GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING

(AUTONOMOUS)

Madhurawada, Visakhapatnam-530048

# INTERNET OF THINGS LAB

6th semester

Submitted by-

M.Devi Pallavi

21131A05E3

# GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING

## (AUTONOMOUS)

## Madhurawada, Visakhapatnam-530048

COLLEGE OF ENGINEERING
(Autonomous)

# CERTIFICATE

**Certified that this is a bonafide record of practical work done**
**By Mummidivarapu Devi Pallavi Roll No. 21131A05E3**
**Of B.Tech 6 th Semester in the Internet Of Things Lab,**

**In the Department of** **Computer science and engineering .**

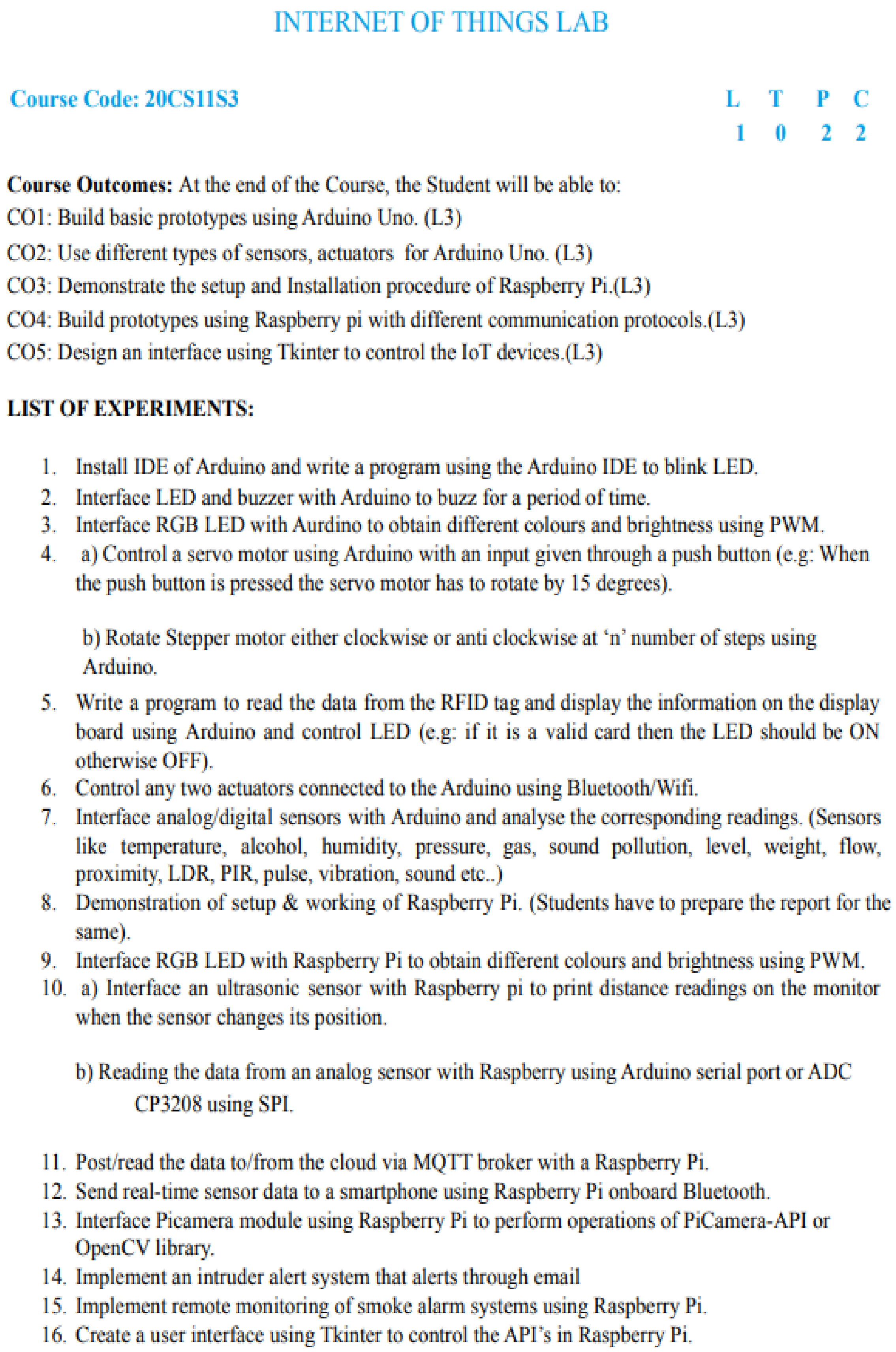
**During the academic year** **2023-24 .**

**No.of Experiments done:12**

**Signature of Faculty**

**Signature of Internal Examiner:**

**Signature of External Examiner:**



# INTERNET OF THINGS LAB

**Course Code: 20CS11S3**

| L | T | P | C |
|---|---|---|---|
| 1 | 0 | 2 | 2 |

**Course Outcomes:** At the end of the Course, the Student will be able to:

CO1: Build basic prototypes using Arduino Uno. (L3)

CO2: Use different types of sensors, actuators for Arduino Uno. (L3)

CO3: Demonstrate the setup and Installation procedure of Raspberry Pi.(L3)

CO4: Build prototypes using Raspberry pi with different communication protocols.(L3)

CO5: Design an interface using Tkinter to control the IoT devices.(L3)

**LIST OF EXPERIMENTS:**

1. Install IDE of Arduino and write a program using the Arduino IDE to blink LED.
2. Interface LED and buzzer with Arduino to buzz for a period of time.
3. Interface RGB LED with Aurdino to obtain different colours and brightness using PWM.
4. a) Control a servo motor using Arduino with an input given through a push button (e.g: When the push button is pressed the servo motor has to rotate by 15 degrees).

   b) Rotate Stepper motor either clockwise or anti clockwise at 'n' number of steps using Arduino.
5. Write a program to read the data from the RFID tag and display the information on the display board using Arduino and control LED (e.g: if it is a valid card then the LED should be ON otherwise OFF).
6. Control any two actuators connected to the Arduino using Bluetooth/Wifi.
7. Interface analog/digital sensors with Arduino and analyse the corresponding readings. (Sensors like temperature, alcohol, humidity, pressure, gas, sound pollution, level, weight, flow, proximity, LDR, PIR, pulse, vibration, sound etc..)
8. Demonstration of setup & working of Raspberry Pi. (Students have to prepare the report for the same).
9. Interface RGB LED with Raspberry Pi to obtain different colours and brightness using PWM.
10. a) Interface an ultrasonic sensor with Raspberry pi to print distance readings on the monitor when the sensor changes its position.

    b) Reading the data from an analog sensor with Raspberry using Arduino serial port or ADC CP3208 using SPI.
11. Post/read the data to/from the cloud via MQTT broker with a Raspberry Pi.
12. Send real-time sensor data to a smartphone using Raspberry Pi onboard Bluetooth.
13. Interface Picamera module using Raspberry Pi to perform operations of PiCamera-API or OpenCV library.
14. Implement an intruder alert system that alerts through email
15. Implement remote monitoring of smoke alarm systems using Raspberry Pi.
16. Create a user interface using Tkinter to control the API's in Raspberry Pi.

# INDEX

# IOT LAB RECORD

## WEEK-1

## Arduino Installation:

## [windows]

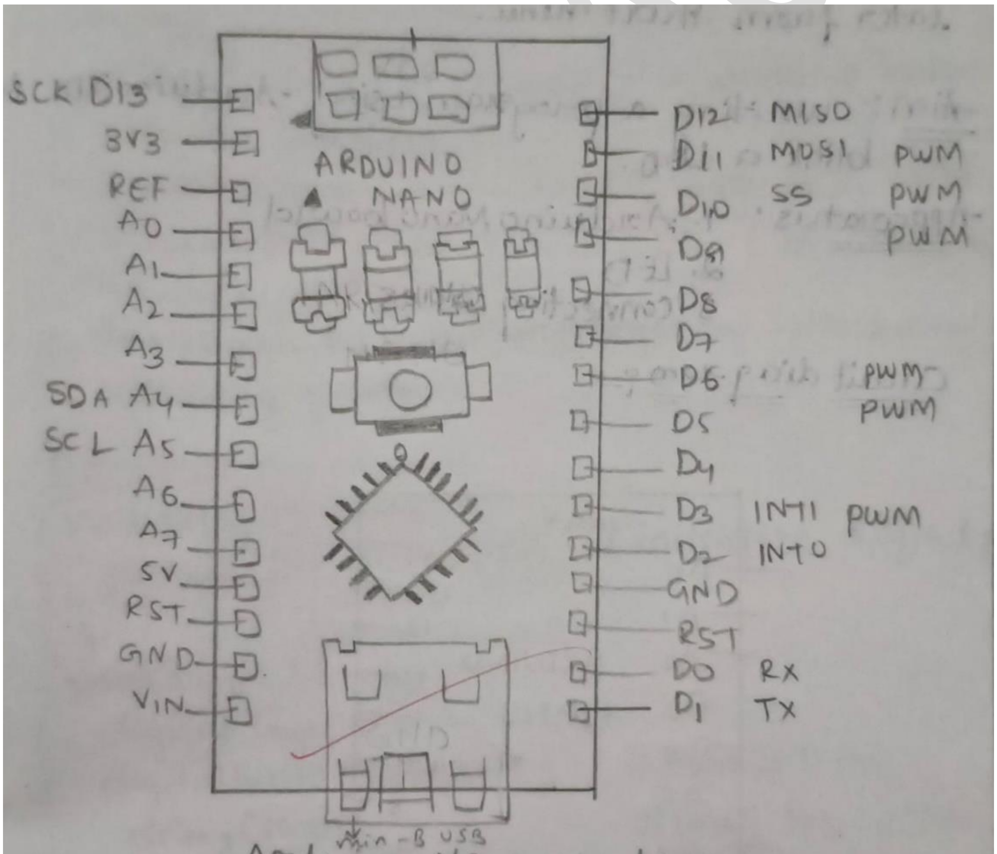
1.Downlaod the latest release of Arduino IDE

2.Double Click the executable file.

3.Follow the instructions in installation guide for installation process.

4.The process will extarct and install all required files to execute properly the arduino software.

5.When completing the setup leave Run Arduino IDE ticked to launch the application or launch it later from Start Menu



Arduino NANO pin diagram.

## 1.Aim:

To implement a Program using Arduino IDE to blink LED.

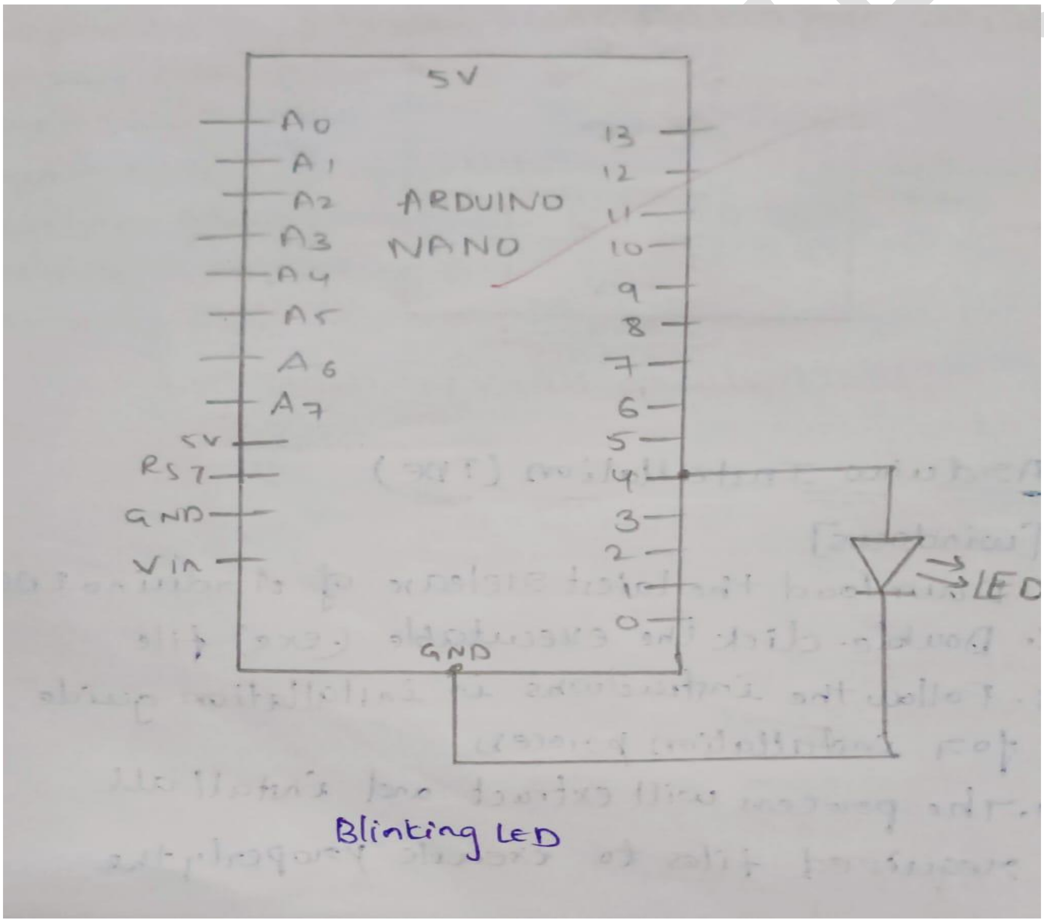
## Apparatus:

1) Arduino Nano Board

2) LED

3) Connecting Wires

## Circuit Diagram:



Blinking LED

## Procedure:

1)Connect LED within a Digital Pin of arduino nano board using connecting wires.

2)After Circuit is built,plug arduino board to the computer.

3)Start the Arduino software(IDE) and write the code.

4)After writing the code,click on verify button to compile the code.

5)Upload the code.

6)After Uploading the code,the LED on the board will light up.

## Program:

```
void setup(){
    pinMode(4,OUTPUT);
}
void loop(){
    digitalWrite(4,HIGH);
    delay(1000);
    digitalWrite(4,LOW);
    delay(1000);
}
```

**OUTPUT:**

LED turns off and on for every second i.e. repeatedly

**Applications:**

Indicator Lights

Safety Alarms

Notification systems

Control systems

## 2. Aim:

To implement a Program Blinking of 2 LED’s using Arduino IDE

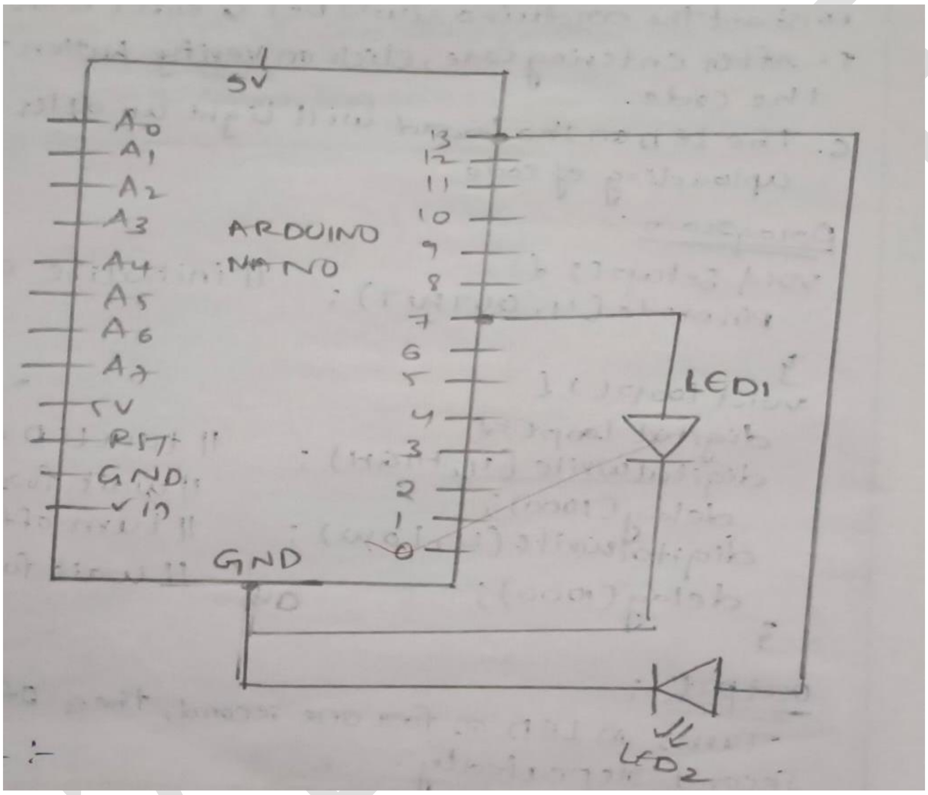
## Apparatus:

1)Arduino Nano Board

2)LED

3)Connecting Wires

## Circuit Diagram:



## Procedure:

1)For blinking 2 LED’s we use 2 digital pins of Arduino nano

2)Connect one LED  to pin 6 and other to pin 7 of arduino nano board using connecting wires.

3)After Circuit is built,plug arduino board to the computer.

3)Start the Arduino software(IDE) and write the code.

4)After writing the code,click on verify button to compile the code.

5)Upload the code.

6)After Uploading the code,the 2 LED’s on the board will blink based on the delay mentioned in the code.

## Program:

```
void setup(){
    pinMode(13,OUTPUT);
    pinMode(7,OUTPUT);
}
void loop(){
    digitalWrite(13,HIGH);
    digitalWrite(7,LOW);
    delay(1000);
    digitalWrite(13,LOW);
    digitalWrite(7,HIGH);
    delay(1000);
}
```

## OUTPUT:

LED connected to pin 6 and pin 7  turns off and on alternatively for every second repeatedly.

**Applications:**

Indicator Lights

Safety Alarms

Notification systems

Control systems

# WEEK-2

## 1.Aim:

Interface LED and buzzer with Arduino to buzz for same time.

## Apparatus:

1.Arduino nano board

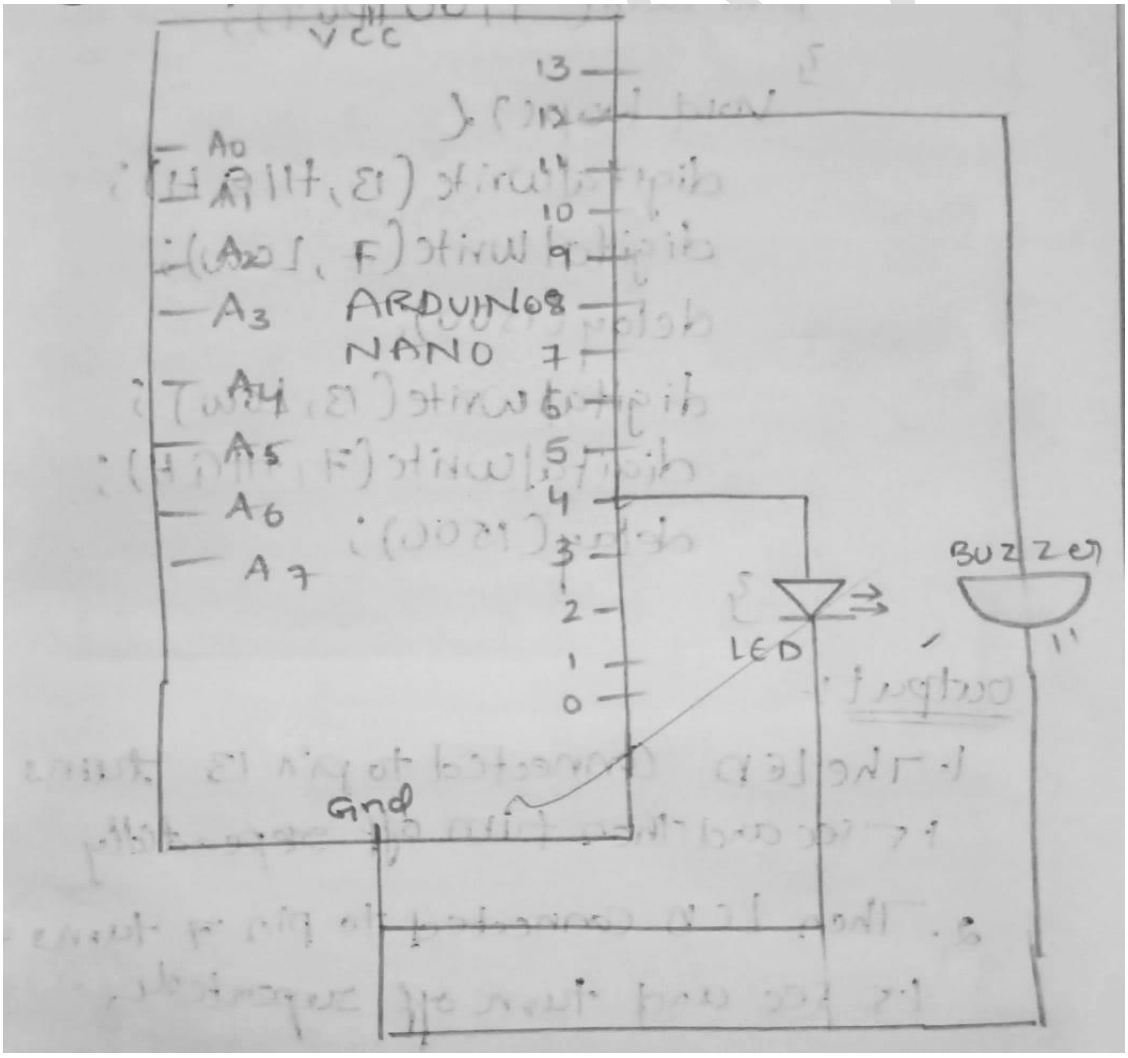
2.LED

3.Buzzer

4.Connecting wires

5.External power supply

## Circuit Diagram:



## Procedure:

1.connect LED to digital pin 4 and buzzer to digital pin 12 of Arduino Nano board with wires.

2.Now connect the power adapter to output connection of the Arduino board

3.plug Arduino board into the computer.

4.start Arduino software and enter the below code

5.After entering the code ,click on verify and compile.

6.led will blink at the same time buzzer also make sound.

## Program:

```
Void setup()
{
pinMode(4,OUTPUT);
pinMode(12,OUTPUT);
}
Void loop()
{
digitalWrite(4,HIGH);
digitalWrite(12,HIGH);
delay(1000);
digitalWrite(4,LOW);
digitalWrite(12,LOW);
delay(1000);
}
```

## OUTPUT:

Both the buzzer and LED works at same time

## 2.Aim:

Interface LED and buzzer with Arduino to buzz at different time.

## Apparatus:

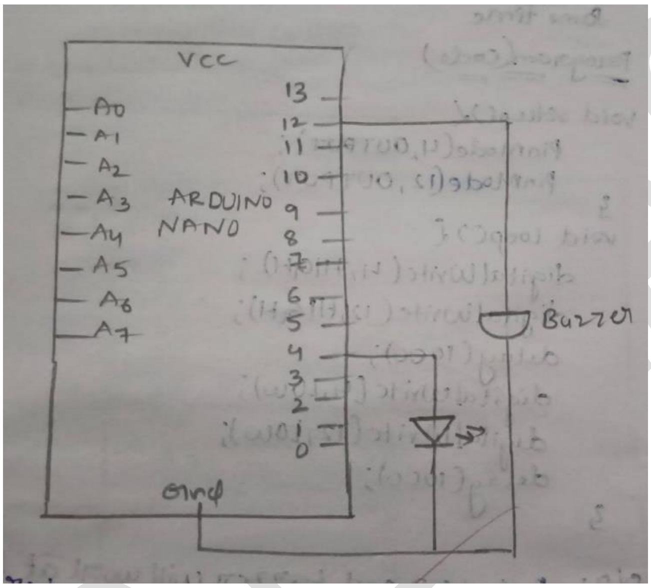
1.Arduino nano board

2.LED

3.Buzzer

4.Connecting wires

## Circuit Diagram:



## Procedure:

1.connect LED to digital pin4 and buzzer to digital pin 12 of Arduino Nano board with wires.

2.Now connect the power adapter to output connection of the Arduino board

3.plug Arduino board into the computer.

4.start Arduino software and enter the below code

5.After entering the code ,click on verify and compile.

6.led will blink at the same time buzzer also make sound.

## Program:

```
Void setup()
{
pinMode(4,OUTPUT);
pinMode(12,OUTPUT);
}
Void loop()
{
digitalWrite(4,HIGH);
digitalWrite(12,LOW);
delay(1000);
digitalWrite(4,LOW);
digitalWrite(12,HIGH);
delay(1000);
}
```

## OUTPUT:

If buzzer works LED OFF and if LED is ON the buzzer stop work

**Applications:**

a. Alarm systems

b. Timer/countdown indicators

c. Simple alert mechanisms

# WEEK-3

## 1.Aim:

To implement a program using arduino nano and make different colours with LED using RGB.

## Apparatus:

1.arduino board

2.4 connecting wires

3.LED

## Procedure:

1.Connect the 4 pins (Gr,R,G,B) of LED to arduino using single pinned 4 wires.

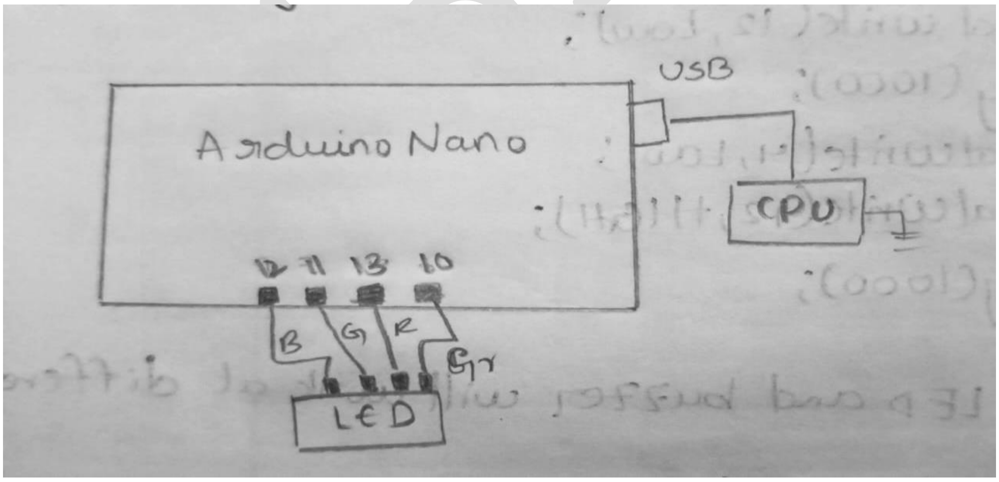
2.plug Arduino to computer.

3.start IDE and enter the code.

4.Verify it and compile the code.

5.upload the code to Arduino.

## Circuit Diagram:



## Program:

```
int red =13;

int green =11;
int blue = 12;
void setup()
{
        pinmode(red,OUTPUT);
         pinmode(green,OUTPUT);
         pinmode(blue,OUTPUT);
}
Void loop()
{
 setRGB(255,0,0);
delay(1000);
setRGB(0,255,0);
delay(1000);
setRGB(0,0,255);
delay(1000);
}
Void setRGB(int r ,int g,int b)
{
   analogWrite(red,r);
   analogWrite(green,g);
   analogWrite(blue,b);
}
```

**OUTPUT:**

We get red,green,blue colours from LED alternatively with a delay of 1sec

## 2.Aim:

Interface RGB LED with Arduino to obtain different colours and brightness using PWM.

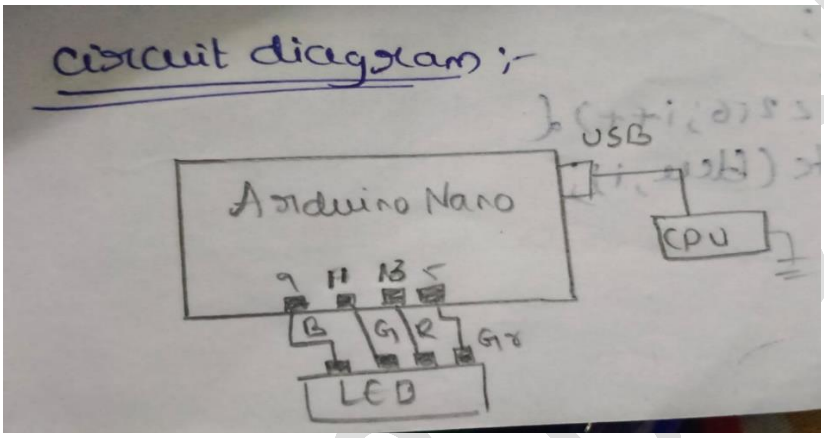
## Apparatus:

1.arduino board

2.4 connecting wires

3.LED

## Circuit Diagram:



## Procedure:

1.Connect the 4 pins (Gr,R,G,B) of LED to arduino using single pinned 4 wires.

2.plug Arduino to computer.

3.start IDE and enter the code.

4.Verify it and compile the code.

5.upload the code to Arduino.

## Program:

```
int red =9;

int green =11;

int blue = 13;

void setup()

{
  pinMode(red,OUTPUT);
  pinMode(green,OUTPUT);
  pinMode(blue,OUTPUT);
}
Void loop()
{
  for(int i = 0;i <= 255;i++)
  {
      analogWrite(red,i);
       delay(50);
   }
for(int i = 255;i >=0;i--)
  {
      analogWrite(red, i);
       delay(50);
   }
for(int i = 0;i <= 255;i++)
  {
      analogWrite(green, i);
       delay(50);
   }
for(int i = 255;i >=0;i--)
  {
      analogWrite(green , i);
       delay(50);
   }
```

```
for(int i = 0;i <= 255;i++)
  {
        analogWrite(blue,i);
         delay(50);
   }
for(int i = 255;i >=0;i--)
  {
        analogWrite(blue , i);
         delay(50);
   }
}
```

## OUTPUT:

Red,green,blue colors are the brightness increased to maximum and gradually decreased to minimum for all 3 colors.

**Applications:**

a. Ambient lighting systems

b. Status indicators

c. Simple data visualization

# WEEK-4

## 1.AIM:

Control a servo motor using Arduino nano with an input given through a push button.

## Apparatus:

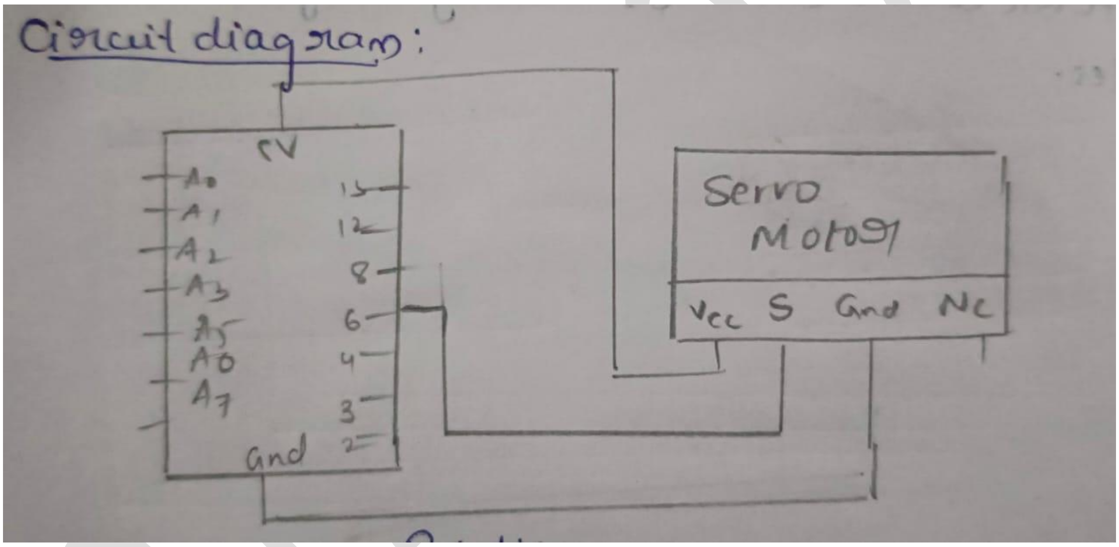
1.arduino nano

2.connecting wires

3.external power supply

4.servo motor

## Circuit Diagram:



## Procedure:

1.connect even pin numbers to servo motor

2.connect external power supply and switch on it.

3.write the code and compile it

4.upload the program to Arduino

5.observe the changes and rotations of the servo motor

## Program:

```
#include<Servo.h>

Servo s;

int p = 6;

void setup(){
     s.attach(p);

}

void loop(){

     for(int i =0;i<=180;i = i+10){

             s.write(i);

            delay(1000);

    }

    for(int i =180;i>=0;i = i-10){

             s.write(i);

            delay(1000);

    }

}
```

## Output:

We have seen that the servo motor rotates for 10 to 10 degrees upto 180 then again it rotates 180 to 0 with 10 degress in decrement

## Applications:

a. Robotics/animatronics

b. Automated mechanisms

c. Simple positioning systems

## 2.Aim:

Rotate stepper motor either clockwise or anticlockwise at n number of steps using Arduino.

## Apparatus:

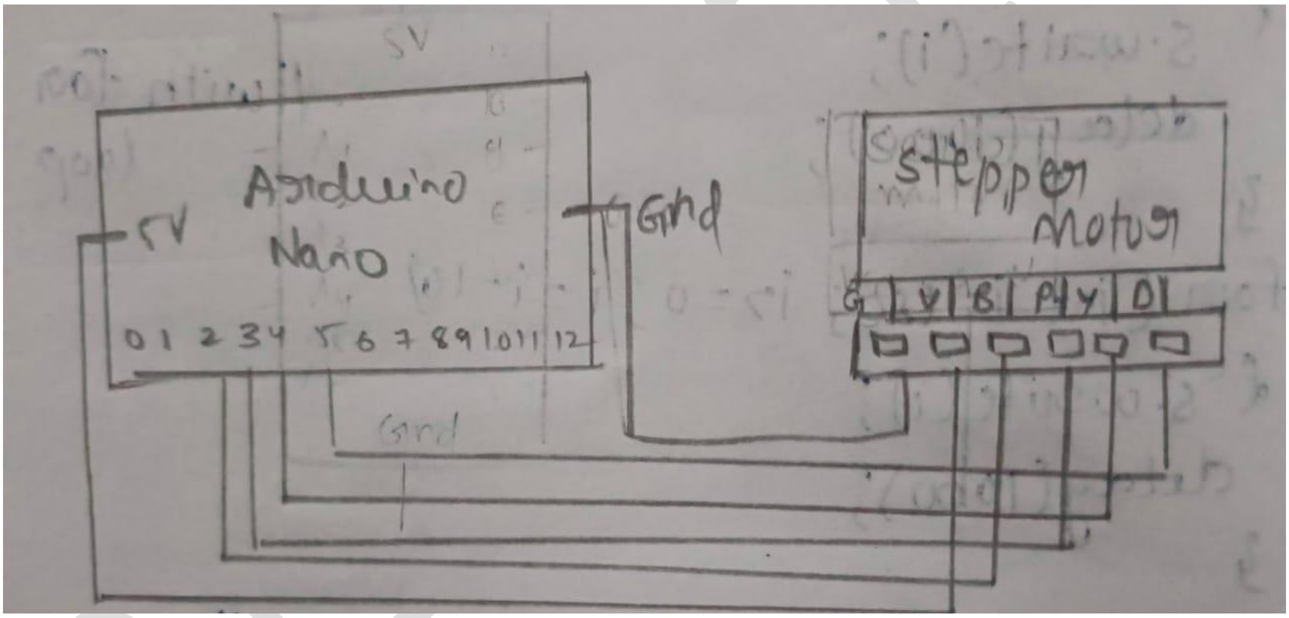
1.arduino nano

2.connecting wires

3.external power supply

4.servo motor

## Circuit Diagram:



## Procedure:

1.Connect 2,4,3,5 pins to stepper motor

2.connect external power suplly and sitch on

3.write code and verify

4.after completion of verify ,compile the code.

5.observe the revolutions of stepper

## Program:

```
#include<Stepper.h>
```

```
Stepper s(200,2,4,3,5);
void setup()
{
     s.setSpeed(60);
     Serial.begin(9600);
}
Void loop()
{
    Serial.println("clockwise");
    s.step(200);
    delay(1000);
    Serial.println("anticlockwise");
    s.step(-200);
    delay(1000);
}
```

## Output:

We seem that stepper motor rotates 360 according to our speed in clockwise direction and in anti clockwise direction

**Applications:**

a. Precision motion control

b. Robotics

c. Instrumentation

# WEEK-5

## Aim:

Interface analog/digital sensors with Arduino and analyse the corresponding readings. (Sensors like temperature, alcohol, humidity, pressure, gas, sound pollution, level, weight, flow, proximity, LDR, PIR, pulse, vibration, sound etc..)

## Apparatus:

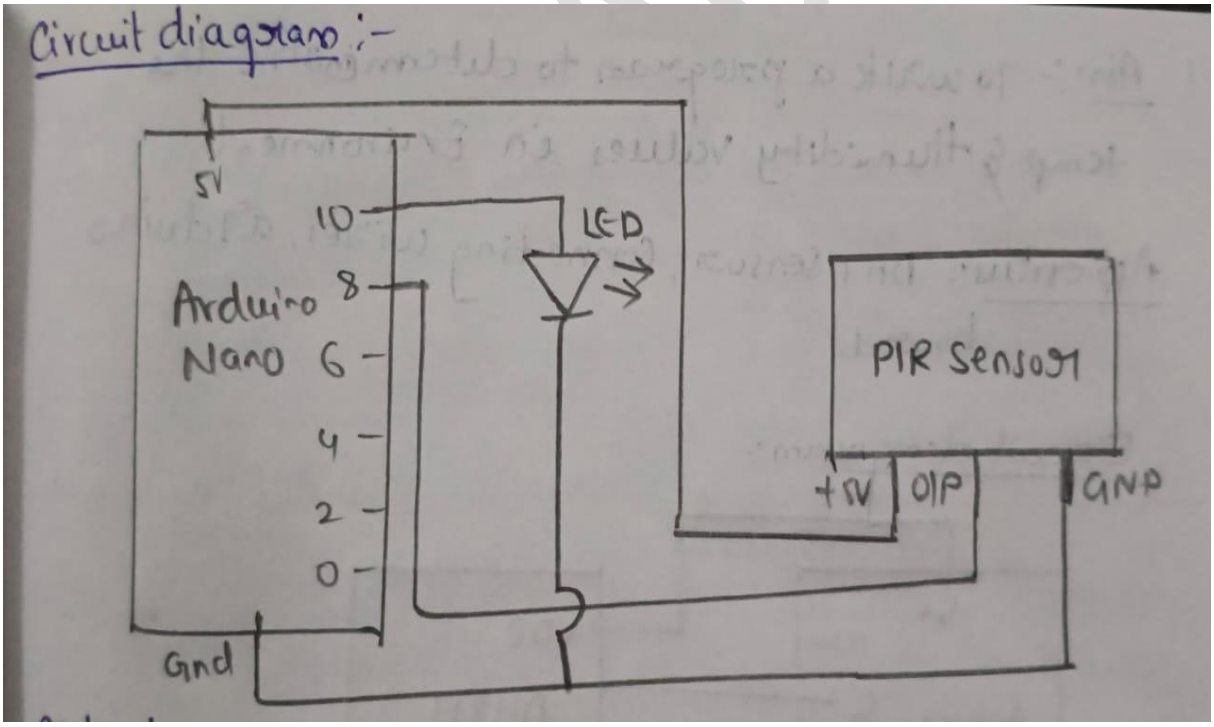
1) Arduino Nano Board

2) LED

3) Connecting Wires

4) PIR Sensor

## Circuit Diagram:



## Procedure:

1)Connect wires to PIR Sensor and Arduino nano board.

2)Connect wires to LED

3)After Circuit is built,plug arduino board to the computer.

4)Start the Arduino software(IDE) and write the code.

5)After writing the code,click on verify button to compile the code.

6)Upload the code to Arduino.

7)Observe the light and movement over PIR sensor.

## Applications:

1)Alarm systems

2)Automated doors

3)Digital Signatures

4)Lightening Control

5)Military Surveillance

6)Motion Detection

7)Thermostats

## Program:

```
void setup(){
    pinMode(9,INPUT);
    pinMode(10,OUTPUT);
    Serial.begin(9600);
}
void loop(){
   int i=digitalRead(9);
   if(i==HIGH){
      Serial.println(“Detected”);
     Serial.println(“millis()/1000);
     digitalWrite(10,HIGH);
     delay(1000);
```

```
 }
 else{
   Serial.println(“Not detected”);
   digitalWrite(10,LOW);
   delay(1000);
  }
 }
```

## OUTPUT:

When the motion is detected,in COM3 port it writes that motion is detected & LED turns “ON” and when the motion is not detected ,it writes “NOT DETECTED” and LED turns off.

```
10:45:45.360 ->NOT DETECTED
10:45:48.368 ->NOT DETECTED
10:47:01.400 ->NOT DETECTED
```

## Applications:

a. Environmental monitoring

b. Industrial process monitoring

c. Data logging systems

## 2.Aim:

To write a program to deterministic the temperature and humidity values in the environment.

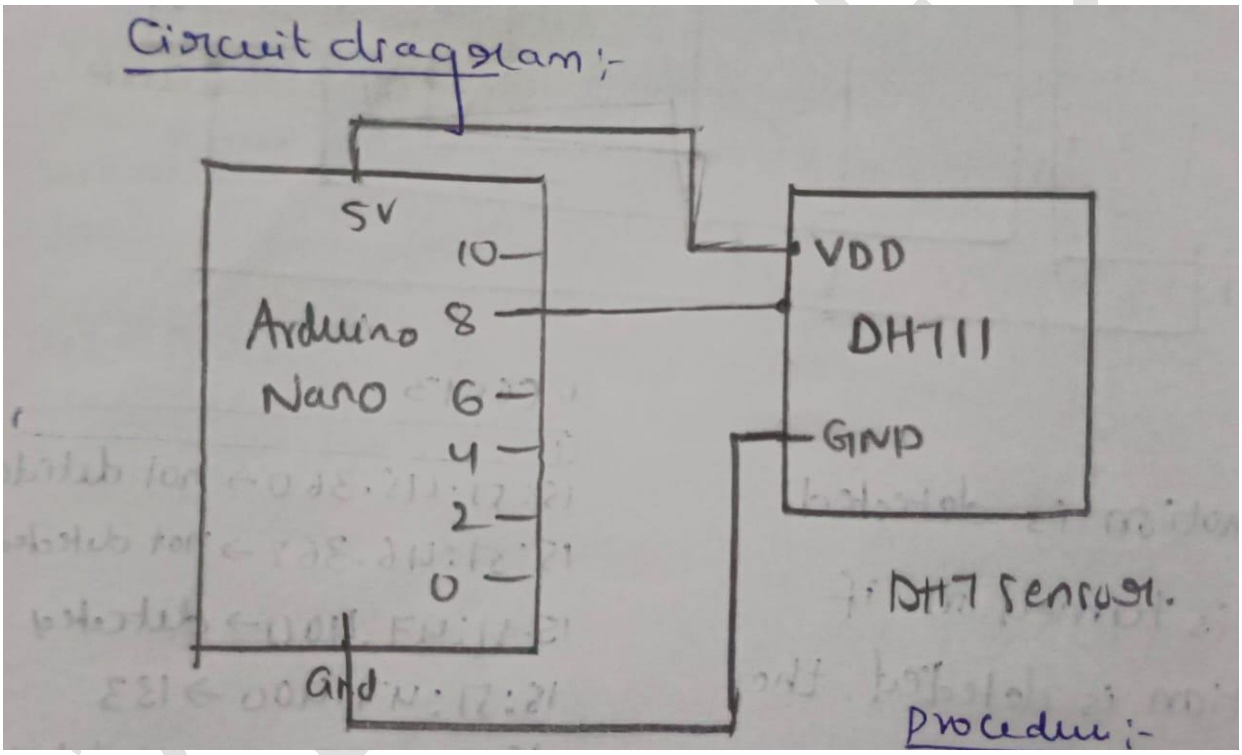
## Apparatus:

1) Arduino Nano Board

2) Connecting Wires

3) DHT Sensor

## Circuit Diagram:



## Procedure:

1)Connect wires to DHT Sensor and Arduino nano board.

2)Write code and compile it.

3)Upload the program to Arduino Nano.

4)Observe the humidity and temperature.

5)Observe the reading values.

## Program:

```
#include <DHT.h>
DHT dht(8, DHT11);
float h,t;
void setup() {
 Serial.begin(9600);
 dht.begin();
 Serial.println("Connect dht");
}

void loop(){
   h=dht.readHumidity();
   t=dht.readTemperature();
   if(isnan(h) || isnan(t)){
       Serial.println("Failed to read from DHT sensor");
       delay(2000);
   }
   else{
       Serial.print("Humidity: ");
       Serial.print(h);
       delay(2000);
       Serial.print("Temperature: ");
       Serial.print(t);
       delay(2000);
   }
}
```

## OUTPUT:

Temperature and Humidity level is detected in environment by DHT sensor.

```
Connect dth:

16:12:36.748->68.00

16.12.38:717->Temperature

16.12.38:717->25.10

16.12.38:717->Humidity

16.12.38:717->68.00
```

### Applications:

a. Environmental monitoring

b. Industrial process monitoring

c. Data logging systems

# Week-6

## 1.Aim:

Controlling actuators connected to Arduino using Bluetooth(Blinking of LED when message is sent)

## Apparatus:

1)Arduino Nano Board

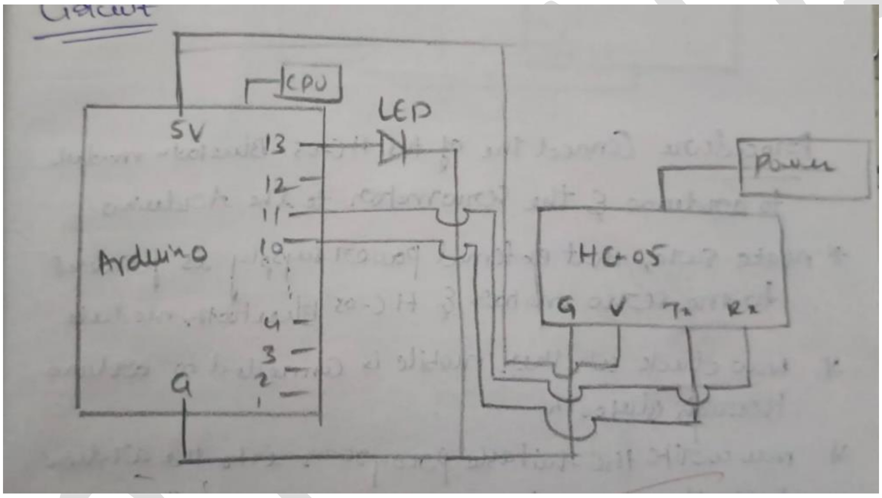
2)Bluetooth Terminal HC-05 App

3)LED

4)Jumper Wires

5)External Power Supply

## Circuit Diagram:



## Procedure:

1)Download Bluetooth connector app and open the connector app bluetooth terminal HC-05 and allow turning on Bluetooth of the device.

2)Search for Bluetooth device and pair device to HC-05 to connect with android app.

3)Connect arduino board pin 13 to the LED.

4)Join the USB cable to CPU of the computer and connect power adapter for external power supply.

5)Open Arduino IDE software and enter below code.

6)After successful compilation,upload the code.

## Program:

```
#include <Software Serial.h>

SoftwareSerial EEBlue(10,11);

void setup() {

 EEBlue.begin(9600);

 Serial.begin(9600);

 Serial.println("Bluetooth devices activated");

 pinMode(13,OUTPUT);

}

void loop(){

   if(EEBlue.available()){

       Serial.write(EEBlue.read());

       digitalWrite(13,HIGH);

   }

   if(Serial.available()){

       EEBlue.write(Serial.read());

   }

}
```

## OUTPUT:

When data is transmitted from the Bluetooth terminal to Arduino device,it turns on the LED and we get output as

->Bluetooth devices are activated

->Message or data that is sent

In system,we can sent information through COM3 Port. m

## 2.Aim:

Controlling Servomotor connected to Arduino using Bluetooth

## Apparatus:

1)Arduino Nano Board

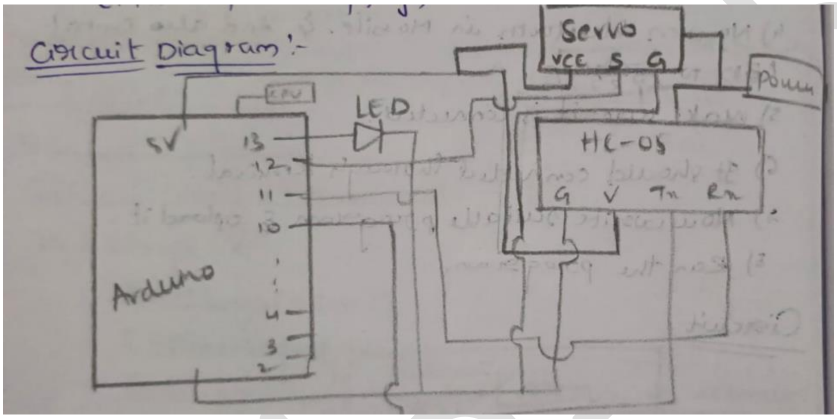
2)Servo Motor

3)Bluetooth terminal HC-05 App

4)Jumper wires

5)External Power Supply

## Circuit Diagram:



## Procedure:

1)Open the Bluetooth terminal HC-05 app and allow turning on Bluetooth of the device.

2)Search for devices and select the pair device for pairing of bluetooth.

3)Connect Servomotor pins to Arduino board pin 6.

4)Join the USB to CPU and connect power adapter for the external power supply.

5)Enter below code in  Arduino IDE software.

6)After successful compilation,upload the code.

## Code:

```
#include <Software Serial.h>
```

```
#include <Servo.h>
SoftwatrSerial EEBlue(10,11);
Servo s1;
int Servopin=6;
void setup() {
  s1.attach(Servopin);
  EEBlue.begin(9600);
  Serial.begin(9600);
  Serial.println("Bluetooth devices activated");
}
void loop(){
  if(EEBlue.parseInt()==1){
     s1.write(180);
  }
  delay(1000);
}
```

**OUTPUT:**

Whenever data is sent from Bluetooth Terminal to Arduino board,servo motor is rotated to 180° with the delay of 1000 msec and we get output a>Bluetooth gates open>Message that is sent

**Applications:**

a. Home/industrial automation

b. Remote control systems

c. Internet of Things (IoT) devices

# week-7

**1.Write a program to read the data from the RFID tag and display the information on the display board using Arduino and control LED (e.g: if it is a valid card then the LED should be ON otherwise OFF).**

**Aim:** A program to read the data from the RFID tag and display the information on the display board using Arduino and control LED

**Apparatus:**

1.Arduino nano board

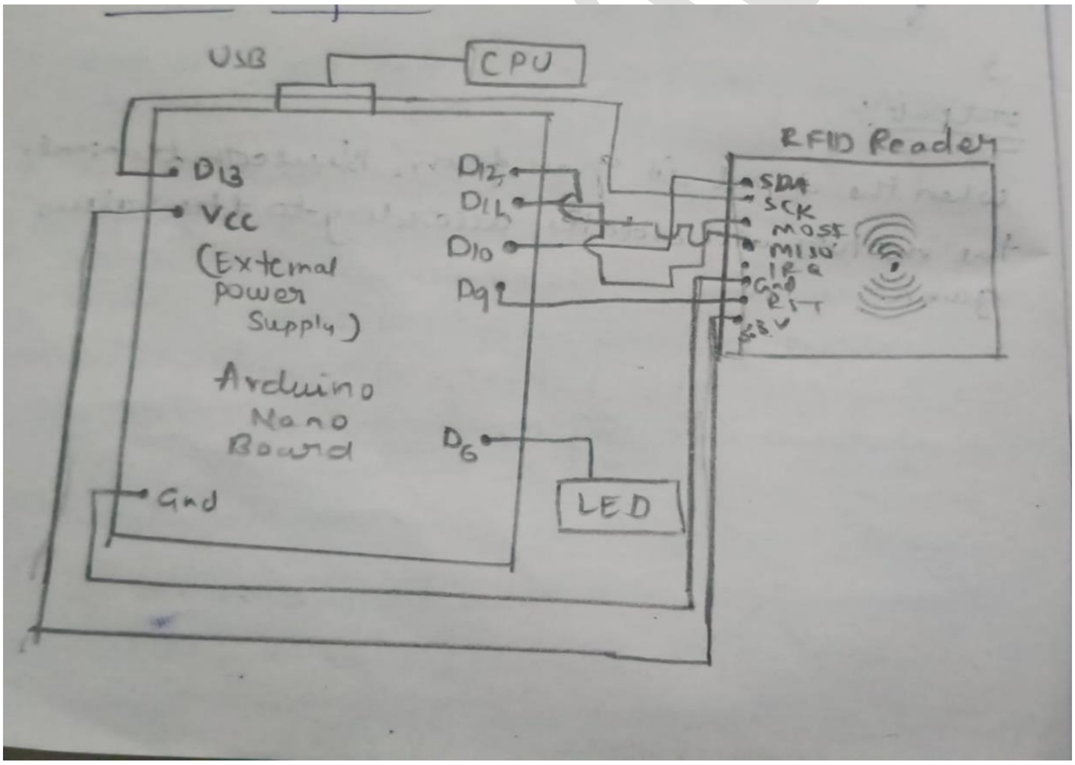
2.RFID readers,card and keychain

3.LED bulb

4.Jumper wires

5.External power supply

**Circuit diagram:**



**Description:**

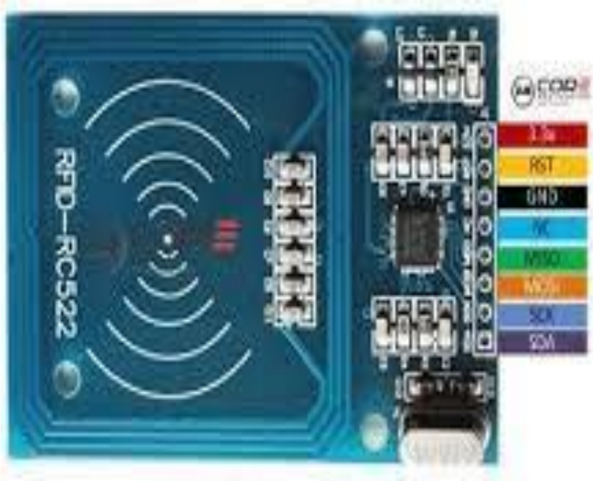
RFID (radio frequency identification) is a form of wireless communication that incorporates the use of electromagnetic or electrostatic coupling in the radio frequency portion of the electromagnetic spectrum to uniquely identify an object, animal or person.

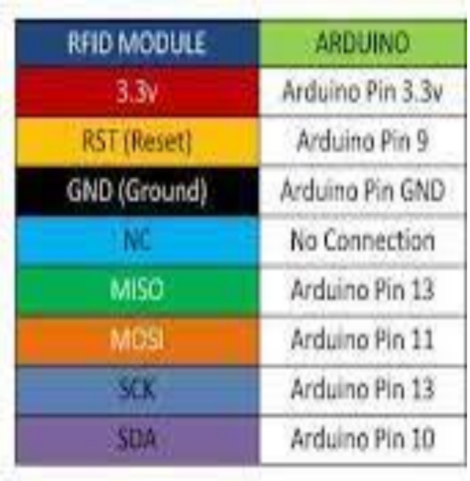
RFID tags are made up of an integrated circuit (IC), an antenna and a substrate. The part of an RFID tag that encodes identifying information is called the RFID inlay.

There are two main types of RFID tags:

- **Active RFID.** An active RFID tag has its own power source, often a battery.
- **Passive RFID.** A passive RFID tag receives its power from the reading antenna, whose electromagnetic wave induces a current in the RFID tag's antenna.

**Pin wiring:**





| RFID MODULE | ARDUINO |
|---|---|
| 3.3v | Arduino Pin 3.3v |
| RST (Reset) | Arduino Pin 9 |
| GND (Ground) | Arduino Pin GND |
| NC | No Connection |
| MISO | Arduino Pin 13 |
| MOSI | Arduino Pin 11 |
| SCK | Arduino Pin 13 |
| SDA | Arduino Pin 10 |

**Procedure:**

1) Connect the Circuit using Arduino nano board and RFID System based on above pin wiring

2) Connect USB cable to cpu and power adapter for external power supply.

3) After having circuit ready, Compiler and upload the code available in Arduino IDE

4)Then open serial monitor.

5)Approximate the RFID Card or keychain to the reader

6) Connect LED to the digital pin 6 of Arduino Nano Board

7) let the reader & tag closer until all Information of card is displayed.

**Program:**

```
#include <SPI.h>
#include <MFRC522.h>
#define RST_PIN 9
#define SS_PIN  10
MFRC522 mfrc522(SS_PIN, RST_PIN);
void setup() {
  Serial.begin(9600);
  while (!Serial);
  SPI.begin();
  mfrc522.PCD_Init();
  delay(4);
  Serial.println(F("Scan PICC to see UID, SAK, type, and data blocks..."));
}
void loop() {
  if ( ! mfrc522.PICC_IsNewCardPresent()) {
    return;
  }
  if ( ! mfrc522.PICC_ReadCardSerial()) {
    return;
  }
  mfrc522.PICC_DumpToSerial(&(mfrc522.uid));
}
```

**Output:**

```
10:17:01.459 -> Card UID: 43 AD FE 05

10:17:01.459 -> Card SAK: 08

10:17:01.506 -> PICC type: MIFARE 1KB

10:17:01.506 -> Sector Block   0  1  2  3   4  5  6  7   8  9 10 11  12 13 14 15
AccessBits

10:17:01.598 ->   15     63   00 00 00 00  00 00 FF 07  80 69 FF FF  FF FF FF FF  [ 0
0 1 ]

10:17:01.691 ->          62   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0
0 0 ]

10:17:01.783 ->          61   00 00 00 00  00 00 00 00  00 00 00 00  00 00  00 00 00
00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]

10:17:01.923 ->   14     59   00 00 00 00  00 00 FF 07  80 69 FF FF  FF FF FF FF  [ 0
0 1 ]

10:17:02.014 ->          58   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0
0 0 ]

10:17:02.096 ->          57   00 00 00 00  00 00 00 00  00 00
```

THe above is the information we can read from the card

The led will glow when the card is approximate to the rfid reader and turn off otherwise

## Applications:

a. Access control systems

b. Asset tracking

c. Attendance monitoring

## WEEK – 8

**Aim:** Demonstration of setup & working of Raspberry Pi. (Students have to prepare the report for the same)

**Setting up Raspberry Pi:**

Here are step-by-step instructions for setting up your Raspberry Pi:

**1. Open the Package and Check Components:**

- Open the Raspberry Pi package and ensure that all components are present. This typically includes the Raspberry Pi board, MicroSD card, power adapter, and any included accessories like a case.

**2. Organize Accessories:**

- Place the accessories such as the power adapter, MicroSD card, and case (if included) in an organized manner for easy access during setup.

**3. Insert MicroSD Card:**

- Insert the MicroSD card into the slot on the Raspberry Pi. Make sure it's the one preloaded with the operating system.

**4. Connect Peripherals:**

- Connect a keyboard, mouse, and monitor to the Raspberry Pi using USB and HDMI ports respectively. If your Raspberry Pi model doesn't have enough USB ports, consider using a USB hub.

**5. Power Up:**

- Plug in the power adapter to a power source and connect it to the Raspberry Pi. The power indicator LED on the Raspberry Pi should illuminate, indicating that it's receiving power.

**6. Initial Configuration:**

- Turn on the Raspberry Pi by pressing the power button or connecting the power supply. Follow the on-screen instructions to configure language, keyboard layout, Wi-Fi (if applicable), and any other initial settings.

**Working of Raspberry Pi:**

1.Explore graphical user interface for easy interaction.

2.we can write code in various languages for different projects.

3.Control external components through general purpose ,Input/Output pins.

4.Develop smart home devices and sensor network for IOT projects.

5.Utilize media center software for multimedia playback.

6.Host websites ,game servers and other network services.

7.Setup automated tasks using scripts and cross jobs.

8. Access your device remotely for management and management purpose**.**

**Applications:**

a. Educational tool for learning embedded systems

b. Prototyping IoT devices

c. Running small server applications

## Week-9

**Aim: Interface RGB LED with Raspberry PI to obtain different colours and brightness using PWM**

**Apparatus:**

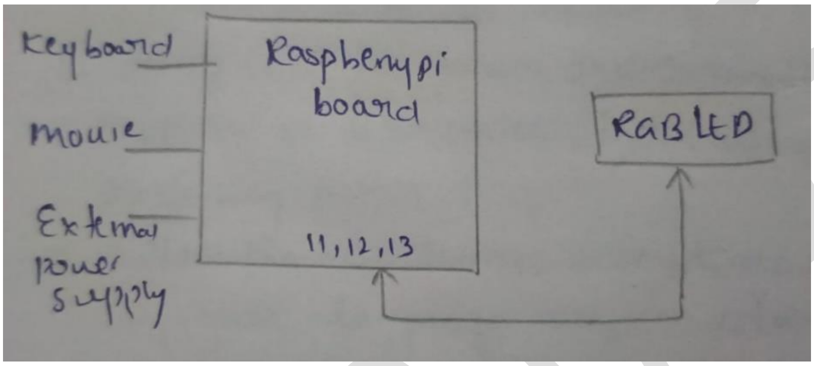
1)Raspberry PI

2)RGB LED

3)Connecting Probes

4)External Power Supply

**Circuit Diagram:**



**Procedure:**

1)Connect the RGB LED to pin numbers 11,12,13 to Raspberry Pi based board using connecting wires.

2)Plug mouse,keyboard to the Pi board.

3)Start Raspberry Pi software ,create newfile with python extension and start writing below code.

4) After entering the code,save the contents and click on 'RUN'

5)The RGB colors glow based on the written program.

**Code:**

```
import RPi.GPIO as g

from time import sleep

g.set warnings (False)

g.setmode (g.BCM)
```

```
red=11
green=12
blue =13
g.setup(red, g.OUT)
9.setup (green, g.OUT)
g setup(blue, g.OUT)
while True:
    g.output (red, g.HIGH)
    g.output (green, g. LOW)
    g.output (blue, g. LOW)
    sleep(2)
    g.output (red, g.LOW)
    g.output (green, g.HIGH)
    g.output (blue, g.LOW)
    sleep (2)
    g.output (red, g.LOW)
    g.output (green, g.LOW)
    g.output (blue, g.HIGH)
    sleep (2)
```

**Output:**

Based on the high value that was send to pin those corresponding color will be emitted out by RGB LED. Here we will observe the Green, Red, Blue Colours Simultaneously one after other.

**Applications:**

a. Smart lighting systems

b. Visual feedback indicators

c. Interactive art installations

# WEEK-10

## 1a.Aim:

To interface an ultrasonic sensor with Raspberry pi to print distance readings on the monitor when the sensor changes its position.

## Apparatus:

1) Raspberry pi-Kit

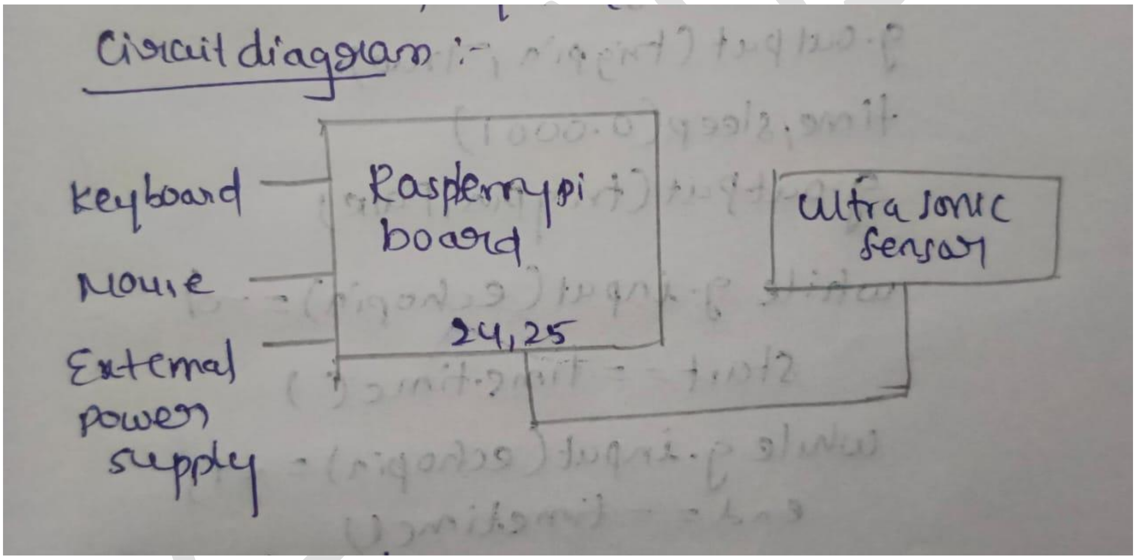
2) Ultrasonic Sensor

3) Connecting Wires

4)External Power Supply

5)System

## Circuit Diagram:



## Procedure:

1)Connect the ultrasonic sensor to pin nu:24,25 to pi-board using connecting wires.

2)Plug: mouse and keyboard to the pi-board.

3)Start the software ,create new file with python extension and start writing below code.

4)After entering the code, save the contents and click on ’RUN’.

5)The sensor will detects the monitor of the object and display the message on monitor.

## Program:

```
import RPi.GPIO as g

import time

g.setwarning(False)

g.setmode(GP.BCM)

trigpin=24

echopin=25

g.setup(trigpin,GP.OUT)

g.setup(echopin,GP.IN)

def distance(trigpin,echopin):

    g,output(trigpin,True)

    time.sleep(0.0001)

    g.output(echopin,False)

    while g.input(echopin)==0:

         start==time.time()

    while g.input(echopin)==1:

         end=time.time()

try:

    duration=end-start

except:

    print(‘Calibrating’)

    return -1

distance=duration*17150

distance=round(distance+1.15,2)
```

```
return distance
while True:
      dist=distance(trigpin,echopin)
      print('Measured distance=()cm'.formar(dist))
      time.sleep(1)
```

**OUTPUT:**

We can see the distance measured and the message in the shell.

```
Calibrating
Measured distance=-1cm
Calibrating
Measured distance=-1cm
Calibrating
Measured distance=-1cm
```

**Applications:**

a. Obstacle detection/avoidance

b. Level monitoring

c. Proximity sensing

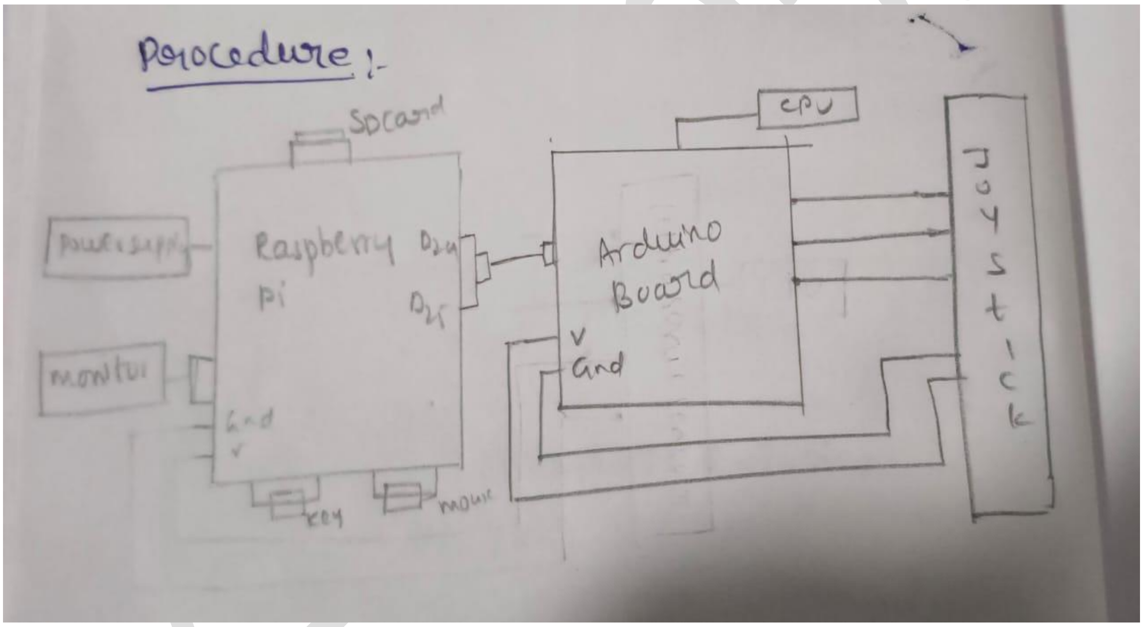
# Week-11

**Aim:** Reading the data from an analog sensor(joy stick) with Raspberry using Arduino serial port or ADC CP3208 using SPI.

**Apparatus:**

1. Arduino Board
2. Analog Sensors (Potentiometers)
3. Connecting Wires
4. USB Cable
5. Raspberry Pi
6. Monitor, Mouse, and Keyboard
7. Python Editor or IDE
8. Power Source

## Circuit:



**Procedure:**

1. Connect the Arduino to the Raspberry Pi using a USB cable.
2. Connect two analog sensors (e.g., potentiometers) to the Arduino's analog pins A0 and A1.
3. Start the Raspberry Pi and open a Python editor or IDE.
4. Create a new Python file with a **.py** extension.
5. Copy the provided Python code and paste it into the Python file.
6. Save the file with an appropriate name and the **.py** extension.
7. Execute the saved Python file on the Raspberry Pi.
8. Monitor the output displayed in the console.

9. The Arduino code reads analog values from two sensors connected to pins A0 and A1, maps these values to a range from -512 to 512, and sends them over serial communication.
10. The Python code running on the Raspberry Pi reads the data sent by the Arduino over the serial port (**/dev/ttyUSB0**), decodes it, and prints it to the console.
11. The output on the Raspberry Pi's console will show the X and Y positions received from the Arduino.

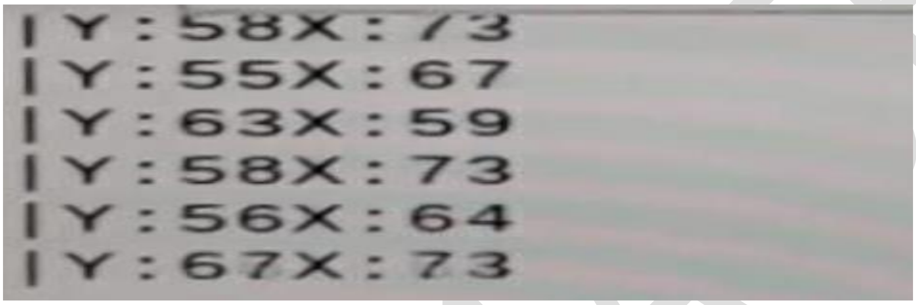
Make sure to adjust the serial port (**'/dev/ttyUSB0'**) in the Python code if your Arduino is connected to a different port. Also, ensure that the baud rate (**9600**) matches the baud rate used in the Arduino code. Additionally, there's a syntax error in the provided Python code (missing **=** sign after **960**). This should be corrected before running the code.

**Program:**

**### On Arduino**

```
int VRx = A0;

int VRy = A1;

int xposition = 0;

int yposition = 0;

int mapx = 0

int mapy = 0;

void setup() {

Serial.begin(9600);

pinMode(VRx, INPUT);

pinMode(VRy, INPUT);

}

void loop(){

xposition = analogRead(VRx);

yposition = analogRead(VRy);
mapx = map(xposition, 0, 1023, -512, 512);

mapy = map(yposition, 0, 1023, -512, 512);

Serial.print("X:");

Serial.print(max),

Serial.print("Y:");

Serial.print(mapy); delay(100);

}
```

**Output:**



```
|Y:58X:73
|Y:55X:67
|Y:63X:59
|Y:58X:73
|Y:56X:64
|Y:67X:73
```

**Applications:**

a. Data acquisition systems

b. Environmental monitoring stations

c. Instrumentation/test equipment

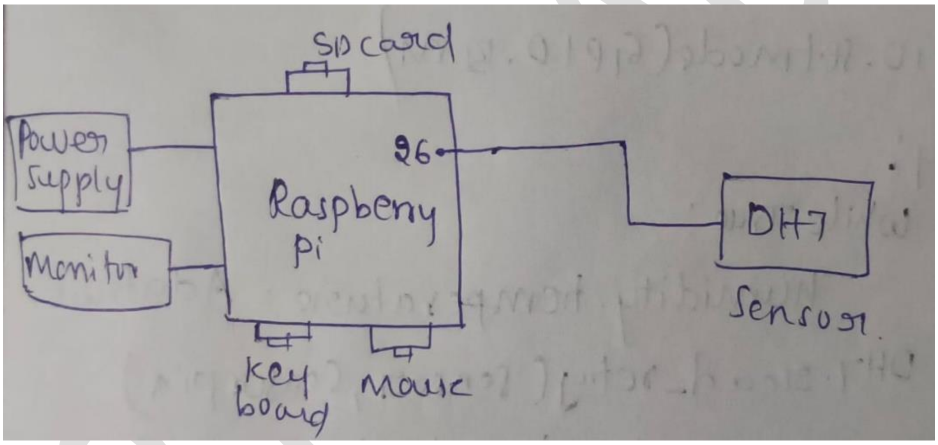
## Week-12

**Aim:** port/read the data to/from the cloud via MQTT broker with a raspberry pi

**Apparatus:**

1. extemal powersupply
2. Raspberrypi
3. Mouse
4. Monitor
5. Wires
6. USB Cable
7. Arduino board
8. Circuit diagram

**Circuit diagram:**



**Procedure:**

1. Create account in things speak

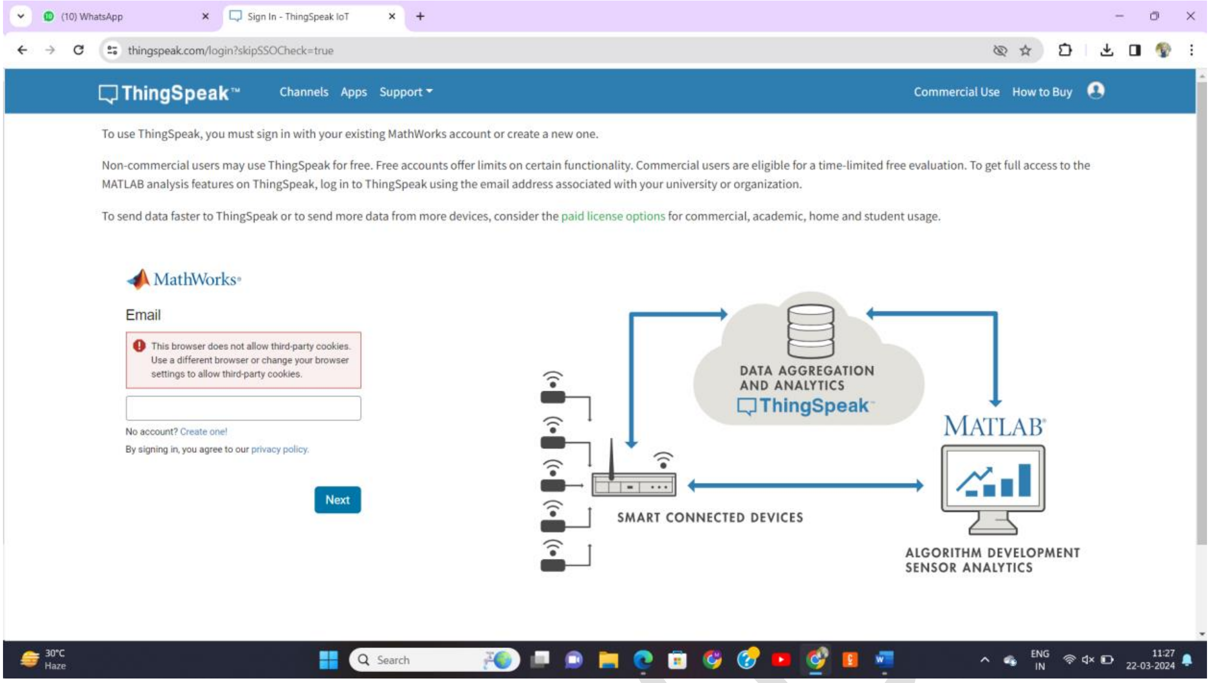
2.login into your acount

3. go to API keys and copy your key

4. and open Raspberry software thanny and write code.

5. According to your code Connect the Pin no.26 to DHT tensor & run it

# 6. observe the results in Thingspeak Private view



To use ThingSpeak, you must sign in with your existing MathWorks account or create a new one.

Non-commercial users may use ThingSpeak for free. Free accounts offer limits on certain functionality. Commercial users are eligible for a time-limited free evaluation. To get full access to the MATLAB analysis features on ThingSpeak, log in to ThingSpeak using the email address associated with your university or organization.

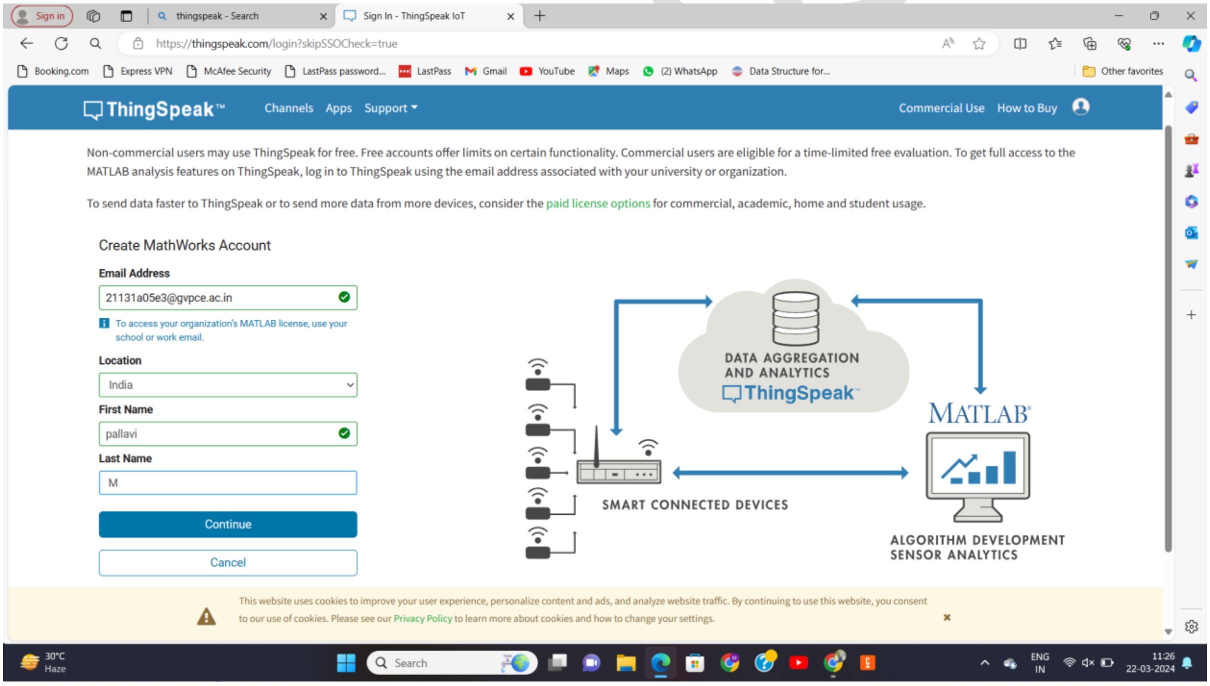
To send data faster to ThingSpeak or to send more data from more devices, consider the paid license options for commercial, academic, home and student usage.

Email

This browser does not allow third-party cookies. Use a different browser or change your browser settings to allow third-party cookies.

No account? Create one!

By signing in, you agree to our privacy policy.



Non-commercial users may use ThingSpeak for free. Free accounts offer limits on certain functionality. Commercial users are eligible for a time-limited free evaluation. To get full access to the MATLAB analysis features on ThingSpeak, log in to ThingSpeak using the email address associated with your university or organization.

To send data faster to ThingSpeak or to send more data from more devices, consider the paid license options for commercial, academic, home and student usage.

Create MathWorks Account

Email Address: 21131a05e3@gvpce.ac.in

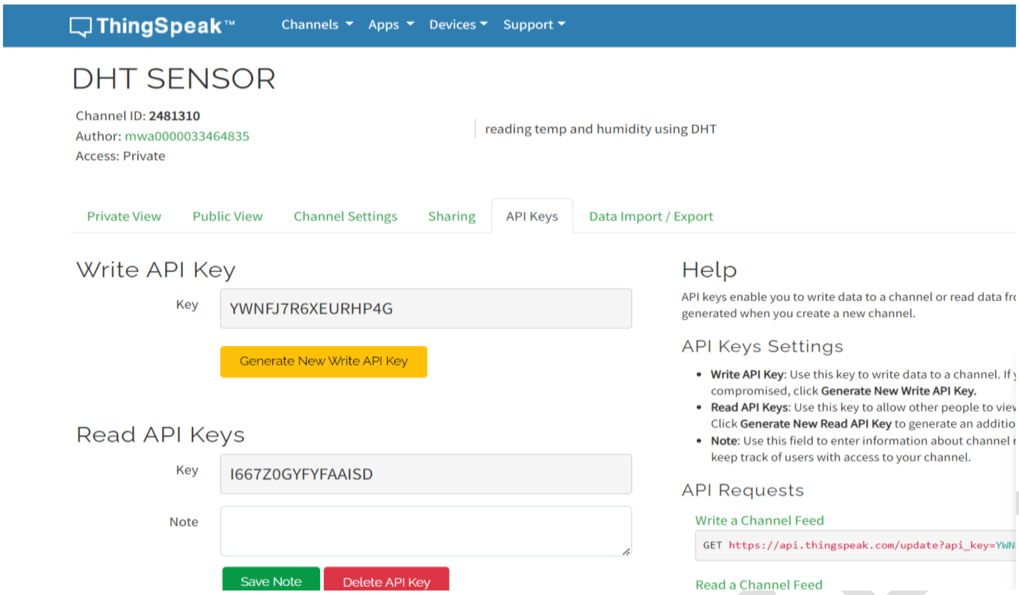
To access your organization's MATLAB license, use your school or work email.

Location: India

First Name: pallavi

Last Name: M

This website uses cookies to improve your user experience, personalize content and ads, and analyze website traffic. By continuing to use this website, you consent to our use of cookies. Please see our Privacy Policy to learn more about cookies and how to change your settings.



DHT SENSOR

Channel ID: **2481310**
Author: mwa0000033464835
Access: Private

reading temp and humidity using DHT

Write API Key

Key: YWNFJ7R6XEURHP4G

Read API Keys

Key: I667Z0GYFYFAAISD

Note:

Help

API keys enable you to write data to a channel or read data fr
generated when you create a new channel.

API Keys Settings

- **Write API Key**: Use this key to write data to a channel. If y
compromised, click **Generate New Write API Key.**
- **Read API Keys**: Use this key to allow other people to view
Click **Generate New Read API Key** to generate an additio
- **Note**: Use this field to enter information about channel r
keep track of users with access to your channel.

API Requests

Write a Channel Feed

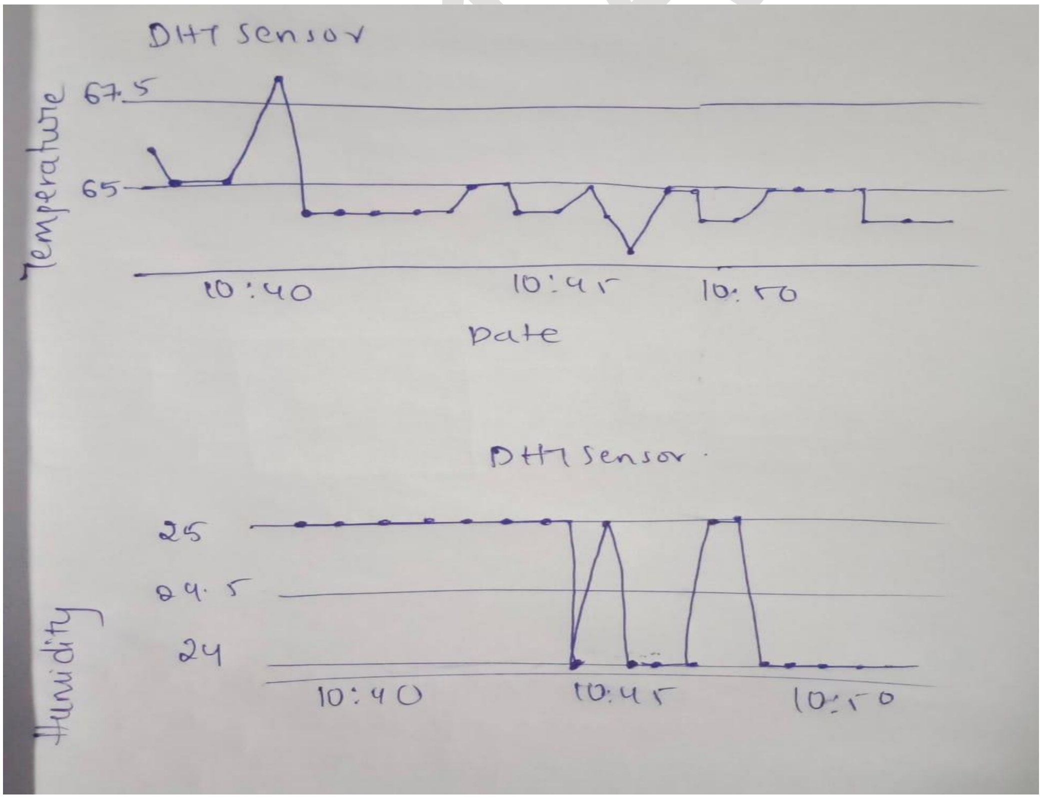
```
GET https://api.thingspeak.com/update?api_key=YWN
```

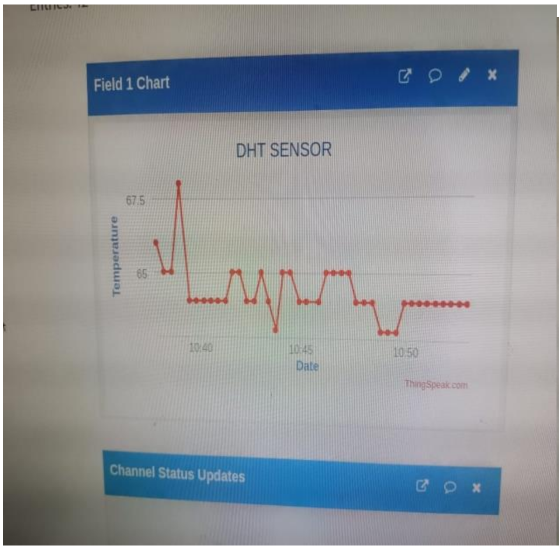
Read a Channel Feed

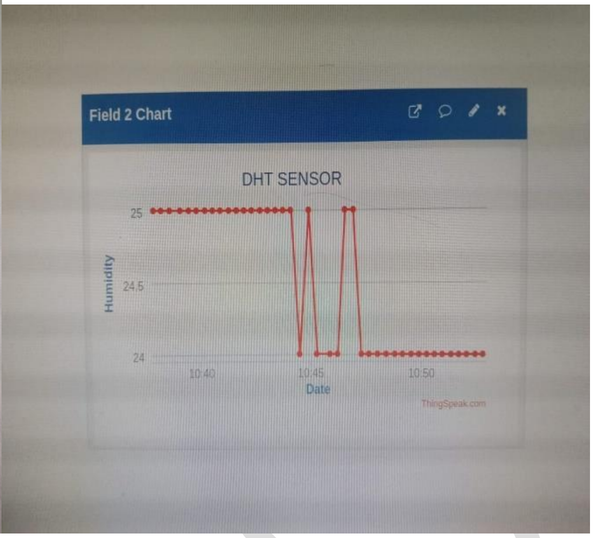
**Program:**

```
Import urllib.request

Import time

Import  RPI.GPIO OS GPIO

import Adafruit-DHT

write API key = '0J3YXQ10 KJL3H JA4’

 base url= https:/lapithingspect.com/ Updates ?api-keys={}”. Farmate (write APIkey)

Sensor = Adafruit_DHT.DHTII

Sensorpin=26

GPIO. set mode(GPIO.BCM)

try:

        While True:

                humidity trmperature=" Adafruit DHT-read-retry (sensor, sensorpin)

                 if humidity is not None and tempers is not None:
                    humidity = '%.2f' humidity
                    temperature: ‘%.2F' % temperature
                    conn=urllib.request.urlopen(baseURL+’&field1={}&field2={}’.format(humidity,temperature))
                    print(conn.read())
                    conn.close()
                    time.sleep(20)
except KeyboardInterrupt:
        GPIO.cleanup()
exit()
```

**Output:**







## Applications:

a. IoT device communication

b. Remote monitoring systems

c. Data logging to cloud