P E S COLLEGE OF ENGINEERING, MANDYA
(An Autonomous Institution under VTU, Belagavi)

Department of Information Science & Engineering

Internet of Things Laboratory manual

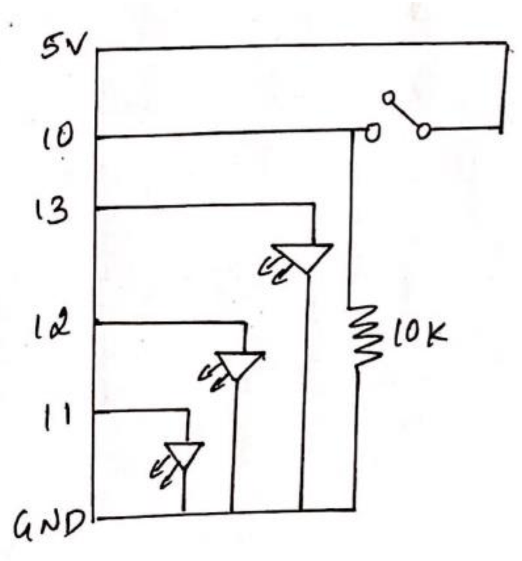
2022

Prepared By

Rakshith N
Assistant Professor
Dept of IS&E
PESCE, Mandya

1. Design a Smart Traffic light System under Arduino UNO which includes a crosswalk button

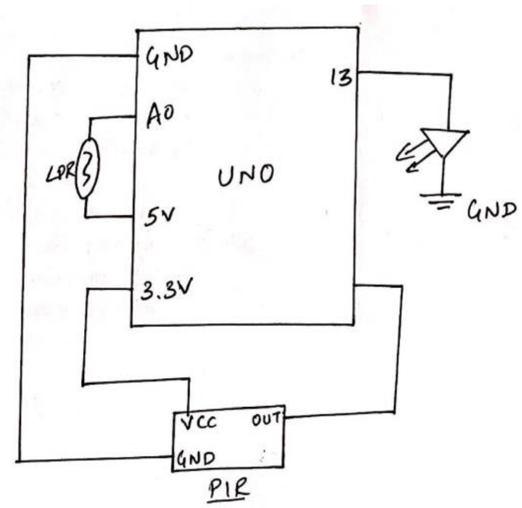
   Components Required: Arduino UNO, Breadboard, Switch, 3LED, Jumper wires, 10k Register



```
int SwitchPin = 10;
int rPin=13;
int yPin=12;
int gPin=11;

boolean ledon = false;

void setup()
{
pinMode(SwitchPin, INPUT);
pinMode (rPin, OUTPUT);
pinMode (yPin, OUTPUT);
pinMode (gPin, OUTPUT);
}
void loop() {
if (digitalRead (SwitchPin) == HIGH)
{
  digitalWrite (rPin, HIGH);
  delay (5000);
  digitalWrite (yPin, LOW);
  delay (5000);
  digitalWrite (gPin, LOW);
  delay (5000);
}
else {
  digitalWrite(rPin, HIGH);
  delay(1000);
  digitalWrite(rPin, LOW);
  delay (1000);
  if (digitalRead (SwitchPin) == HIGH)
  {
    digitalWrite(rPin, HIGH);
    delay (5000);
    digitalWrite (rPin, LOW);
  }
  digitalWrite(yPin, HIGH);
  delay(1000);
  digitalWrite(yPin, LOW );
  delay(1000);
  if (digitalRead (SwitchPin) == HIGH)
  {
    digitalWrite(rPin, HIGH);
    delay(5000);
    digitalWrite(yPin, LOW);
    digitalWrite(yPin, HIGH);
    delay(1000);
    digitalWrite(yPin, LOW);
    delay(1000);
  }
  digitalWrite(gPin, HIGH);
  delay(1000);
  digitalWrite(gPin, LOW);
  delay(1000);
  if(digitalRead (SwitchPin) == HIGH)
  {
    digitalWrite(rPin, HIGH);
    delay(5000);
    digitalWrite(rPin, LOW);
  }
  }
}
```

2. Develop an application using Arduino UNO to modulate an LED using an LDR and PIR sensor

Components Required: Arduino UNO, Breadboard, Jumper wires, LDR, LED and PIR Sensor

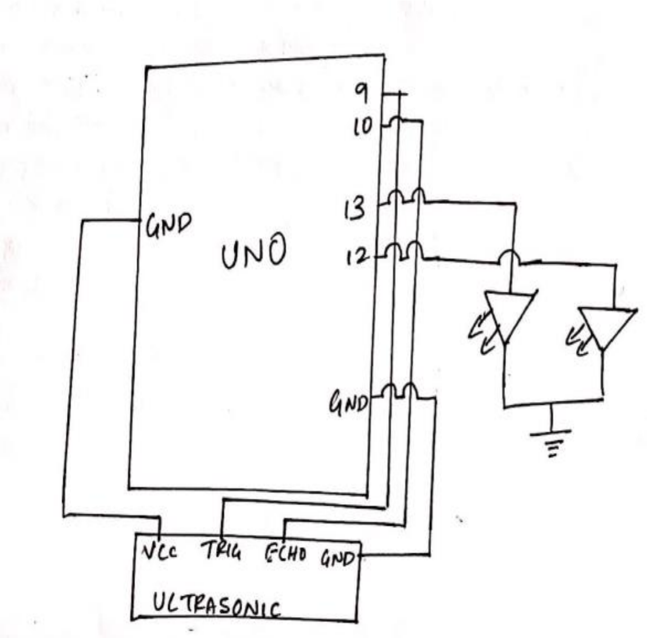


```
int pir = 10;
int ledPin = 13;
int ldr = 0;

void setup() {
  analogReference (DEFAULT);
  Serial.begin(9600);
  pinMode (ledPin, OUTPUT);
  pinMode (pir, INPUT);
  Serial.begin(9600);
}

void loop()
{
  int s =digitalRead (pir);
  Serial.println(s);
  Serial.println(analogRead(ldr));
  delay (500);
  int I=0;
  if (analogRead (ldr) >300)
  {
    if (s = HIGH)
    {
       if (analogRead (ldr) >300 && analogRead (ldr) < 400)
       {
          I=I+50;
          digitalWrite(ledPin, I);
       }
       if (analogRead (ldr )>400 && analogRead(ldr) <500)
       {
          I=I+100;
          digitalWrite (ledPin, I);
       }
       if (analogRead(ldr) >500 && analogRead(ldr) <600)
    {
     I=I+150;
     digitalWrite(ledPin,I);
    }
  }
  else{
   digitalWrite(ledPin, LOW);

  }
 }

}
```

3. Design a Smart Dustbin System using Arduino UNO and Ultrasonic Sensor

Components Required: Arduino UNO, Breadboard, Jumper wires, ultrasonic sensor, LED

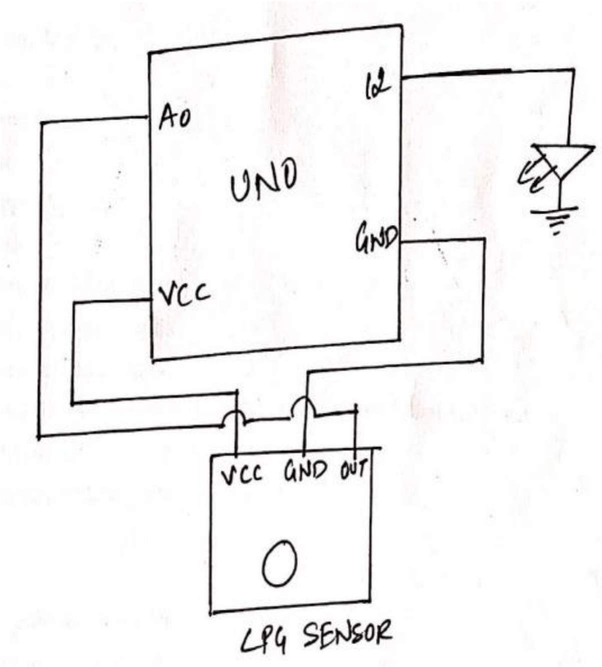
```
const int trigpin = 9;
const int echopin = 10;
int ledpin1 = 13;
int ledpin2 = 12;
long duration;
int distance;

void setup()
{
  pinMode (trigpin, OUTPUT);
  pinMode (echopin, INPUT);
  pinMode (ledpin1, OUTPUT);
  pinMode(ledpin2, OUTPUT);
  Serial.begin(9600);

}
```



```
void loop()
{
  digitalWrite (trigpin, LOW);
  delayMicroseconds (2);
  digitalWrite(trigpin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin, LOW);
  duration = pulseIn(echopin, HIGH);
  distance=duration * 0.034/2;
  Serial.print ("Distance");
  Serial.println(distance);
  if (distance <10)
  {
    Serial.println("Completely Empty");
    digitalWrite(ledpin1, HIGH);
    digitalWrite(ledpin2, LOW);
    delay (5000);
    digitalWrite(ledpin1, LOW);

  }
  if (distance >10 && distance <30)
  {
    Serial.println("Half Empty");
    digitalWrite (ledpin1, HIGH);
    digitalWrite (ledpin2, LOW);
    delay (2000);
  }
  else
  {
    Serial.println("Completely full");
    digitalWrite(ledpin2, HIGH);
    digitalWrite(ledpin1, LOW);
    delay(5000);
    digitalWrite(ledpin2, LOW);
  }

}
```

4. Write a program to interface LPG sensor to Arduino UNO and read the values of a sensor in the given environment and turn on the buzzer if petroleum gas is detected

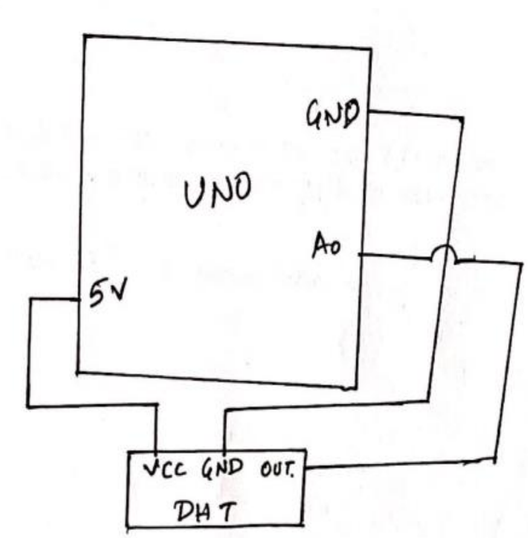
Components Required: Arduino UNO, Breadboard, Jumper wires, Buzzer, LPG Sensor



```
int pot=A0;

int val=0;
int led = 12;
int buzzer=8;

void setup ()
{
  Serial.begin(9600);
  pinMode(pot, INPUT);
  pinMode (led, OUTPUT);
  pinMode (buzzer, OUTPUT);

}

void loop()
{
  val=analogRead(pot);
  Serial.println(val);
  delay(1000);
  if (val <300)
  {
    digitalWrite(led,HIGH);
    delay(1000);
    digitalWrite(buzzer, HIGH);
  }
  else{
    digitalWrite(led, LOW);
    delay (1000);
    digitalWrite(buzzer, LOW);
  }

}
```

5. Write a program to interface DHT11 to Arduino UNO and read the values of temperature and humidity in the given environment and turn on the LED if the temperature value met the threshold value

Components Required: Arduino UNO, Breadboard, Jumper wires, DHT11



```
//Download SimpleDHT library in Arduino IDE

#include <SimpleDHT.h>

int pinDHT11=2;
SimpleDHT11 dht11 (pinDHT11);
const int Sensor_pin=A1;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  Serial.println("==============");
  Serial.println("Simple DHT11");
  byte temperature = 0;
  byte humidity = 0;
  int err=SimpleDHTErrSuccess;

  if((err=dht11.read(&temperature,&humidity, NULL)!= SimpleDHTErrSuccess))
  {
    Serial.print("Read DHT11 failed, err = ");
    Serial.print(SimpleDHTErrCode(err));
    Serial.print(SimpleDHTErrDuration(err));
    delay(1000);
    return;
  }
  Serial.print("Sample OK ");
  Serial.print((int) temperature);
  Serial.print("*c");
  Serial.print((int) humidity);
  Serial.print ("H");
  delay (1500);

  float moisture_percentage;
  int Sensor_analog;
  Sensor_analog =analogRead (Sensor_pin);
  moisture_percentage=(100-((Sensor_analog / 1023.00) * 100));
  Serial.print("Moisture Percentage= ");
  Serial.print(moisture_percentage);
  Serial.print ("% \n\n");
  delay(1000);
  if (moisture_percentage > 45 && moisture_percentage <=48)
  {
    Serial.println("Low water Content");
  }
  if (moisture_percentage >=49 && moisture_percentage <=52)
  {
    Serial.println("Moderate Water Content");
  }
  if(moisture_percentage >=53)
  {
    Serial. println ("HIGH Water Content");
  }
}
```

6. Develop an application using Raspberry pi to read the temperature of CPU and store the data in cloud using Ethernet/WiFi

Components Required: Raspberry Pi, Connecting cables

step1: open terminal and install

```
sudo apt-get install httplib
sudo apt-get instal urllib
```

step 2: login/sign up for ThingSpeak Cloud
visit www.thinspeak.com

step 3: create a Channel for your Data

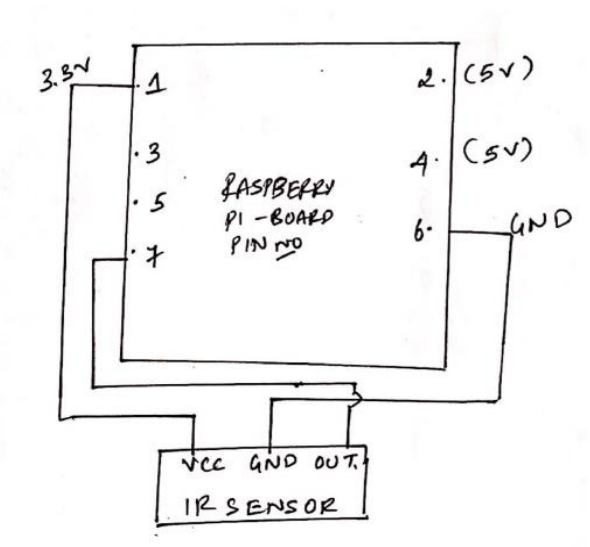
Once sign In after your account verification, create a new channel by clicking "New channel". Enter the Name and Description of the data you want to upload on this channel

step 4: Getting API Key in ThingSpeak
Click on API keys button to get your Unique API key

```
import httplib
import urllib
import time
key = "ABCD"  # Put your API Key here
def thermometer():
    while True:
        #Calculate CPU temperature of Raspberry Pi in Degrees C
        temp = int(open('/sys/class/thermal/thermal_zone0/temp').read())
        params = urllib.urlencode({'field1': temp, 'key':key })
        headers = {"Content-typZZe": "application/x-www-form-
urlencoded","Accept": "text/plain"}
        conn = httplib.HTTPConnection("api.thingspeak.com:80")
        try:
            conn.request("POST", "/update", params, headers)
            response = conn.getresponse()
            print temp
            print response.status, response.reason
            data = response.read()
            conn.close()
        except:
            print "connection failed"
        break
if __name__ == "__main__":
        while True:
                thermometer()
```

7. Develop an application using Raspberry Pi to detect an object using IR sensor and send the data to cloud using Ethernet/WiFi

Components Required: Raspberry Pi, Connecting cables, IR Sensor



```
import urllib

import httplib2
import RPi.GPIO as GPIO

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(2, GPIO.IN)


key="  "

def infra():
    while True:
        ir=GPIO.input(2)
        http=httplib2.Http()
        url='https://api.thingspeak.com/update?api_key='+str(key)+'&field1='+str(ir)
        try:
            response,data=http.request(url, "GET")
            print(ir)
            print(response.status, response, reason)
        except:
            print("connection failed")
            break

infra()
```