SMART STREET LIGHT MANAGEMENT SYSTEM

INTRODUCTION:

A huge amount of electrical power of many countries is consumed in lighting the streets. However, vehicles pass with very low rate in specific periods of time and parts of the streets are not occupied by vehicles over time. In this project, we propose a system that automatically dim the light for the parts of the streets having no vehicles and intensify the light for these parts once there are some vehicles that are going to come. Logically, this system may save a large amount of the electrical power. In addition, it may increase the lifetime of the lamps. This system automatically controls and monitors the light of the streets. It can intensify only the parts that have vehicles and help on the maintenance of the lighting equipment’s. Conventional street lighting systems in areas with a low frequency of passer by are online most of the night without purpose. The consequence is that a large amount of power is wasted meaninglessly. The present system is like, the street lights will be switched on in the evening before the sun sets and they are switched off the next day morning after there is sufficient light on the roads. This project gives the best solution for electrical power wastage. In this project, sensors used are, Light Dependent Resistor (LDR) to indicate a day/night time and the IR sensors to detect the movement on the street. The Arduino Uno is used to control the street light system.

Problem statement:

Our aim is to achieve power saving in the case of street light by creating a model based on IOT that senses the light in surroundings and on the street light if the light intensity is low

Challenges:

* Loss of net connectivity
* Obstacle encountered in front of IR sensor
* Obstacle covering LDR sensor
* Defective in equipment’s
* Delays in analysing readings

Solution:

We want to design an IOT application that senses the intensity of light, if intensity goes down by threshold value it sends signal and switch on the street lights, and these lights glow at half the intensity, and IR sensors are placed on the sides of road, these are used to sense the motion, if IR sensor senses motion it sends a signal so that street lights glow at full intensity. If the light intensity increases more than threshold then system sends signal which turns off the lights

WORKING:

The proposed system has four conditions :

1. When light is present in ambiance and no vehicle is present :-

When light is present in ambiance the LDR sensor senses the light and sends the signal to ARDUNIO board that so that no lights will be glowing

1. When light is present in ambiance and vehicle is in motion :-

When vehicle passes by IR sensor it sends the signals to ARDUNIO board but the LDR sensor detects light so it sends signals so that the lights will not glow

1. When light is not present and vehicle is not moving :-

When light was not present LDR sensor sends signals so that the lights will glow at some intensity not full intensity

1. When light is not present and vehicle is not moving :-

When light was not present LDR sensor sends signals so that the lights will glow at some intensity not full intensity, but when IR sensor detects movement of vehicle it will send signal to ARDUNIO board so that lights will glow at full intensity

COMPONENTS :

* ARDUNIO UNO BORD
* LDR SENSOR
* IR SENSORS
* 10 mm LED’S
* CONNECTION WIRES
* BREAD BORD
* ESP 8266 WIFI module

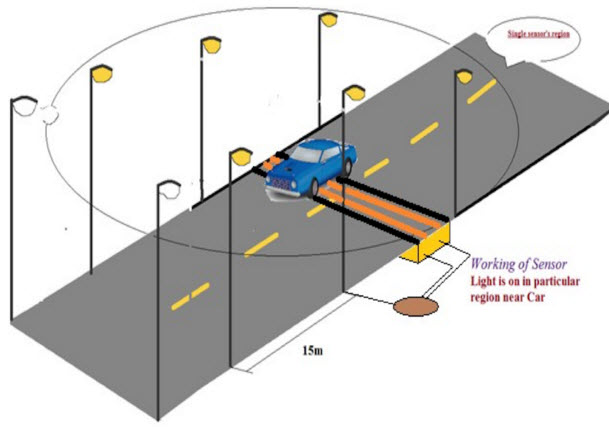
Technologies:

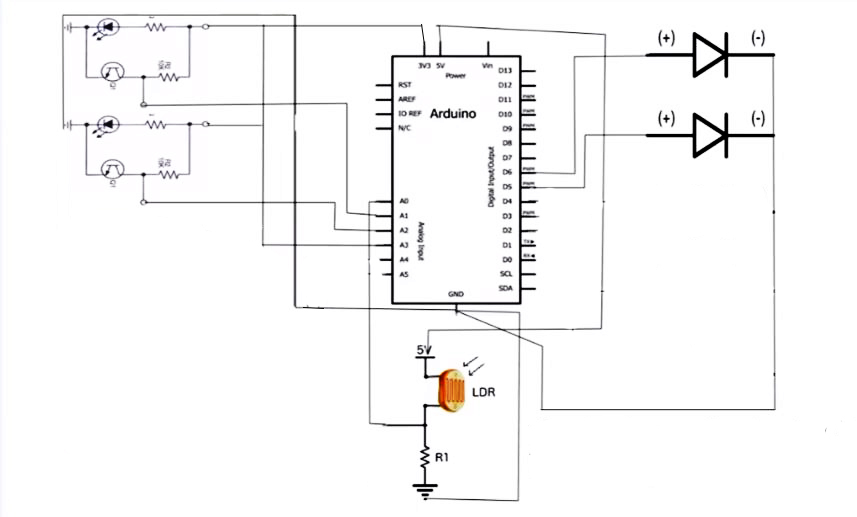
* Cloud : thing speak
* Arduino

Protocols:

* TCP
* Wi-Fi
* HTTP

Diagram representation



Circuit diagram: 

Data flow diagram:

Data to android app

Data stored in cloud

Data from sensors

Code:

#include <SoftwareSerial.h>

SoftwareSerial espSerial = SoftwareSerial(3,2);

String apiKey = "TEE245AGMVXKO3Z8"; // replace with your channel's thingspeak WRITE API key

String ssid="rahul"; // Wifi network SSID

String password ="rahul999"; // Wifi network password

boolean DEBUG=true;

int led = 11;

int led1 = 10;

int led2 = 5;

int led3 = 6;

int led4 = 9;

int ldr = A0;

int ir = A1;

int ir1 = A2;

int ir2 = A3;

int ir3 = A4;

int ir4 = A5;

//======================================================================== showResponce

void showResponse(int waitTime){

long t=millis();

char c;

while (t+waitTime>millis()){

if (espSerial.available()){

c=espSerial.read();

if (DEBUG) Serial.print(c);

}

}

}

//========================================================================

boolean thingSpeakWrite(float value1, float value2){

String cmd = "AT+CIPSTART=\"TCP\",\""; // TCP connection

cmd += "184.106.153.149"; // api.thingspeak.com

cmd += "\",80";

espSerial.println(cmd);

if (DEBUG) Serial.println(cmd);

if(espSerial.find("Error")){

if (DEBUG) Serial.println("AT+CIPSTART error");

return false;

}

String getStr = "GET /update?api\_key="; // prepare GET string

getStr += apiKey;

getStr +="&field1=";

getStr += String(value1);

getStr +="&field2=";

getStr += String(value2);

getStr += "\r\n\r\n";

// send data length

cmd = "AT+CIPSEND=";

cmd += String(getStr.length());

espSerial.println(cmd);

if (DEBUG) Serial.println(cmd);

delay(100);

if(espSerial.find(">")){

espSerial.print(getStr);

if (DEBUG) Serial.print(getStr);

}

espSerial.print(getStr);

if (DEBUG) Serial.print(getStr);

else{

espSerial.println("AT+CIPCLOSE");

// alert user

if (DEBUG) Serial.println("AT+CIPCLOSE");

return false;

}

return true;

}

//================================================================================ setup

void setup() {

DEBUG=true; // enable debug serial

Serial.begin(9600);

pinMode (led,OUTPUT);

pinMode (led1,OUTPUT);

pinMode (led2,OUTPUT);

pinMode (led3,OUTPUT);

pinMode (led4,OUTPUT);

pinMode (ldr,INPUT);

pinMode (ir,INPUT);

pinMode (ir1,INPUT);

pinMode (ir2,INPUT);

pinMode (ir3,INPUT);

pinMode (ir4,INPUT);

espSerial.begin(115200); // enable software serial

// Your esp8266 module's speed is probably at 115200.

// For this reason the first time set the speed to 115200 or to your esp8266 configured speed

// and upload. Then change to 9600 and upload again

//espSerial.println("AT+RST"); // Enable this line to reset the module;

//showResponse(1000);

//espSerial.println("AT+UART\_CUR=9600,8,1,0,0"); // Enable this line to set esp8266 serial speed to 9600 bps

//showResponse(1000);

espSerial.println("AT+CWMODE=1"); // set esp8266 as client

showResponse(1000);

espSerial.println("AT+CWJAP=\""+ssid+"\",\""+password+"\""); // set your home router SSID and password

showResponse(5000);

if (DEBUG) Serial.println("Setup completed");

}

// ====================================================================== loop

void loop() {

// Read sensor values

float t = 0;

float h = 0;

t = analogRead(A0);

h = analogRead(A2) + analogRead(A1);

h = h/2;

Serial.print(analogRead(A0));

Serial.println("\n");

int ldrStatus = analogRead (ldr);

if (ldrStatus <=300)

{

digitalWrite(led,HIGH);

analogWrite(led,255/5);

digitalWrite(led1,HIGH);

analogWrite(led1,255/5);

Serial.println("a1");

Serial.print(analogRead(A1));

Serial.println("\n");

Serial.println("a2");

Serial.print(analogRead(A2));

Serial.println("\n");

if(analogRead(A1)<500){

digitalWrite(led,HIGH);

}

else{

digitalWrite(led,HIGH);

analogWrite(led,255/5);

}

if(analogRead(A2)<500){

digitalWrite(led1,HIGH);

}

else{

digitalWrite(led1,HIGH);

analogWrite(led1,255/5);

}

}

else{

digitalWrite(led,LOW);

digitalWrite(led1,LOW);

}

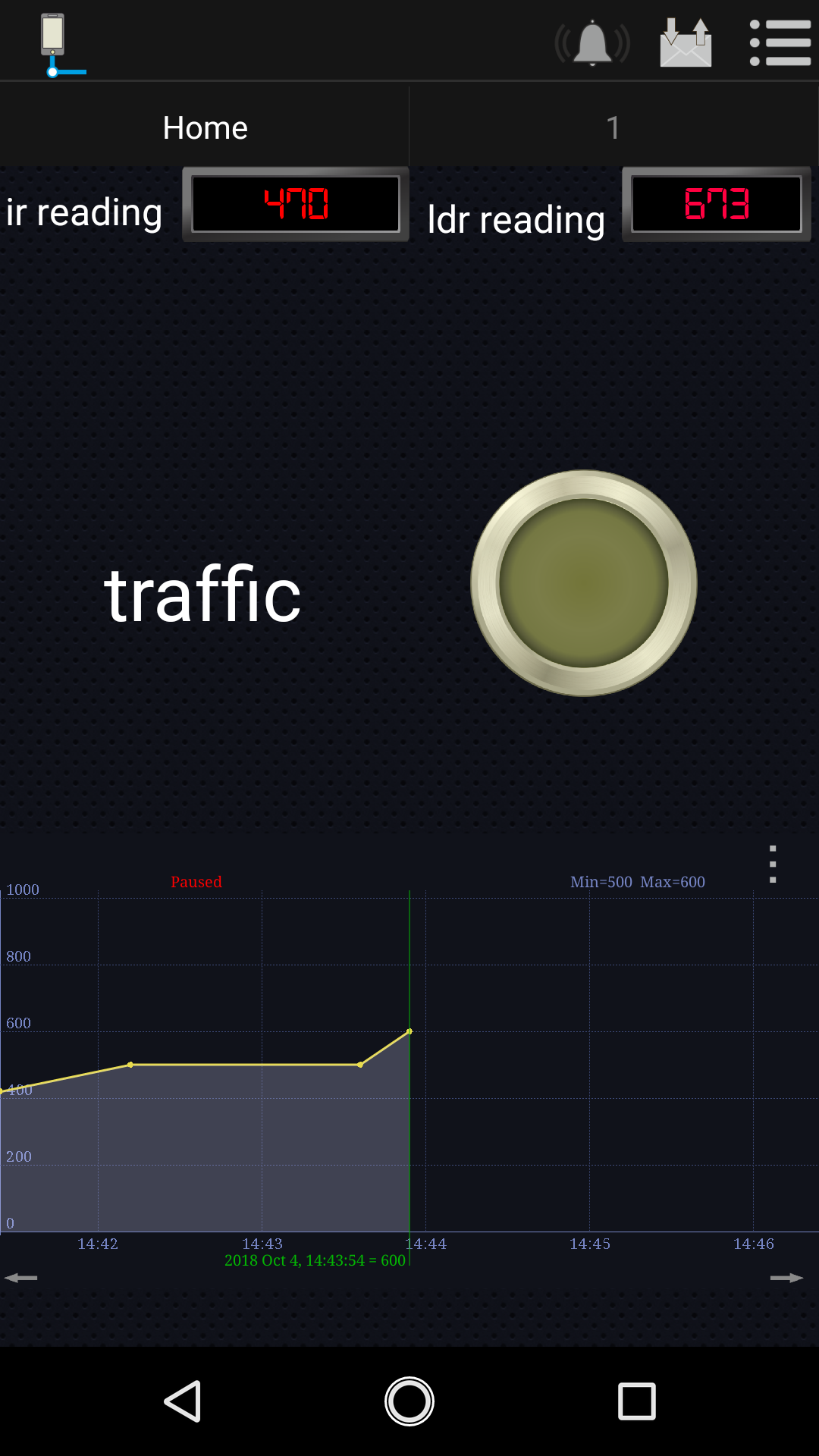
thingSpeakWrite(t,h);

delay(20000);

// thingspeak needs 15 sec delay between updates,

}

Interface design:



END USER BENEFITS :

* Reduction of human interference in controlling the operation of street lights
* Can monitor and analyse how much time the streetlights are on
* traffic in that road can be calculated
* Saving of electricity

Future enhancement:

This model can be further enhanced to send data to all users who subscribed to this channel and live traffic notification can also be sent to users and sending messages and alerts can also be implemented, if something is not working properly alarm can be implemented

References:

* <https://thingspeak.com>
* <https://www.circuito.io/>
* <https://www.circuitlab.com/editor/#?id=6t78uk>
* <https://ubidots.com/>
* <https://io.adafruit.com/>
* <https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266WiFi>
* <https://www.arduino.cc/en/main/documentation>