Internet Of Things Lab

WEEK - 1

LED LIGHTS:-

A):-

Aim:- To implement a program to blink LED for every one second.

Program:-

```
void setup(){
        pinMode(8, OUTPUT);
}
void loop(){
        digitalWrite(8,1);
        delay(1000);
        digitalWrite(8,0);
        delay(1000);
}
```

Output:- On board LED blinks for every one second.

B):-

Aim:- To implement a program to blink 2 LEDs for every one second **Alternatively**.

Program:-

```
void setup(){
        pinMode(8, OUTPUT);
        pinMode(9, OUTPUT);
}
void loop(){
        digitalWrite(8,1);
        digitalWrite(9,0);
        delay(1000);
        digitalWrite(8,0);
        digitalWrite(9,1);
        delay(1000);
}
```

Output:- On board 2 LEDs blink for every one second Alternatively.

C):-

Aim:- To implement a program to change the **intensity** of LED.

Program:-

```
void setup(){
        pinMode(8, OUTPUT);
}
void loop(){
        for(int i=0;i<=255;i++){
                analogWrite( 8, i);
                delay(100);
        }
        for(int i=255;i>=0;i--){
                analogWrite( 8, i);
                delay(100);
        }
}
```

Output:- On board LED glows with varying intensity.

WEEK – 2

BUZZER:-

Aim:- To implement a program to Switch LED and Buzzer ON for every second.

Program:-

```
void setup(){
        pinMode(8, OUTPUT);
        pinMode(10, OUTPUT);
}
void loop(){
        digitalWrite(8,1);
        digitalWrite(10,1);
        delay(1000);
        digitalWrite(8,0);
        digitalWrite(10,0);
        delay(1000);
}
```

Output:- On board LED glows with Buzzer sound for every one second.

WEEK – 3

RGB:-

Aim:- To implement a program to obtain RED, BLUE and GREEN colors of RGB LED.

Program:-

```
int c;
void setup(){
        Serial.begin( 9600 );
        pinMode(8, OUTPUT);
        pinMode(9, OUTPUT);
        pinMode(10, OUTPUT);
}
void loop(){
        for(c=1;c<=3;c++){
                if(c==1){
                        digitalWrite(8,1);
                        digitalWrite(9,0);
                        digitalWrite(10,0);
                }
                else if(c==2){
                        digitalWrite(8,0);
                        digitalWrite(9,1);
                        digitalWrite(10,0);
                }
                else{
                        digitalWrite(8,0);
                        digitalWrite(9,0);
                        digitalWrite(10,1);
                }
                delay(1000);
                Serial.println(c);
        }
}
```

Output:- The LED colors Red, Green and Blue will lightup simultaneously one after the another.

WEEK – 4

RGB:-

A):-

Aim:- To implement a program to obtain different colors of RGB LED through user input.

Program:-

```
string color = " ";
void setup(){
        Serial.begin( 9600 );
        pinMode(8, OUTPUT);
        pinMode(9, OUTPUT);
        pinMode(10, OUTPUT);
}
void loop(){
        color = Serial.readString();
        color.trim();
        if(color = 'green' || color = 'GREEN'){
                rgb( 0, 255, 0);
        }
        if(color = 'red' || color = 'RED'){
                rgb( 255, 0, 0);
        }
        if(color = 'blue' || color = 'BLUE'){
                rgb(0, 0, 255);
        }
        if(color = 'yellow' || color = 'YELLOW'){
                rgb( 255, 0, 255);
        }
        if(color = 'cyan' || color = 'CYAN'){
                rgb( 0, 255, 255);
        }
        if(color = 'magenta' || color = 'MAGENTA'){
                rgb(255, 255, 0);
        }
        if(color = 'purple' || color = 'PURPLE'){
                rgb( 128, 128, 0);
        }
        if(color = 'navyblue' || color = 'NAVYBLUE'){
                rgb( 0, 128, 0);
        }
        if(color = 'brown' || color = 'BROWN'){
                rgb( 165, 42, 42);
        }
        if(color = 'sage' || color = 'SAGE'){
```

```
                rgb(178, 136, 172);
        }
    }
    void rgb(int x, int y, int z){
        analogWrite(8, x);
        analogWrite(9, y);
        analogWrite(10, z);
    }
```

Output:- The color of RGB LED changes as per the given string.

B):-

Aim:- To implement a program to change the intensity of RGB LED lights.

Program:-

```
    void setup(){
        pinMode(8, OUTPUT);
        pinMode(9, OUTPUT);
        pinMode(10, OUTPUT);
    }
    void loop(){
        for(int i=0;i<=255;i++){
            analogWrite( 8, i);
            analogWrite( 9, i);
            analogWrite( 10, i);
            delay(100);
        }
        for(int i=255;i>=0;i--){
            analogWrite( 8, i);
            analogWrite( 9, i);
            analogWrite( 10, i);
            delay(100);
        }
    }
```

Output:- The intensity of LED increases and then decreases continously.

C):-

Aim:- To implement a program to change the intensity of LED lights.

Program:-

```
int i;
void setup(){
        pinMode(8, OUTPUT);
        pinMode(9, OUTPUT);
        pinMode(10, OUTPUT);
}
void loop(){
        for(int i=0;i<=255;i++){
                for(int i=0;i<=255;i++){
                        for(int i=0;i<=255;i++){
                                RGB(i, j, k);
                                delay(100);
                        }
                }
        }
}
void RGB(int R, int G, int B){
        analogWrite(8, R);
        analogWrite(9, G);
        analogWrite(10, B);
}
```

Output:- The intensity of the colors RED, GREEN and BLUE will starts to increase and changes.

WEEK – 5

SERVO MOTOR:-

A):-

Aim:- To implement a program to rotate Servo motor 90° for every 500ms.

Program:-

```
#include <servo.h>
Servo s1;
int Servopin = 8;
void setup(){
        s1.attach(Servopin);
}
void loop(){
        s1.write(90);
        delay(500);
}
```

Output:- The Servo Motor needle points at 90° continously for every 500ms.

B):-

Aim:- To implement a program to rotate Servo motor 180° in Clockwise direction.

Program:-

```
#include <servo.h>
Servo s1;
int Servopin = 8;
void setup(){
        s1.attach(Servopin);
}
void loop(){
        for(int i=0 ; i<=180 ;i += 45){
                s1.write(i);
                delay(500);
        }
}
```

Output:- The Servo Motor needle rotates in 180° in the clockwise direction passing for 500ms at each 45° increment.

C):-

Aim:- To implement a program to rotate Servo motor 180° in antiClockwise direction.

Program:-

```
#include <servo.h>
Servo s1;
int Servopin = 8;
void setup(){
        s1.attach(Servopin);
}
void loop(){
        for(int i=180 ; i>=0 ;i -= 45){
                s1.write(i);
                delay(500);
        }
}
```

Output:- The Servo Motor needle rotates in 180° in the counter clockwise direction passing for 500ms at each 45° increment.

WEEK – 6

STEPPER MOTOR:-

A):-

Aim:- To implement a program to rotate Stepper motor rotate in clockwise direction.

Program:-

```
#include <Stepper.h>
int motorspeed = 5;
Stepper mystepper(2048, 2, 4, 3, 5);
void setup() {
        mystepper.setSpeed(motorspeed);
}

void loop() {
        mystepper.step(512);
        delay(2000);
}
```

Output:- The stepper motor starts to rotate in the clockwise direction in a stepwise manner.

B):-

Aim:- To implement a program to rotate Stepper motor rotate either in clockwise or anti-clockwise direction to ‘m’ number of steps.

Program:-

```
#include <Stepper.h>
int motorspeed = 5;
Stepper mystepper(2048, 2, 4, 3, 5);
void setup() {
        mystepper.setSpeed(motorspeed);
}

void loop() {
        mystepper.step(512);
        delay(2000);
        mystepper.step(-1024);
        delay(2000);
}
```

Output:- The stepper motor starts to rotate in the clockwise direction at rotates back in anti-clockwise direction with specified step counts and delays.

WEEK – 7

BLUETOOTH MODULE:-

A):-

Aim:- To implement a program to send and receive messages using Bluetooth module and arduino.

Program:-

```
#include <SoftwareSerial.h>
SoftwareSerial EEBlue(10, 11);
void setup() {
        Serial.begin(9600);
        EEBlue.begin(9600);
        Serial.println("Bluetooth is ready");
}
void loop() {
        if (EEBlue.available()) {
                Serial.write(EEBlue.read());
        }
        if (Serial.available()) {
                EEBlue.write(Serial.read());
        }
}
```

Output:- The messages are sent & received by the HC-05 adapter to the receiver & vice-versa with a following IP connected through the Bluetooth.

WEEK – 8

DHT11 SENSOR:-

Aim:- To implement DHT11 sensor for measuring humidity and temperature.

Program:-

```
#include <DHT.h>
DHT dht(8, DHT11);
void setup() {
  Serial.begin(9600);
  dht.begin();
  Serial.println("Starting DHT test");
  delay(2000);

}

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

Output:- The readings of the humidity are done by DHT11 sensor by the thermistor in it. It converts them into digital signals and print them.

PIR SENSOR

Aim:- To implement the working of the PIR sensor for detecting motion of an object.

Program:-

```
int sensorData;
void setup() {
  Serial.begin(9600);
  pinMode(9, OUTPUT);
  pinMode(8, INPUT);
}
void loop() {
```

```
  sensorData = digitalRead(8);
  if (sensorData == HIGH) {
    digitalWrite(9, HIGH);
    Serial.println("Sensor activated");
    Serial.println("Motion detected");
    delay(50);
  } else {
    digitalWrite(9, LOW);
    delay(50);
  }
}
```

Output:- The PIR sensor is used to detect the motion. When the motion is detected the LED will turn on.

WEEK – 9

RASPBERRY PI:

LED BLINKING:-

Aim:- To implement the Raspberry PI & implement the LED Light blinking.

Program:-

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

ledPinOne = 12
ledPinTwo = 13

GPIO.setup(ledPinOne, GPIO.OUT)
GPIO.setup(ledPinTwo, GPIO.OUT)

try:
   while True:
      GPIO.output(ledPinOne, GPIO.HIGH)
      time.sleep(1)
      GPIO.output(ledPinTwo, GPIO.HIGH)
      time.sleep(1)
      GPIO.output(ledPinOne, GPIO.LOW)
      time.sleep(1)
      GPIO.output(ledPinTwo, GPIO.LOW)
      time.sleep(1)

except KeyboardInterrupt:
   GPIO.cleanup()
   exit()
```

Output:- The Led starts to blink alternative for every second controlled by raspberry pi

RGB :-

Aim:- To implement the Raspberry PI & implement the RGB.

Program:-

```
import RPi.GPIO as GPIO
import time
print("RGB LED TEST")
print("Connect 10 11 12 13 to G R Gr B of RGB LED")
i = input('Enter the color: ')
```

```
GPIO.setmode(GPIO.BCM)
buzzer = 14
commonCathode = 10
red = 11
green = 12
blue = 13
def ledColour(colour="none"):
    for x in range(10, 14):
        GPIO.setup(x, GPIO.OUT)
        GPIO.output(x, GPIO.LOW)
    if colour == "red":
        GPIO.output(red, GPIO.HIGH)
    elif colour == "green":
        GPIO.output(green, GPIO.HIGH)
    elif colour == "blue":
        GPIO.output(blue, GPIO.HIGH)
while True:
    ledColour(i)
    time.sleep(2)
```

Output:- when we enter the colour it will show the respective color.

WEEK – 10

BUZZER :-

Aim:- To implement the Buzzer using Raspberry PI

Program:-

```
import RPi.GPIO as GPIO
import time
print("BUZZER TEST")
print("Connect G 5V 24 25 to G V S N of Buzzer")
buzzerPin = 24  # Assign the correct GPIO pin number
GPIO.setmode(GPIO.BCM)
GPIO.setup(buzzerPin, GPIO.OUT)
while True:
   GPIO.output(buzzerPin, GPIO.HIGH)
   time.sleep(0.381)  # Adjust the sleep time as needed
   GPIO.output(buzzerPin, GPIO.LOW)
   time.sleep(0.381)  # Adjust the sleep time as needed
```

Output:- buzzer sounds at an interval of 0.381 secs.

10B)

Aim: Interface an ultrasonic sensor with raspberry pi to print distance

Program:

```
import RPi.GPIO as GPIO
import time
def distance(trigpin, echopin):
   GPIO.output(trigpin, True)
   time.sleep(0.0001)
   GPIO.output(trigpin, False)
   while GPIO.input(echopin) == 0:
      pulse_start = time.time()
   while GPIO.input(echopin) == 1:
      pulse_end = time.time()
```

```
    try:
        pulse_duration = pulse_end - pulse_start
    except:
        print('Calibrating')
        return -1

    distance = pulse_duration * 17150
    distance = round(distance + 1.15, 2)

    return distance

GPIO.setmode(GPIO.BCM)
trigpin = 24
echopin = 25
GPIO.setup(trigpin, GPIO.OUT)
GPIO.setup(echopin, GPIO.IN)

while True:
    dist = distance(trigpin, echopin)
    print('Measured distance = {}cm'.format(dist))
    time.sleep(0.01)
```

Output: The ultrasonic measures the distance for every 0.01 seconds.

WEEK – 11

SERVO :-

Aim:- To implement the Servo motor using Raspberry PI

Program:-

```
import RPi.GPIO as GPIO
import time
print("=== SERVO MOTOR TEST ===")
print("Connect G 5V 24 25 to G V S N of Servo")
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
servoPin = 24
GPIO.setup(servoPin, GPIO.OUT)
p = GPIO.PWM(servoPin, 50)
p.start(0)
try:
   while True:
      for i in range(2, 11, 1):
         p.ChangeDutyCycle(1)  # left -90 deg position
         time.sleep(0.5)
except KeyboardInterrupt:
   p.stop()
   GPIO.cleanup()
```

Output:- servo motor rotates at regular intervals by a constant step.

WEEK – 12

DHT11:-

Aim:- To implement the DHT11 sensor to measure temperature and humidity using Raspberry PI

Program:-

```
import RPi.GPIO as GPIO
import Adafruit_DHT
import time
print("=== DHT11 TEST ===")
print("Connect G SV 24 25 to G VS N of DHT11")
GPIO.setmode(GPIO.BCM)
dhtPin = 24
sensor = Adafruit_DHT.DHT11
try:
   while True:
      humidity, temperature = Adafruit_DHT.read_retry(sensor, dhtPin)
      print("Humidity: ", humidity, "Temperature:", temperature)
      time.sleep(0.25)
except KeyboardInterrupt:
   GPIO.cleanup()
```

Output:- servo motor rotates at regular intervals by a constant step.

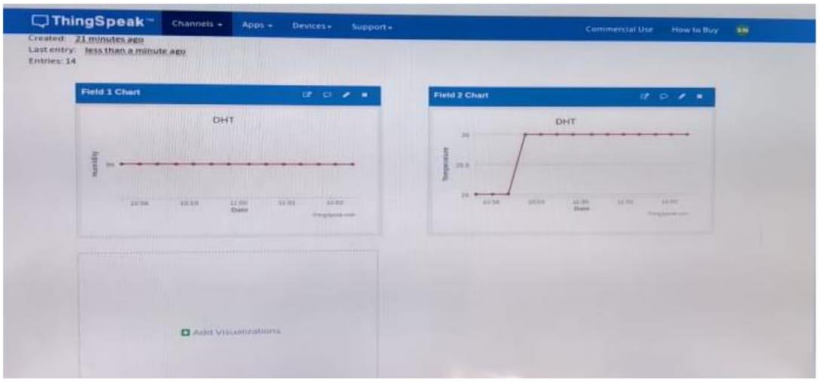
WEEK-13

AIM: To post the data to the cloud via MQTT Broker with a raspberry pi

Program:

```
import urllib.request
import time
import RPi.GPIO as GPIO
import Adafruit_DHT
writeAPIKey = 'FNAIGOOVH990KRLW'
baseURL = "https://api.thingspeak.com/update?api_key={}".format(writeAPIKey)
sensor = Adafruit_DHT.DHT11
sensorPin = 18
GPIO.setmode(GPIO.BCM)
try:
    while True:
        humidity, temperature = Adafruit_DHT.read_retry(sensor, sensorPin)
        if humidity is not None and temperature is not None:
            humidity = '{:.2f}'.format(humidity)
            temperature = '{:.2f}'.format(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:


ThingSpeak
Channels
Apps
Devices
Support
Commercial Use
How to Buy
Created: 21 minutes ago
Last entry: less than a minute ago
Entries: 14
Field 1 Chart
DHT
Humidity
Date
Field 2 Chart
DHT
Temperature
Date
Add Visualizations