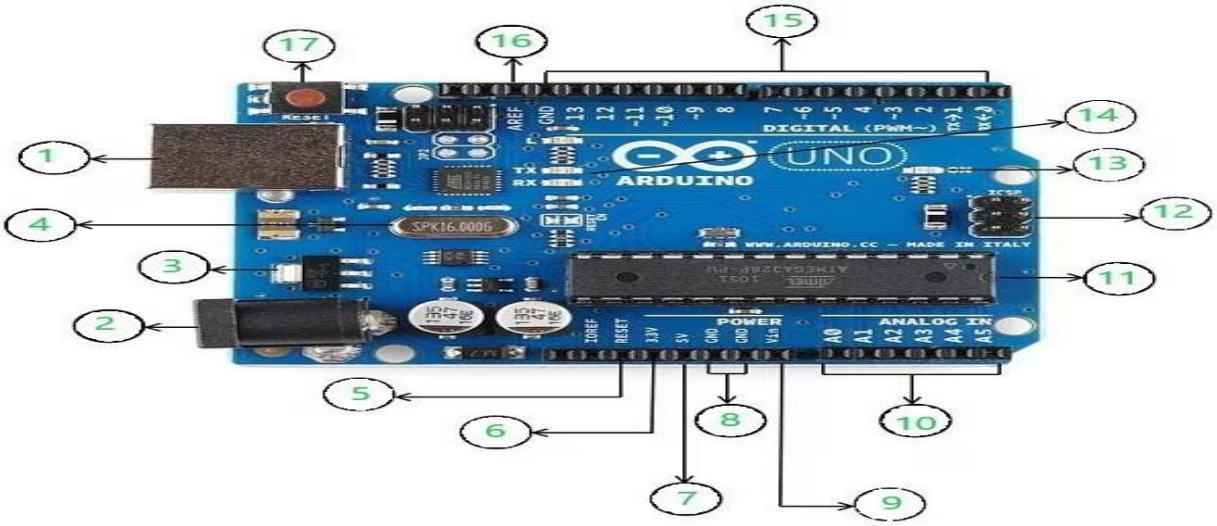
- A single-board computer (SBC) is a complete computer built on a single circuit board, with microprocessor(s), memory, input/output (I/O) and other features required of a functional computer.
- Single-board computers are commonly made as demonstration or development systems, for educational systems, or for use as embedded computer controllers.
- Many types of home computers or portable computers integrate all their functions onto a single printed circuit board.

Arduino

- Arduino is an open-source electronics platform based on easy-to-use hardware and software.
- Arduino boards are able to read inputs – light on a sensor, a finger on a button, or a Twitter message – and turn it into an output – activating a motor, turning on an LED, publishing something online.
- You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.



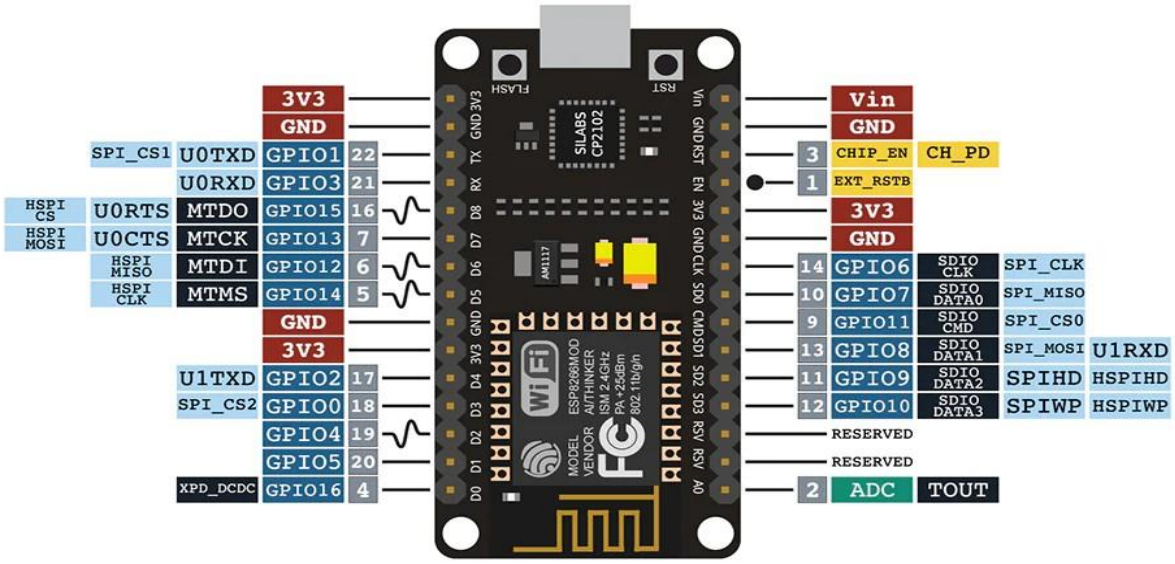
Arduino UNO



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

NodeMCU

- NodeMCU Development board is featured with wifi capability, analog pin, digital pins, and serial communication protocols.
- The NodeMCU ESP8266 development board comes with the ESP-12E module containing the ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor.
- This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs.
- Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects.
- NodeMCU can be powered using a Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.

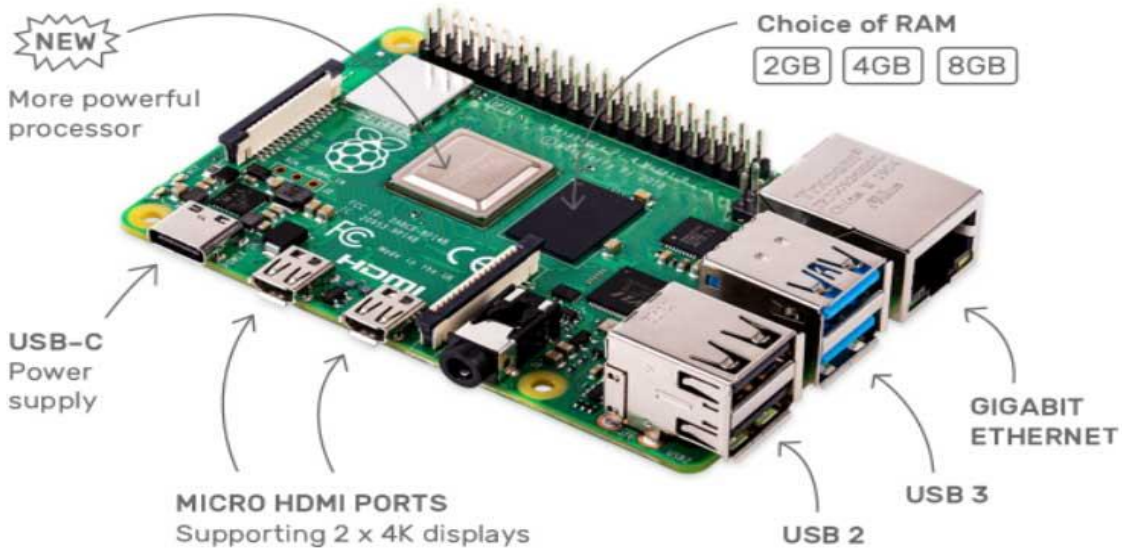


- POWER PIN: The power pins consist of one VIN pin and three 3.3V pins.
- I2C pins: These are used to connect all of your project's I2C sensors and peripherals. I2C Master and Slave are both supported.
- GPIO PINS: The ESP8266 NodeMCU includes 17 GPIO pins that may be programmatically assigned to different tasks like I2C, I2S, UART, PWM, IR Remote Control, LED Light, and Button.
- GROUND: It is a ground pin of the ESP8266 NodeMCU development board.
- ADC CHANNEL: analog to digital converter.The NodeMCU has a 10-bit precision SAR ADC built-in.
- UART PINS:universal asynchronous receiver-transmitter, is one of the most used device-to-device communication protocols
- SPI PINS: A Serial Peripheral Interface (SPI) bus is a system for serial communication, which uses up to four conductors, commonly three
- PWM PINS: There are four channels of Pulse Width Modulation on the PCB (PWM). The PWM output can be programmatically implemented and utilized to drive digital motors and LEDs.
- CONTROL PINS: are used to control the ESP8266 microcontroller. Chip's Enable pin (EN), Reset pin (RST), and WAKE pin is among these pins.

Raspberry pi

- The Raspberry Pi is a fully integrated computer (palmtop) mounted on a circuit board measuring approximately 7 cm x 5.5cm.
- It is a small, capable device that enables people of all ages to scan a computer and learn to edit in languages such as Scratch and Python.
- It can do everything you would expect a desktop computer to do, from browsing online and playing high-definition video, creating spreadsheets, word processing, and playing games.

- The Raspberry Pi has the ability to interact with the outside world and has been used in many digital



maker projects, from music machines and parental finders to weather stations and tweeting birdhouses with infra-red cameras.

- The Raspberry Pi has a Broadcom BCM2835system on chip (SoC), which includes the ARM1176JZF-S 700 MHz processor, which was later upgraded to Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC 1.5GHz.
- Originally shipped with 256 megabytes of RAM, later upgraded to 4GB.
- Does not include a built-in hard disk, but uses an SD card for boot and long-term storage.
- OS Support: Linux-based (Fedora, Raspbian, Debian, ArchLinux ARM, etc.

Function of GPIO pins in raspberry pi

GPIO stands for **General Purpose Input Output**

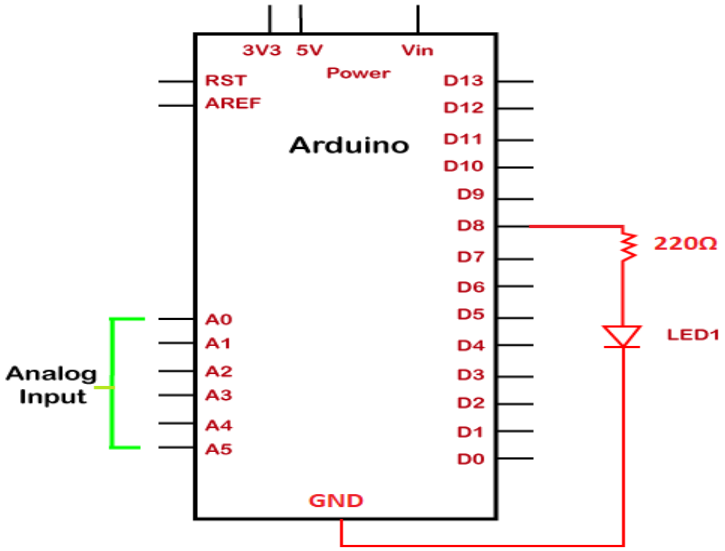
The GPIO pins allow **the Raspberry Pi to control and monitor the outside world by being connected to electronic** circuits

LED interfacing with Arduino

Components of the project

1. ArduinoUNO
2. 1 x Breadboard
3. 2 x Jump wires
4. 1 x LED
5. 1 x Resistor of 220 Ohm.

Structure of the project



We need to install the Arduino IDE, to begin with the coding, which is already discussed.

Open the IDE and start with the coding, which is given below:

```
1. void setup ()
2. {
3. pinMode ( 8, OUTPUT);  .
4. }
5. void loop ()
6. {
7. digitalWrite (8, HIGH);
8. delay(1000);
9. digitalWrite (8, LOW);
10. delay(500);
11. }
```

Procedure

- Attach an LED on the breadboard. We need to plug-in the two terminals of an LED into the holes of the breadboard.
- Connect the resistor in series with the LED
- Connect the left leg of the resistor (connected in series with red LED) to the digital output pin of the UNO board, i.e., PIN **8**.
- Connect the negative/shorter terminal (Cathode) of the LED to the GND pin of the UNO board using the wire.
- Write the program code on Arduino IDE.and execute the code and take the output

IR Sensor interfacing with Arduino

components

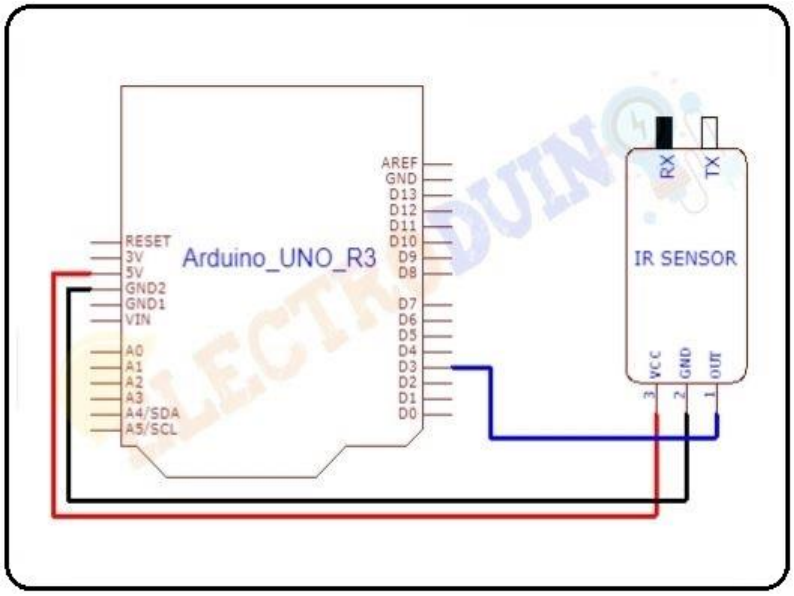
IR Sensor

LED and Resistor Kit

Breadboard

Jumper wires

USB cable type A/B



Program

```
int IRSensor = 3;
int LED = 11;
void setup()
{
pinMode (IRSensor, INPUT);
pinMode (LED, OUTPUT);
Serial.begin (9600); communication
}
void loop()
{
 int Sensordata = digitalRead (IRSensor);
Serial.print("Sensor value:");
Serial.println(Sensordata);
 if (Sensordata == 0)
 {
digitalWrite(LED, HIGH);
}
 else  {
digitalWrite(LED, LOW); // LED LOW
```

}}

Procedure

- Attach an IR Sensor,LED on the breadboard.
- We need to plug-in the 3 terminals of IR Sensor(VCC,GND,OUT)

  and 2 terminals of LED with 1k resistor into the holes of the

  breadboard.

- Connect the OUT pin of IR into digital output pin of the arduino

  UNO board, i.e., PIN 3.

- Connect the left leg of the resistor (connected in series with red

  LED) to the digital output pin of the UNO board, i.e., PIN 11.

- Connect the VCC of IR into 5v Vin pin of the UNO board.
- Connect the GND pin of IR into GND pin of the UNO board.
- Write the program code on Aurdino IDE.and execute the code and take the output

Raspberry Pi IR Sensor Interface

Now that we have seen a little bit about the IR Sensor Module and its connections, we will proceed with interfacing IR Sensor with Raspberry Pi.

The Raspberry Pi IR Sensor Interface can be converted into a Proximity Detector, where the application will detect if the object is too close to the sensor.

Components Required

- Raspberry Pi 3 Model B
- IR Sensor
- 5V Buzzer
- Mini Breadboard
- Connecting Wires
- Power Supply
- Computer

The IR Sensor Module has only three Pins: VCC, GND and Data. Connect the VCC and GND pins of the IR Sensor to +5V and GND pins of the Raspberry Pi.

Then connect the Data pin of the IR Sensor to GPIO23 i.e. Physical Pin 16 of the Raspberry Pi.

In order to indicate the alarm, I have used a simple 5V Buzzer. Connect one terminal of the buzzer to GND of Raspberry Pi and the other terminal (usually marked +) to GPIO24 i.e. Physical Pin 18 of Raspberry Pi.

Code

```
import RPi.GPIO as GPIO

import time

sensor = 16

buzzer = 18

GPIO.setmode(GPIO.BOARD)

GPIO.setup(sensor,GPIO.IN)

GPIO.setup(buzzer,GPIO.OUT)

GPIO.output(buzzer,False)

print "IR Sensor Ready....."

print " "

try:

   while True:

      if GPIO.input(sensor):

          GPIO.output(buzzer,True)

          print "Object Detected"

          while GPIO.input(sensor):

              time.sleep(0.2)

      else:

          GPIO.output(buzzer,False)

except KeyboardInterrupt:

    GPIO.cleanup()
```

Python

Python is a popular programming language

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Built-in Data Types

In programming, data type is an important concept.

Variables can store data of different types, and different types can do different things.

Python has the following data types built-in by default, in these categories:

Text Type:	str
Numeric Types:	int, float, complex
Sequence Types:	list, tuple, range
Mapping Type:	dict
Set Types:	set, frozenset
Boolean Type:	bool
Binary Types:	bytes, bytearray, memoryview
None Type:	NoneType

python Loops

Python has two primitive loop commands:

- while loops
- for loops

while Loop

With the while loop we can execute a set of statements as long as a condition is true.

```
while condition:
    # body of while loop
```

For Loops

A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).

This is less like the for keyword in other programming languages, and works more like an iterator method as found in other object-orientated programming languages.

With the for loop we can execute a set of statements, once for each item in a list, tuple, set etc

```
for val in sequence:
    # statement(s)
```

PYTHON FUNCTIONS

A function is a block of code which only runs when it is called.

You can pass data, known as parameters, into a function.

A function can return data as a result.

In Python a function is defined using the def keyword

```
def my_function():
    print("Hello from a function")
```

Calling a Function

To call a function, use the function name followed by parenthesis:

```
def my_function():
    print("Hello from a function")
my_function()
```

Arguments

Information can be passed into functions as arguments.

Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.

The following example has a function with one argument (fname). When the function is called, we pass along a first name, which is used inside the function to print the full name:

```
def myfunction(fname):
  print(fname)

myfunction("Emil")
myfunction("Tobias")
myfunction("Linus")
```

Number of Arguments

By default, a function must be called with the correct number of arguments. Meaning that if your function expects 2 arguments, you have to call the function with 2 arguments, not more, and not less.

```
def my_function(fname, lname):
    print(fname + " " + lname)

my_function("Emil", "Refsnes")
```

Constructors in Python

- Constructors are generally used for instantiating an object.
- The task of constructors is to initialize(assign values) to the data members of the class when an object of the class is created.
- In Python the __init__() method is called the constructor and is always called when an object is created.

Syntax of constructor declaration :

```
Def __init__(self):
    # body of the constructor
```

Types of constructors :

- **Default constructor:** The default constructor is a simple constructor which doesn't accept any arguments. Its definition has only one argument which is a reference to the instance being constructed.
- **Parameterized constructor**: constructor with parameters is known as parameterized constructor. The parameterized constructor takes its first argument as a reference to the instance being constructed known as self and the rest of the arguments are provided by the programmer.

Advantages of using constructors in Python:

- **Initialization of objects**: Constructors are used to initialize the objects of a class. They allow you to set default values for attributes or properties, and also allow you to initialize the object with custom data.
- **Easy to implement**: Constructors are easy to implement in Python, and can be defined using the __init__() method.
- **Better readability:** Constructors improve the readability of the code by making it clear what values are being initialized and how they are being initialized.
- **Encapsulation**: Constructors can be used to enforce encapsulation, by ensuring that the object's attributes are initialized correctly and in a controlled manner.

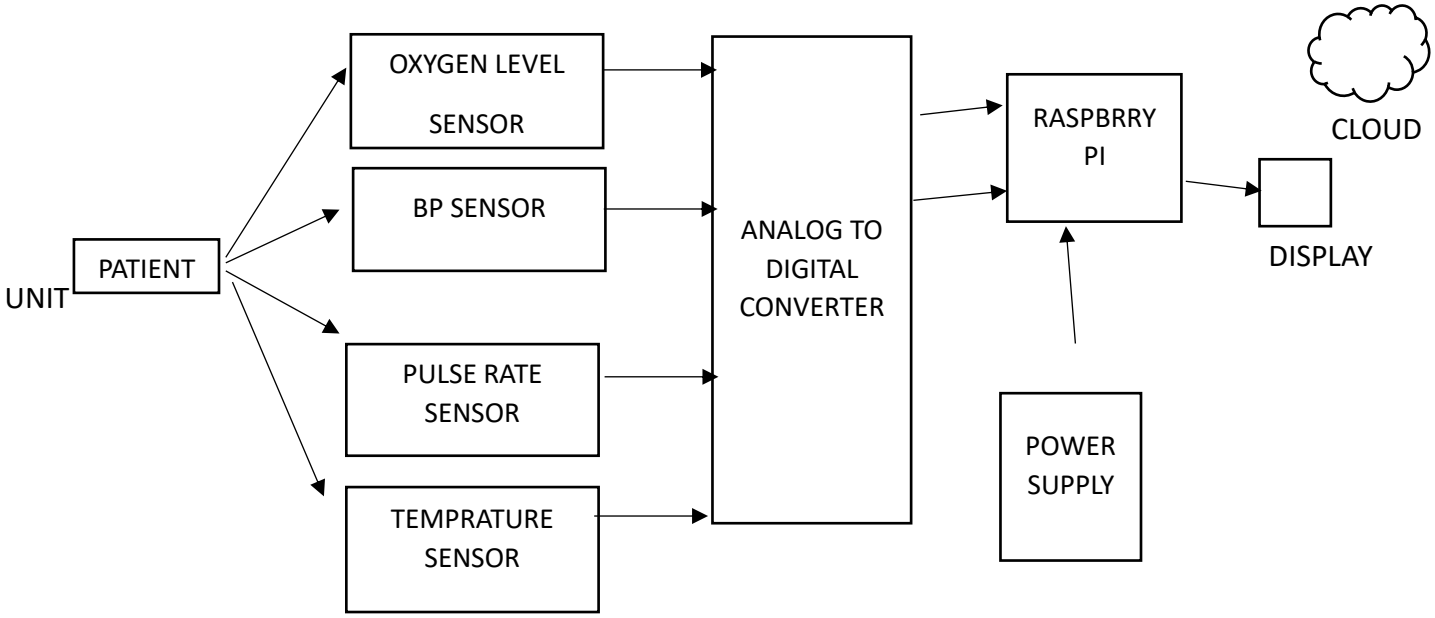
IOT APPLICATIONS

Smart Transportation:

Smart transportation uses IoT technology to optimize traffic flow, reduce congestion, and enhance road safety. IoT sensors collect data on traffic conditions, and real-time analysis allows for dynamic traffic management. IoT-enabled traffic lights adjust signal timings based on traffic flow, while connected vehicles feature advanced safety systems.

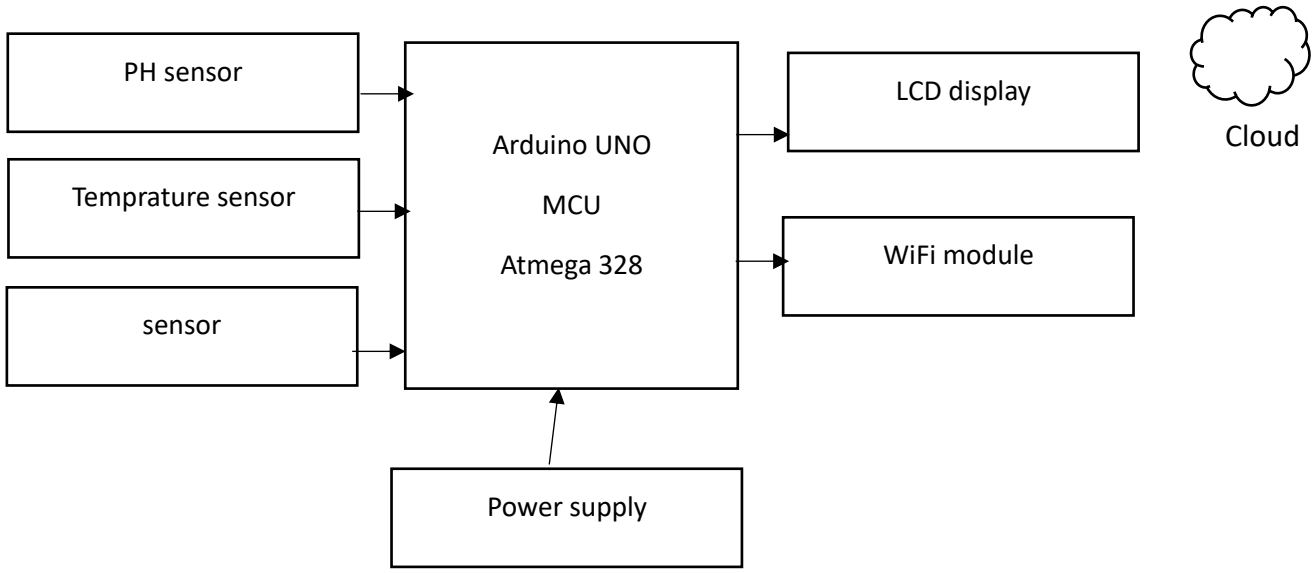
Smart Healthcare:

IoT in healthcare enables remote patient monitoring and efficient medical equipment management. Wearable devices track vital signs and send data to healthcare providers, allowing continuous monitoring. IoT sensors keep track of the condition and location of medical equipment, improving patient care and health outcomes.



Smart Water through IoT:

IoT sensors are deployed in water systems to monitor water quality, detect leaks, and manage resources efficiently. These sensors ensure safe water for consumption and environmental sustainability. They also enable rapid responses to water system issues.



Smart Warehouse Monitoring:

In logistics and supply chain management, IoT optimizes warehouse operations and inventory management. IoT sensors provide real-time data on inventory and storage conditions, reducing costs and improving efficiency.

Smart Retail :

IoT transforms retail with real-time inventory management, personalized shopping experiences, and automated checkout. RFID tags enable inventory monitoring, beacons offer personalized recommendations, and automated checkout systems streamline the shopping process, enhancing customer satisfaction.

Smart Driver Assistance System:

IoT-based systems integrated into vehicles improve road safety and convenience. They use sensors, cameras, and radar to provide features like lane departure warnings and adaptive cruise control, reducing accidents and enhancing driving safety.