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 4		
Faculty	Dr. Karthikeyan B		

School of Electronics Engineering ,VIT, Vellore

1)

AIM →

To perform the following operation –

Consider the following four road junction for monitoring the traffic.

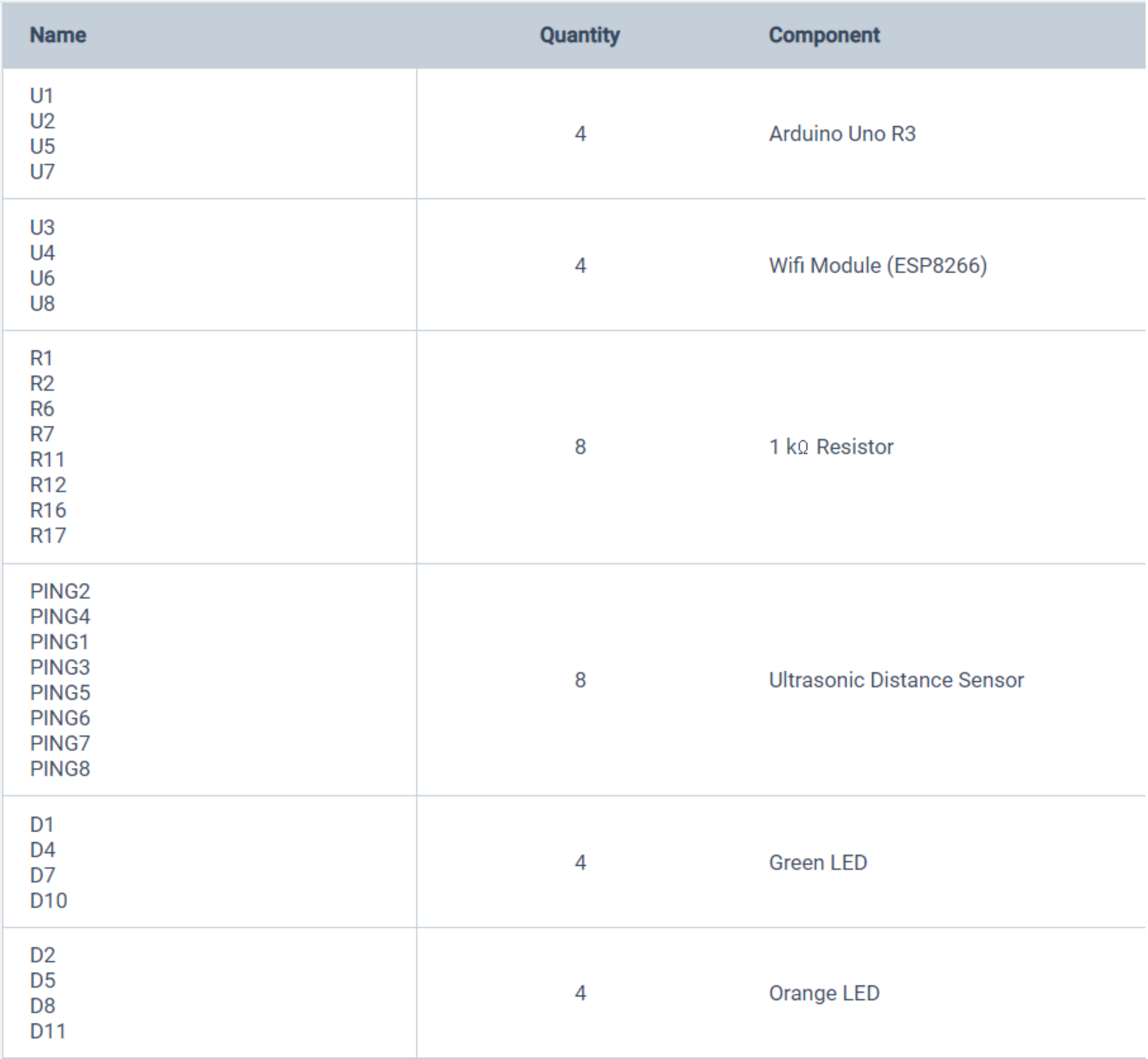
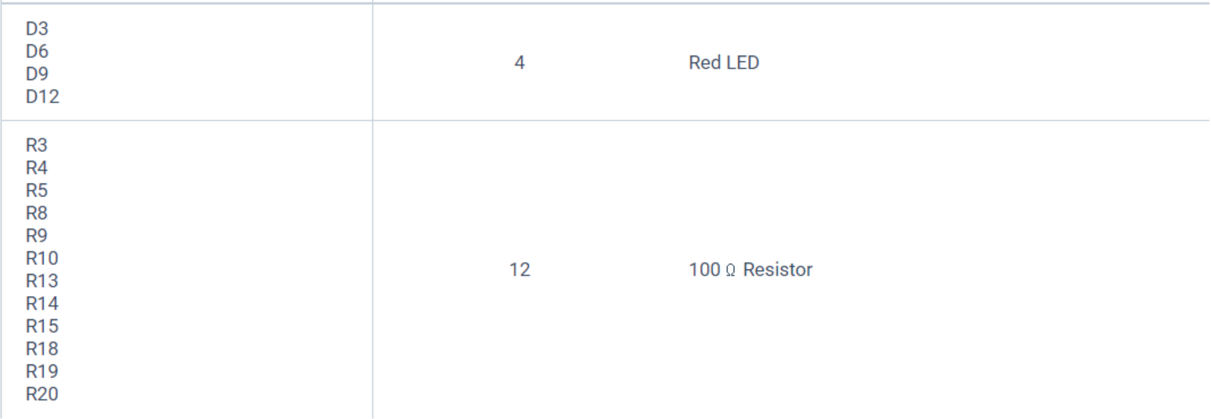
• Here each lane is having Two ultrasonic sensors[US_BEGIN & US_END] which are interfaced to ARDUINO.

• When RED light of the SIGNAL is ON then US_BEGIN should get ON. If US_BEGIN detects the car, it means a vehicle is arrived in the lane. The Arduino is waiting for US_END to detect a vehicle. US_END is placed some distance far from the US_BEGIN indicating that we are monitoring the traffic between US_BEGIN and US_END. If the US_END detects the vehicle for more than 15seconds continuously, then it is the situation that more vehicles arrived in the lane. Now this Arduino should stop signalling RED and change to ORNAGE and GREEN immediately as Traffic in this particular lane has reached the maximum limit[Traffic signal of other lane must change according this lane signal change from RED to GREEN]. The same setup and logic should follow for other LANES.

• How many times Lane has reached the maximum capacity must be sent to thingspeak.

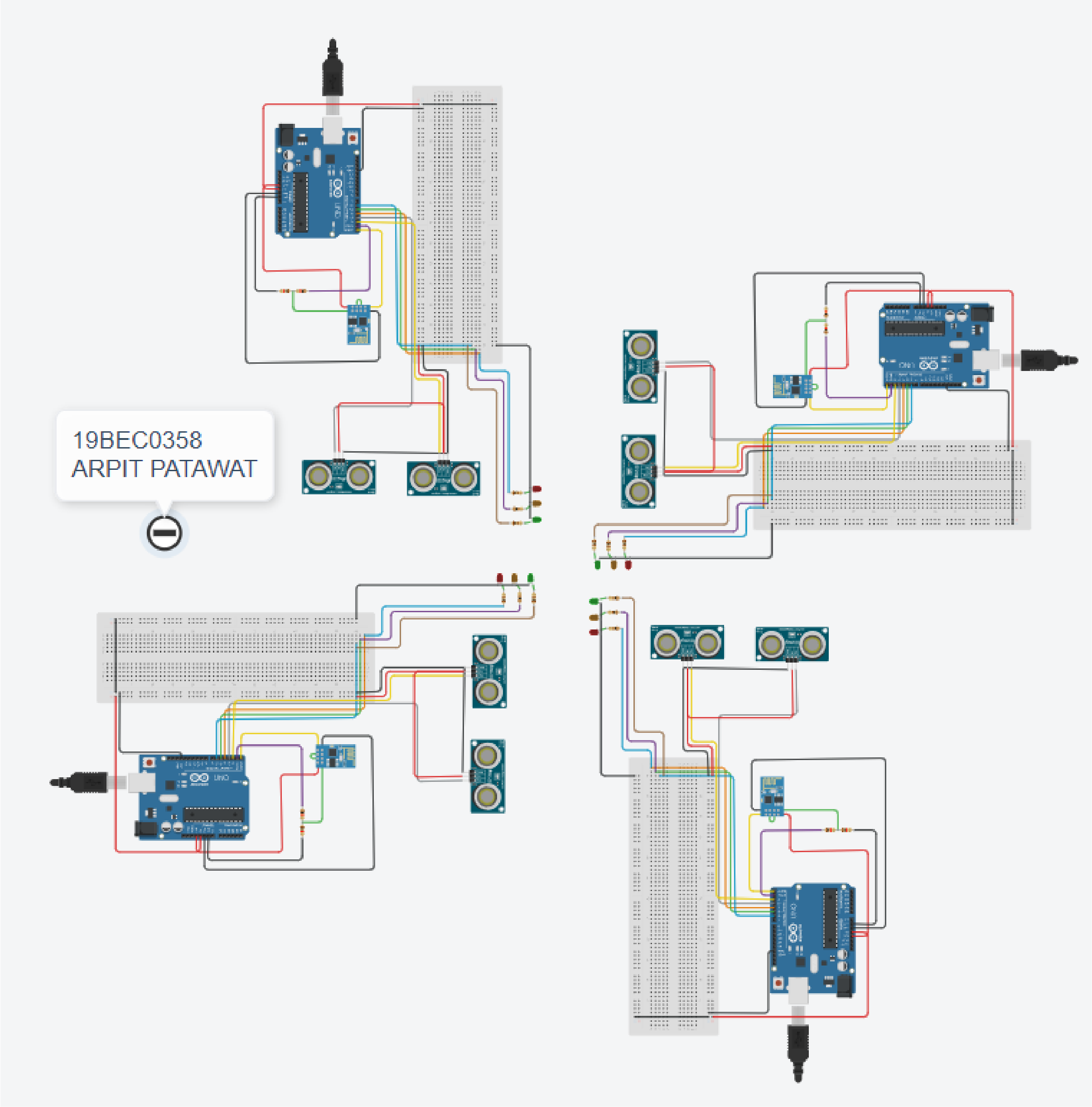
• Create four field in the THINGSPEAK for four LANE and indicate the number of times each lane reached maximum capacity. The design of the circuit may require more than 1 Arduino as there will be more Ultrasonic sensors[approx.8].

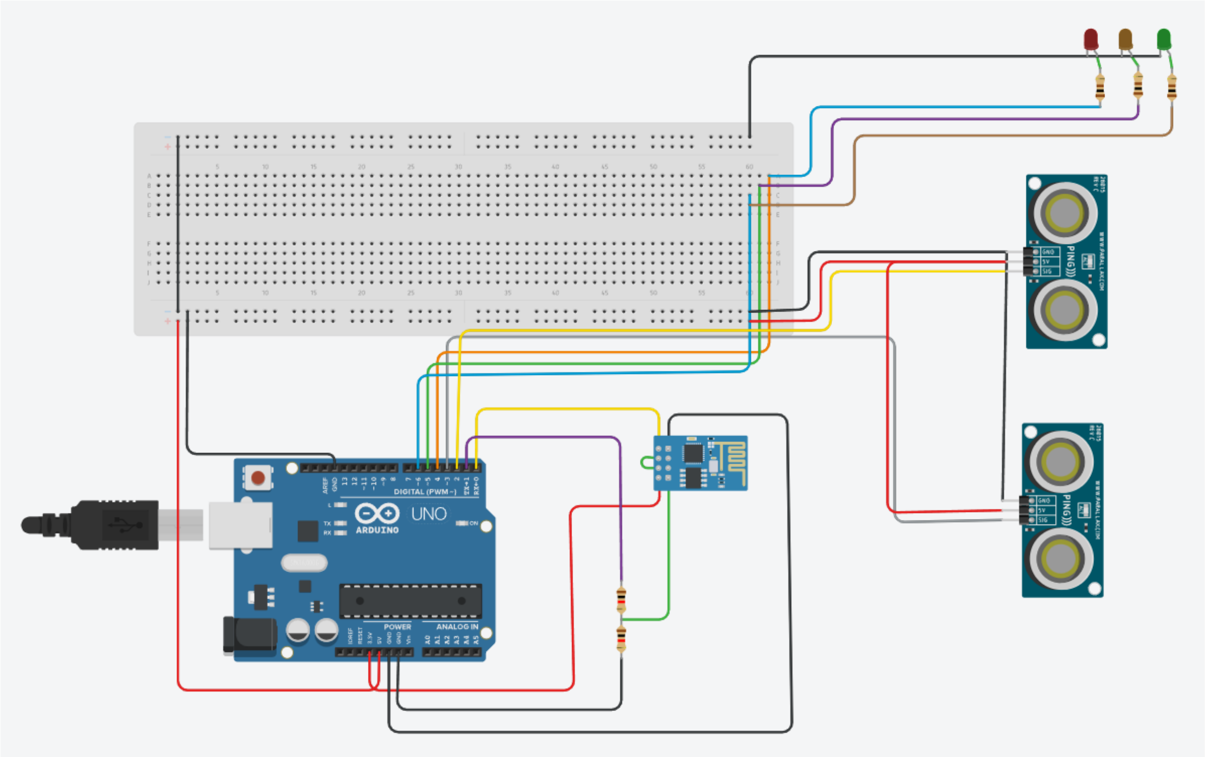
COMPONENT LIST→

Name	Quantity	Component
U1 U2 U5 U7	4	Arduino Uno R3
U3 U4 U6 U8	4	Wifi Module (ESP8266)
R1 R2 R6 R7 R11 R12 R16 R17	8	1 kΩ Resistor
PING2 PING4 PING1 PING3 PING5 PING6 PING7 PING8	8	Ultrasonic Distance Sensor
D1 D4 D7 D10	4	Green LED
D2 D5 D8 D11	4	Orange LED
D3 D6 D9 D12	4	Red LED
R3 R4 R5 R8 R9 R10 R13 R14 R15 R18 R19 R20	12	100 Ω Resistor

CIRCUIT DIAGRAM →



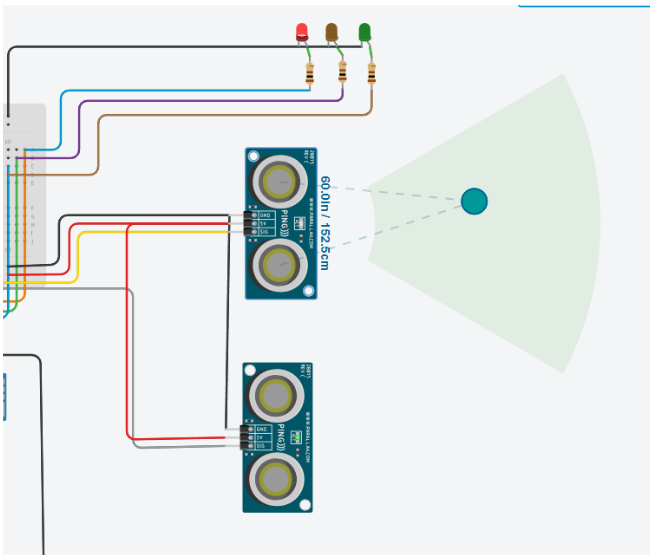


CODE → this code will be used for all the Arduino but with different field value

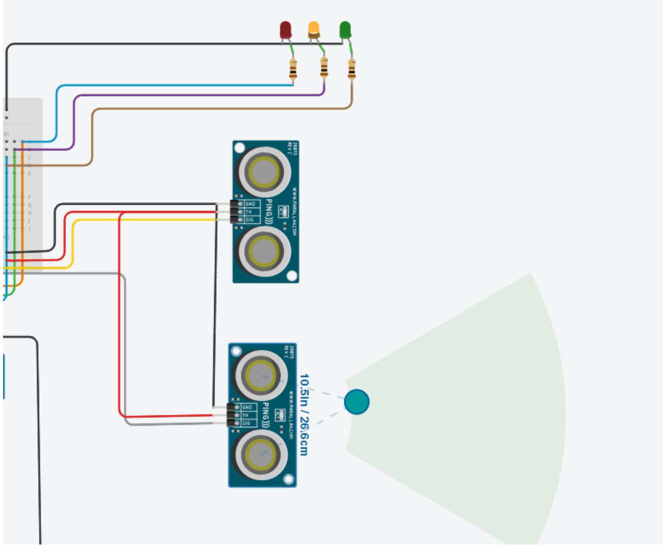
```
long duration;
int distancecm;
long distancein;
const unsigned long eventInterval = 15000;
unsigned long previousTime = 0;
int count;
String ssid     = "Simulator Wifi";  // SSID to connect to
String password = ""; // Our virtual wifi has no password
String host     = "api.thingspeak.com"; // Open Weather Map API
const int httpPort   = 80;
String url     = "/update?api_key=J9N71ZN8230QCGEA&field1=";
int cm = 0;

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 anydata(int num) {

  // Construct our HTTP call
  String httpPacket = "GET " + url + String(num) + " 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;


}


void setup() {
  Serial.begin(9600);
  setupESP8266();
  pinMode(2, OUTPUT); //signal for Us_BEGIN
  pinMode(6, OUTPUT); //green led
  pinMode(5, OUTPUT); //yellow led
  pinMode(4, OUTPUT); //red led
  pinMode(3,OUTPUT); //signal for US_END

}

void loop() {
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  float begin1 = 0.01741 * readUltrasonicDistance(2,2);
  if(begin1 < 100){
       float begin2 = 0.01741 * readUltrasonicDistance(3,3);
  if(begin2 < 100){
   Serial.println("both on");
  unsigned long currentTime = millis();
   while(begin2 < 100){
   if(millis() - currentTime >= eventInterval) {
    pass();
   }
    begin2 = 0.01741 * readUltrasonicDistance(3,3);
   }
  }
 }

}

long readUltrasonicDistance(int triggerPin, int echoPin)
{
 pinMode(triggerPin, OUTPUT);  // Clear the trigger
 digitalWrite(triggerPin, LOW);
 delayMicroseconds(2);
 // Sets the trigger pin to HIGH state for 10 microseconds
 digitalWrite(triggerPin, HIGH);
 delayMicroseconds(10);
 digitalWrite(triggerPin, LOW);
 pinMode(echoPin, INPUT);
 // Reads the echo pin, and returns the sound wave travel time in microseconds
 return pulseIn(echoPin, HIGH);
}

void pass(){
 count += 1;
 anydata(count);
digitalWrite(4,LOW);
digitalWrite(5,HIGH);
delay(3000);
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
delay(3000);
 }
```

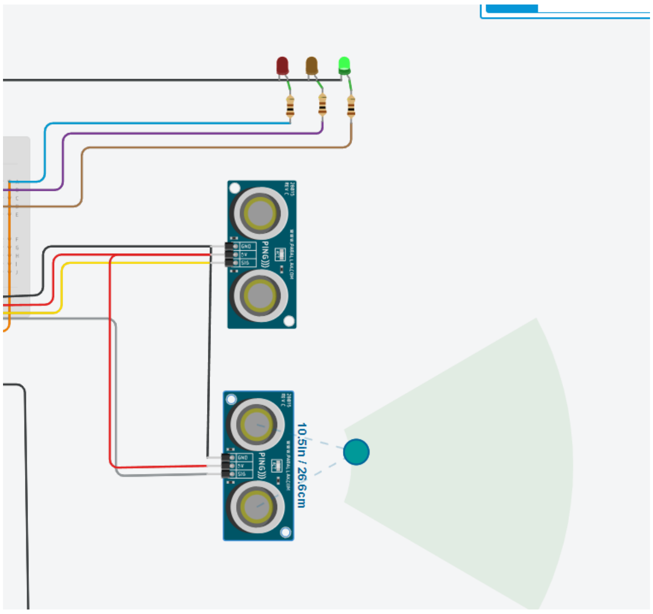
OUTPUT →

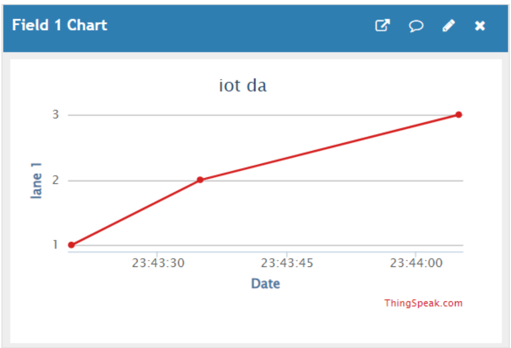
1. Initally light will be green and US_BEGIN will detect if there is a car within 100 cm of sensor if not then light will remain red

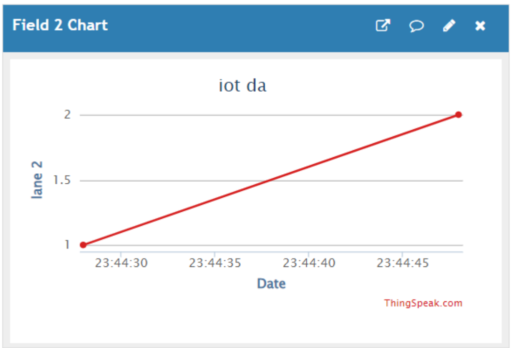


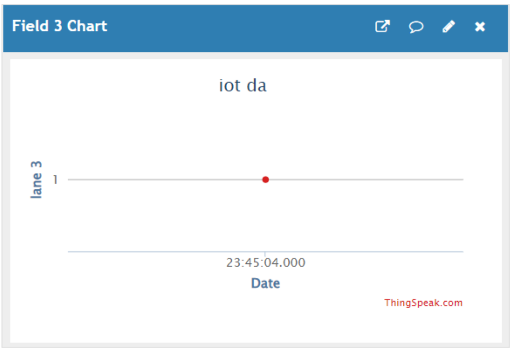
2. when US_BEGIN detect any car then US_END will start working and detect car withing 100 cm of sensor distance and if it detect any car then the signal will turn orange then green.

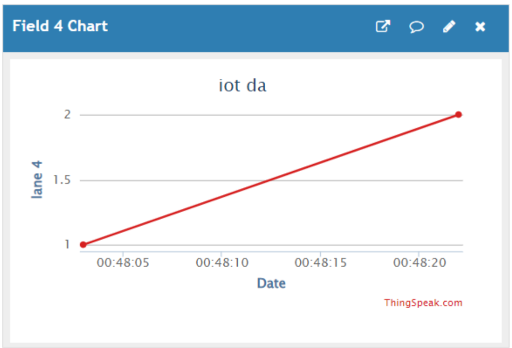





Field 1 Chart
iot da
lane 1
3
2
1
23:43:30
23:43:45
23:44:00
Date
ThingSpeak.com


Field 2 Chart
iot da
lane 2
2
1.5
1
23:44:30
23:44:35
23:44:40
23:44:45
Date
ThingSpeak.com


Field 3 Chart
iot da
lane 3
1
23:45:04.000
Date
ThingSpeak.com


Field 4 Chart
iot da
lane 4
2
1.5
1
00:48:05
00:48:10
00:48:15
00:48:20
Date
ThingSpeak.com

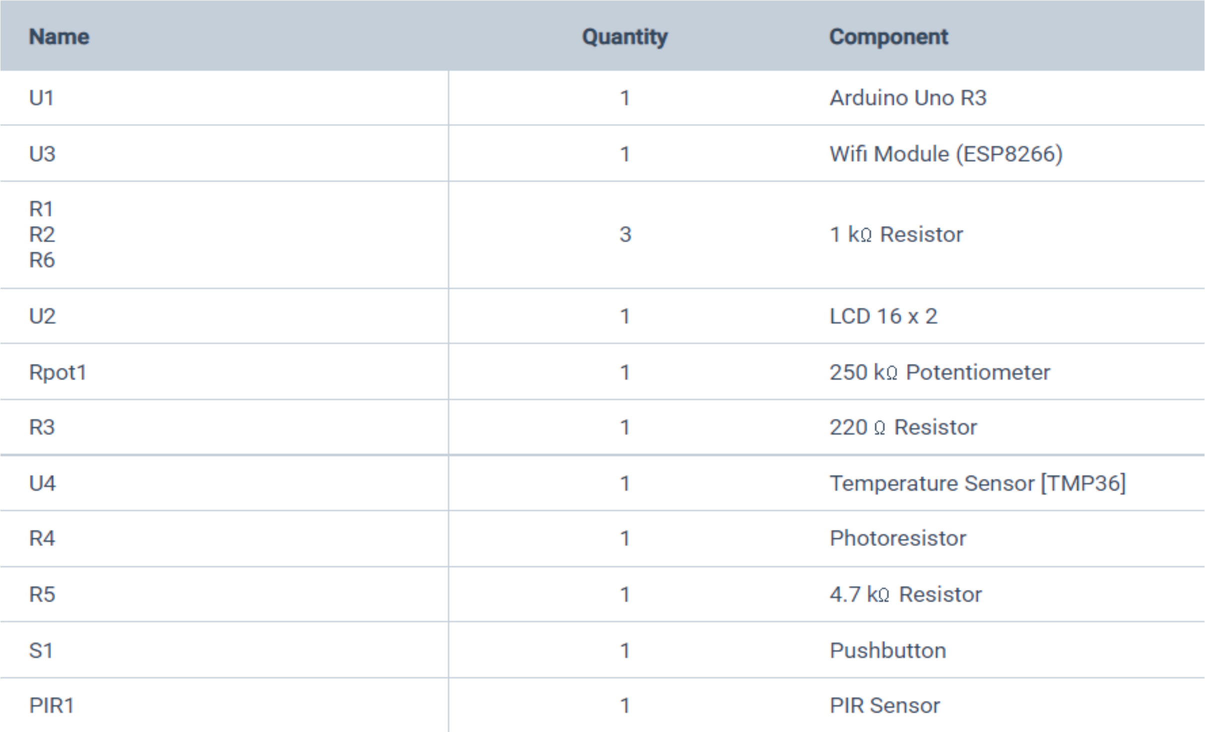
2)

AIM →

To perform the following operation –

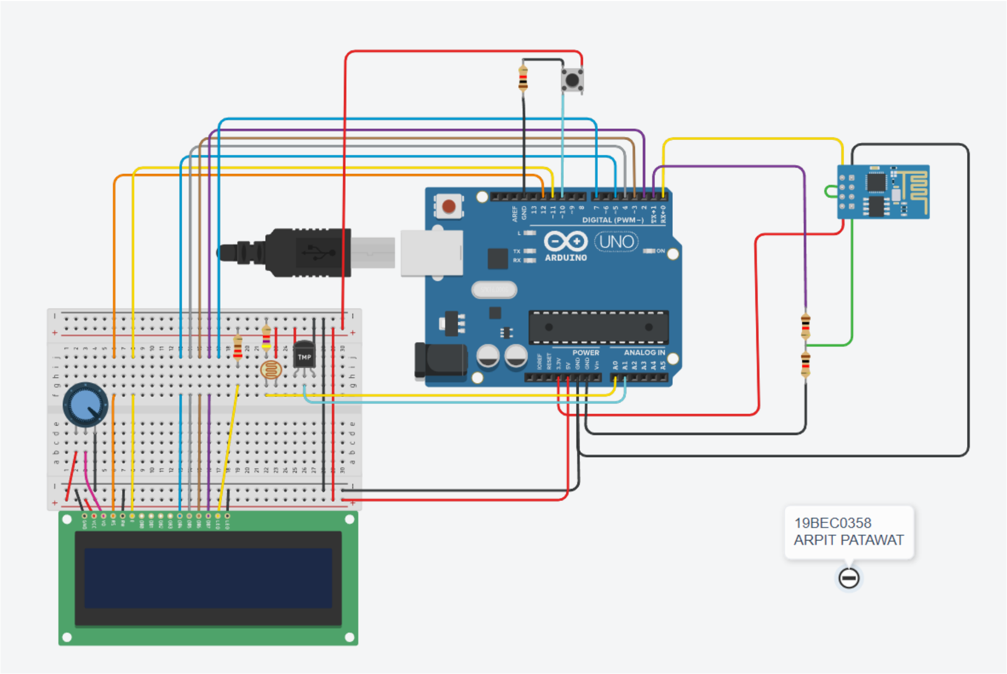
There are pulse sensor(photo-resister), Temperature sensor present in the circuit. Using this circuit, we can measure pulse and temperature of a person. It will ask to hold the button and wait for a while till the sensors capture your temperature of body and pulse. Send all the values to Thingspeak

COMPONENT LIST→



Name	Quantity	Component
U1	1	Arduino Uno R3
U3	1	Wifi Module (ESP8266)
R1 R2 R6	3	1 kΩ Resistor
U2	1	LCD 16 x 2
Rpot1	1	250 kΩ Potentiometer
R3	1	220 Ω Resistor
U4	1	Temperature Sensor [TMP36]
R4	1	Photoresistor
R5	1	4.7 kΩ Resistor
S1	1	Pushbutton
PIR1	1	PIR Sensor

CIRCUIT DIAGRAM →

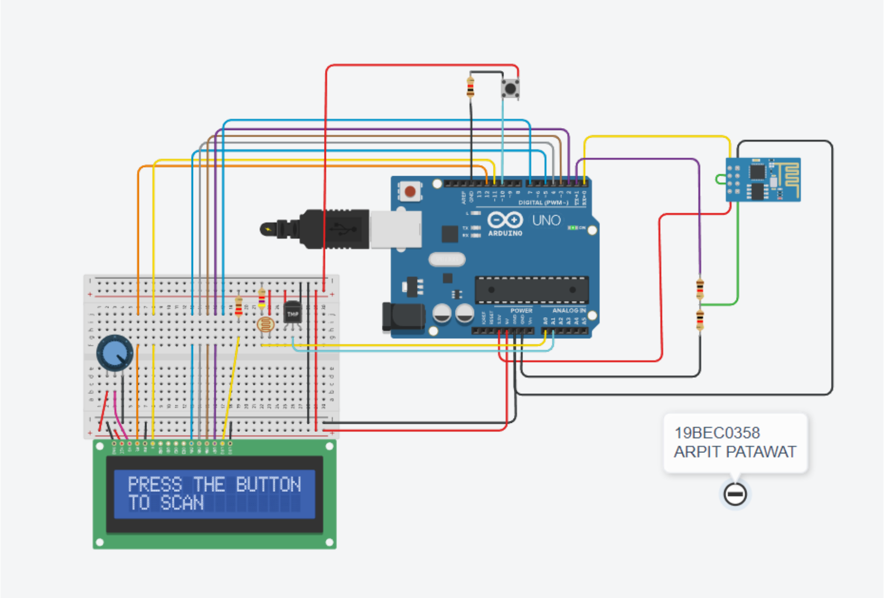


```
#include <LiquidCrystal.h>
#include <Servo.h>
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int sensorValue = 0;
int celsius = 0;
int fahrenheit = 0;
int button=0;

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=J9N71ZN8230QCGEA&field1=";
String url1     = "/update?api_key=J9N71ZN8230QCGEA&field2=";


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 anydata(int num) {
  // Construct our HTTP call
  String httpPacket = "GET " + url + String(num) + " 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;
}

void anydata1(int num) {
  // Construct our HTTP call
  String httpPacket = "GET " + url1 + String(num) + " 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;
}


void pulse(){
sensorValue = analogRead(A0);
Serial.print("pulse ");
Serial.println(sensorValue);
lcd.clear();
lcd.setCursor(0,0);
 lcd.print("Pulse - ");
 lcd.print(sensorValue);
 lcd.print(" BPM");
 anydata(sensorValue);
}

void temp(){
celsius = map(((analogRead(A1) - 20) * 3.04), 0, 1023, -40, 125);
fahrenheit = ((celsius * 9) / 5 + 32);
  Serial.print(celsius);
  Serial.println(" C");
  Serial.print(fahrenheit);
 Serial.println(" F");
lcd.setCursor(0,1);
```

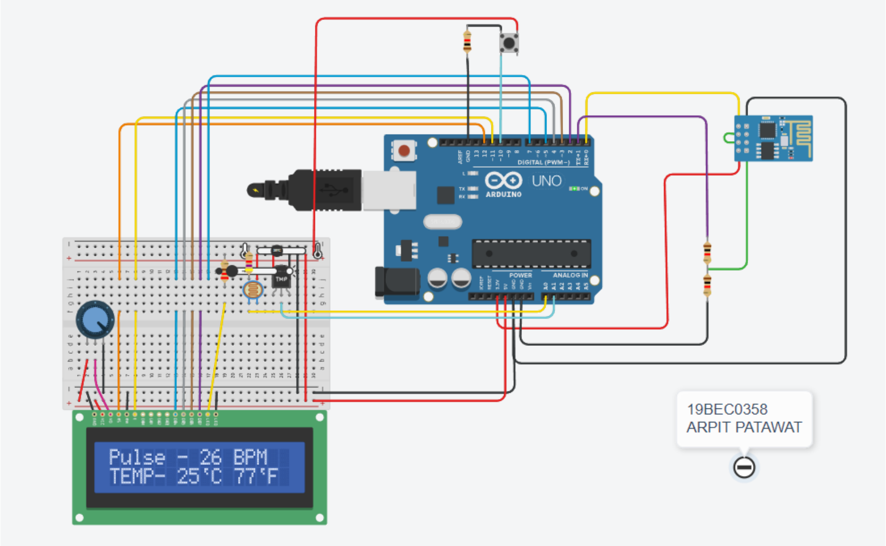
```
 lcd.print("TEMP- ");
 lcd.print(celsius);
 lcd.print("`C ");
 lcd.print(fahrenheit);
 lcd.print("`F");
 delay(15000);
 anydata1(celsius);
}

void btn(){
button=digitalRead(10);
}
void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
  lcd.clear();
  // Print a message to the LCD.
  lcd.print("PRESS THE BUTTON");
  lcd.setCursor(0,1);
  lcd.print("TO SCAN");
  setupESP8266();
  pinMode(A0, INPUT);
  pinMode(A1, INPUT);
  pinMode(10,INPUT);
}

void loop() {
 btn();
  if(button == HIGH){
  pulse();
  temp();
  }

}
```

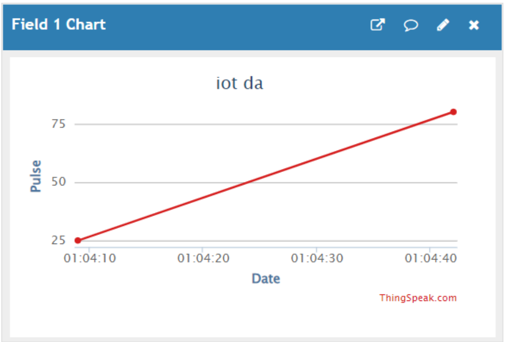
OUTPUT →

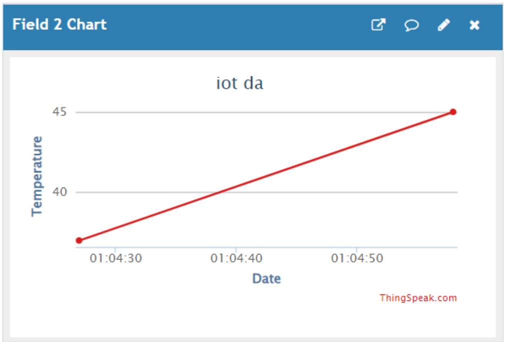
1. At the start , it waill ask for input



2.when we will press and hold the push button it will take the data and display it to lcd and also update to thingspeak







--XXXXX--