Smart Street light Management

Using IBM Cloud

S. S. Harshitta, Sanjana Karra, RayeesaRuhi,

[harshittasara24@gmail.com](mailto:harshittasara24@gmail.com), [sanjanakarra9@gmail.com](mailto:sanjanakarra9@gmail.com), [rayeesaruhi1@gmail.com](mailto:rayeesaruhi1@gmail.com),

G. Narayanamma institute of G. Narayanamma institute of G. Narayanamma institute of

technology & science. technology & science. technology & science.

**ABSTRACT**

Streetlights are among a city’s strategic assets providing safe roads, inviting public areas, and enhanced security in homes, businesses, and city centres. They’re usually very costly to operate and they use in average 40% of a city’s electricity spending. it’s becoming crucial that municipalities, highway companies and other streetlight owners deploy control systems to dim the lights at the right light level at the right time, to automatically identify lamp and electrical failures and enable real time control.

Our project aims to control the intensity of the street lights with individual area control, total area control and dimming of the street lights as the street light plays a major role in safe road transport along with that if the street lights are turned on unnecessary there will be a huge lose in city’s electricity, As they consumption lots of power.

**PROBLEM STATEMENT**

1. To reduce power loss.
2. To make safe road transport.
3. Smart transport management.
4. Decrease human efforts.
5. Individual Area control.
6. Total Area control.
7. Dimming of lights.

**PROJECT WORKING PROCESS:**

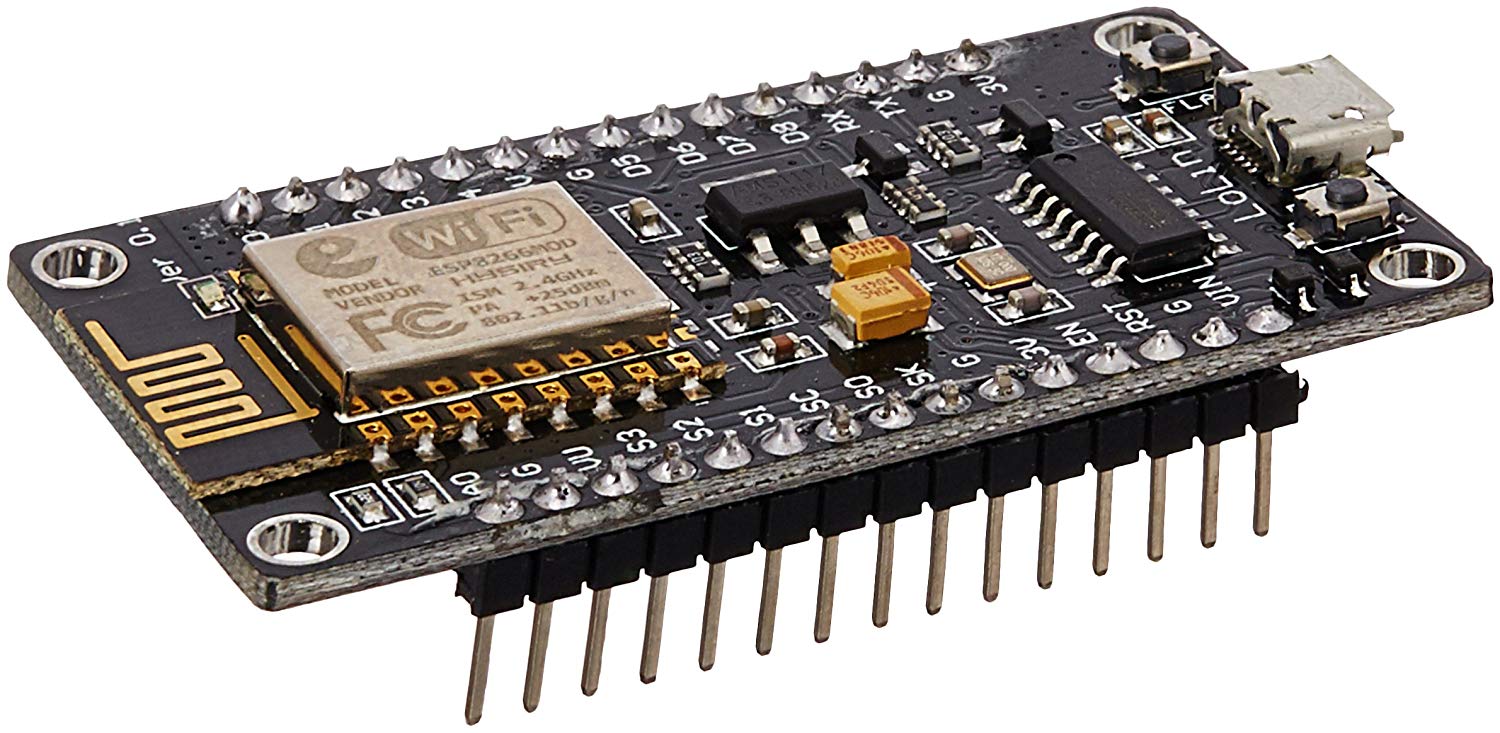
* LDR(Light Dependent Resistor)-  It is a variable resistor and changes its  resistance in a proportion to the light exposed to it.
* It’s resistance decreases with the intensity of light.
* The attached LED glows in analog mode according to the LDR Values.
* The attached LED remains OFF for all the values below Threshold limit. You can set your own threshold limit. In this programme we have given 800 as threshold. You can set the threshold to any value between 0 and 1023.
* Connecting Node MCU with IBM Cloud using Arduino Programming
* Stores the command send and the value of resistor .
* Creating a Node RED UI to receive and send data to the device .
* Individual and Area wise Street lights can be controlled.
* The required street lights can even be made dimmer .
* Reduces Manual effort .
* Accessing Sensor data from anywhere in the world .

**ABOUT COMPONENTS USED (SPESIFICATIONS AND WORKING PRINCIPLES, DATASHEETS)**

**1.NodeMCU Devise 1.0(ESP8266).**

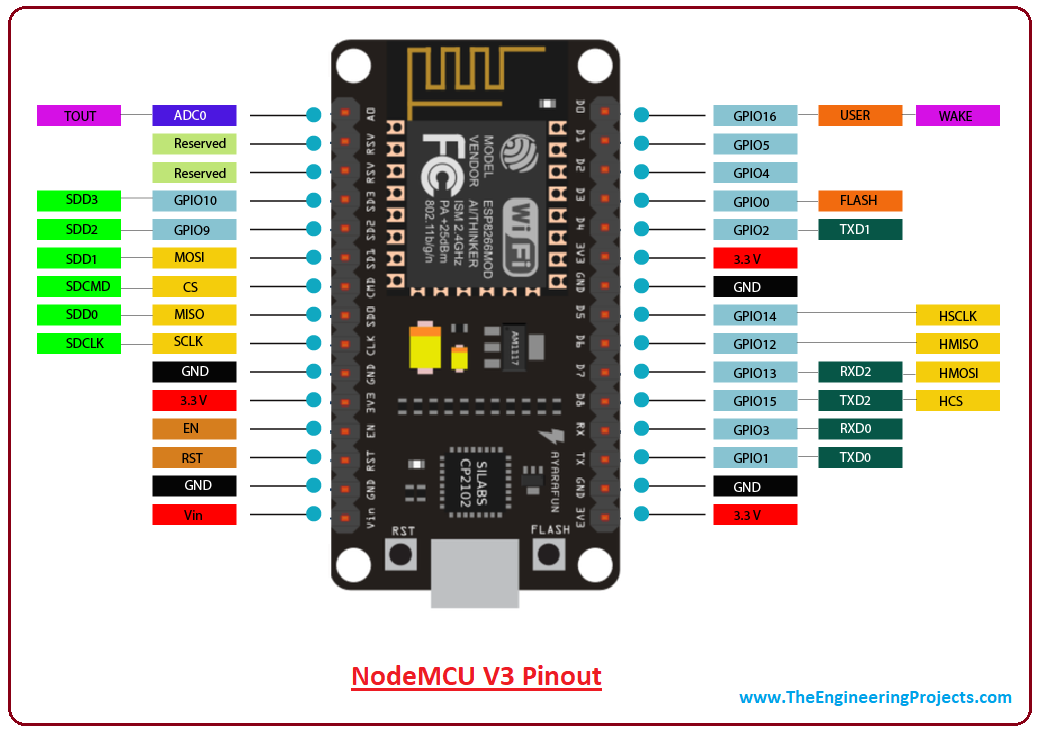
**Principle:**

**NodeMCU** is an open source IoT platform. It includes firmware which runs on the**ESP8266** Wi-Fi SoC from Espressif Systems, and hardware, which is based on the ESP-12 module. The term “**NodeMCU**” by default refers to the firmware rather than the dev kits. The firmware uses the **Lua** scripting language.

**NODEMCU(ESP8266)**

**How NodeMCU Works:**

The **ESP8266 can** be controlled from your local Wi-Fi network or from the internet (after port forwarding). The ESP-01 module has GPIO pins that **can** be programmed to turn an LED or a relay ON/OFF through the internet. The module **can** be programmed using an Arduino/USB-to-TTL converter through the serial pins (RX,TX).



* Node MCU development board is an open-source IOT development kit.
* It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12E module.
* The term "NodeMCU" by default refers to the firmware rather than the dev kits. The firmware uses the Lua scripting language.
* It is a low cost hardware platform available for development of IOT applications

ESP8266 is high integration wireless SOC θ ESP-12E Wi-Fi module is developed by Ai-thinker Team

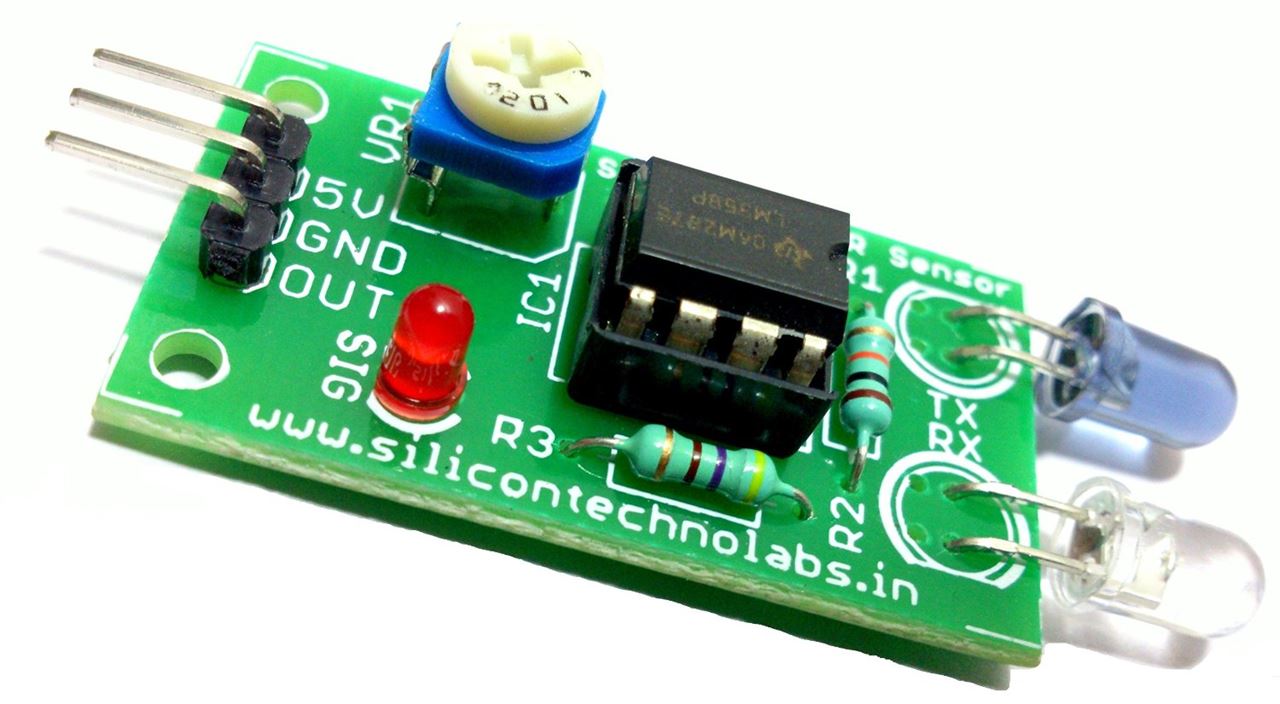
* Integrated low power 32-bit MCU - Tensilica L106, with 4MB Flash memory, 128KB SRAM
* Clock speed support 80 MHz and over clock 160 MHz,
* Integrated 10-bit ADC and 13 GPIO’s
* Integrated TCP/IP protocol stack
* Supports IEEE 802.11 b/g/n , Wi-Fi 2.4 GHz, WPA/WPA2 , +20dBm output power
* Support STA/AP/STA+AP operation modes θ Integrated UART, I2C, I2S, IRDA, PWM, GPIO, SDIO 2.0, (H) SPI interface
* Deep sleep power<10μA, Standby power consumption less than 1.0mW
* Supports the RTOS (Real-Time Operating System)

**2.IR sensor**.

Working Principle:

An [infrared sensor](https://www.elprocus.com/ir-remote-control-basics-operation-application/) is an electronic device, that emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. These types of sensors measures only infrared radiation, rather than emitting it that is called as a [passive IR sensor](https://www.elprocus.com/passive-infrared-pir-sensor-with-applications/). Usually in the infrared spectrum, all the objects radiate some form of thermal radiations. These types of radiations are invisible to our eyes, that can be detected by an infrared sensor. The emitter is simply an IR LED ([Light Emitting Diode](https://www.elprocus.com/explain-different-types-leds-working-applications-engineering-students/)) and the detector is simply an IR photodiode which is sensitive to IR light of the same wavelength as that emitted by the IR LED. When IR light falls on the photodiode, The resistances and these output voltages, change in proportion to the magnitude of the IR light received.

**IR SENSOR**



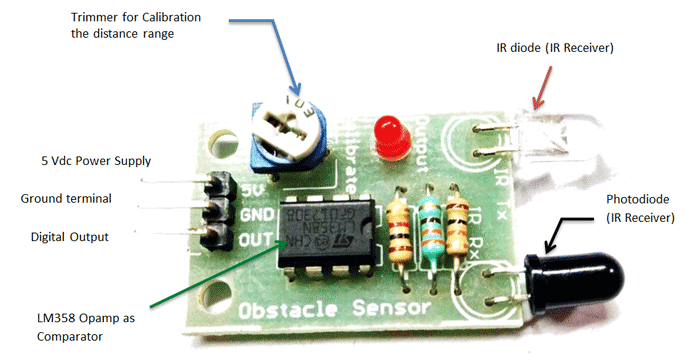
### **Pin Configuration**

|  |  |
| --- | --- |
| **Pin Name** | **Description** |
| VCC | Power Supply Input |
| GND | Power Supply Ground |
| OUT | Active High Output |

### **IR Sensor Module Features**

* 5VDC Operating voltage
* I/O pins are 5V and 3.3V compliant
* Range: Up to 20cm
* Adjustable Sensing range
* Built-in Ambient Light Sensor
* 20mA supply current
* Mounting hole

### **Brief about IR Sensor Module**



The IR sensor module consists mainly of the IR Transmitter and Receiver, Opamp, Variable Resistor (Trimmer pot), output LED in brief.

**IR LED Transmitter**

[IR LED](https://components101.com/ir-led-pinout-datasheet) emits light, in the range of Infrared frequency. IR light is invisible to us as its wavelength (700nm – 1mm) is much higher than the visible light range. IR LEDs have light emitting angle of approx. 20-60 degree and range of approx. few centimetres to several feet’s, it depends upon the type of IR transmitter and the manufacturer. Some transmitters have the range in kilometres. IR LED white or transparent in colour, so it can give out amount of maximum light.

**Photodiode Receiver**

Photodiode acts as the IR receiver as its conducts when light falls on it. Photodiode is a semiconductor which has a P-N junction, operated in Reverse Bias, means it start conducting the current in reverse direction when Light falls on it, and the amount of current flow is proportional to the amount of Light. This property makes it useful for IR detection. Photodiode looks like a LED, with a black colour coating on its outer side, Black colour absorbs the highest amount of light.

**LM358 Opamp**

[LM358](https://components101.com/ic-lm358-pinout-details-datasheet) is an Operational Amplifier (Op-Amp) is used as voltage comparator in the IR sensor. the comparator will compare the threshold voltage set using the preset (pin2) and the photodiode’s series resistor voltage (pin3).

Photodiode’s series resistor voltage drop > Threshold voltage = Opamp output is High

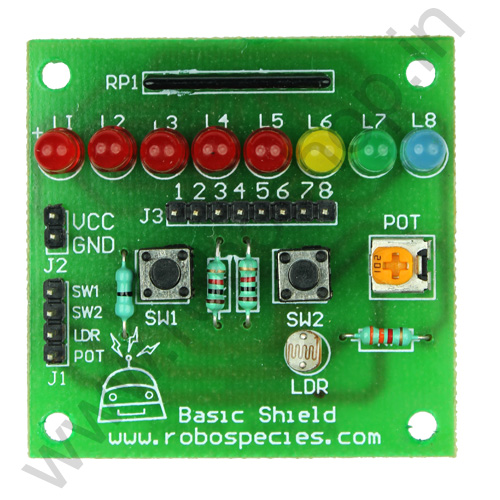
Photodiode’s series resistor voltage drop < Threshold voltage = Opamp output is Low

When Opamp's output is **high** the LED at the Opamp output terminal **turns ON** (Indicating the detection of Object).

**Variable Resistor**

The variable resistor used here is a preset. It is used to calibrate the distance range at which object should be detected.

**3.Basic Shield.**



A Light Dependent Resistor (**LDR**) or a photo resistor is a device whose resistivity is a **function** of **the** incident electromagnetic radiation. Hence, they are light sensitive devices. They are also called as photo conductors, photo conductive cells or simply photocells.

Working Principle of LDR. This resistor works on the principle of photo conductivity. It is nothing but, when the light falls on its surface, then the material conductivity reduces and also the **electrons** in the valence band of the device are excited to the conduction band.

**FINAL CODE:**

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

int value,fdata;

const char\* ssid = "Sanjana Karra";

const char\* password = "nandu624";

#define ORG "qfbvzt"

#define DEVICE\_TYPE "karra"

#define DEVICE\_ID "6919"

#define TOKEN "1234567890"

String command;

String flag;

String area;

String both;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char topic[] = "iot-2/cmd/home/fmt/String";

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;

void callback(char\* topic, byte\* payload, unsigned int payloadLength);

//Serial.println(clientID);

#define LED D2

#define OP D3

String command1;

WiFiClient wifiClient;

PubSubClient client(server, 1883, callback, wifiClient);

void setup() {

Serial.begin(115200);

Serial.println();

pinMode(LED,OUTPUT);

pinMode(OP,INPUT);

wifiConnect();

mqttConnect();

}

void loop() {

if(area=="area1"||"both"){

flag=command;

if(flag=="lighton" ||fdata==1){

value=analogRead(A0);

if(value>=900){

int OP=digitalRead(OP);

value=(value/20);

analogWrite(LED,value);

if(OP==1){

digitalWrite(LED,HIGH);

}

}else

digitalWrite(LED,LOW);

}

if(flag=="lightoff"||fdata==0)

digitalWrite(LED,LOW);

}

if (!client.loop()) {

mqttConnect();

}

delay(100);

}

void wifiConnect() {

Serial.print("Connecting to "); Serial.print(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.print("nWiFi connected, IP address: "); Serial.println(WiFi.localIP());

}

void mqttConnect() {

if (!client.connected()) {

Serial.print("Reconnecting MQTT client to "); Serial.println(server);

while (!client.connect(clientId, authMethod, token)) {

Serial.print(".");

delay(500);

}

initManagedDevice();

Serial.println();

}

}

void initManagedDevice() {

if (client.subscribe(topic)) {

Serial.println("subscribe to cmd OK");

} else {

Serial.println("subscribe to cmd FAILED");

}

}

void callback(char\* topic, byte\* payload, unsigned int payloadLength) {

Serial.print("callback invoked for topic: "); Serial.println(topic);

int value=analogRead(A0);

Serial.print("the resistor value is");

Serial.println(value);

// digitalWrite(LED,value);

for (int i = 0; i < payloadLength; i++) {

//Serial.println((char)payload[i]);

command1 += (char)payload[i];

}

loop();

command=getValue(command1,'$',0);

area=getValue(command1,'$',1);

Serial.println(command);

Serial.println(area);

if(command=="lighton"){

fdata=1;

}

else if(command=="lightoff"){

fdata=0;

}

command="";

Serial.println(area);

}

String getValue(String data, char separator, int index)

{

int found = 0;

int strIndex[] = { 0, -1 };

int maxIndex = data.length() - 1;

for (int i = 0; i <= maxIndex && found <= index; i++) {

if (data.charAt(i) == separator || i == maxIndex) {

found++;

strIndex[0] = strIndex[1] + 1;

strIndex[1] = (i == maxIndex) ? i+1 : i;

}

}

return found > index ? data.substring(strIndex[0], strIndex[1]) : "";

}

**INSTALLING PROCESS**

1. Top of Form
2. Arduino IDE.
3. IBM Watson Cloud Services.
4. MIT App Inventor-2.