CEIS 114 COURSE PROJECT

SMART IOT TRAFFIC CONTROLLER

FINAL REVIEW

Presented by: Niral Patel

INTRODUCTION

IoT Traffic Controller

We developed the smart Traffic Controller with ESP 32

What does my created system do?

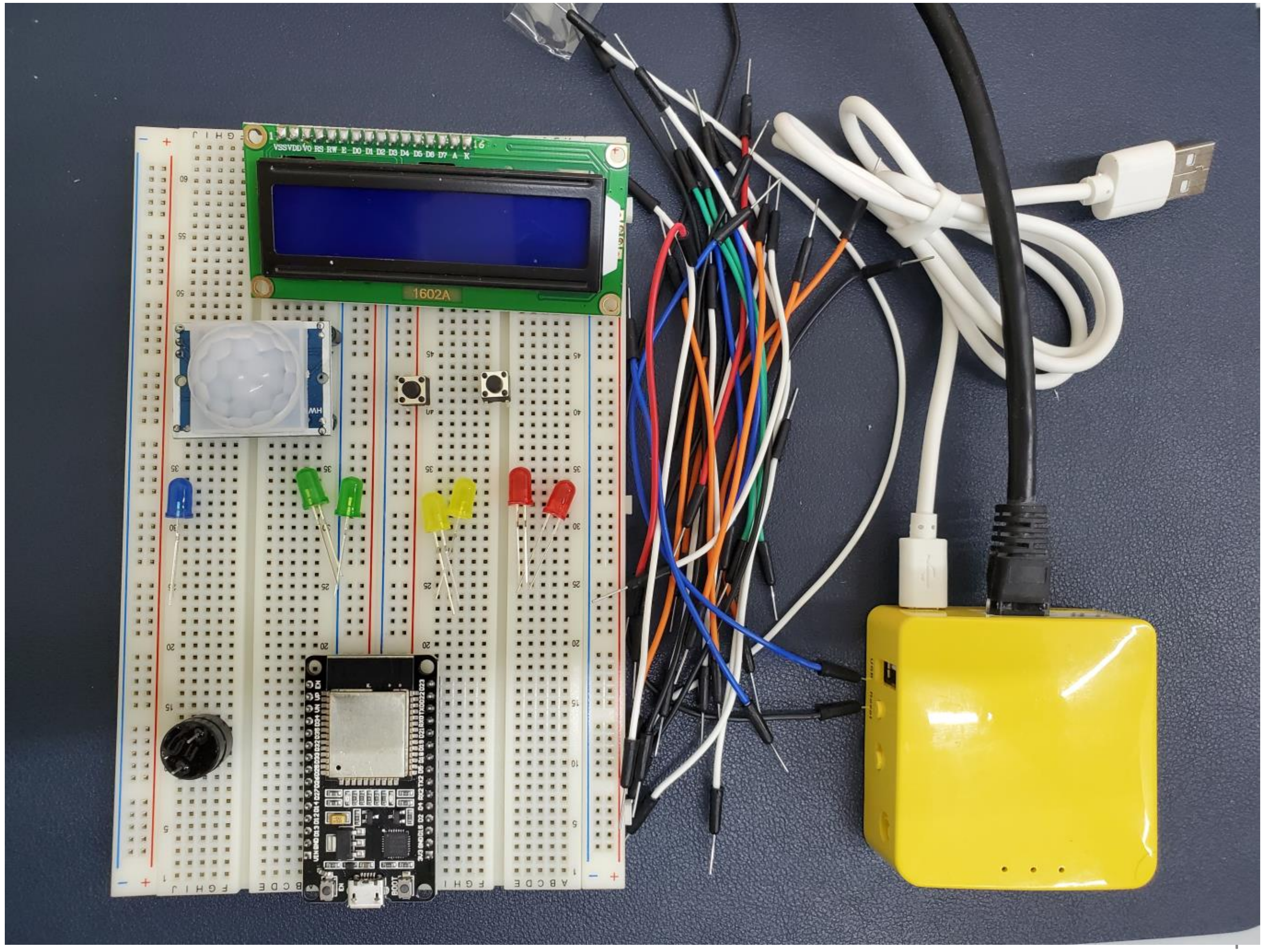
- It will control Traffic Light
- It will display on led when to walk and not to walk
- It will change all the light red and give signal to pedestrian walk on push of Button
- Fire/police department can control the traffic light with push off the Button when they have emergency

PROJECT 1

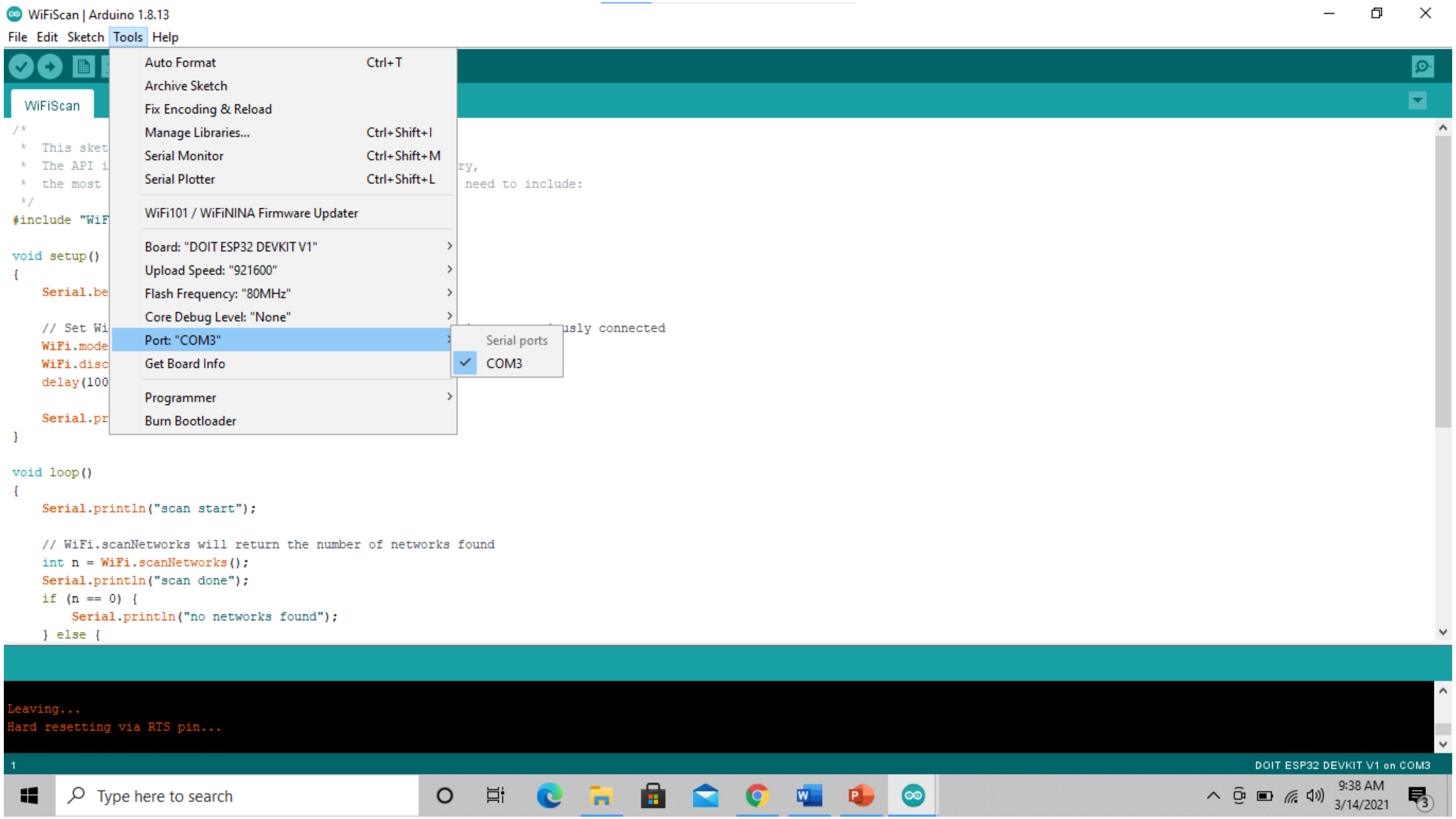
Project Plan for IoT Traffic Controller

PARTS I USED

ESP 32 Board
Colored LEDs: Red, Yellow, Green, and Blue
220 Ohm Resistors (optional)
Wires
Breadboard(s)
LCD Unit with I2C Adapter
Active Buzzer
Mini Router
Push Button(s)
PIR Motion Sensor



INSTALLATION OF ARDUINO IDE



WiFiScan | Arduino 1.8.13
File Edit Sketch Tools Help
WiFiScan
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Get Board Info
Programmer
Burn Bootloader
Serial ports
COM3
Type here to search
9:38 AM
3/14/2021

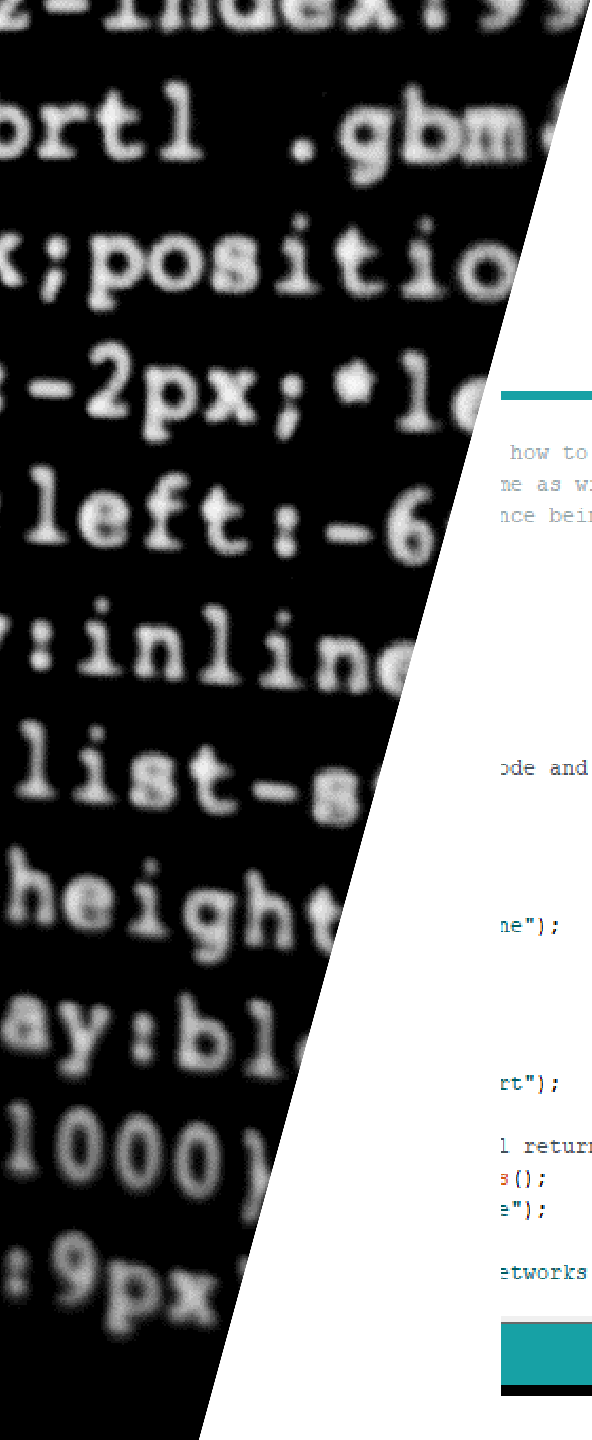
Board: "DOIT ESP32 DEVKIT V1"
Upload Speed: "921600"
Flash Frequency: "80MHz"
Core Debug Level: "None"
Port: "COM3"
✓ COM3

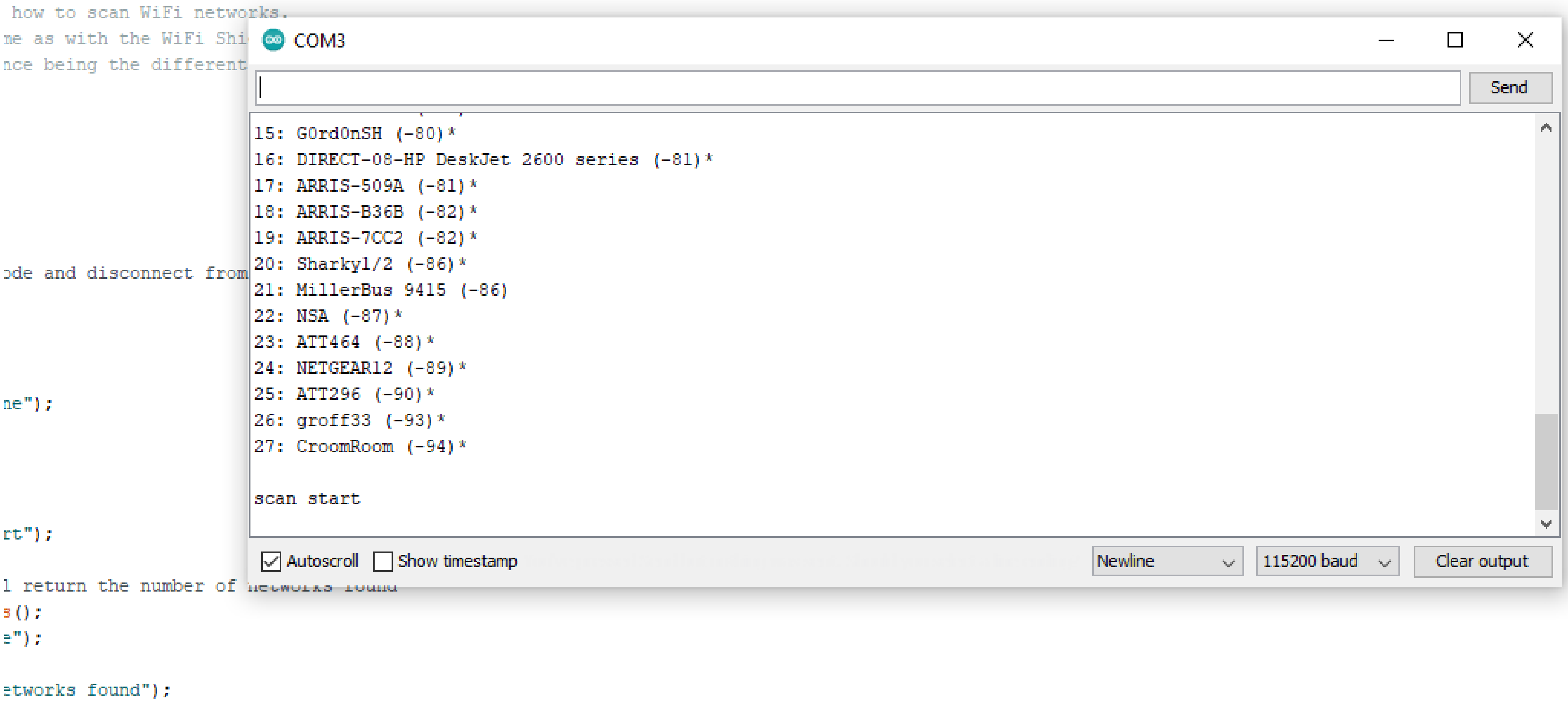
```
/*
 *  This sket                                      ry,
 *  The API i
 *  the most                                        need to include:
 */
#include "WiF

void setup()
{
    Serial.be

    // Set Wi                                                      usly connected
    WiFi.mode
    WiFi.disc
    delay(100

    Serial.pr
}

void loop()
{
    Serial.println("scan start");

    // WiFi.scanNetworks will return the number of networks found
    int n = WiFi.scanNetworks();
    Serial.println("scan done");
    if (n == 0) {
        Serial.println("no networks found");
    } else {
```

```
Leaving...
Hard resetting via RTS pin...
```

1 DOIT ESP32 DEVKIT V1 on COM3

ESP32 WIFI SCAN





```
 how to scan WiFi networks.
ne as with the WiFi Shi
nce being the different

ode and disconnect from

ne");

rt");

l return the number of networks found
s();
e");

etworks found");
```

COM3

```
15: G0rd0nSH (-80)*
16: DIRECT-08-HP DeskJet 2600 series (-81)*
17: ARRIS-509A (-81)*
18: ARRIS-B36B (-82)*
19: ARRIS-7CC2 (-82)*
20: Sharkyl/2 (-86)*
21: MillerBus 9415 (-86)
22: NSA (-87)*
23: ATT464 (-88)*
24: NETGEAR12 (-89)*
25: ATT296 (-90)*
26: groff33 (-93)*
27: CroomRoom (-94)*

scan start
```

☑ Autoscroll ☐ Show timestamp

Newline

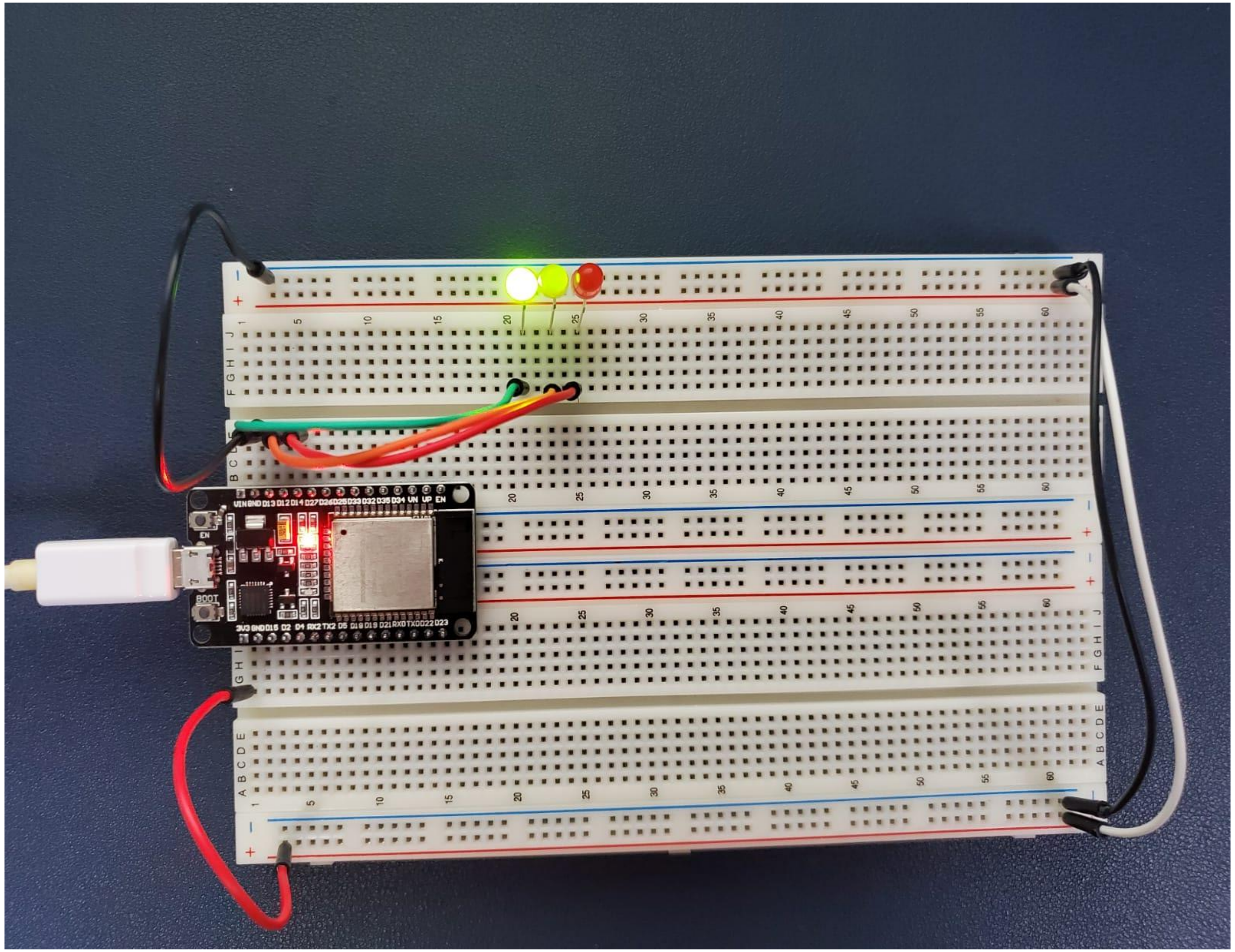
115200 baud

PROJECT 2

SETING UP THE FIRST TRAFFIC LIGTS

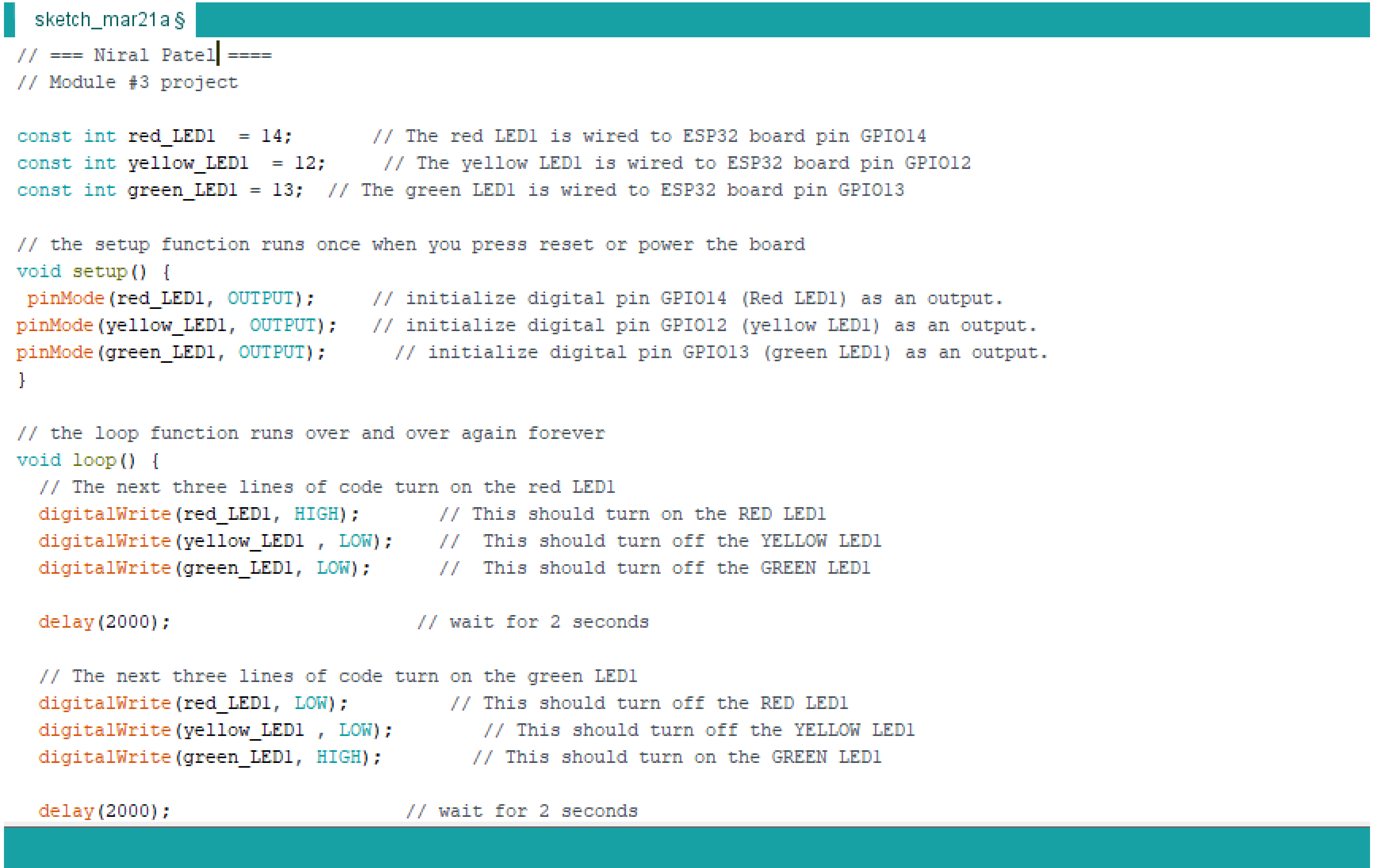
```
z-index:9
ortl .gbm
;positio
-2px;*l
left:-6
:inline
list-s
height
ay:bl
1000)
:9px
```

PICTURE OF CIRCUIT WITH WORKING LEDS



VIN GND D13 D12 D14 D27 D26 D25 D33 D32 D35 D34 VN VP EN
EN
BOOT
3V3 GND D15 D2 D4 RX2 TX2 D5 D18 D19 D21 RX0 TX0 D22 D23

SCREENSHOT OF CODE IN ARDUINO IDE

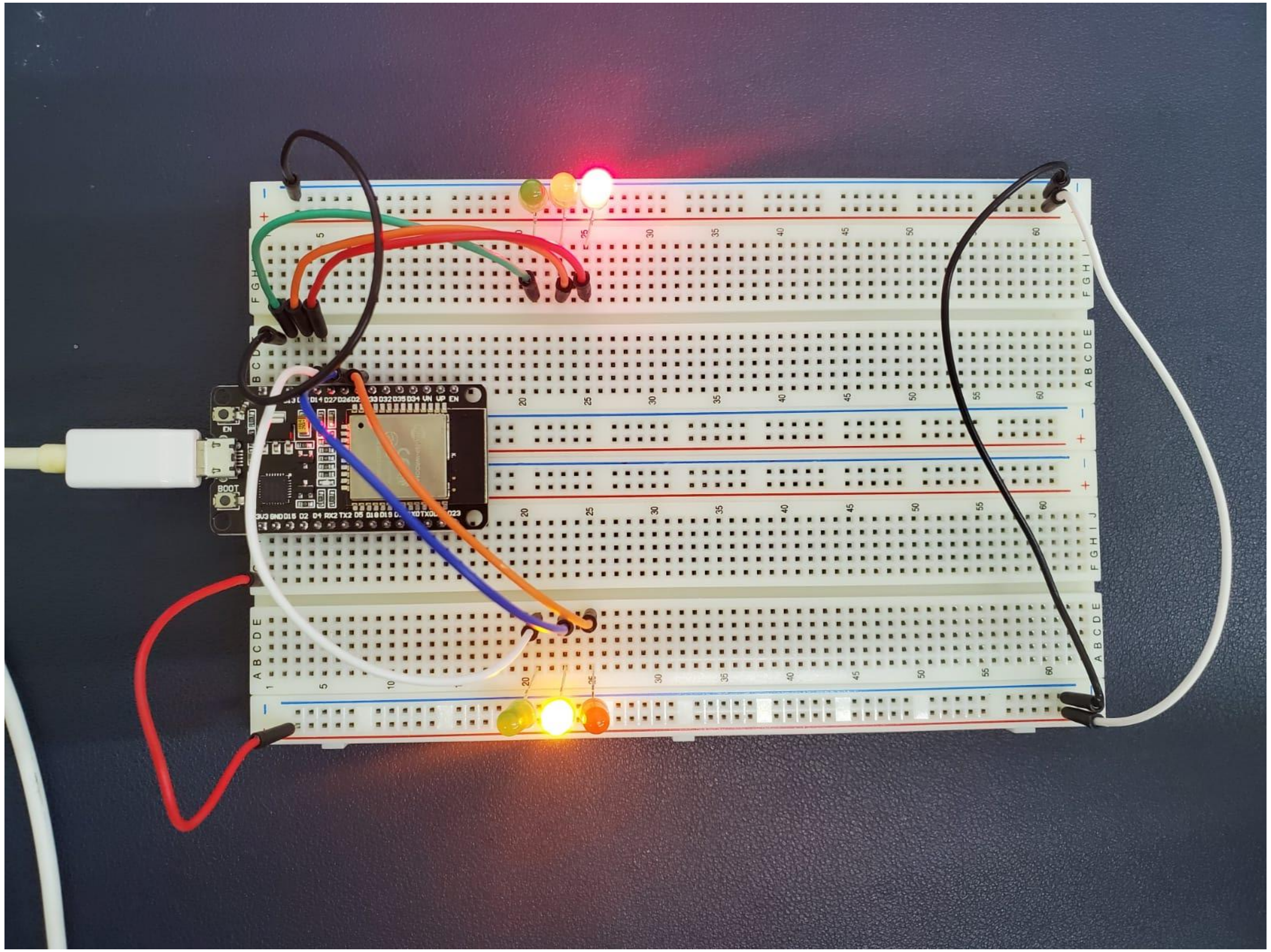


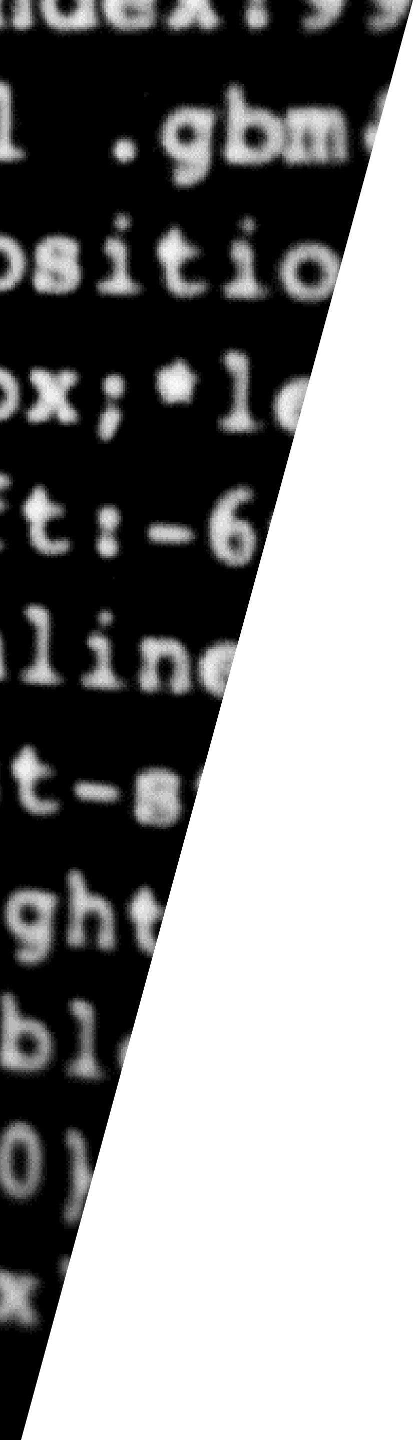
```
// === Niral Patel ====
// Module #3 project

const int red_LED1  = 14;       // The red LED1 is wired to ESP32 board pin GPIO14
const int yellow_LED1  = 12;     // The yellow LED1 is wired to ESP32 board pin GPIO12
const int green_LED1 = 13;  // The green LED1 is wired to ESP32 board pin GPIO13

// the setup function runs once when you press reset or power the board
void setup() {
 pinMode(red_LED1, OUTPUT);     // initialize digital pin GPIO14 (Red LED1) as an output.
pinMode(yellow_LED1, OUTPUT);   // initialize digital pin GPIO12 (yellow LED1) as an output.
pinMode(green_LED1, OUTPUT);      // initialize digital pin GPIO13 (green LED1) as an output.
}

// the loop function runs over and over again forever
void loop() {
  // The next three lines of code turn on the red LED1
  digitalWrite(red_LED1, HIGH);       // This should turn on the RED LED1
  digitalWrite(yellow_LED1 , LOW);    //  This should turn off the YELLOW LED1
  digitalWrite(green_LED1, LOW);      //  This should turn off the GREEN LED1

  delay(2000);                      // wait for 2 seconds

  // The next three lines of code turn on the green LED1
  digitalWrite(red_LED1, LOW);         // This should turn off the RED LED1
  digitalWrite(yellow_LED1 , LOW);        // This should turn off the YELLOW LED1
  digitalWrite(green_LED1, HIGH);        // This should turn on the GREEN LED1

  delay(2000);                     // wait for 2 seconds
```

PROJECT 3

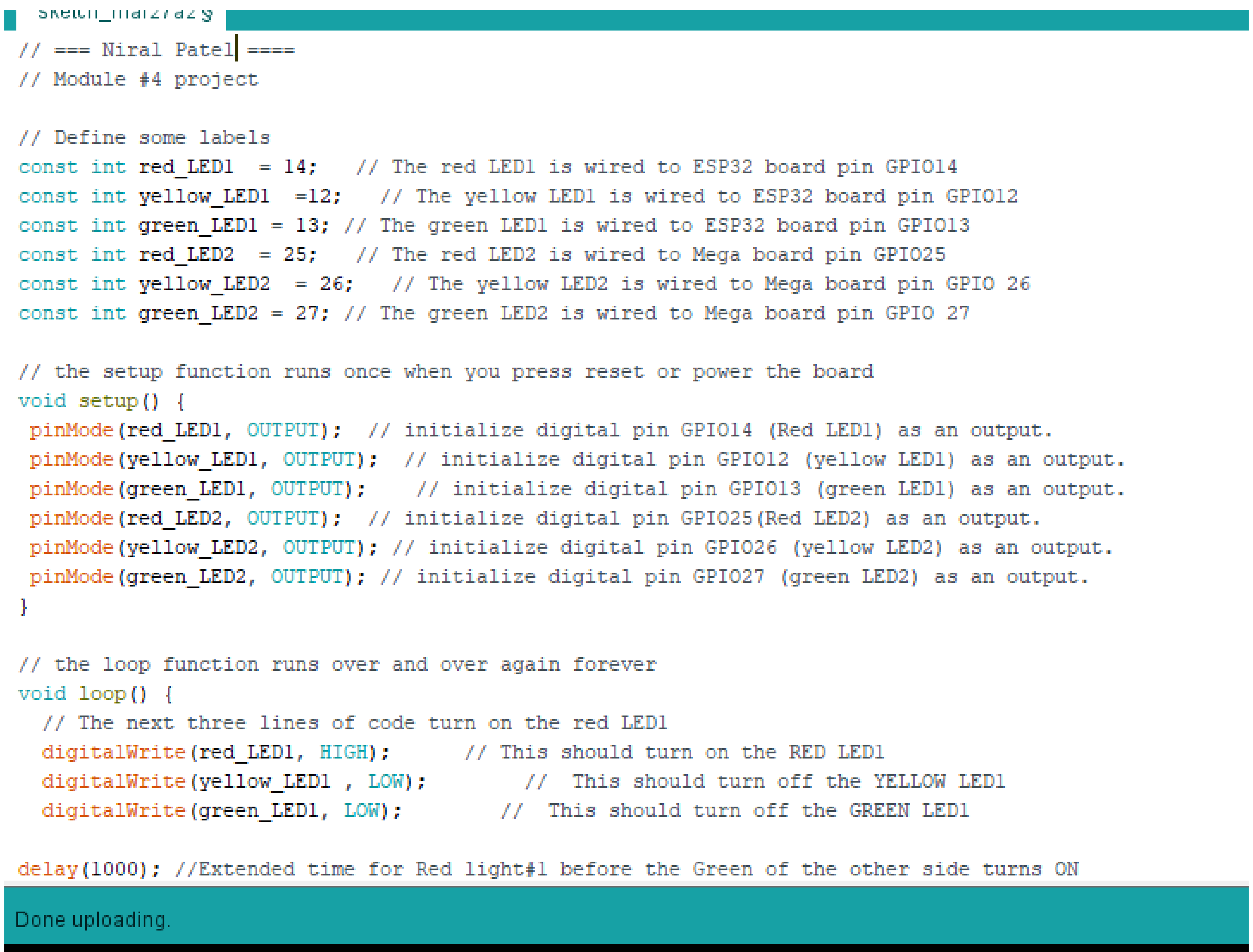
ADDING 2ND PARE OF LIGHTS

PICTURE OF CIRCUIT WITH WORKING LEDS





SCREENSHOT OF CODE IN ARDUINO IDE



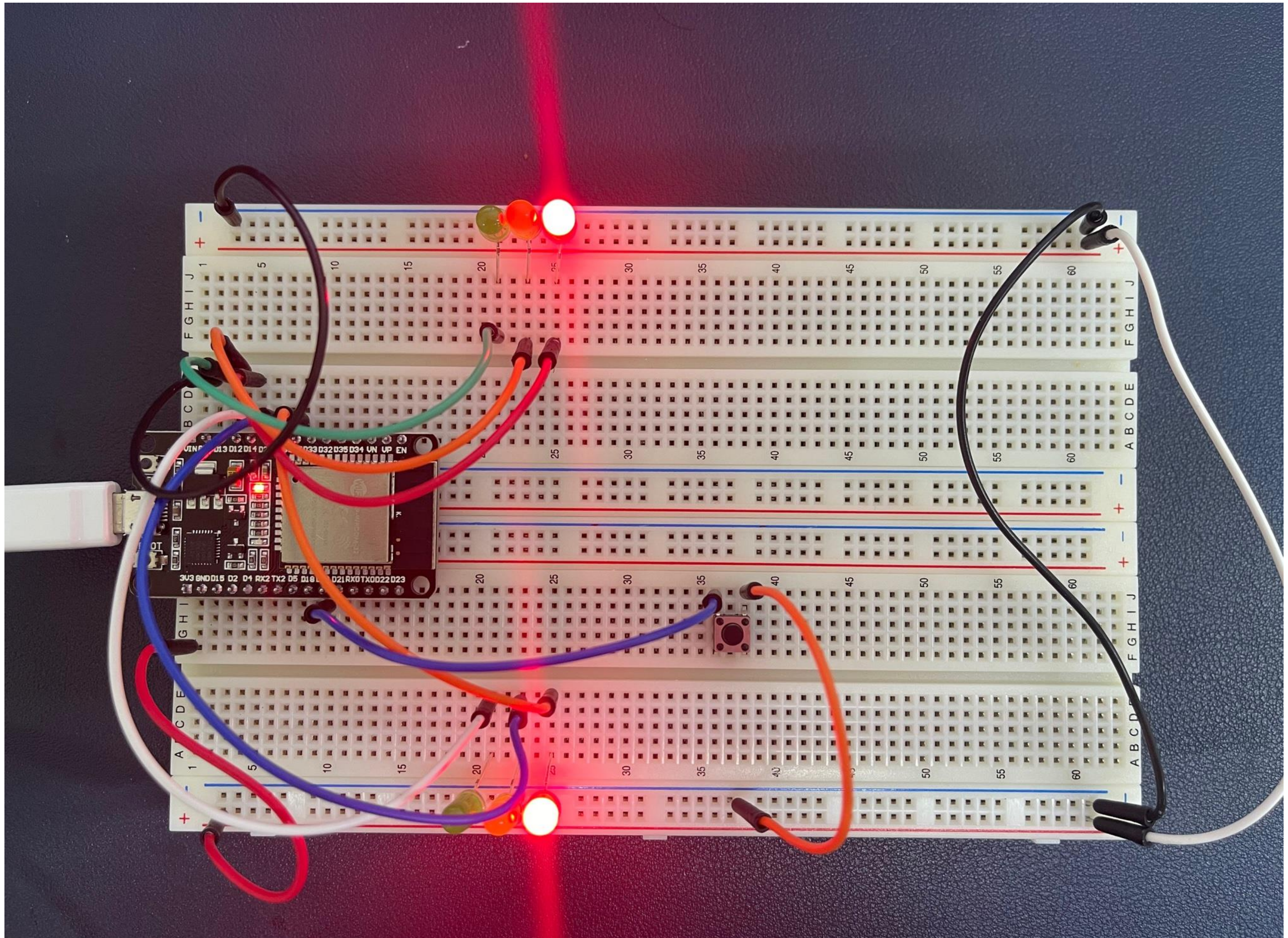
```
// === Niral Patel ====
// Module #4 project

// Define some labels
const int red_LED1  = 14;   // The red LED1 is wired to ESP32 board pin GPIO14
const int yellow_LED1  =12;   // The yellow LED1 is wired to ESP32 board pin GPIO12
const int green_LED1 = 13; // The green LED1 is wired to ESP32 board pin GPIO13
const int red_LED2  = 25;   // The red LED2 is wired to Mega board pin GPIO25
const int yellow_LED2  = 26;   // The yellow LED2 is wired to Mega board pin GPIO 26
const int green_LED2 = 27; // The green LED2 is wired to Mega board pin GPIO 27

// the setup function runs once when you press reset or power the board
void setup() {
 pinMode(red_LED1, OUTPUT);  // initialize digital pin GPIO14 (Red LED1) as an output.
 pinMode(yellow_LED1, OUTPUT);  // initialize digital pin GPIO12 (yellow LED1) as an output.
 pinMode(green_LED1, OUTPUT);    // initialize digital pin GPIO13 (green LED1) as an output.
 pinMode(red_LED2, OUTPUT);  // initialize digital pin GPIO25(Red LED2) as an output.
 pinMode(yellow_LED2, OUTPUT); // initialize digital pin GPIO26 (yellow LED2) as an output.
 pinMode(green_LED2, OUTPUT); // initialize digital pin GPIO27 (green LED2) as an output.
}

// the loop function runs over and over again forever
void loop() {
  // The next three lines of code turn on the red LED1
  digitalWrite(red_LED1, HIGH);      // This should turn on the RED LED1
  digitalWrite(yellow_LED1 , LOW);        //  This should turn off the YELLOW LED1
  digitalWrite(green_LED1, LOW);        //  This should turn off the GREEN LED1

delay(1000); //Extended time for Red light#1 before the Green of the other side turns ON
```

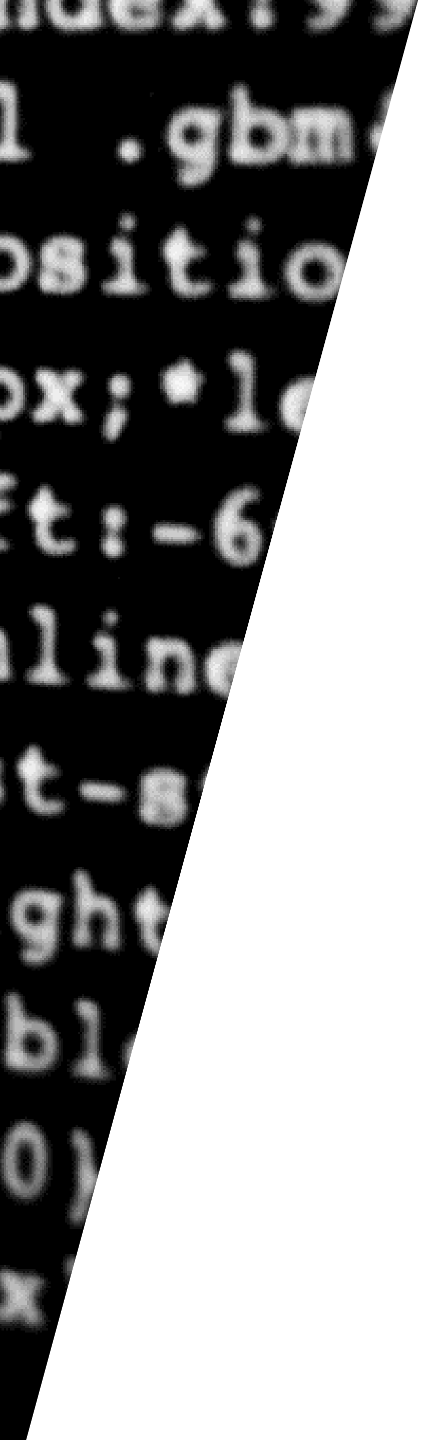
Done uploading.

PROJECT 4

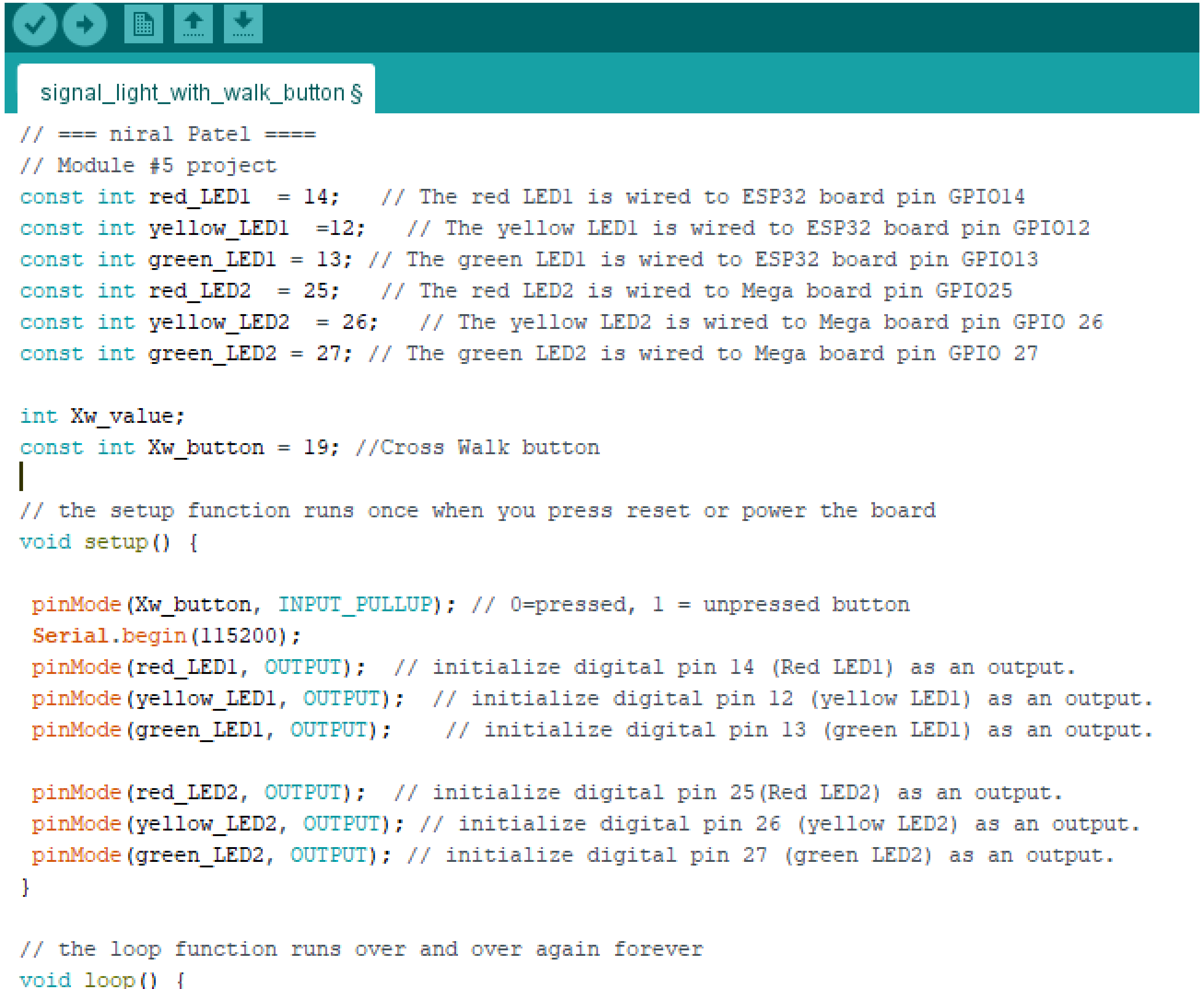
ADDDING PUSH BUTTON

PICTURE OF CIRCUIT WITH WORKING LEDS

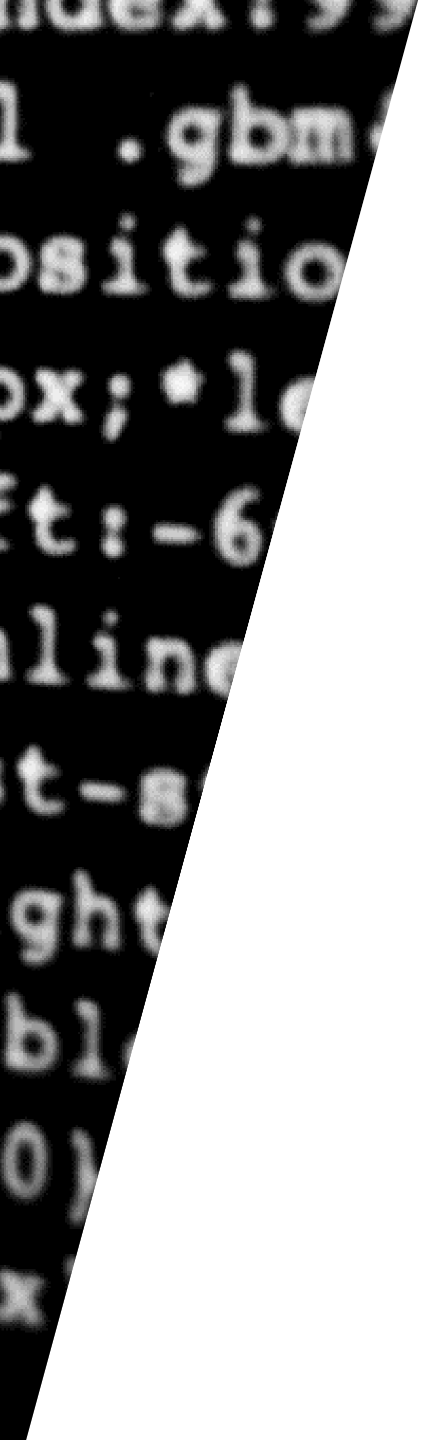




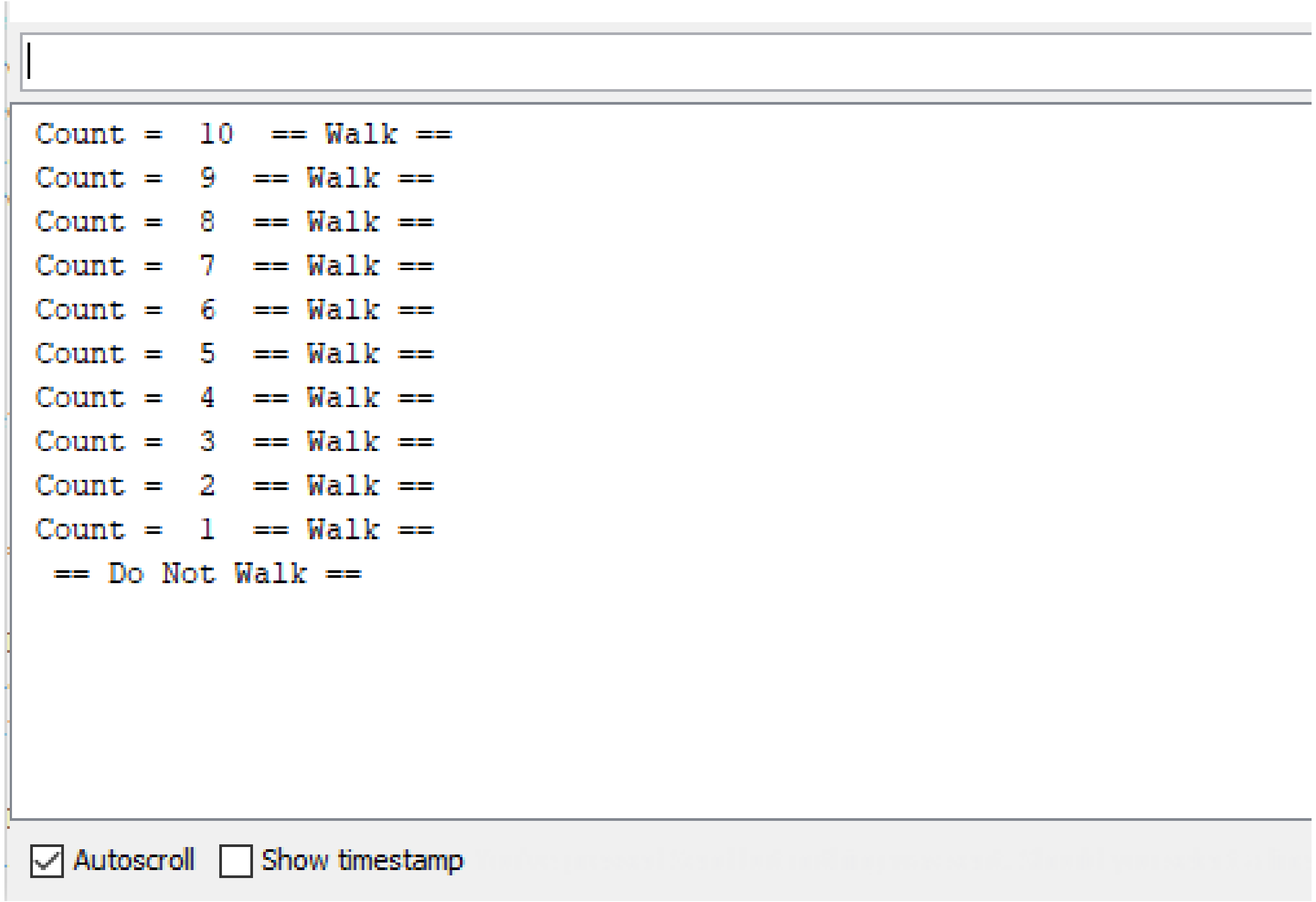
SCREENSHOT OF CODE IN ARDUINO IDE



```
// === niral Patel ====
// Module #5 project
const int red_LED1  = 14;   // The red LED1 is wired to ESP32 board pin GPIO14
const int yellow_LED1  =12;   // The yellow LED1 is wired to ESP32 board pin GPIO12
const int green_LED1 = 13; // The green LED1 is wired to ESP32 board pin GPIO13
const int red_LED2  = 25;   // The red LED2 is wired to Mega board pin GPIO25
const int yellow_LED2  = 26;   // The yellow LED2 is wired to Mega board pin GPIO 26
const int green_LED2 = 27; // The green LED2 is wired to Mega board pin GPIO 27

int Xw_value;
const int Xw_button = 19; //Cross Walk button

// the setup function runs once when you press reset or power the board
void setup() {

 pinMode(Xw_button, INPUT_PULLUP); // 0=pressed, 1 = unpressed button
 Serial.begin(115200);
 pinMode(red_LED1, OUTPUT);  // initialize digital pin 14 (Red LED1) as an output.
 pinMode(yellow_LED1, OUTPUT);  // initialize digital pin 12 (yellow LED1) as an output.
 pinMode(green_LED1, OUTPUT);    // initialize digital pin 13 (green LED1) as an output.

 pinMode(red_LED2, OUTPUT);  // initialize digital pin 25(Red LED2) as an output.
 pinMode(yellow_LED2, OUTPUT); // initialize digital pin 26 (yellow LED2) as an output.
 pinMode(green_LED2, OUTPUT); // initialize digital pin 27 (green LED2) as an output.
}

// the loop function runs over and over again forever
void loop() {
```



SCREENSHOT OF SERIAL MONITOR IN ARDUINO IDE



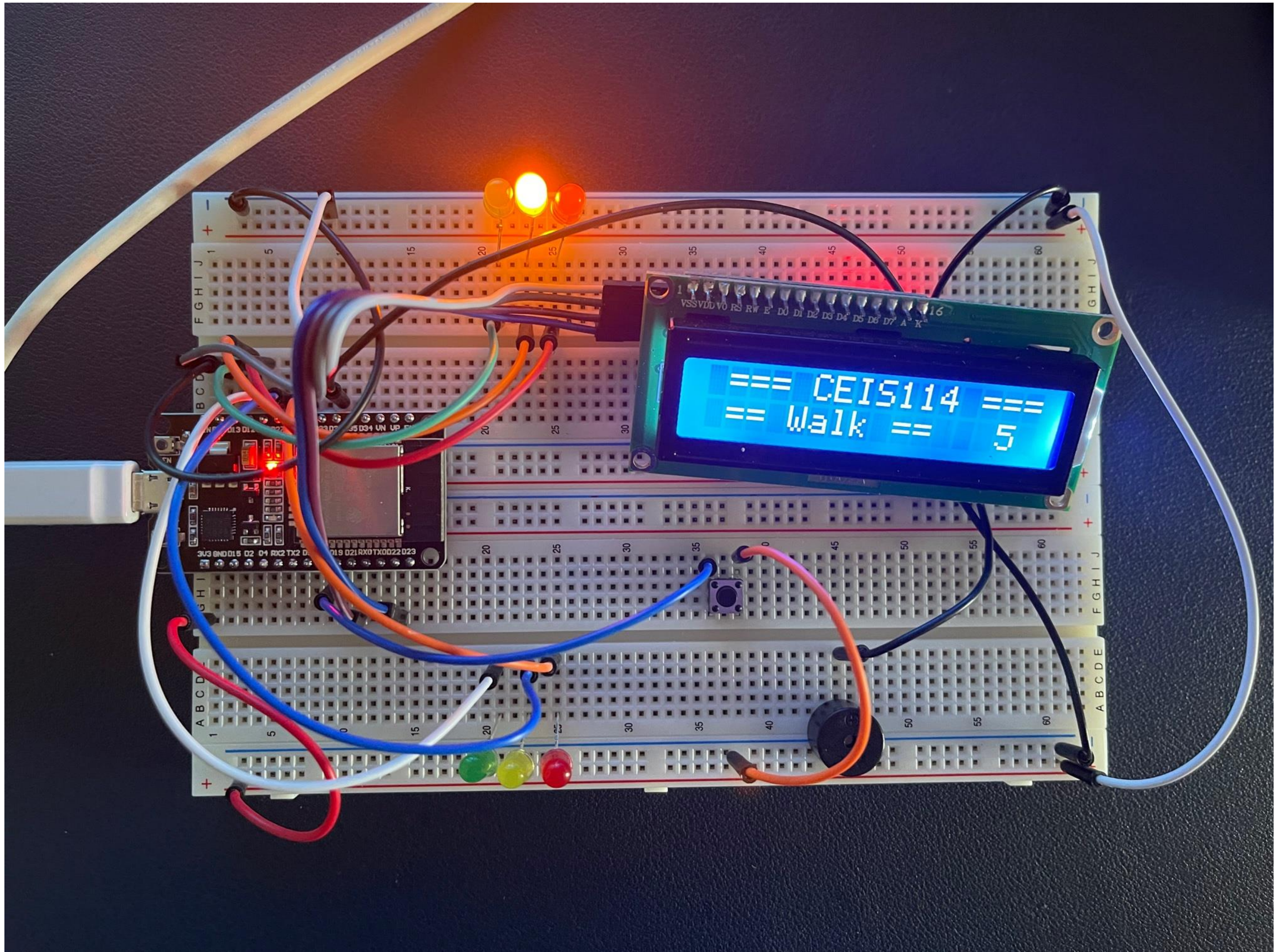
```
Count =  10  == Walk ==
Count =  9  == Walk ==
Count =  8  == Walk ==
Count =  7  == Walk ==
Count =  6  == Walk ==
Count =  5  == Walk ==
Count =  4  == Walk ==
Count =  3  == Walk ==
Count =  2  == Walk ==
Count =  1  == Walk ==
 == Do Not Walk ==
```

☑ Autoscroll □ Show timestamp

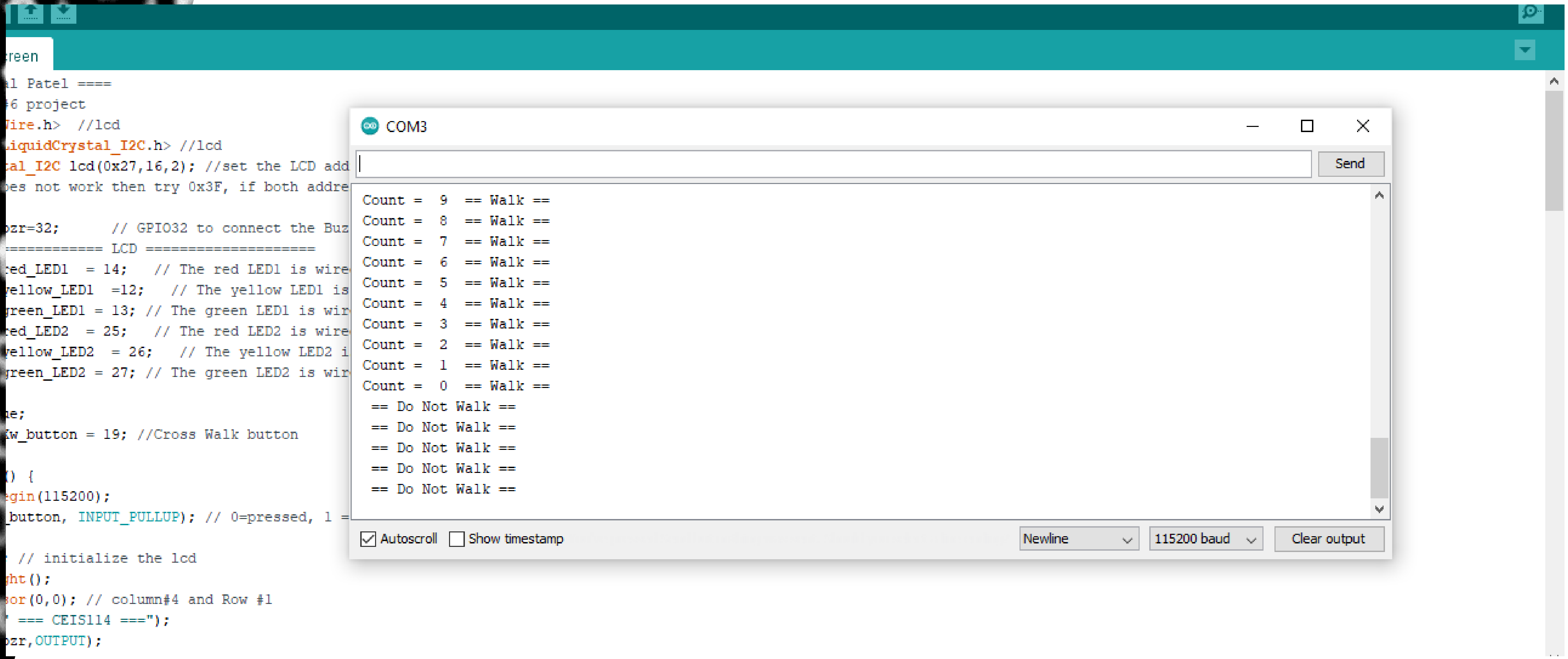
PROJECT 5

ADDING LED SCREEN AND BIPPER

CIRCUIT WITH LIGHT ON/OFF



SCREENSHOT OF SERIAL MONITOR IN ARDUINO IDE



```
l Patel ====
#6 project
Wire.h>  //lcd
LiquidCrystal_I2C.h> //lcd
tal_I2C lcd(0x27,16,2); //set the LCD add
oes not work then try 0x3F, if both addre

bzr=32;      // GPIO32 to connect the Buz
============ LCD ====================
red_LED1  = 14;   // The red LED1 is wire
yellow_LED1  =12;   // The yellow LED1 is
green_LED1 = 13; // The green LED1 is wir
red_LED2  = 25;   // The red LED2 is wire
yellow_LED2  = 26;   // The yellow LED2 i
green_LED2 = 27; // The green LED2 is wir

ne;
Xw_button = 19; //Cross Walk button

() {
egin(115200);
_button, INPUT_PULLUP); // 0=pressed, 1 =

; // initialize the lcd
ght();
sor(0,0); // column#4 and Row #1
" === CEIS114 ===");
bzr,OUTPUT);
```

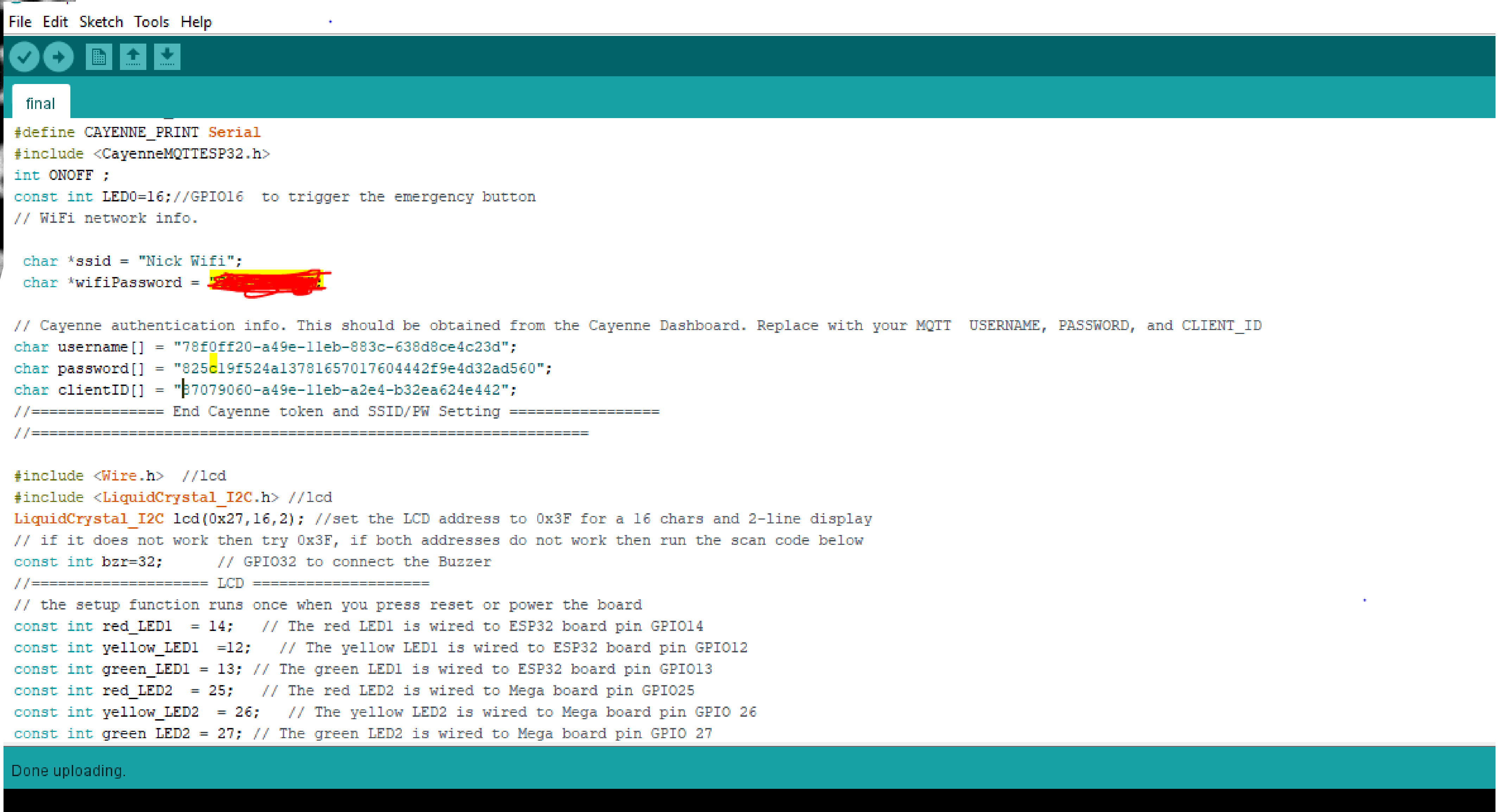
```
Count =  9  == Walk ==
Count =  8  == Walk ==
Count =  7  == Walk ==
Count =  6  == Walk ==
Count =  5  == Walk ==
Count =  4  == Walk ==
Count =  3  == Walk ==
Count =  2  == Walk ==
Count =  1  == Walk ==
Count =  0  == Walk ==
 == Do Not Walk ==
 == Do Not Walk ==
 == Do Not Walk ==
 == Do Not Walk ==
 == Do Not Walk ==
```

☑ Autoscroll ☐ Show timestamp | Newline | 115200 baud | Clear output

PROJECT 6

ADDIN BLUE LIGHT AND CLOUD CONTROLING

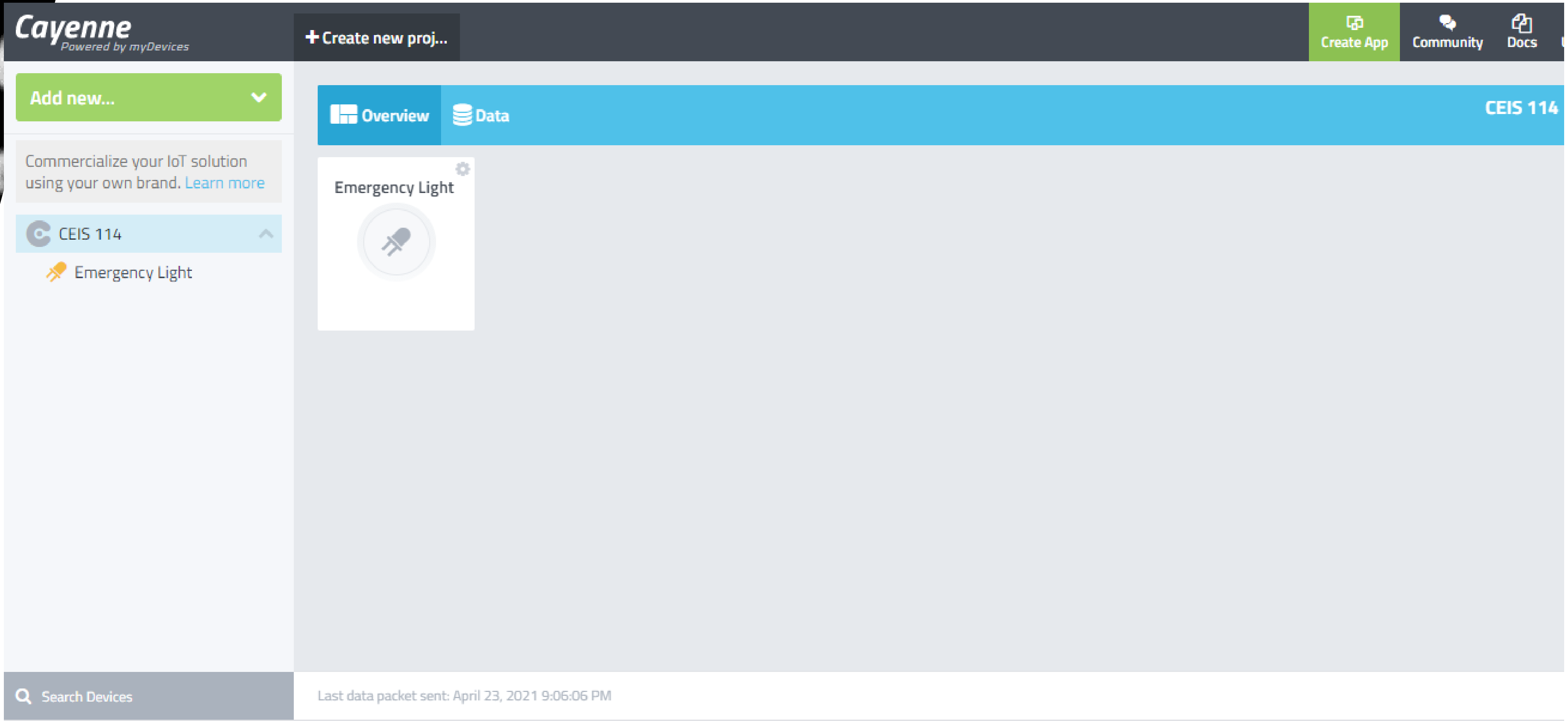
SCREENSHOT OF CODE IN ARDUINO IDE



```
#define CAYENNE_PRINT Serial
#include <CayenneMQTTESP32.h>
int ONOFF ;
const int LED0=16;//GPIO16  to trigger the emergency button
// WiFi network info.

 char *ssid = "Nick Wifi";
 char *wifiPassword = [redacted]

// Cayenne authentication info. This should be obtained from the Cayenne Dashboard. Replace with your MQTT  USERNAME, PASSWORD, and CLIENT_ID
char username[] = "78f0ff20-a49e-11eb-883c-638d8ce4c23d";
char password[] = "825c19f524a137816570176044442f9e4d32ad560";
char clientID[] = "87079060-a49e-11eb-a2e4-b32ea624e442";
//=============== End Cayenne token and SSID/PW Setting =================
//===============================================================

#include <Wire.h>  //lcd
#include <LiquidCrystal_I2C.h> //lcd
LiquidCrystal_I2C lcd(0x27,16,2); //set the LCD address to 0x3F for a 16 chars and 2-line display
// if it does not work then try 0x3F, if both addresses do not work then run the scan code below
const int bzr=32;     // GPIO32 to connect the Buzzer
//==================== LCD ====================
// the setup function runs once when you press reset or power the board
const int red_LED1  = 14;   // The red LED1 is wired to ESP32 board pin GPIO14
const int yellow_LED1  =12;   // The yellow LED1 is wired to ESP32 board pin GPIO12
const int green_LED1 = 13; // The green LED1 is wired to ESP32 board pin GPIO13
const int red_LED2  = 25;   // The red LED2 is wired to Mega board pin GPIO25
const int yellow_LED2  = 26;   // The yellow LED2 is wired to Mega board pin GPIO 26
const int green LED2 = 27; // The green LED2 is wired to Mega board pin GPIO 27
```

Done uploading.

SCREENSHOT EMERGENCY BUTTON



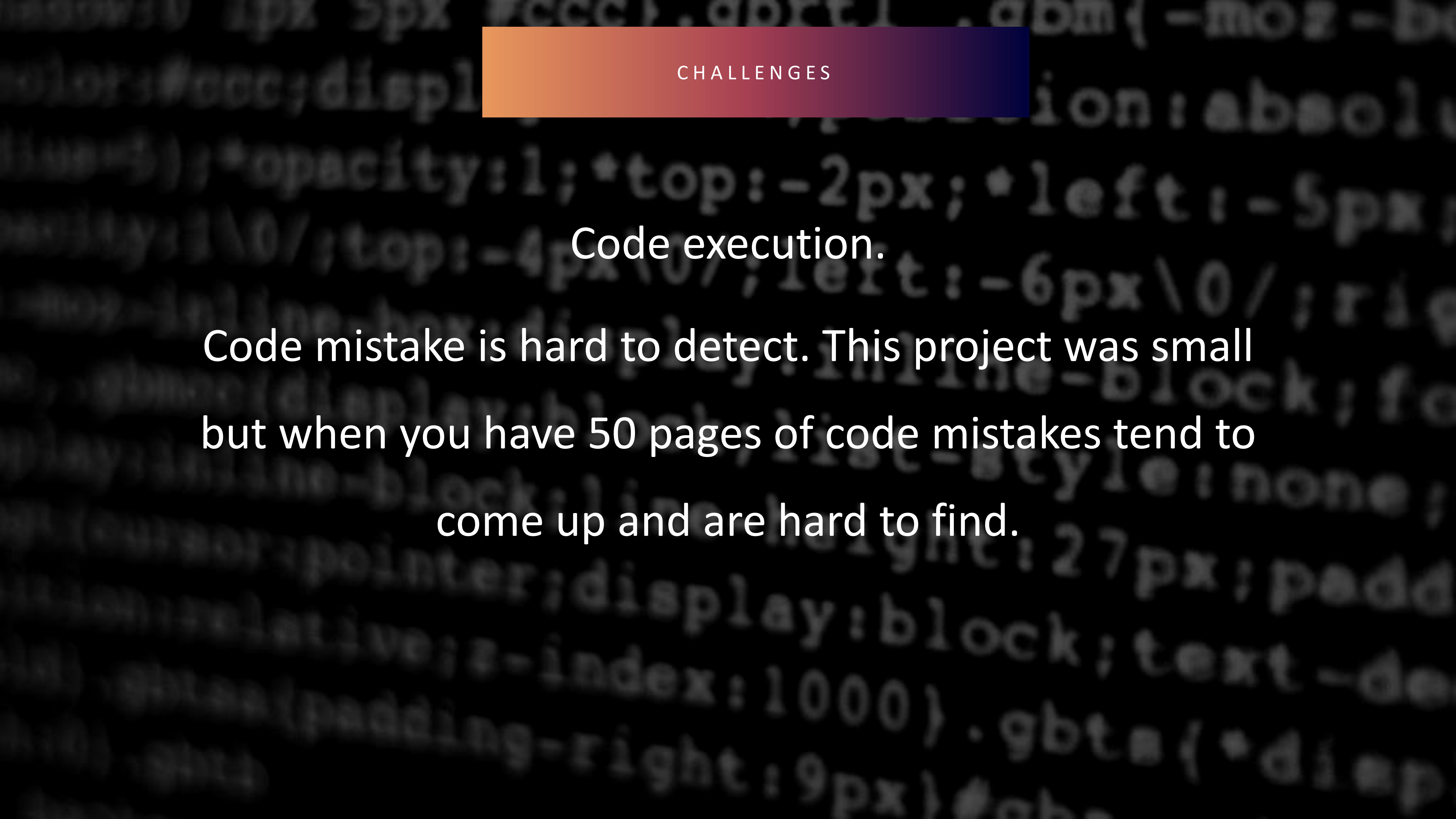
Commercialize your IoT solution using your own brand. Learn more

CEIS 114

Emergency Light

Emergency Light

Last data packet sent: April 23, 2021 9:06:06 PM



CHALLENGES

Code execution.

Code mistake is hard to detect. This project was small but when you have 50 pages of code mistakes tend to come up and are hard to find.

CAREER SKILLS ACQUIRED

Knowledge about the Adriano/ sensor that can use.

Learn how to code for it and execute it.

How to monitor in serial monitor what is happening with the system

Find the error and where to look for it.

In conclusion, I learn how to operate and work with Arduino. New coding skill, how to plan the project and execute. Create new and online access key. That I can control it online

☎ +1 (630) 317 4732

Thank you.

✉ Nira1988@gmail.com