VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Reg.No	19BEC0358		
Student Name	ARPIT PATAWAT		
Course Code	ECE3502	**Slot & Semester**	L37+L38 WINTER -- 2021-22
Course Name	Iot Domain Analyst		
Program Title	Exercise 2		
Faculty	Dr. Karthikeyan B		

School of Electronics Engineering ,VIT, Vellore

AIM →

To perform the following operation –

•Based on the person presence, which will be detected by PIR, automatically turns ON bulb

•If the Potentiometer is less than the threshold value [water level sensor indicator], switch ON DC motor automatically indicating WATER MOTOR is ON to pump water

•If Smoke Sensor detect smoke, Buzzer must be ON

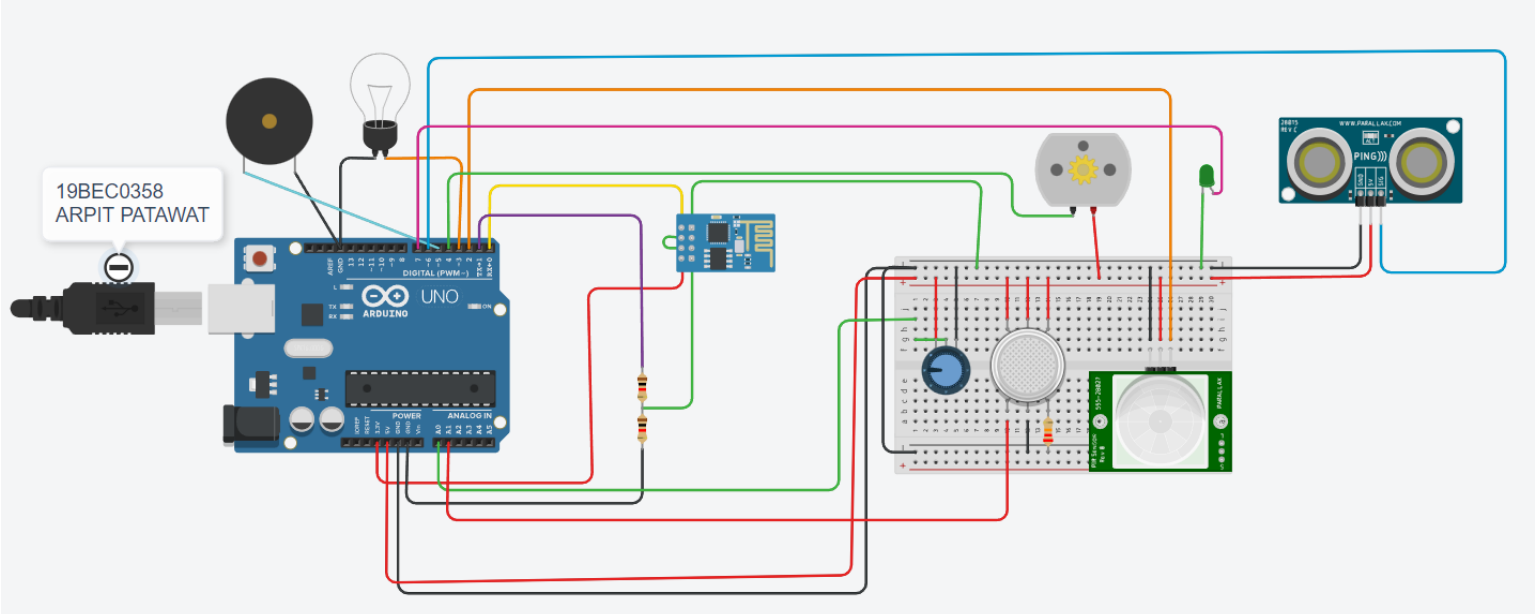
•Ultrasonic sensor detects the presence of any objects close to the main DOOR and SWITCH ON indicating that objects are close to main door.

•In Thingspeak, BULB[ON/OFF], DC MOTOR[ON/OFF], SMOKE SENSOR, ULTRASONIC sensor BOOLEAN data need to displayed as at what time these sensors/actuators responded.

COMPONENT LIST→

Name	Quantity	Component
U1	1	Arduino Uno R3
U3	1	Wifi Module (ESP8266)
R1 R2	2	1 kΩ Resistor
L1	1	Light bulb
M1	1	DC Motor
PIR1	1	9.117045552011653 , -170.13934414248678 , -176.687910748008 , -171.20131622160136 PIR Sensor
Rpot2	1	250 kΩ Potentiometer
GAS1	1	Gas Sensor
R3	1	22 kΩ Resistor
PIEZO1	1	Piezo
PING1	1	Ultrasonic Distance Sensor
D1	1	Green LED

CIRCUIT DIAGRAM →



CODE →

```
String ssid     = "Simulator Wifi";  // SSID to connect to
String password = ""; // Our virtual wifi has no password
String host     = "api.thingspeak.com"; // Open Weather Map API
const int httpPort   = 80;
String url     = "/update?api_key=OMPQYIH9IVI2HZBS&field";


int sensorState = 0;
float thresholda = 0.25;
float thresholdb = 0.75;

int setupESP8266(void) {
  // Start our ESP8266 Serial Communication
  Serial.begin(115200);   // Serial connection over USB to computer
  Serial.println("AT");   // Serial connection on Tx / Rx port to ESP8266
  delay(10);       // Wait a little for the ESP to respond
  if (!Serial.find("OK")) return 1;

  // Connect to 123D Circuits Simulator Wifi
  Serial.println("AT+CWJAP=\"" + ssid + "\",\"" + password + "\"");
  delay(10);       // Wait a little for the ESP to respond
  if (!Serial.find("OK")) return 2;

  // Open TCP connection to the host:
  Serial.println("AT+CIPSTART=\"TCP\",\"" + host + "\"," + httpPort);
  delay(50);       // Wait a little for the ESP to respond
  if (!Serial.find("OK")) return 3;

  return 0;
}




void setup() {
  pinMode(2, INPUT);
  pinMode(A0,INPUT);
  pinMode(A1,INPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(7,OUTPUT);
  Serial.begin(9600);

  setupESP8266();
```

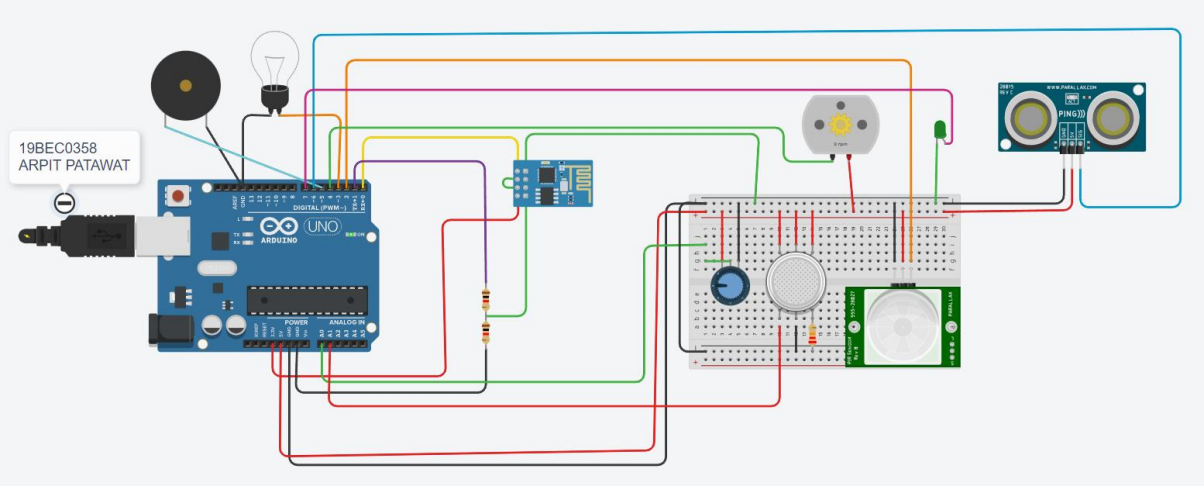
```
}

void loop() {
 digitalWrite(4,HIGH);
 PIR();
 WATER();
 GAS();
 DISTANCE();

 //delay(1000);
}

void PIR(){
  sensorState = digitalRead(2);
  if (sensorState == HIGH) {
    digitalWrite(3, HIGH);
    Serial.println("Motion Detected!");
    anydata(1,1);
  } else {
    digitalWrite(3, LOW);
    anydata(0,1);
  }
  delay(10);
}

void WATER(){
float thresholda = 0.25;
float thresholdb = 0.75;
float val = analogRead(A0);
float per = val / 1023;
Serial.print("Water level is ");
Serial.println(per*100);
if(per < thresholda){
    digitalWrite(4, LOW);
           anydata(1,2);
    while(per < thresholdb){
      val = analogRead(A0);
             per = val / 1023;
      digitalWrite(4, LOW);
      anydata(1,2);
      Serial.print("Water level is ");
      Serial.println(per*100);
    }
  }
  digitalWrite(4, HIGH);
  anydata(0,2);
```

```
}

void GAS(){
int gas = analogRead(A1);
Serial.print("value of gas sensor is: ");
Serial.println(gas,DEC);
anydata(gas,3);
if (gas> 800){
   digitalWrite(5, HIGH );
   Serial.print("UNSAFE GAS DETECTED");
}
else {
  digitalWrite(5, LOW);

}
}

void DISTANCE(){
pinMode(6, OUTPUT);
digitalWrite(6, LOW);
delayMicroseconds(2);
digitalWrite(6, HIGH);
delayMicroseconds(5);
digitalWrite(6, LOW);
pinMode(6, INPUT);
long duration , cm;
duration = pulseIn(6,HIGH);
cm =  microsecondsToCentimeters(duration);
Serial.print("Distance: ");
Serial.print(cm);
Serial.println("cm");
anydata(cm,4);
  if(cm < 100){
  digitalWrite(7,HIGH);
  Serial.println("Human Presence Detected");
  anydata(cm,2);
  }
  else{digitalWrite(7,LOW);}
}

long microsecondsToCentimeters(long microseconds) {
return microseconds / 29 / 1.955;
}

void anydata(int value, int field ) {
  // Construct our HTTP call
```

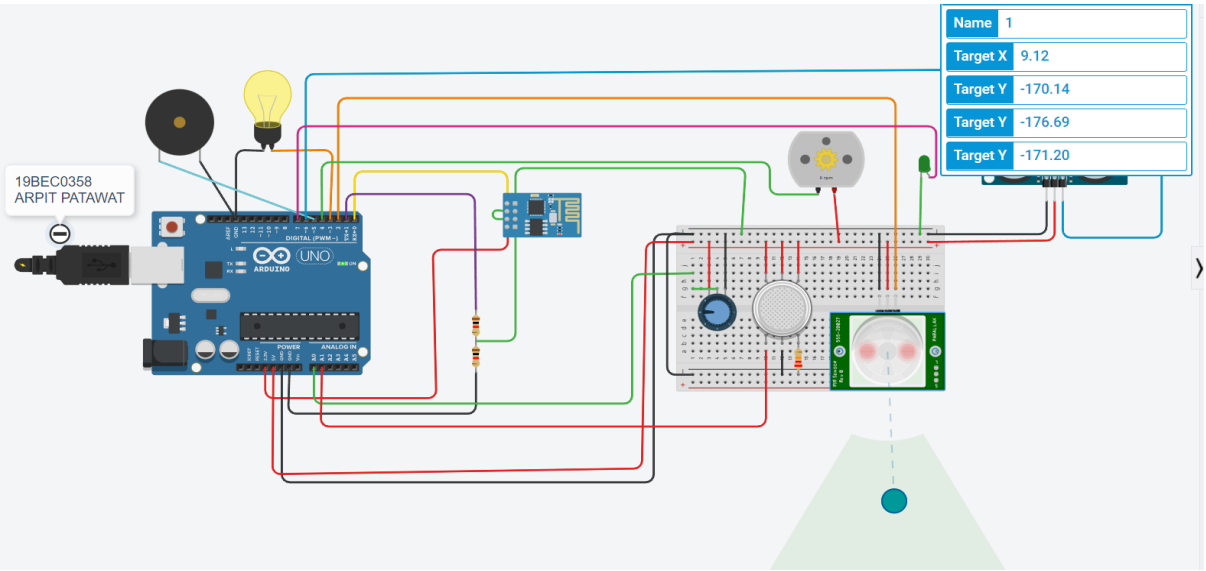
```
 String httpPacket = "GET " + url + String(field)+ "=" + String(value) + " HTTP/1.1\r\nHost: " + host + "\r\n\r\n";
 int length = httpPacket.length();

 // Send our message length
 Serial.print("AT+CIPSEND=");
 Serial.println(length);
 delay(10); // Wait a little for the ESP to respond if (!Serial.find(">")) return -1;

 // Send our http request
 Serial.print(httpPacket);
 delay(10); // Wait a little for the ESP to respond
 if (!Serial.find("SEND OK\r\n")) return;
 delay(10000);

}
```

OUTPUT →

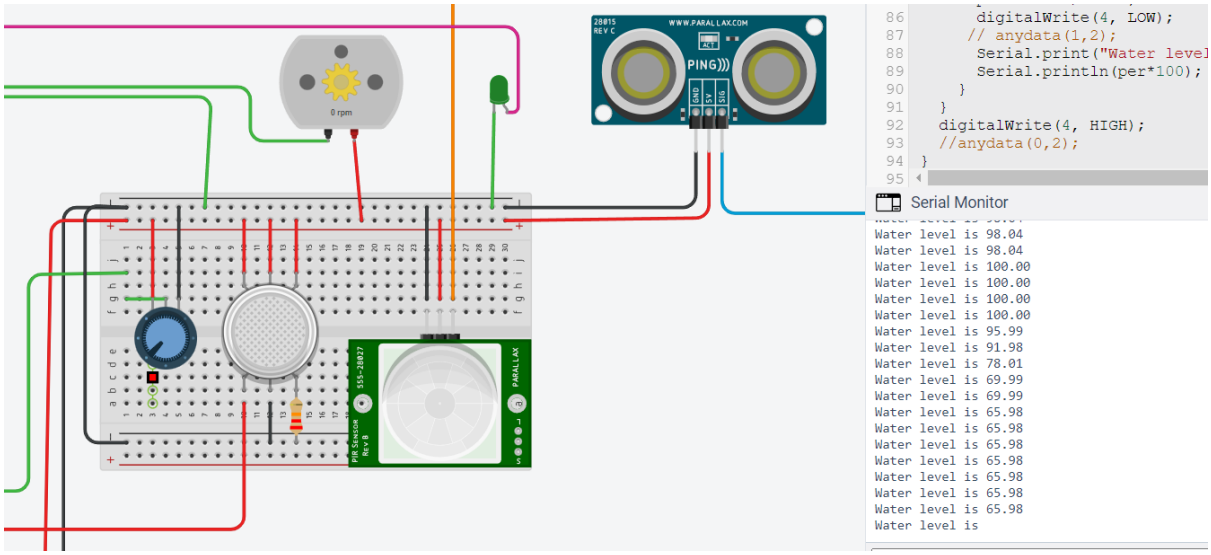
1.when no motion is detected



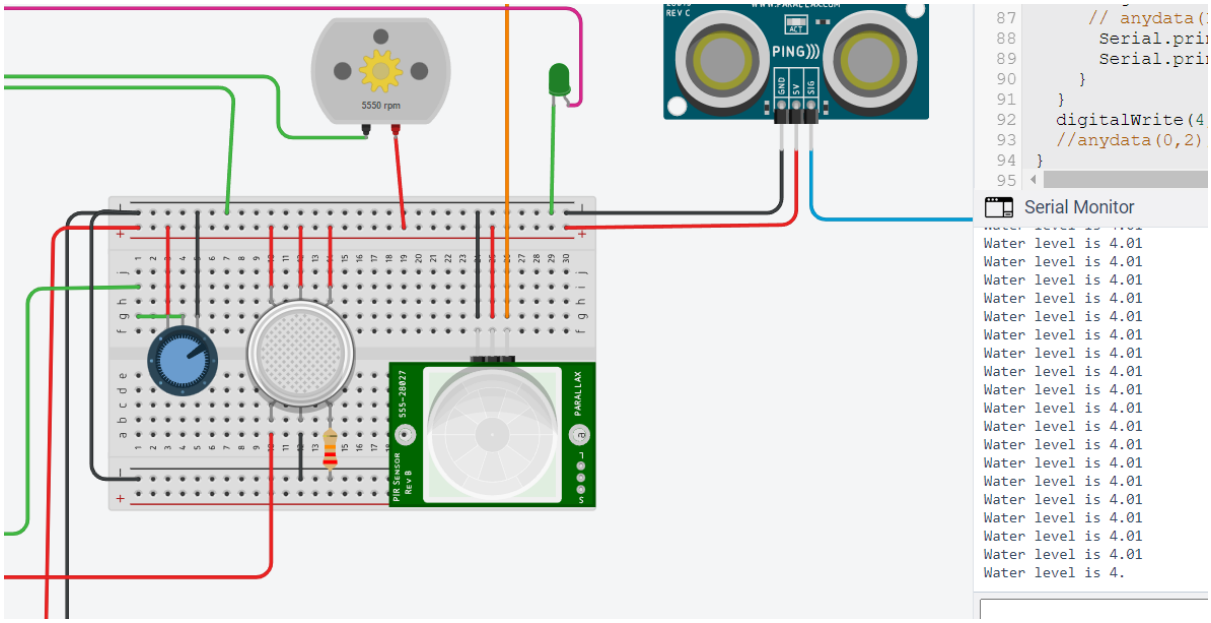
When motion is detected, bulb turns on.



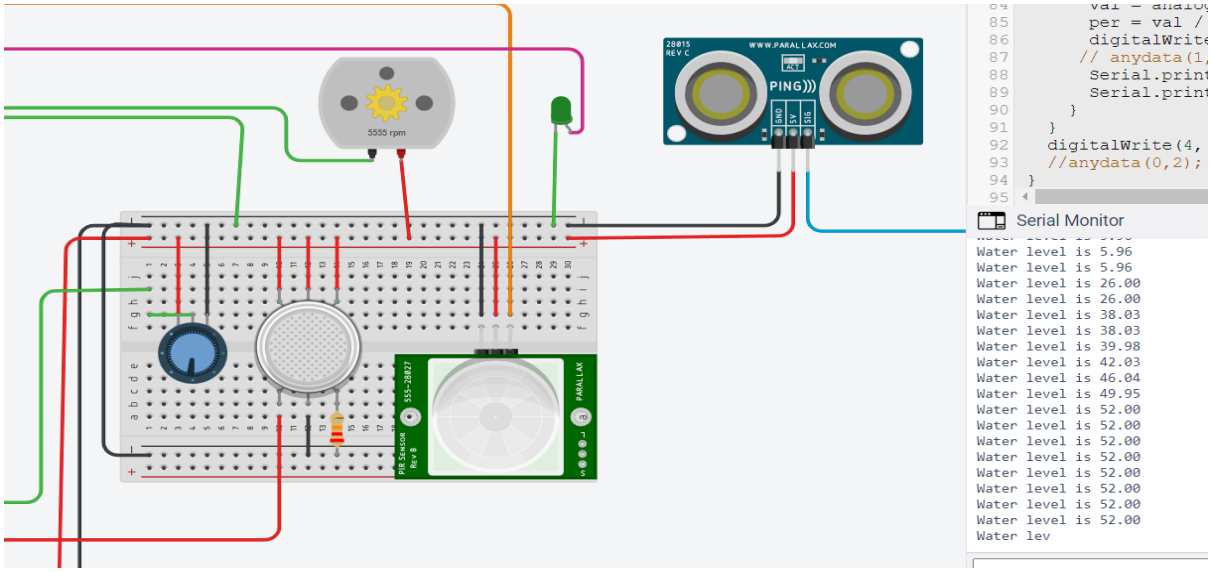
2.when the value of potentiometer is greater than > 25% which mean turn off motor



Once the value reaches < 25% then motor will turn on

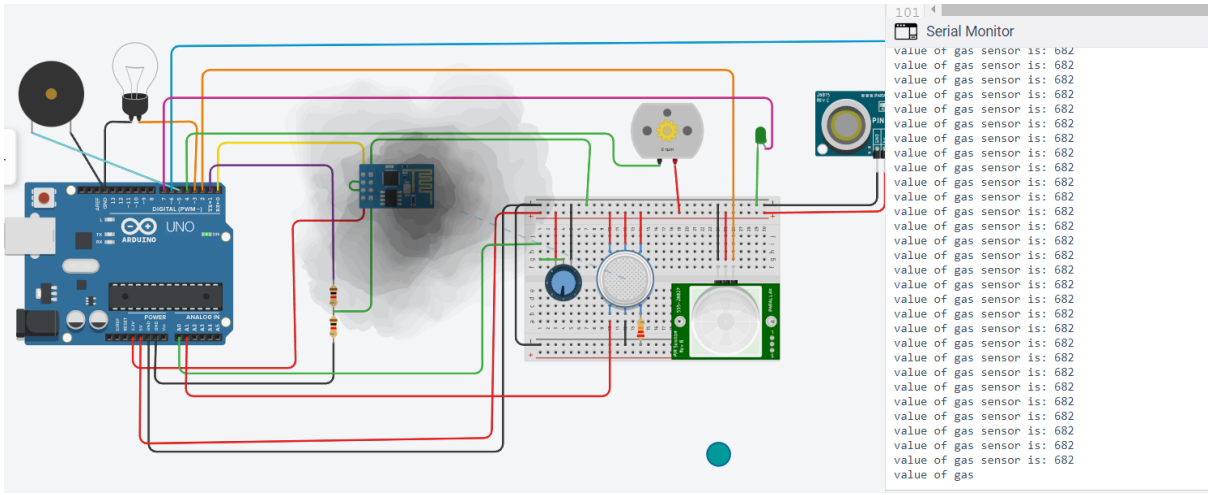


It will remain on until the tank is filled > 75%

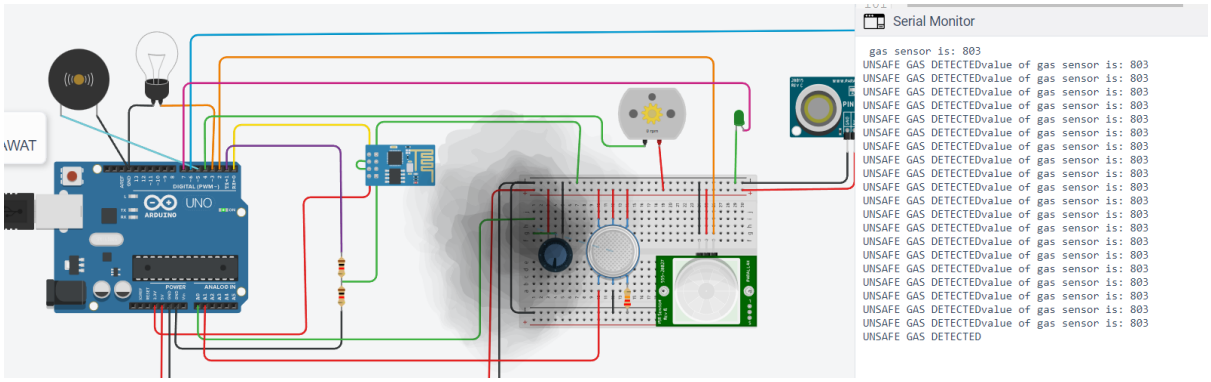


It will turn off once it reaches > 75%
Note –since I am checking value continuously, I am not sending data to thingspeak cloud for this part alone. When all 4 parts are working together then I will send.

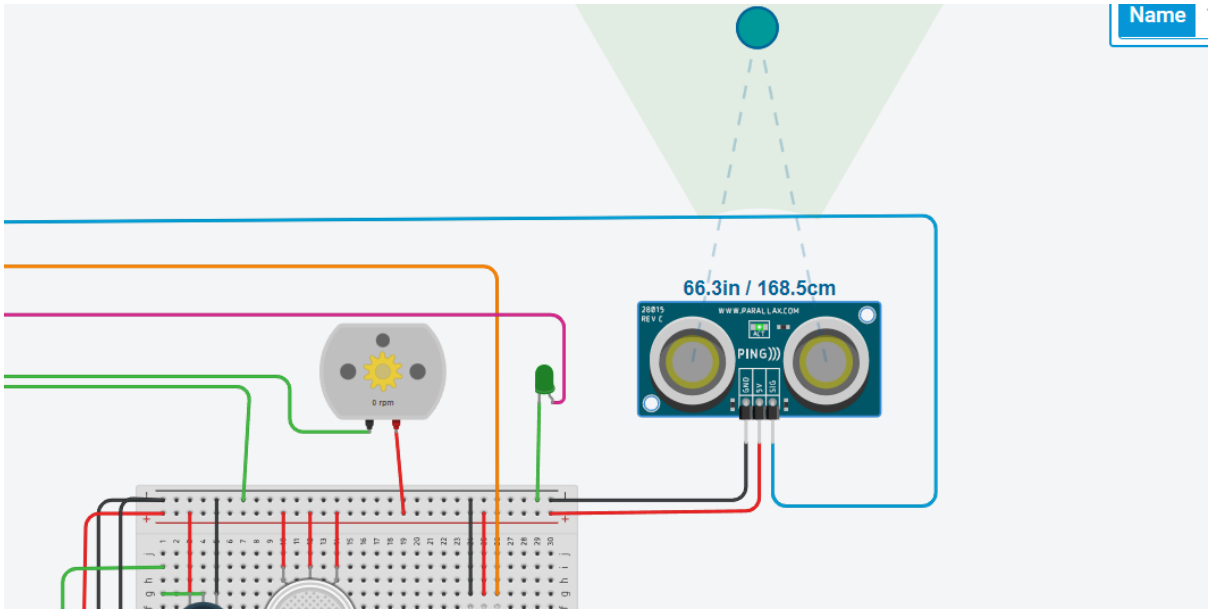
3.when gas sensor value is <800 then buzzer remains off



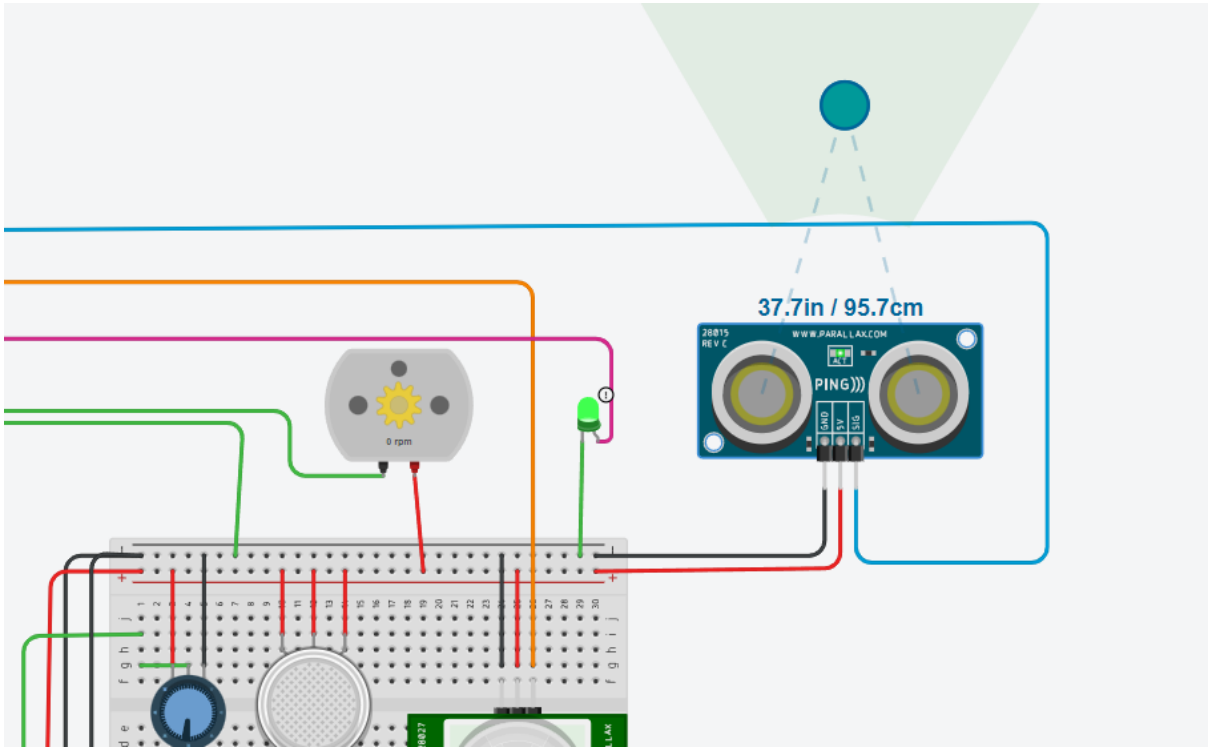
As soon as the sensor reads value > 800 , buzzer turns on.



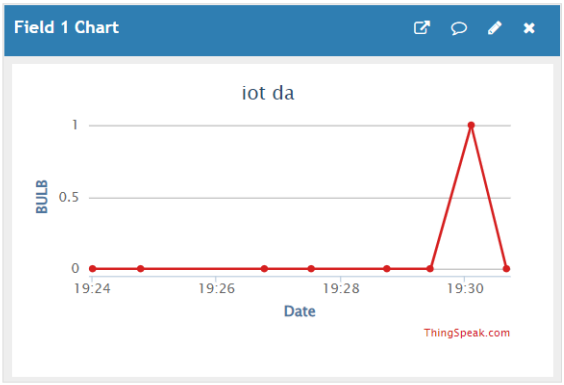
4.when the distance between ultrasonic sensor and person is > 100 cm then led remains off

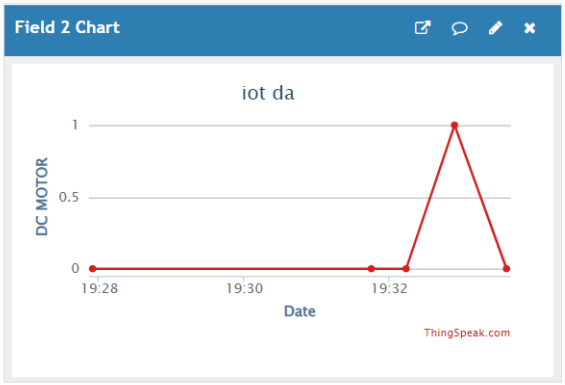


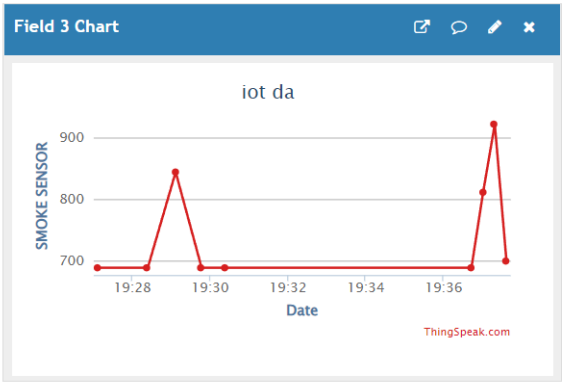
As soon the distance reduces below 100 cm then led turns on

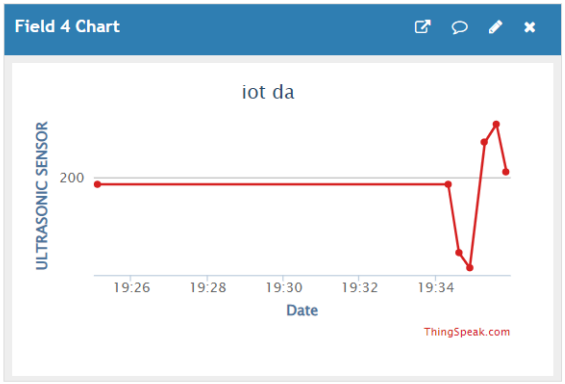


5.SENDING ALL DATA TO THINGSPEAK









Serial Monitor

```
AT+CWJAP="Simulator Wifi",""
AT+CIPSTART="TCP","api.thingspeak.com",80
AT+CIPSEND=84
GET /update?api_key=OMPQYIH9IVI2HZBS&field1=0 HTTP/1.1
Host: api.thingspeak.com

Water level is 5.96
AT+CIPSEND=84
GET /update?api_key=OMPQYIH9IVI2HZBS&field2=1 HTTP/1.1
Host: api.thingspeak.com

AT+CIPSEND=84
GET /update?api_key=OMPQYIH9IVI2HZBS&field2=1 HTTP/1.1
Host: api.thingspeak.com

Water level is 90.03
AT+CIPSEND=84
GET /update?api_key=OMPQYIH9IVI2HZBS&field2=0 HTTP/1.1
Host: api.thingspeak.com

value of gas sensor is: 688
AT+CIPSEND=86
GET /update?api_key=OMPQYIH9IVI2HZBS&field3=688 HTTP/1.1
Host: api.thingspeak.com

Distance: 97cm
AT+CIPSEND=85
GET /update?api_key=OMPQYIH9IVI2HZBS&field4=97 HTTP/1.1
Host: api.thingspeak.com
```

----------XXXXX----------