CS578 Project: Smart Light Using IOT

Due on Wednesday, September 29, 2022

Siddharth Charan 190101085

Main objective

Main objective of assignment is to create smart light using IOT. This project is for learning purpose and main aim to incorporate as much IOT components as possible.

Smart Light Controller Working plan and purpose:
As we know that mostly light is controlled by switch on and off buttons, which user need to press in order to change state of light from on to off and vice versa. This leads to some disadvantages, if user is not willing to switch off lights either he/she has no time or he/she tends to ignore it. It leads to excessive use of electricity.

Hence; we are trying to solve this problem using IOT. We will create a light controlling system which will sense the environment using 2 sensors and will measure if:
1.It is day or night(How much light is already in room)
2, Is someone in room(Is need for Light is there or not?)

And based on that if controller/system finds need to turn light on it, will turn light on automatically. And when there is no need it will automatically turn lights off.

Implemented Attributes

From assignment we were asked to perform following tasks, first I have listed them and then I have shown how I have done each of these steps in next part of this section:

1. **These objectives/attributes/features should be implemented:**
a. Sensor should sense the environment.
b. Actuator should act upon receiving any command.
c. Internet connectivity to access the cloud.
d. Data visualization through LCD/cloud application.
e. Cloud application / In-home developed application to control the full system

2.At least these required hardware software components should be present in the implementation:
a.Sensor and Actuator
b.Arduino / NodeMCU / any other MCU
c.ESP8266 / ESP32 / Ethernet shield / any other IoT networking device(but not smartphone/laptop)
d.Any could application for storing data, running data analysis, giving command to actuators, etc.
e.Smartphone / PC / Laptop / Tablet will be used as Router / Gateway to connect with the Internet3

3. At least the following data should be implemented.
a.Reading Sensor Data and Display in Computer console/LCD e.g. Sensors -> Arduino -> Arduino serial monitor / LCD
b.Transmit the data to cloud storage using Internet and display it using cloud application

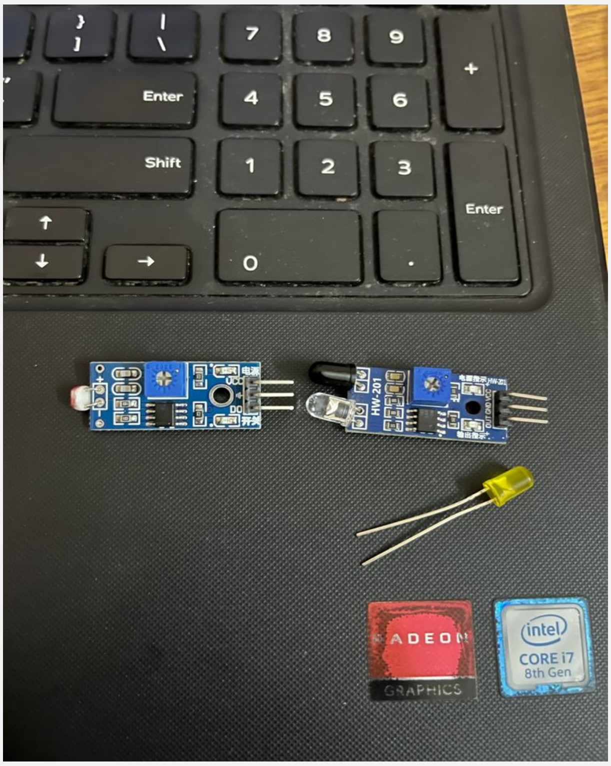
e.g. Arduino -> Router/Gateways (e.g. Smartphone) -> Cloud -> Smartphone or Laptop using cloud application to display
c.Command should come from the cloud application to act on any actuator e.g. Cloud application -> Router/Gateways -> NodeMCU -> Arduino -> Actuators
d.Sending notification/alert to administrator/user's email or smartphone e.g. Cloud application -> Internet -> Smartphone/ Email(to inform any message/information)

I worked on to implement each of these attributes.
1. The objetive/features implemented as asked in assignment:
(a). Sensor should sense the environment:
For this 2 sensors are used
● LDR Sensor: Used for measuring intensity of light around us.
● IR Sensor: Used for measuring any stir in environmnt.

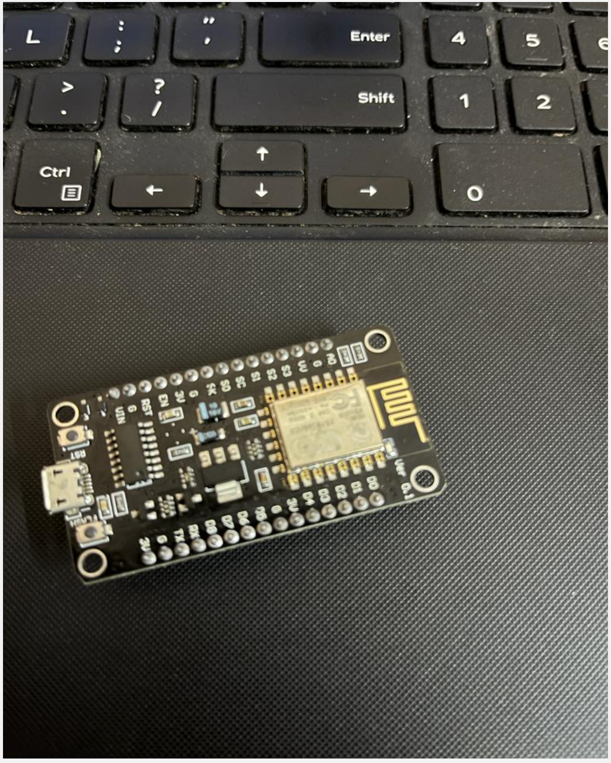
b. Actuator should act upon receiving any command.
Here; whenever there is need of light; lights are turned on, hence working like actuators.
c.Internet connectivity to access the cloud.
Here, we are connected to internet using ESP8266. Hence; this part is also done.
d. Data visualization:
Data visualization is done through cloud application. There; we can visualize if light is on or not and if it is in on state then; what is the value of light it is generating. It is being visualized using a graph, which is constantly updated after some time period.

e. Cloud application to control the full system:
For this again we have used the same application, which enables us to turn on and off lights manually, and hence it has control over the system.

2. These hardware and software components should are there in implementation, which assignment demanded:
A. Sensor and actuator:
As mentioned above we used sensor namely LDR and Ultrasonic. And LED is used as actuator.



B. Arduino/NodeMCU/Other MCU :
I used NodeMCU for this purpose.



C. Networking device:

ESP8266 was used for this purpose, it is already inbuilt in NodeMCU.

D. Cloud application:
I used "Thingspeak" library and cloud application.

E. Router
In demo I used PC but anything like Mobile or Table can be used also.

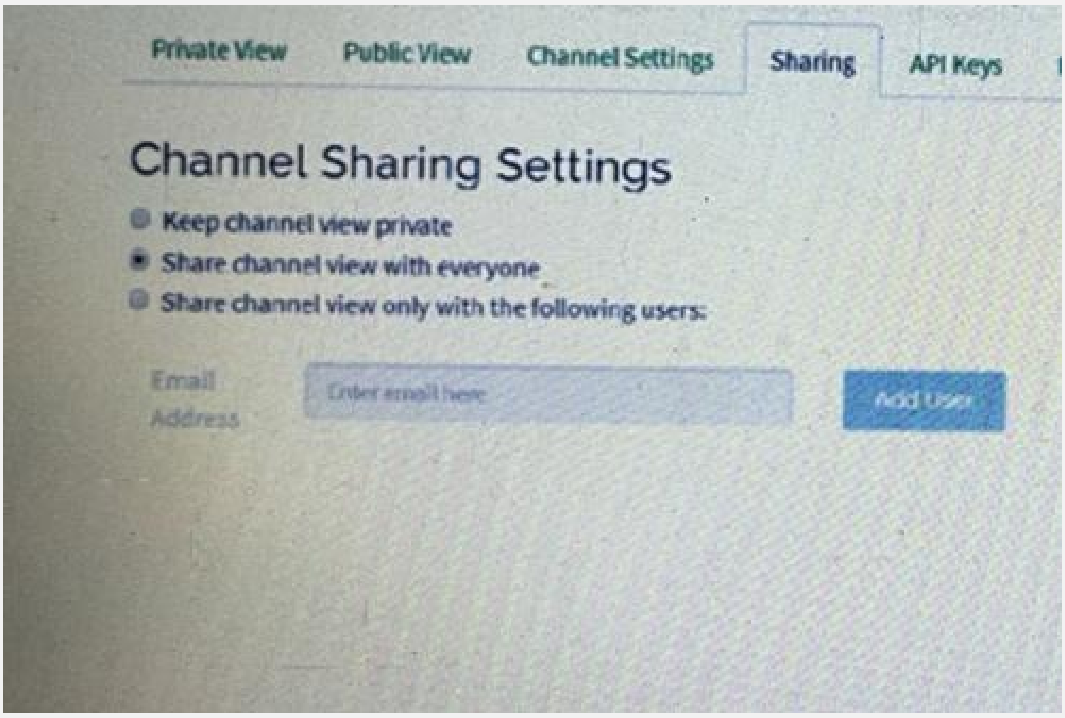
3. These data flows have been implemented:
(a) Reading data and display it:
I am reading data from sensors and actually displaying it in serial monitor.(Diagram given in next section)

(b) Transmit the data:
I am actually transmitting the data from Arduino to thingspeak using inbuilt ESP8266 in NodeMCU and using API I am showing the data on display about LED states.

(c) Then; from cloud, based on that data LED stated will be changed. For this we created new field in thingsstate, which represents led light, here if its value is 0. LED light will be turned off, else it will be turned on.

```
Mon = ThingSpeak.readIntField(myChannelNumber, 5, myReadAPIKey);
if(Mon==1) digitalWrite(led1,HIGH);
else digitalWrite(led1,LOW);
```

(d) Explicit notification system to user/administrator is not given. But; **he/she can easily see it using cloud application as it has a system to view details about process from anywhere on any device**.



Channel Sharing Settings

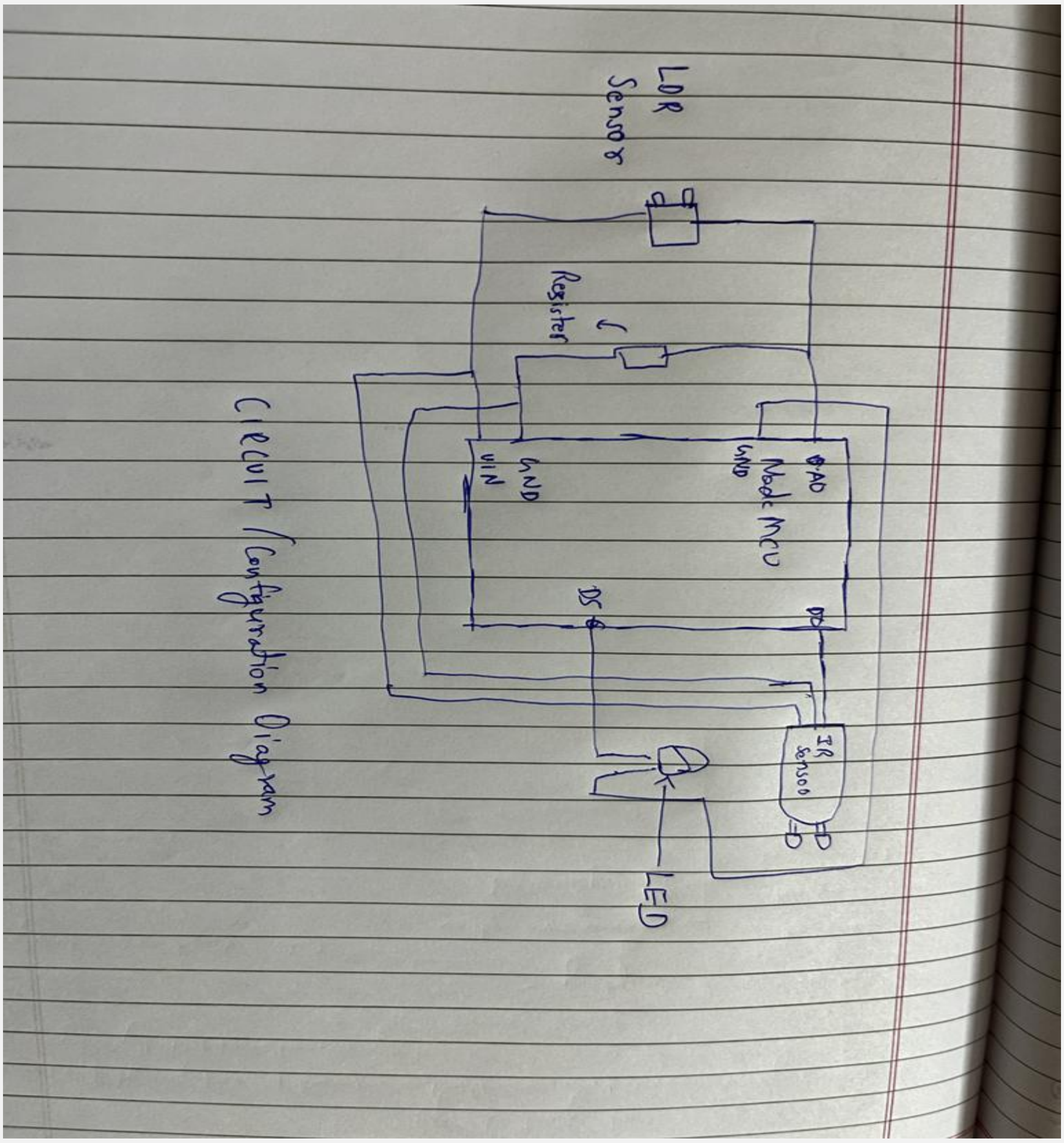
○ Keep channel view private

● Share channel view with everyone

○ Share channel view only with the following users:

Configuration Diagrams

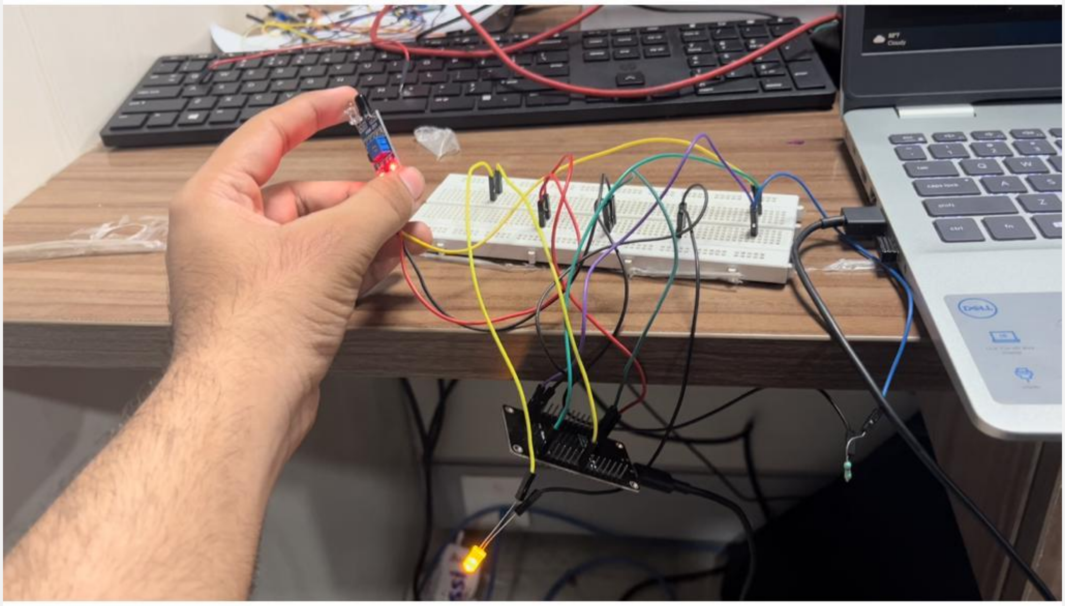
Configuration diagram is as following:



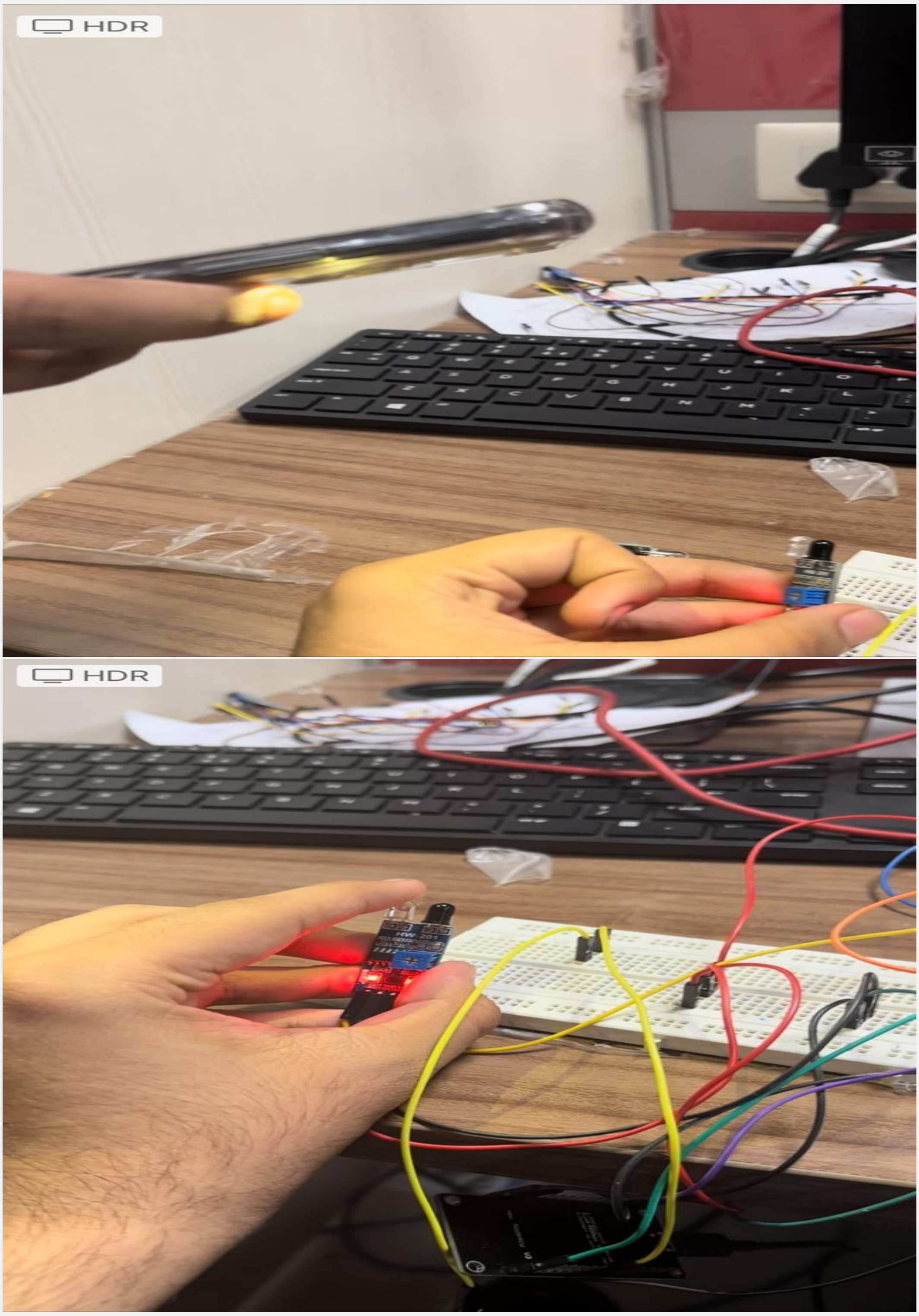
Sample Outputs

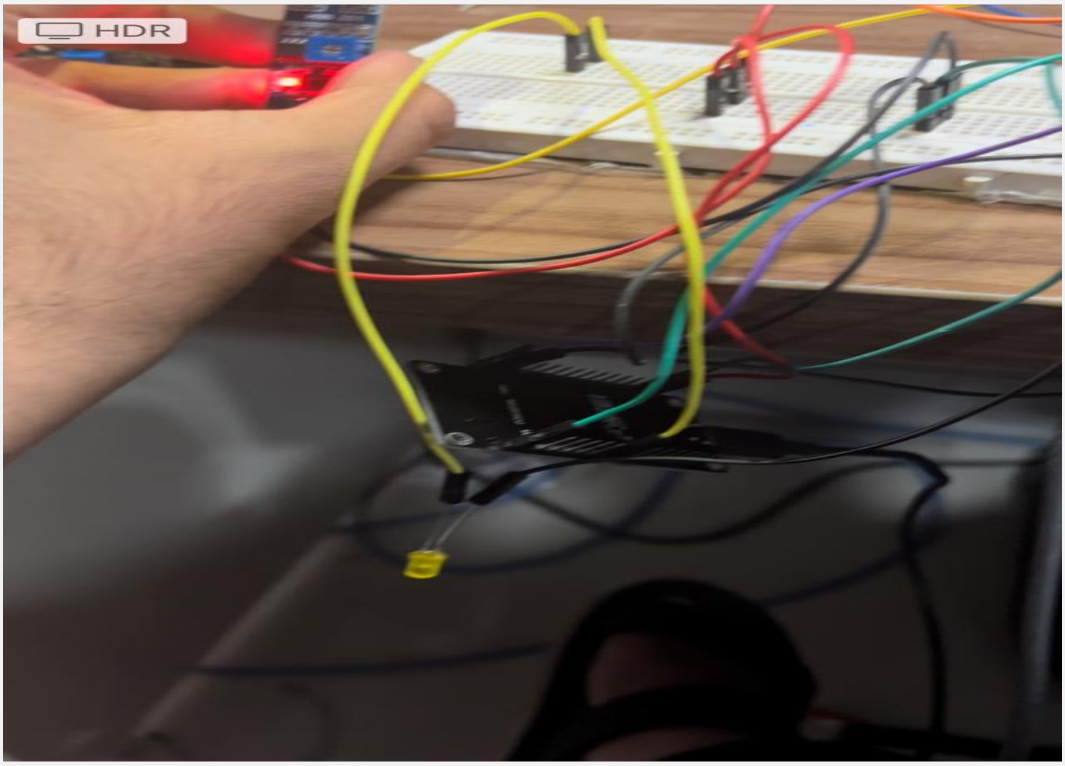
So, main output's are like below:
(1) light is off when IR sensor doesn't see anything:

(2) Light in on condition when sensor sees something:



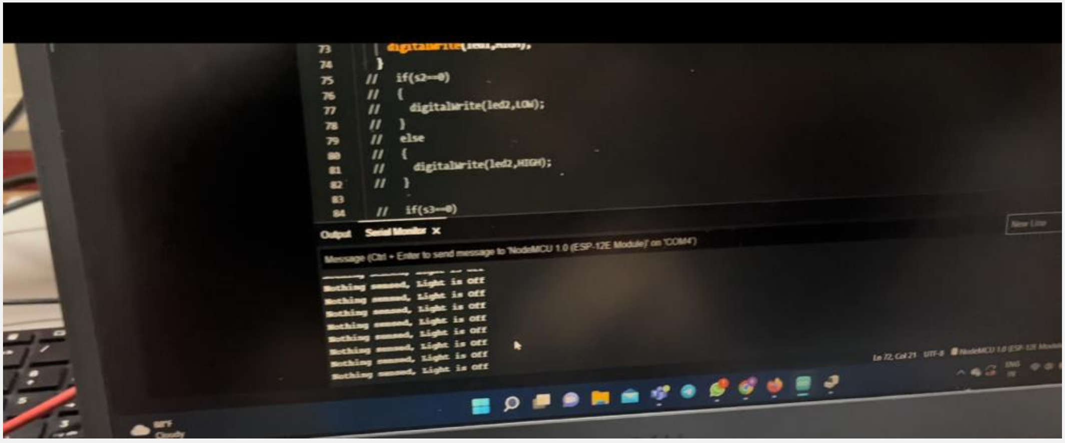
(3) Light is off even when IR sensor sees something, but ldr sensor senses light greater then the limit.


HDR
HDR
HW-201



(4) serial Monitor Output.

(a) When Light is off:



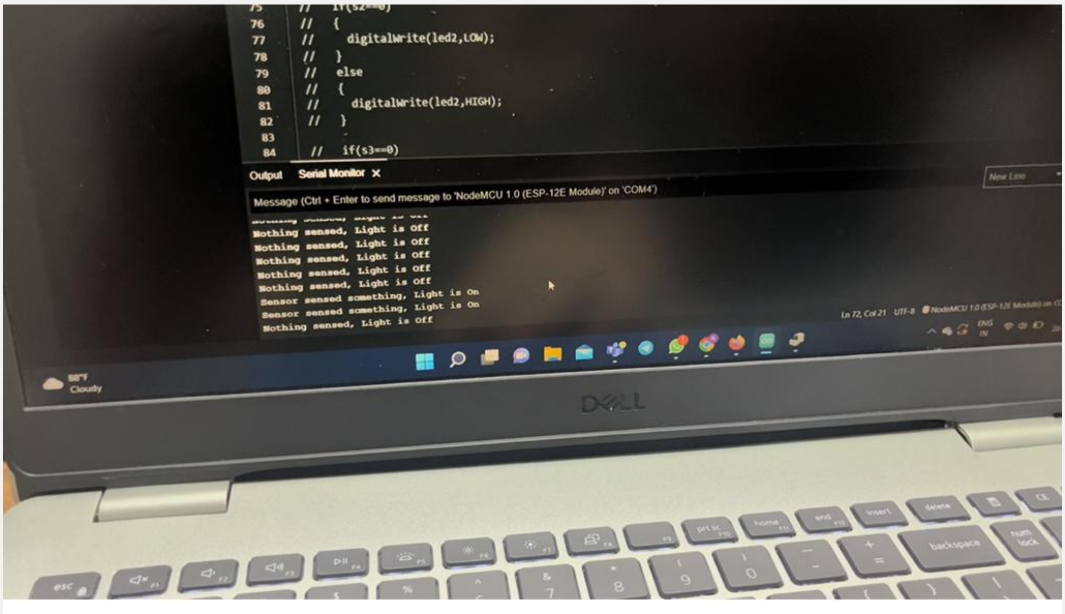
```
    digitalWrite(led1,HIGH);
  }
//  if(s2==0)
//  {
//    digitalWrite(led2,LOW);
//  }
//  else
//  {
//    digitalWrite(led2,HIGH);
//  }

//  if(s3==0)
```

Message (Ctrl + Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM4')

```
Nothing sensed, Light is off
Nothing sensed, Light is off
Nothing sensed, Light is off
Nothing sensed, Light is off
Nothing sensed, Light is off
Nothing sensed, Light is off
Nothing sensed, Light is off
Nothing sensed, Light is off
```

(b) When Light is on:



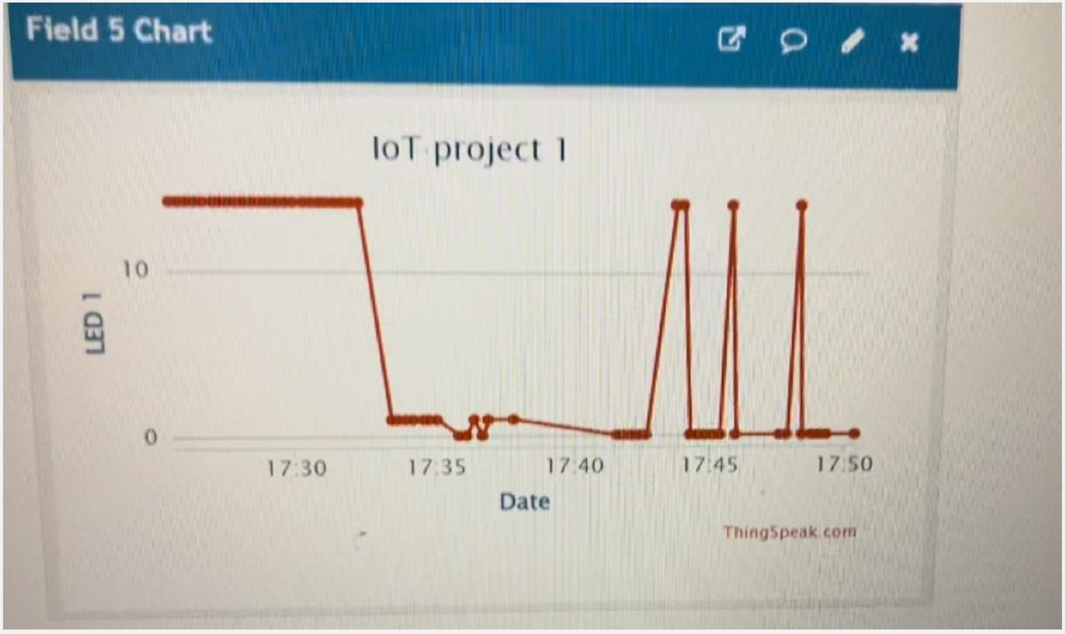
```
//   if(s2==0)
//   {
//     digitalWrite(led2,LOW);
//   }
//   else
//   {
//     digitalWrite(led2,HIGH);
//   }

//   if(s3==0)
```

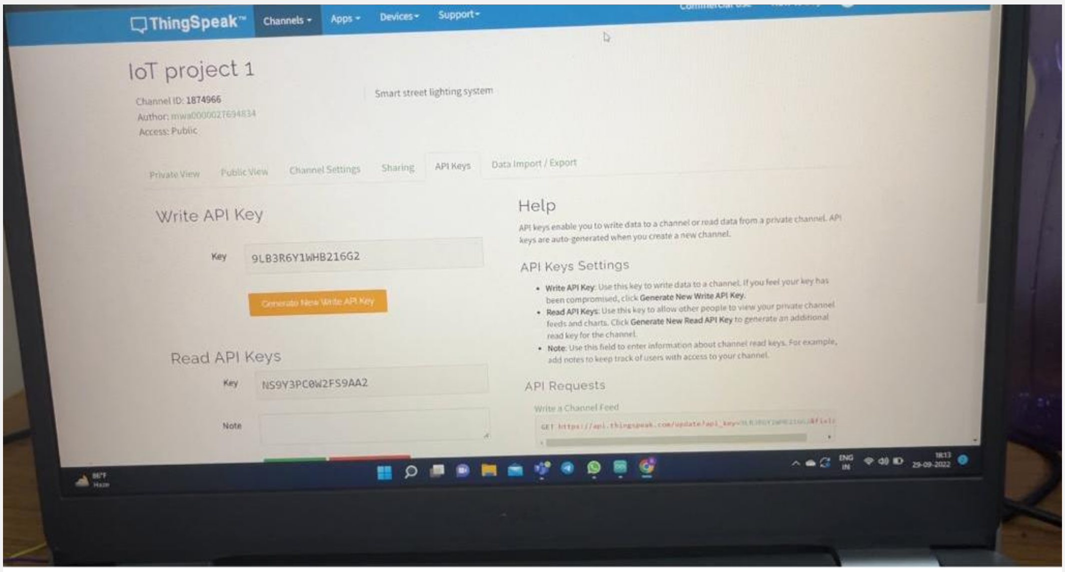
Output
Serial Monitor
Message (Ctrl + Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM4')

```
Nothing sensed, Light is off
Nothing sensed, Light is off
Nothing sensed, Light is off
Nothing sensed, Light is off
Nothing sensed, Light is off
Sensor sensed something, Light is On
Sensor sensed something, Light is On
Nothing sensed, Light is off
```

(5) Cloud Application showing data.



(6) controlling application using cloud.



IoT project 1

Channel ID: 1874966
Author: mwa0000027694834
Access: Public

Smart street lighting system

Write API Key

Key: 9LB3R6Y1WHB216G2

Read API Keys

Key: NS9Y3PC0W2FS9AA2

Note:

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

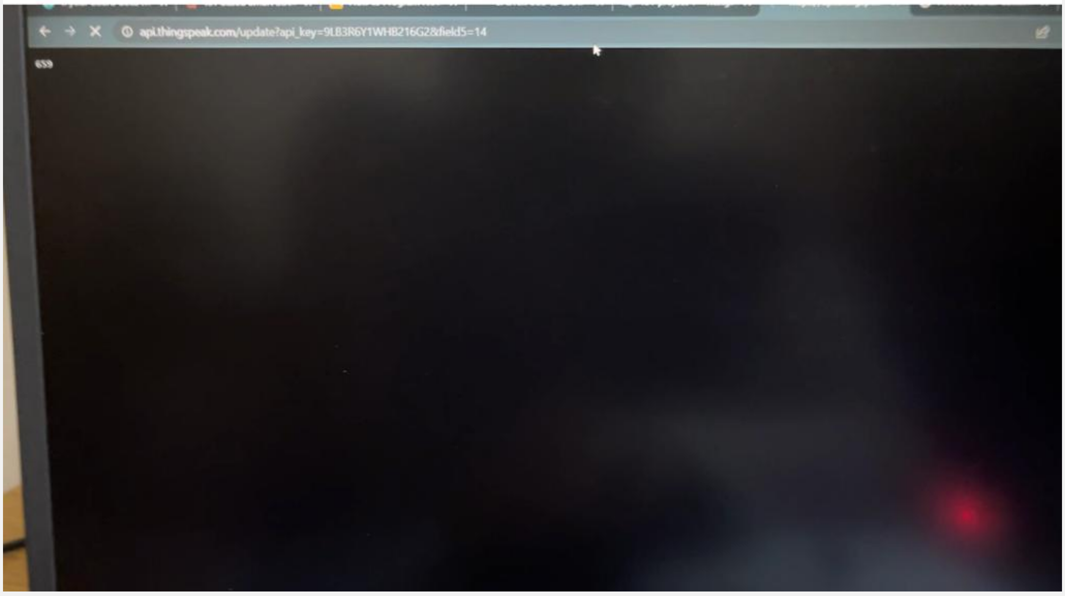
API Keys Settings

- **Write API Key**: Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key**.
- **Read API Keys**: Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- **Note**: Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

API Requests

Write a Channel Feed

```
GET https://api.thingspeak.com/update?api_key=9LB3R6Y1WHB216G2&field
```



api.thingspeak.com/update?api_key=9LB3R6Y1WHB216G2&field5=14

```
659
```

Codes

```
include <ESP8266WiFi.h>;
include <WiFiClient.h>;
include <ThingSpeak.h>;

const char* ssid = "Siddharth";
const char* password = "Siddu12";

int myChannelNumber = 12345;
const char * myWriteAPIKey = "A123F65Tf34";
const char * myReadAPIKey = "R345728203";
```

```
int Monitor;

int ir = D0;
int led = D5;

int ldr = A0;
int val =0;

void setup() {
Serial.begin(9600);
delay(10);
pinMode(ir,INPUT);
pinMode(led,OUTPUT);

WiFi.begin(ssid, password);
ThingSpeak.begin(client);
}

void loop() {
int s = not(digitalRead(ir));
val = analogRead(ldr);

Serial.println(s);

if(s==0  val<1200) digitalWrite(led1,LOW);
else digitalWrite(led1,LOW);

ThingSpeak.writeField(myChannelNumber, 1,val, myWriteAPIKey);
ThingSpeak.writeField(myChannelNumber, 2,s, myWriteAPIKey);
ThingSpeak.writeField(myChannelNumber, 5,led1, myWriteAPIKey);

Mon = ThingSpeak.readIntField(myChannelNumber, 5, myReadAPIKey);

if(Mon==1) digitalWrite(led1,HIGH);
else digitalWrite(led1,LOW);
}
```