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 3		
Faculty	Dr. Karthikeyan B		

School of Electronics Engineering ,VIT, Vellore

1.

AIM →

To perform the following operation –

•Two ultrasonic sensors [US_ENTRY& US_EXIT] are interfaced to ARDUINO, one for entry and another for exit. •When US_ENTRY detects the car, servo motor needs to rotate 90 degree implies that the gate is opened.

•On the LCD, display must be as below for one vehicle entered the parking areao Total slot 10oAvailable slot 9

•Also the information regarding available slot for parking must be send to Thingspeak.

•When US_EXIT detects a car, then the following must be displayed from the previous display Total slot 10oAvailable slot 10

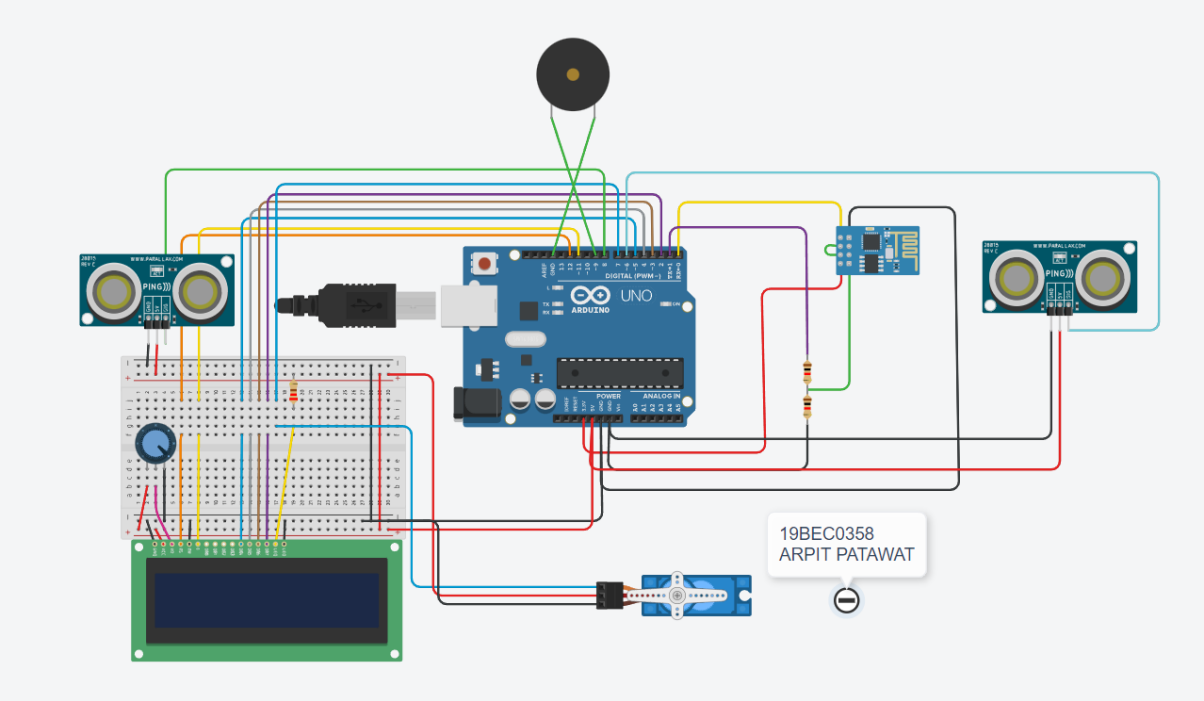
•Also the information regarding available slot for parking must be send to Thingspeak.

•After the 10thvehicle entered the parking area and if there is NO available parking slot, BUZZER must ON for random delay.

COMPONENT LIST→

Name	Quantity	Component
U3	1	Wifi Module (ESP8266)
R1 R2	2	1 kΩ Resistor
U4	1	Arduino Uno R3
U5	1	LCD 16 x 2
Rpot1	1	250 kΩ Potentiometer
R3	1	220 Ω Resistor
PING1 PING2	2	Ultrasonic Distance Sensor
SERVO1	1	Positional Micro Servo
PIEZO1	1	Buzzer [Piezo small]

CIRCUIT DIAGRAM →



CODE →

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

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 pos = 0;
Servo servo_9;
int cm = 0;
int empty = 10;

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;
```

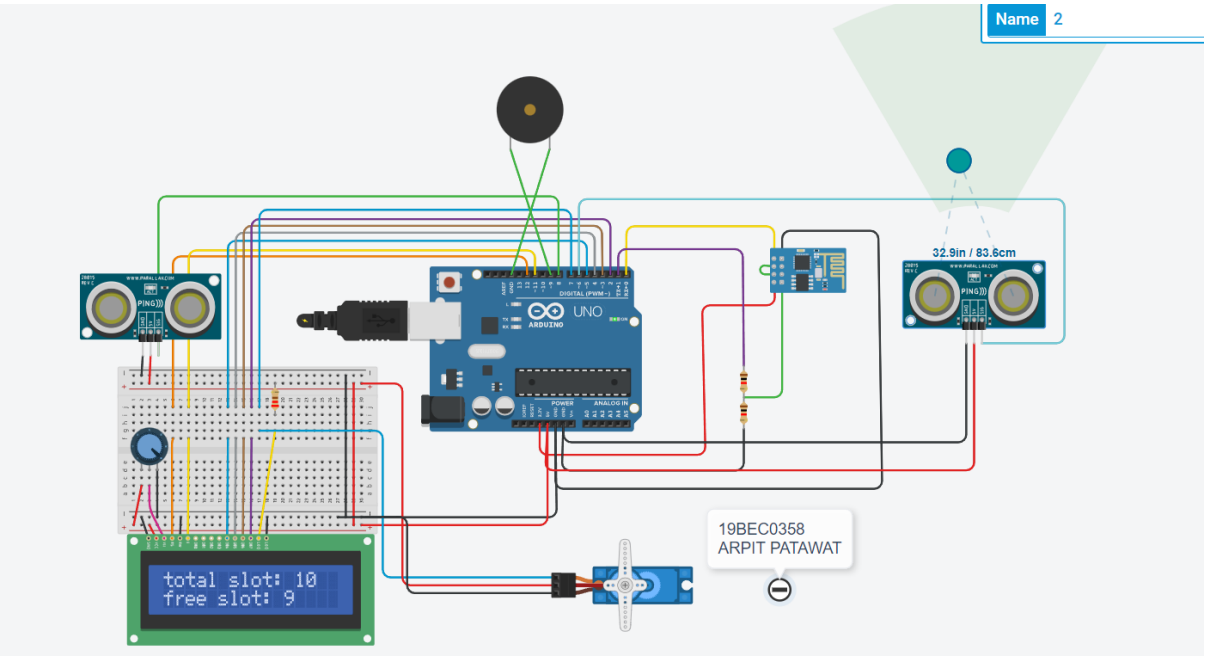
```
  // Open TCP connection to the host:
  Serial.println("AT+CIPSTART=\"TCP\",\"" + host + "\"," + httpPort);
  delay(50);       // Wait a little for the ESP to respond
  if (!Serial.find("OK")) return 3;

  return 0;
}

void anydata(void) {
  // Construct our HTTP call
  String httpPacket = "GET " + url + String(empty) + " 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;


}

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 ultrasonicin()
{
  // measure the ping time in cm
  if (empty < 1) {buzzer(); goto label;}
  cm = 0.01723 * readUltrasonicDistance(6,6);
  lcd.setCursor(0, 1);
```

```
  if(cm < 100){

          empty = empty - 1;

    lcd.print("free slot: ");
    lcd.print(empty);
    for (pos = 0; pos <= 90; pos += 1) {
    // tell servo to go to position in variable 'pos'
    servo_9.write(pos);
    // wait 15 ms for servo to reach the position
    delay(15); // Wait for 15 millisecond(s)
  }
    anydata();
    delay(10000);
    //delay(5000);
  for (pos = 90; pos >= 0; pos -= 1) {
    // tell servo to go to position in variable 'pos'
    servo_9.write(pos);
    // wait 15 ms for servo to reach the position
    delay(15); // Wait for 15 millisecond(s)
  }

  }
  label: delay(100); // Wait for 100 millisecond(s)
}


void ultrasonicout()
{
  // measure the ping time in cm
  cm = 0.01723 * readUltrasonicDistance(8,8);
  lcd.setCursor(0, 1);
  if(cm < 100){
    empty = empty + 1;
    if (empty>10) {empty = 10;}
    anydata();
    lcd.print("free slot: ");
    lcd.print(empty);
    for (pos = 0; pos <= 90; pos += 1) {
    // tell servo to go to position in variable 'pos'
    servo_9.write(pos);
    // wait 15 ms for servo to reach the position
    delay(15); // Wait for 15 millisecond(s)
  }

    delay(10000);
  for (pos = 90; pos >= 0; pos -= 1) {
```

```
   // tell servo to go to position in variable 'pos'
   servo_9.write(pos);
   // wait 15 ms for servo to reach the position
   delay(15); // Wait for 15 millisecond(s)
  }

  }
  delay(100); // Wait for 100 millisecond(s)
}

void buzzer(){
  digitalWrite(9,HIGH);
  delay(1000);
  digitalWrite(9,LOW);
  delay(1000);
}
void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("total slot: 10");
  servo_9.attach(7, 500, 2500);
  servo_9.write(0);
  setupESP8266();
  pinMode(9,OUTPUT);
}

void loop() {
  ultrasonicin();
  ultrasonicout();
}
```

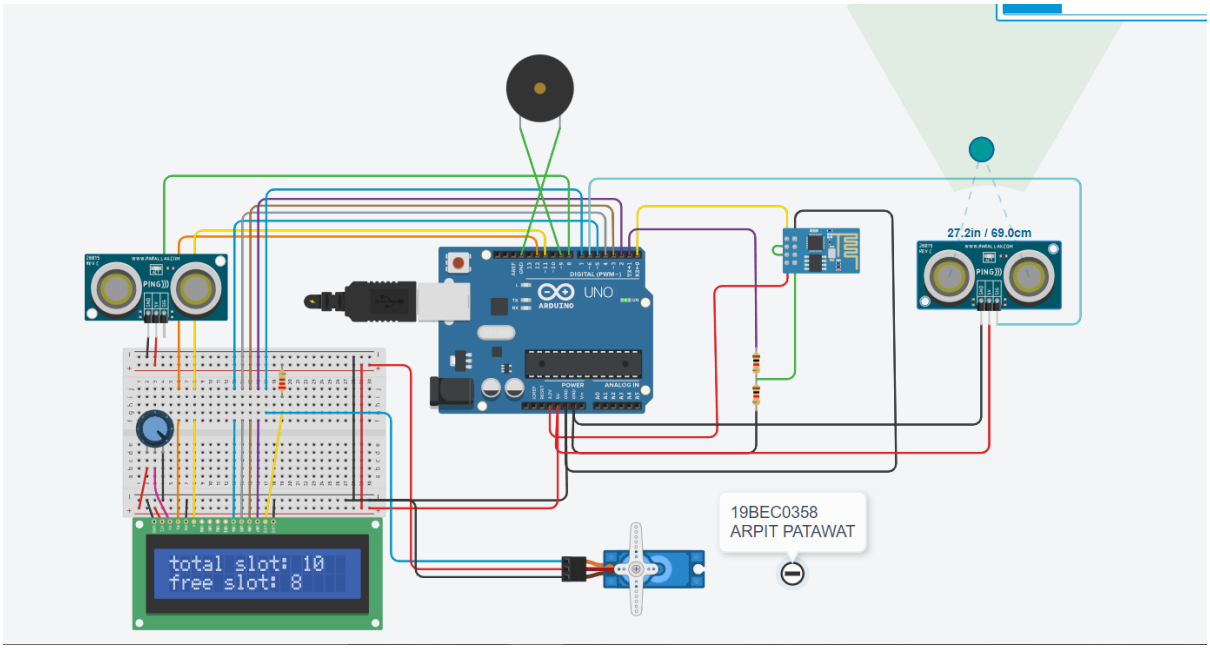
OUTPUT →

1.WHEN A CAR ENTERS THROUGH ENTRY, SERVO WILL ROTATE AND AVAILABLE SLOT = 10

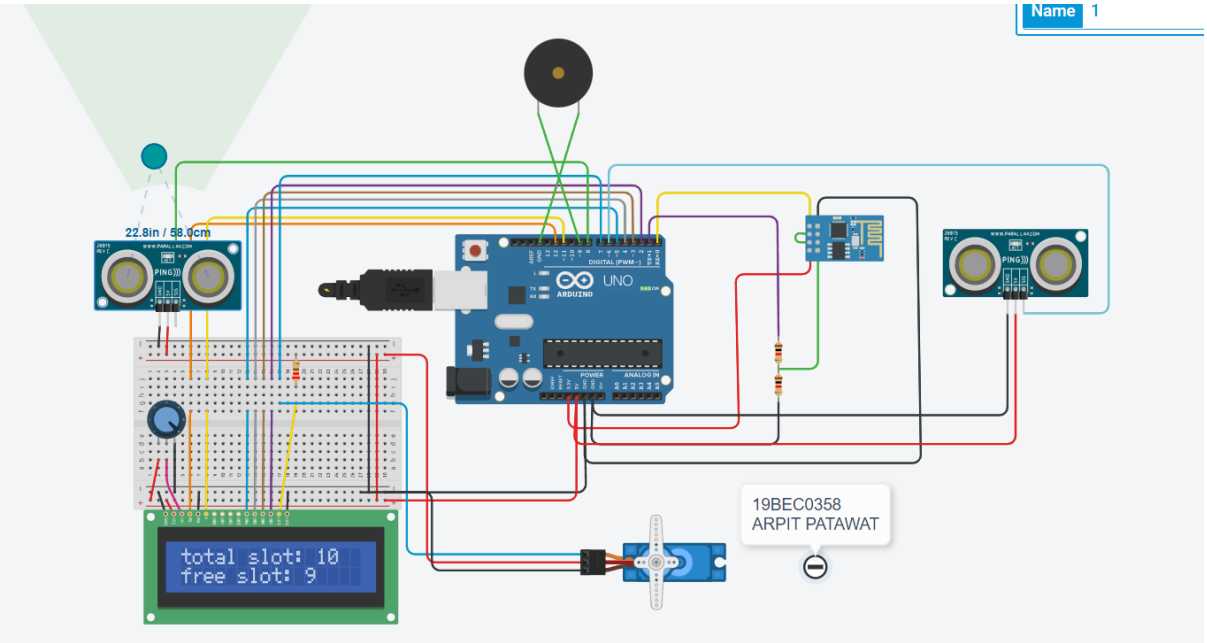


Servo motors rotate for 90 degree and till the time car enters, we send data to Thingspeak then servo motor again rotate anti clockwise.

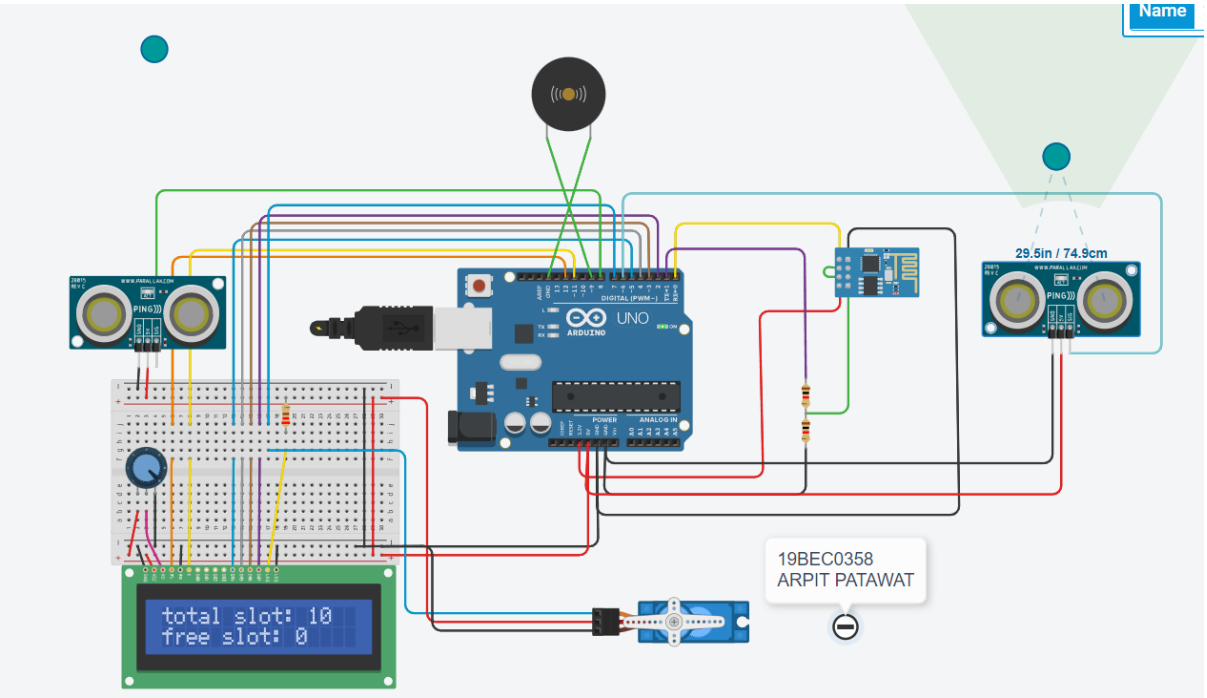
2. WHEN ANOTHER CAR ENTERS –



3. WHEN 1 CAR LEAVES –



4. WHEN ALL SLOTS ARE FULL –



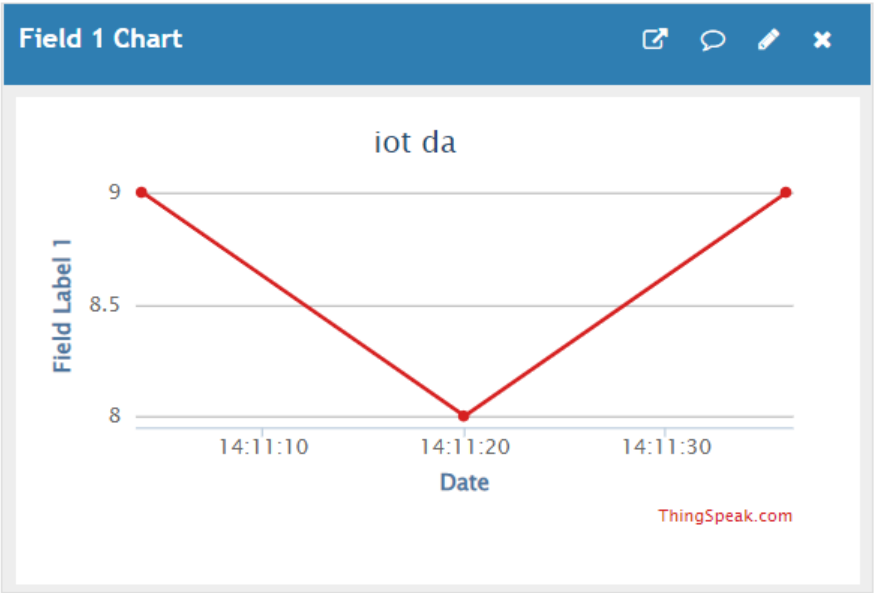
Buzzer will start making sound.

Serial Monitor

```
AT+CIPSEND=84
GET /update?api_key=J9N71ZN8230QCGEA&field1=9 HTTP/1.1
Host: api.thingspeak.com

AT+CIPSEND=84
GET /update?api_key=J9N71ZN8230QCGEA&field1=8 HTTP/1.1
Host: api.thingspeak.com

AT+CIPSEND=84
GET /update?api_key=J9N71ZN8230QCGEA&field1=9 HTTP/1.1
Host: api.thingspeak.com
```

Channel Stats

Created: about 2 hours ago
Last entry: less than a minute ago
Entries: 3



2.

AIM →

To perform the following operation –

•When human is moving near toa PIR sensor, the present and next LEDS GLOW (street lights are personified as LEDs here)

•The system also enables effective use of energy as lights “on” only when object detected by the PIR.

•Count of number of objects passed according to time are sent to THINGSPEAK

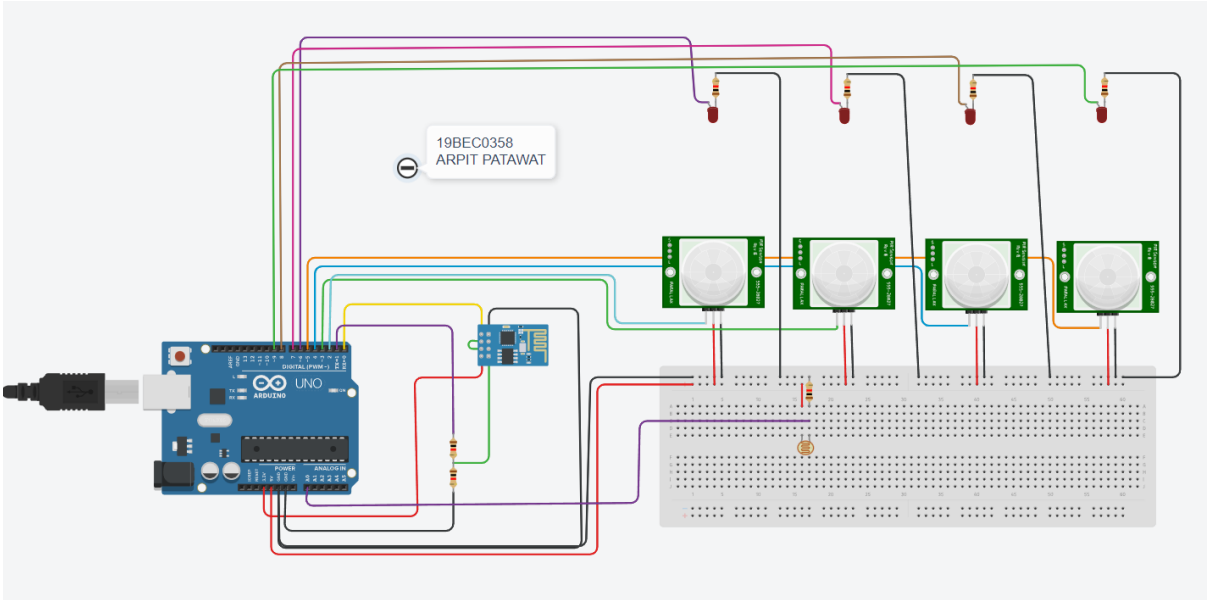
•When LDR is low => it is night => circuit works, When LDR is high => it is day => no use of circuit (no use of streetlights under sun)

•The above steps is automatically switching ON and OFF the LEDS [street light]and sending the count to thingspeak.

COMPONENT LIST→

Name	Quantity	Component
U1	1	Arduino Uno R3
U3	1	Wifi Module (ESP8266)
R1 R2 R5 R6 R7 R8 R10	7	1 kΩ Resistor
PIR1	1	26.566443338987597 , -156.50161399688898 , -148.18208781680295 PIR Sensor
PIR2	1	-2.0744346787018912 , -145.50078528746764 , -142.69918171207178 PIR Sensor
PIR3	1	16.45679110208414 , -147.7961253898643 , -158.9571684756836 PIR Sensor
PIR4	1	-22.776333312469887 , -157.74535739963716 , -167.33970472665453 PIR Sensor
D1 D2 D3 D4	4	Red LED
R9	1	Photoresistor

CIRCUIT DIAGRAM →



CODE →

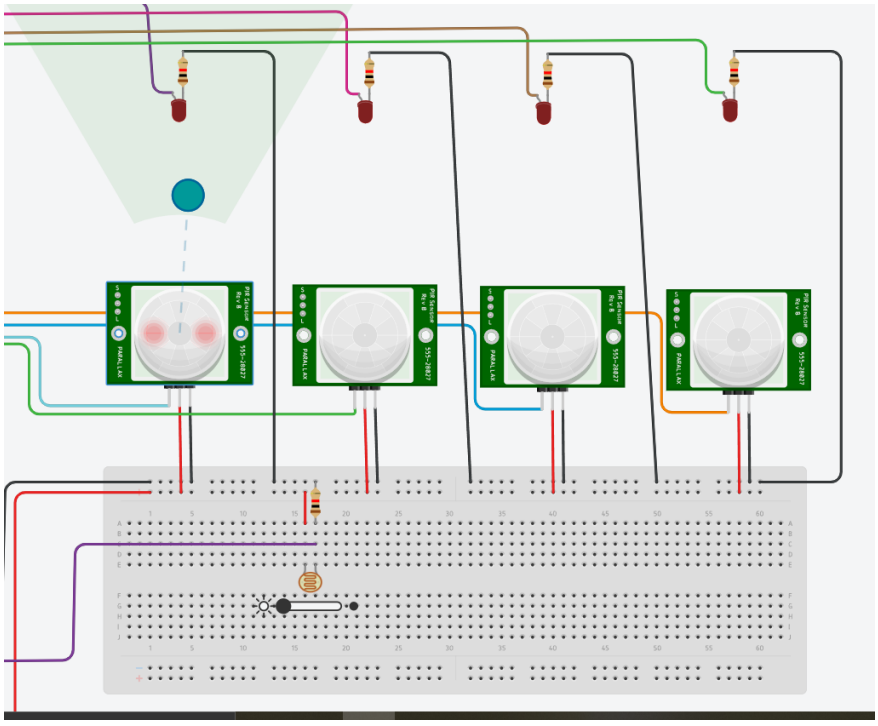
```
//Make sure to subscribe Technomekanics:)
String ssid     = "Simulator Wifi";  // SSID to connect to
String password = ""; // Our virtual wifi has no password
String host     = "api.thingspeak.com"; // Open Weather Map API
const int httpPort   = 80;
String url     = "/update?api_key=KZOIMGT4UKD66G5B&field1=";

int sensorState_1 = 0;
int sensorState_2 = 0;
int sensorState_3 = 0;
int sensorState_4 = 0;
int sensorValue;
int count;
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(void) {



  // Construct our HTTP call
  String httpPacket = "GET " + url + String(count) + " 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(115200);
          setupESP8266();
          pinMode(2, INPUT);
          pinMode(3, INPUT);
          pinMode(4, INPUT);
          pinMode(5, INPUT);
          pinMode(6,OUTPUT);
    pinMode(7,OUTPUT);
    pinMode(8,OUTPUT);
    pinMode(9,OUTPUT);
          pinMode(A0,INPUT);


}

void loop() {
  sensorValue = analogRead(A0);
```

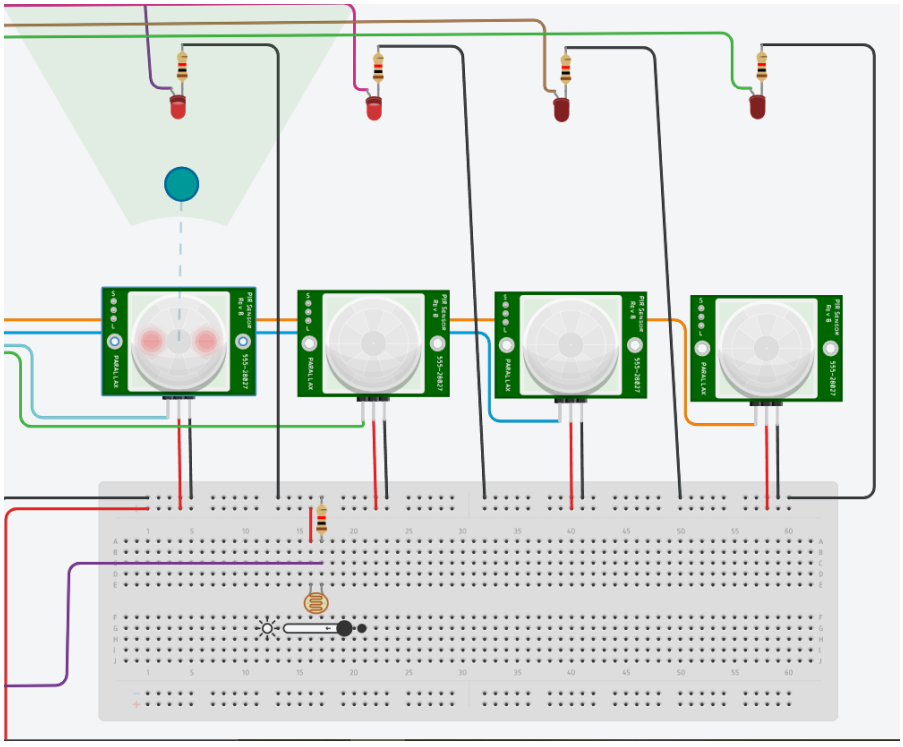
```
  Serial.println(sensorValue);
  if(sensorValue > 515) {
    allow();
    goto label; }
  sensorState_1 = digitalRead(2);
  sensorState_2 = digitalRead(3);
  sensorState_3 = digitalRead(4);
  sensorState_4 = digitalRead(5);
  if(sensorState_1 == HIGH) {
    digitalWrite(6, HIGH);
          digitalWrite(7, HIGH);
          digitalWrite(8, LOW);
    digitalWrite(9, LOW);
  }

  else if (sensorState_2 == HIGH) {
    digitalWrite(6, LOW);
          digitalWrite(7, HIGH);
          digitalWrite(8, HIGH);
    digitalWrite(9, LOW);
  }

  else if(sensorState_3 == HIGH) {
    digitalWrite(6, LOW);
          digitalWrite(7, LOW);
          digitalWrite(8, HIGH);
    digitalWrite(9, HIGH);
  }

  else if (sensorState_4 == HIGH) {
    digitalWrite(6, LOW);
          digitalWrite(7, LOW);
          digitalWrite(8, LOW);
    digitalWrite(9, HIGH);
    count++;
    anydata();
  }
  label: delay(1000);
 }

void allow(){
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  digitalWrite(8, LOW);
  digitalWrite(9, LOW);
}
```

OUTPUT →

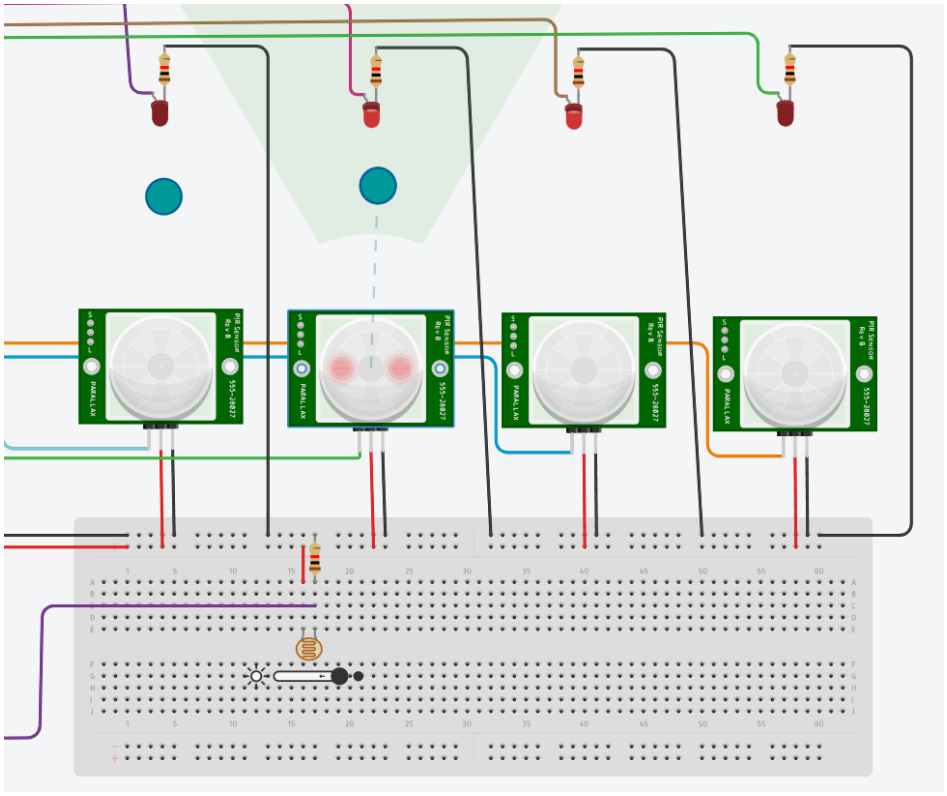
1.When its daytime then all the lights will remain off →

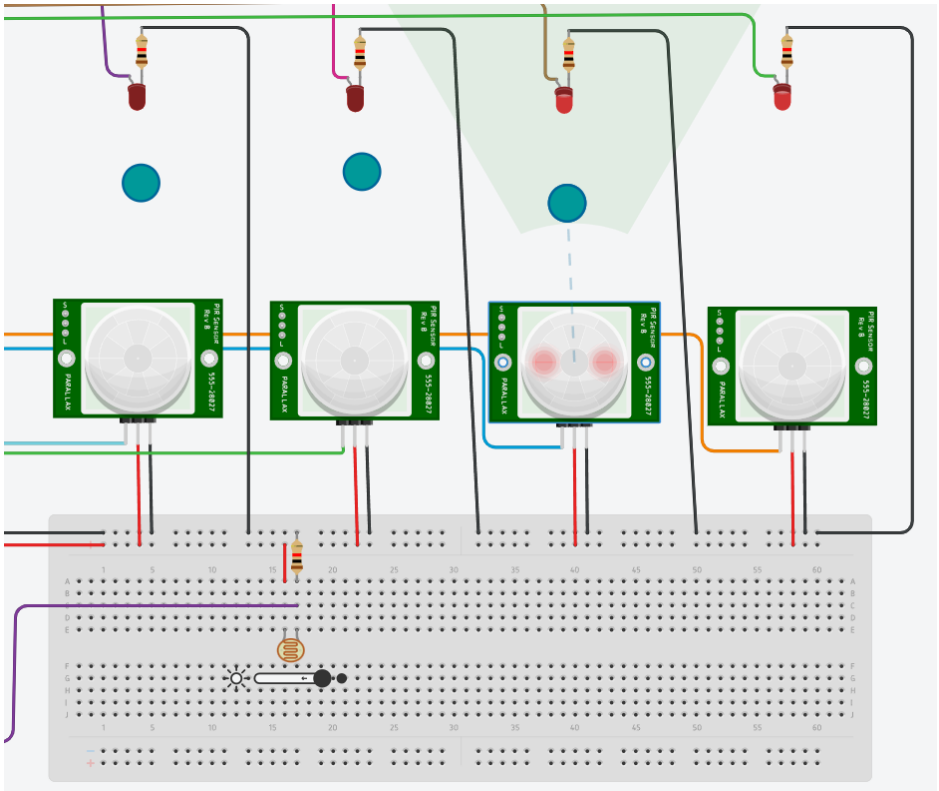


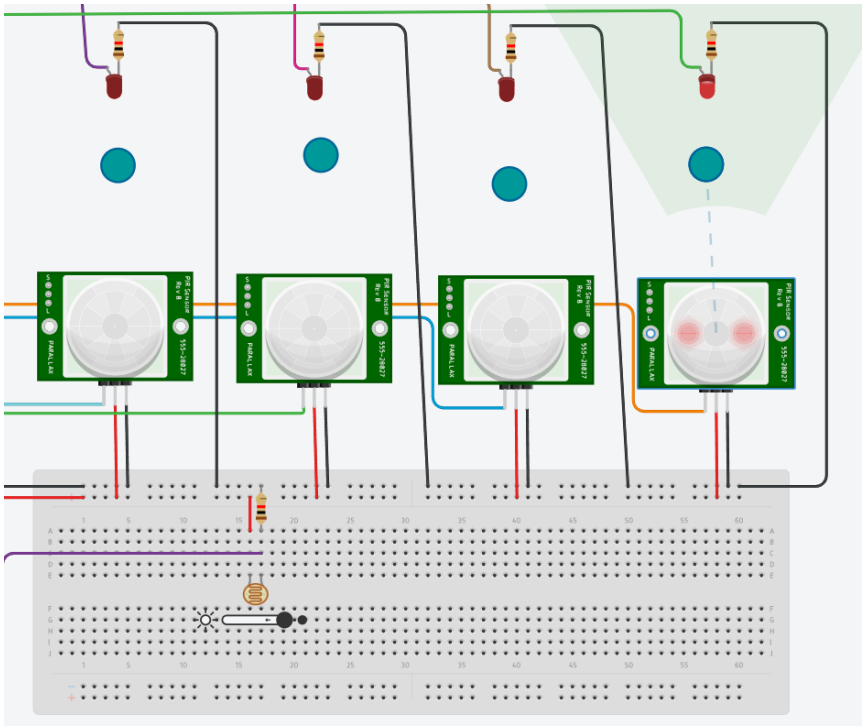
2.in night time, whenever vehicle passes through first PIR sensor then first and second light will turn on

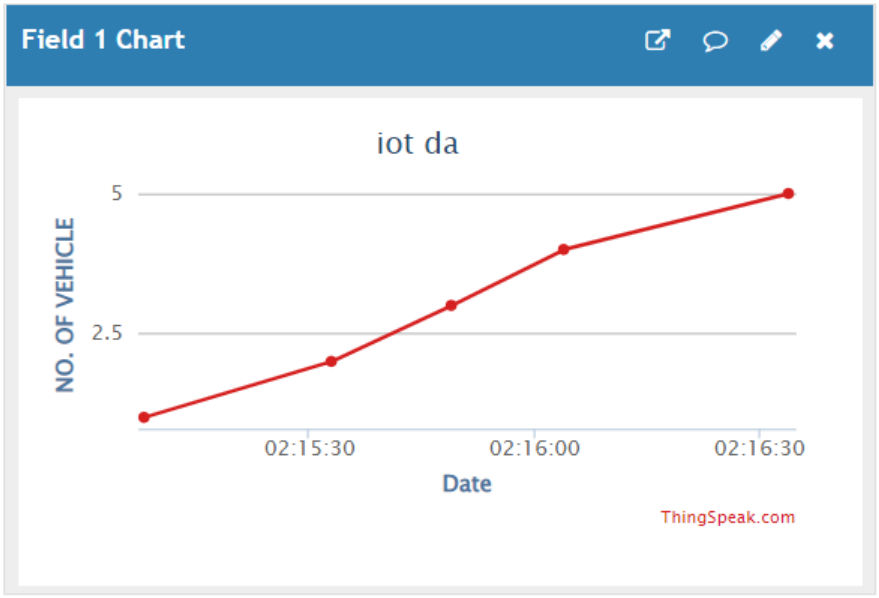


3. same goes for 2nd, 3rd and 4th sensor








Field 1 Chart
iot da
5
2.5
NO. OF VEHICLE
02:15:30
02:16:00
02:16:30
Date
ThingSpeak.com

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