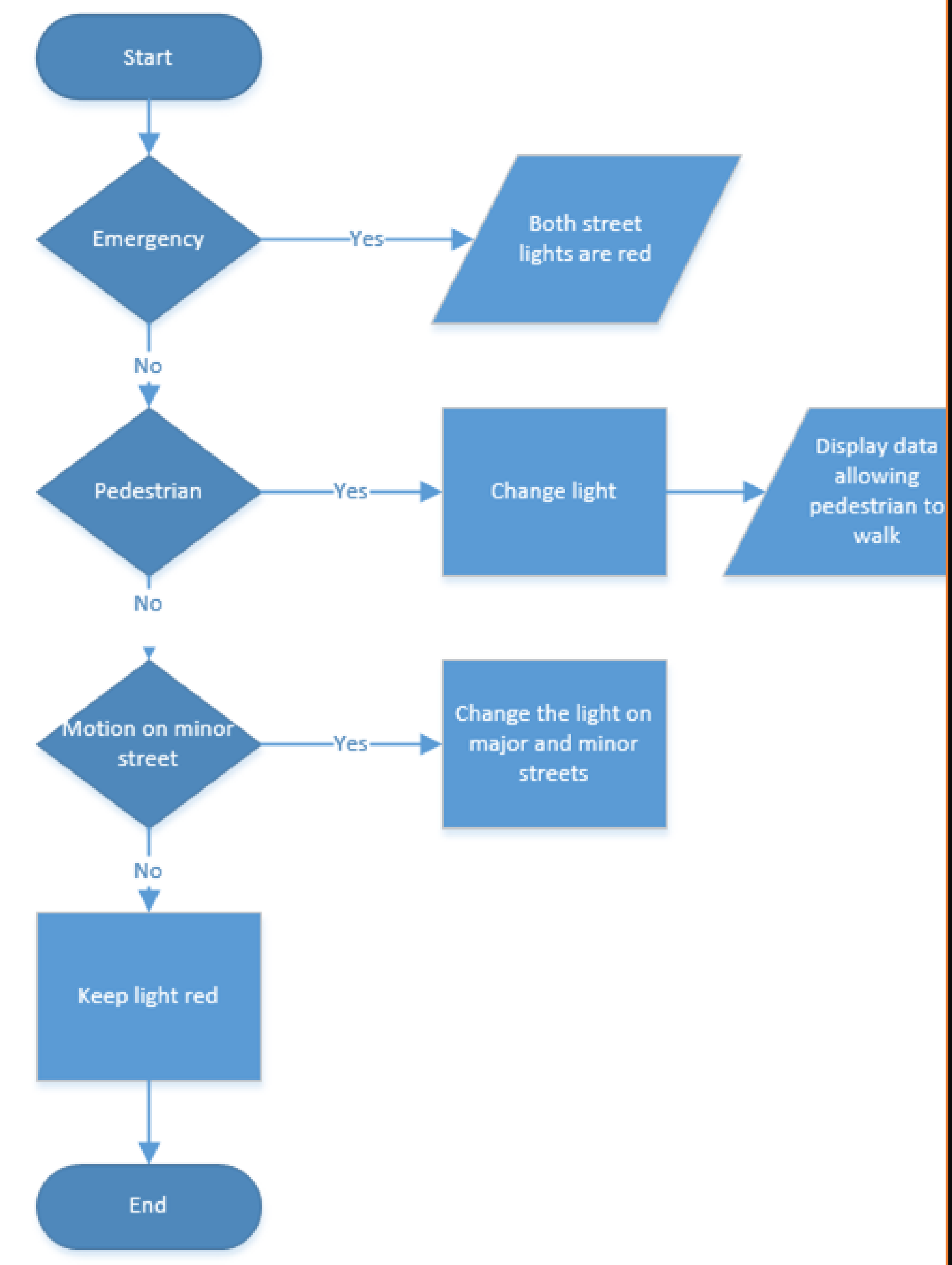
CEIS 114

Creating the Traffic Controller

Introduction

- A Traffic Control ensures safe pedestrian crosswalks as coordinating with traffic to keep everything running smoothly.

Flowchart

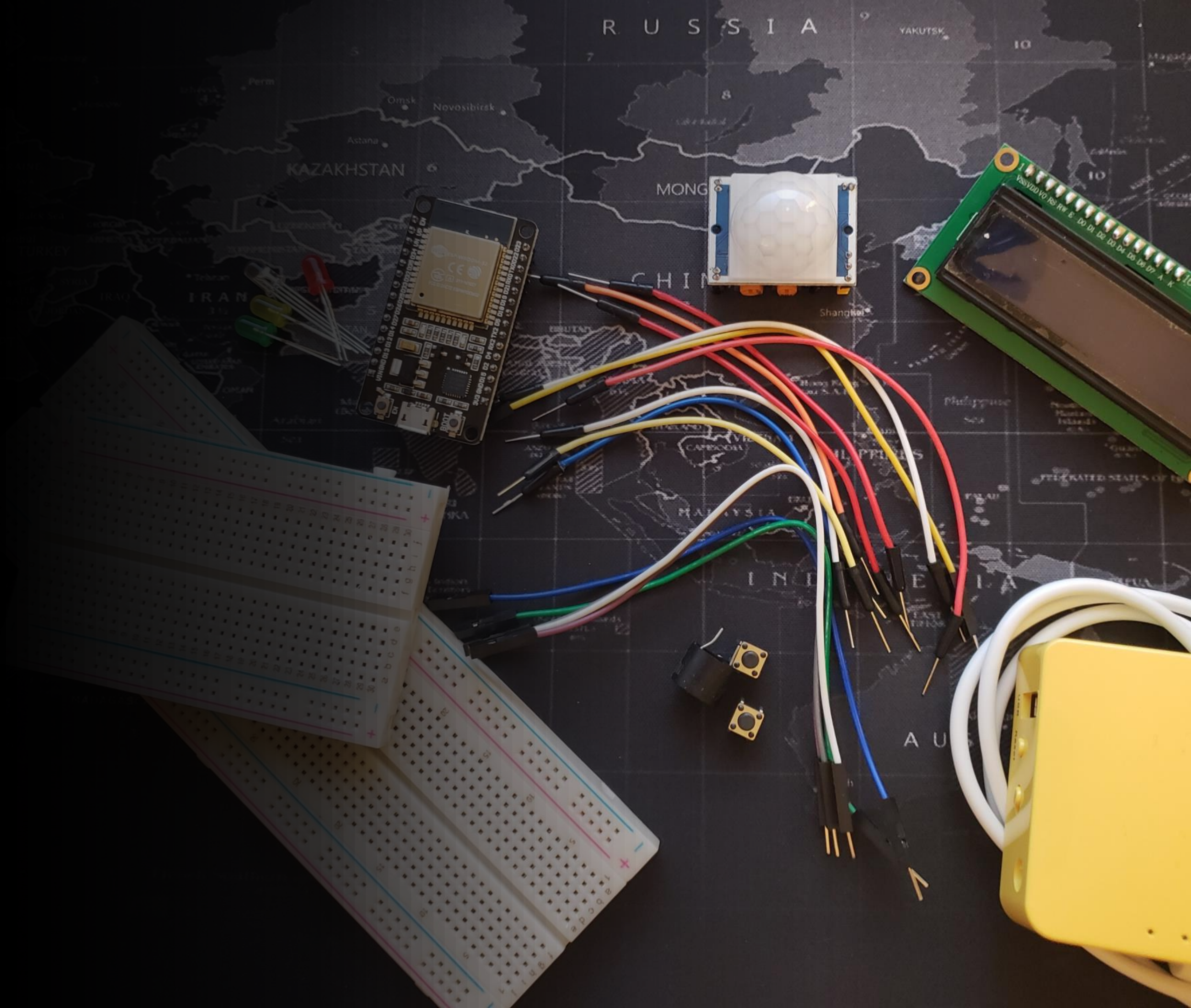

Start
Emergency
Yes
Both street lights are red
No
Pedestrian
Yes
Change light
Display data allowing pedestrian to walk
No
Motion on minor street
Yes
Change the light on major and minor streets
No
Keep light red
End

Input – Processing - Output

Input	Process	Output
Emergency Pedestrian on crosswalk Motion on minor street	Change the light to red Change light for pedestrian Change lights	Walk/Do not walk

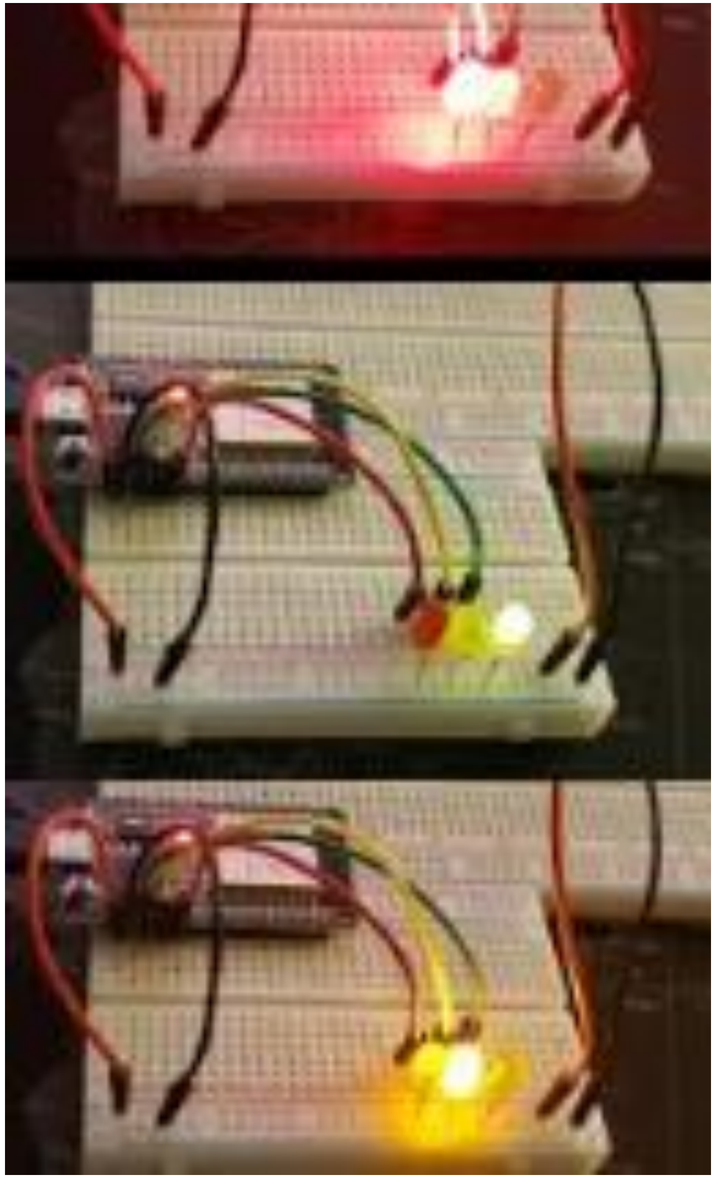
Inventory

- ESP32 Board
- Colored LEDs: Red, Yellow, Green, and Blue
- 220 Ohm Resistors (optional)
- Wires
- Breadboard
- LCD Unit
- Buzzer
- Mini Router
- Push Buttons – 2
- Motion Detector



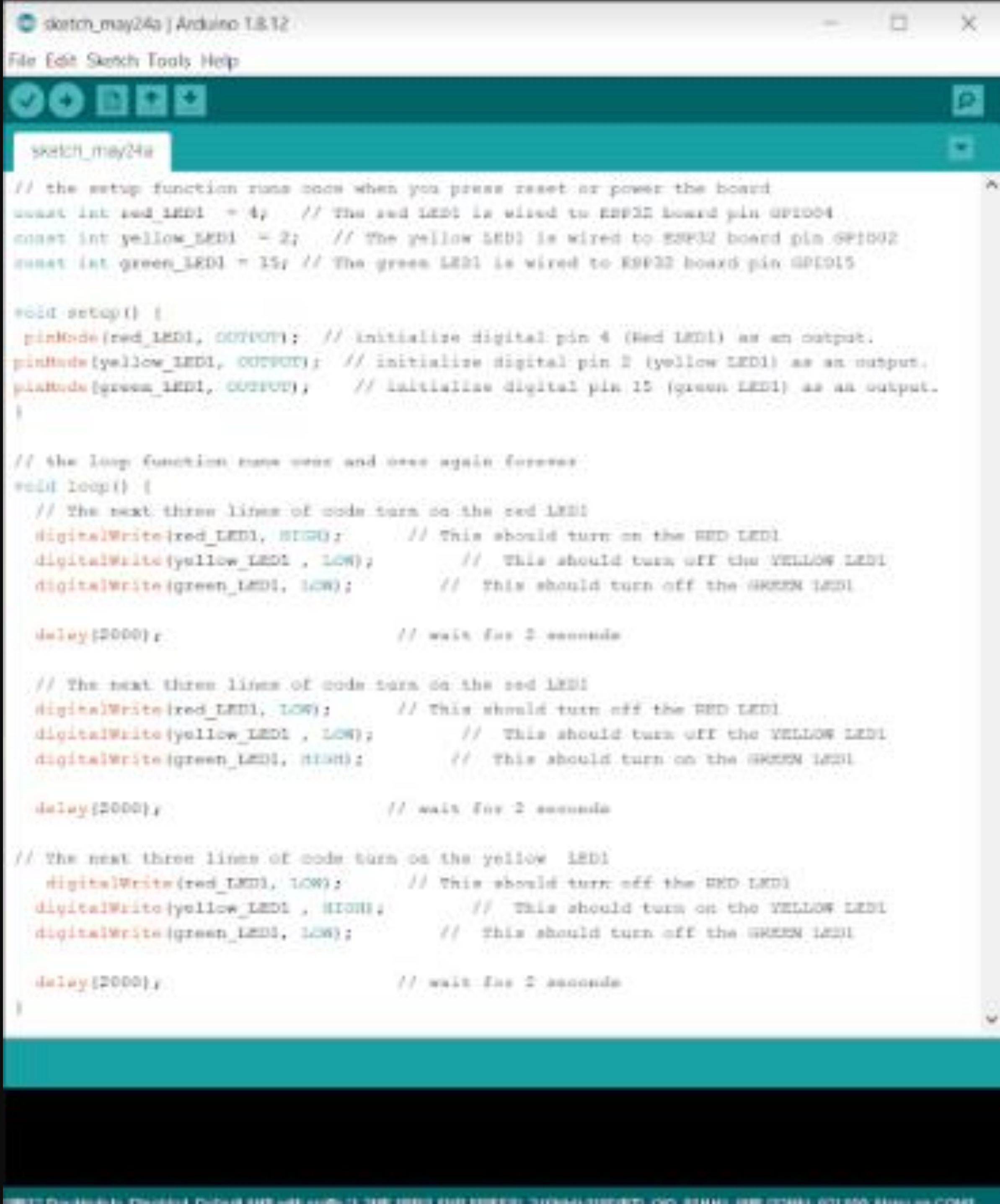
Circuit with 3 LEDs

- ESP 32 Board
- Colored LEDs: Red, Yellow and Green
- Wires
- Breadboard



Arduino IDE

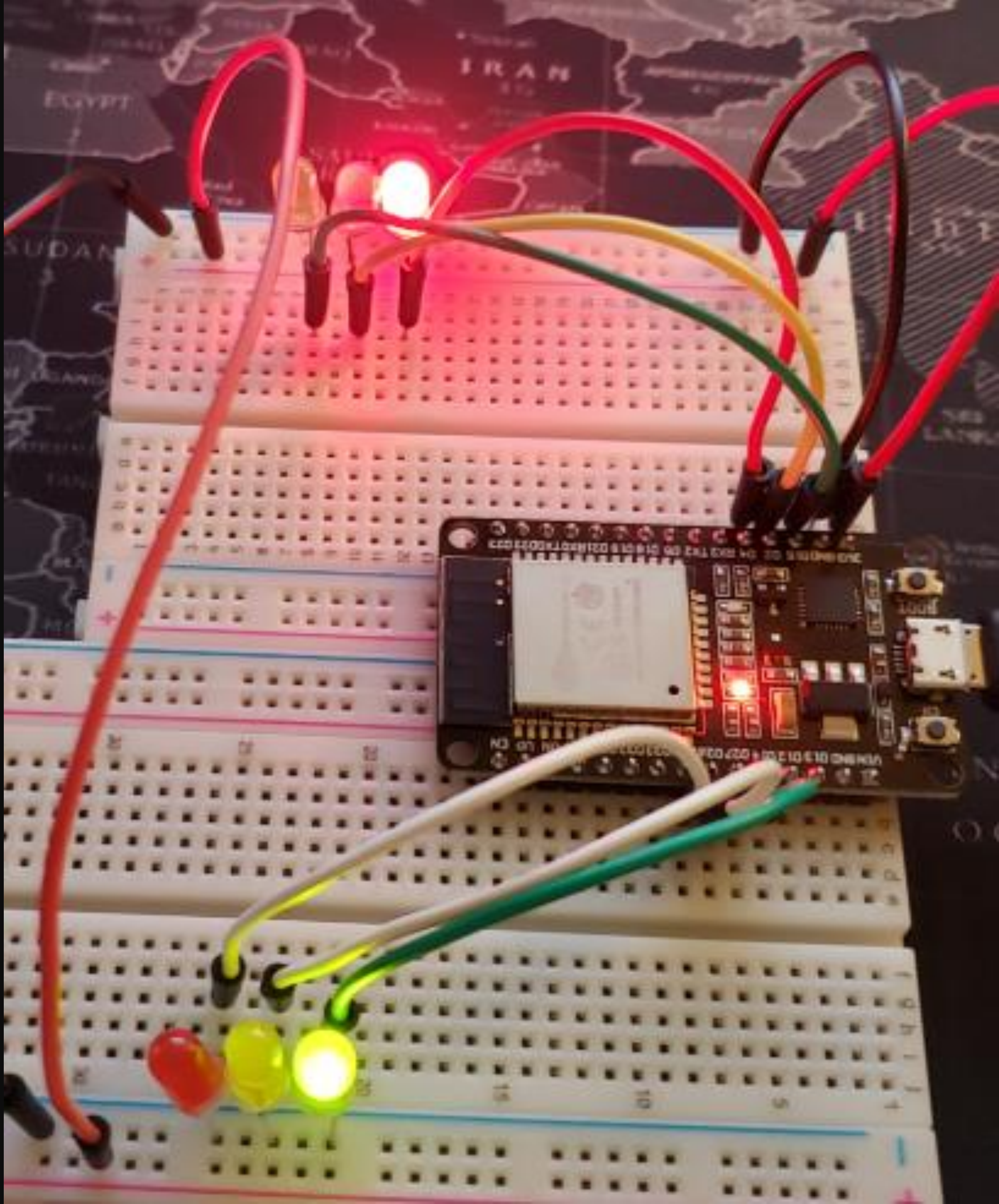
- Code for a single controller



```
// the setup function runs once when you press reset or power the board
const int red_LED1  = 4;   // The red LED1 is wired to ESP32 board pin GPIO04
const int yellow_LED1  = 2;   // The yellow LED1 is wired to ESP32 board pin GPIO02
const int green_LED1 = 15; // The green LED1 is wired to ESP32 board pin GPIO15

void setup() {
 pinMode(red_LED1, OUTPUT);  // initialize digital pin 4 (Red LED1) as an output.
pinMode(yellow_LED1, OUTPUT);  // initialize digital pin 2 (yellow LED1) as an output.
pinMode(green_LED1, OUTPUT);    // initialize digital pin 15 (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 red 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

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

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

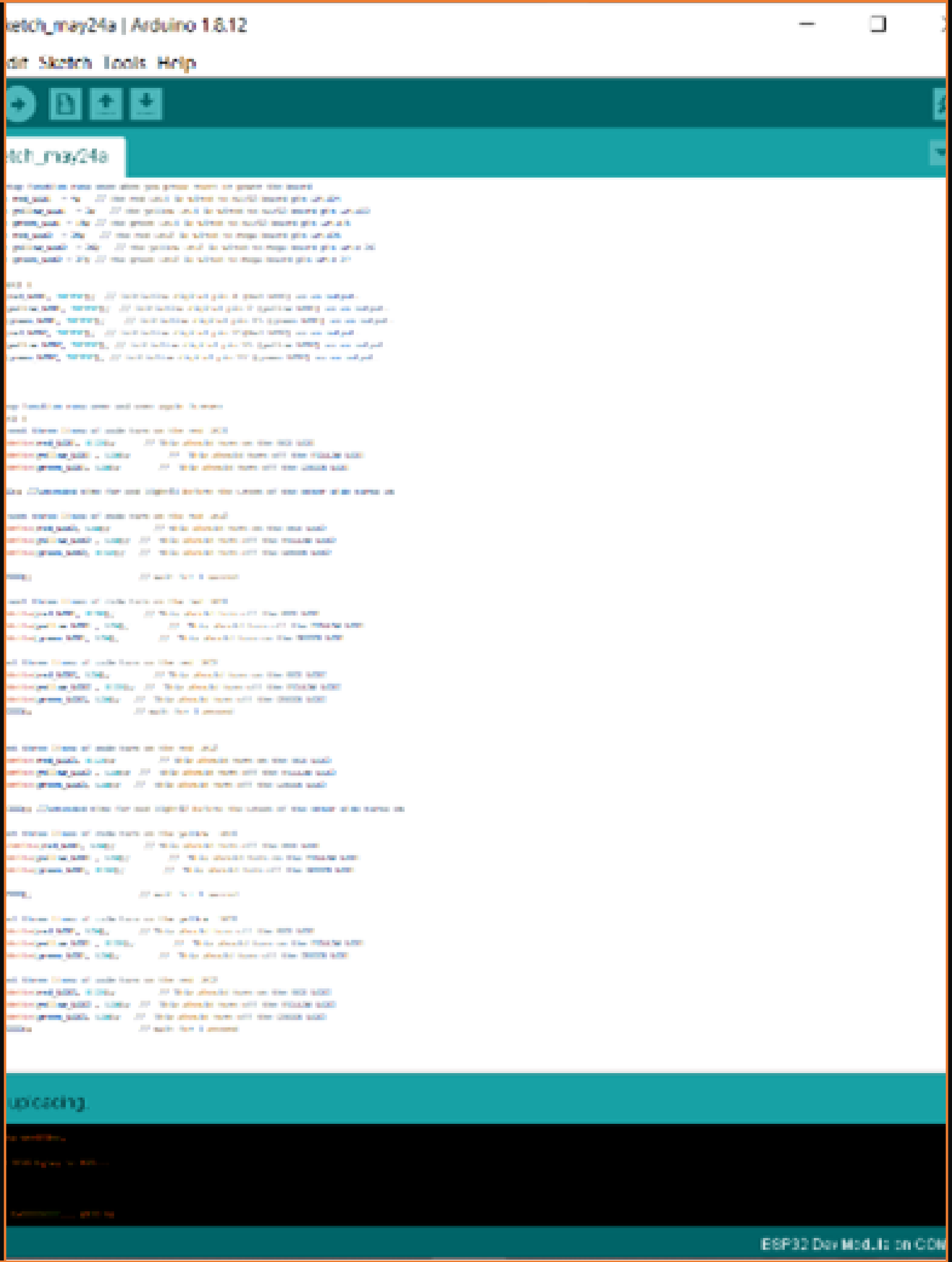
Circuit with 2 Controllers

- ESP32 Board
- Two sets of colored LEDs: Red, Yellow and Green
- 220 Ohm Resistors (optional)
- Wires
- Breadboard



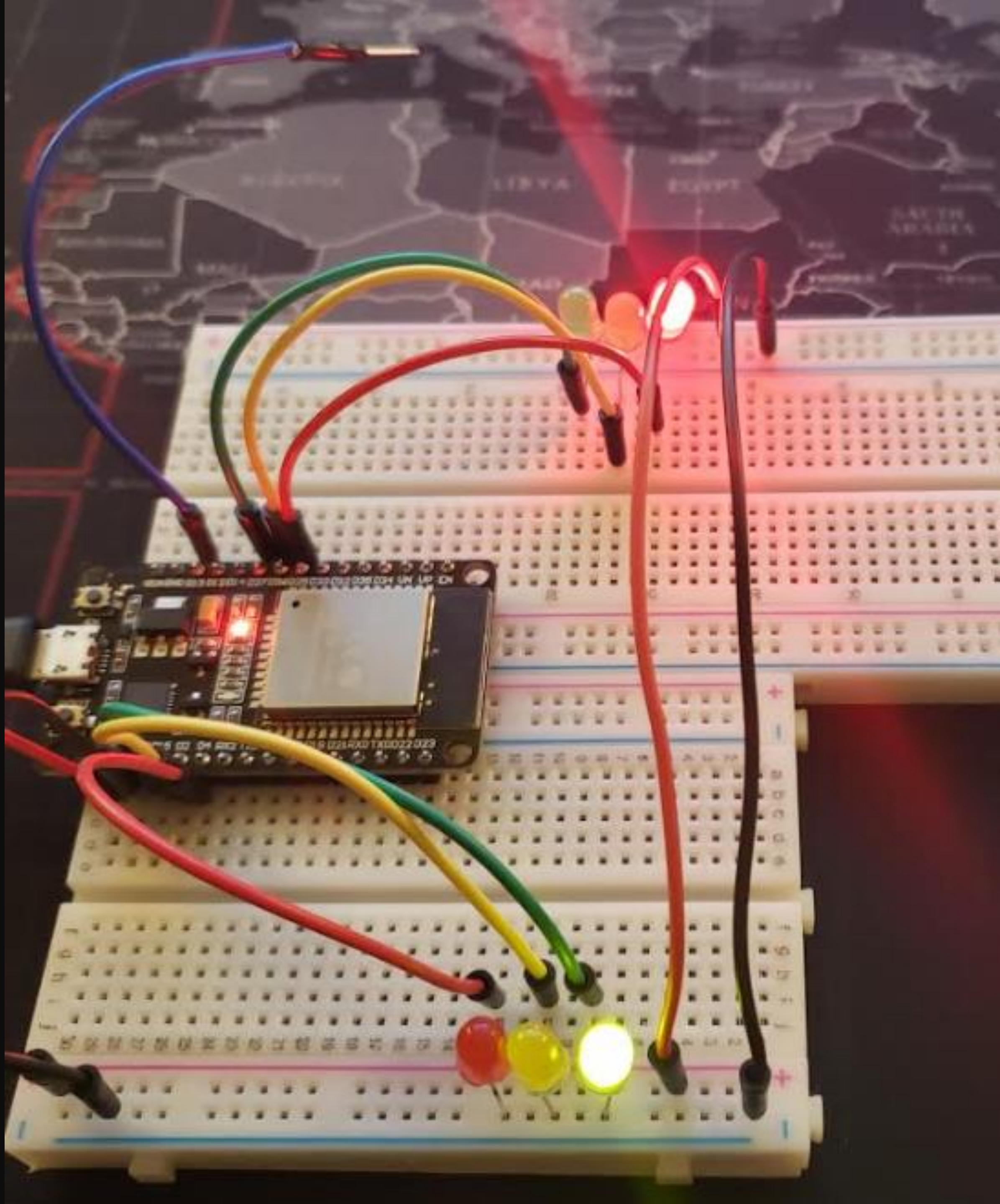
Arduino IDE

Code in Arduino IDE showing 2 controllers



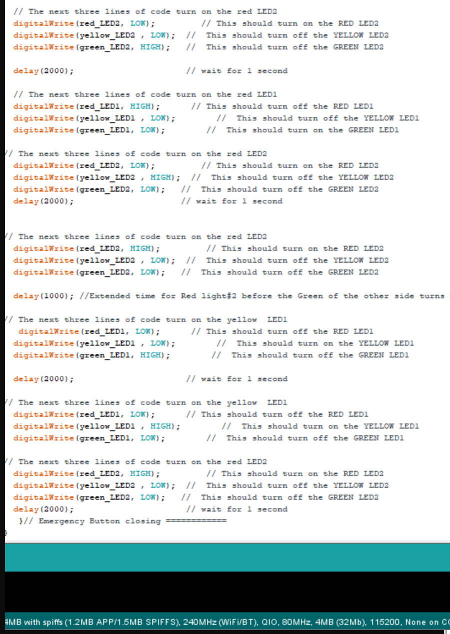
Circuit with 2 sets of controllers and touch sensor

- ESP32 Board
- Two sets of Colored LEDs: Red, Yellow and Green
- Touch Sensor (Wire connected to GPIO 13)
- Wires
- Breadboard



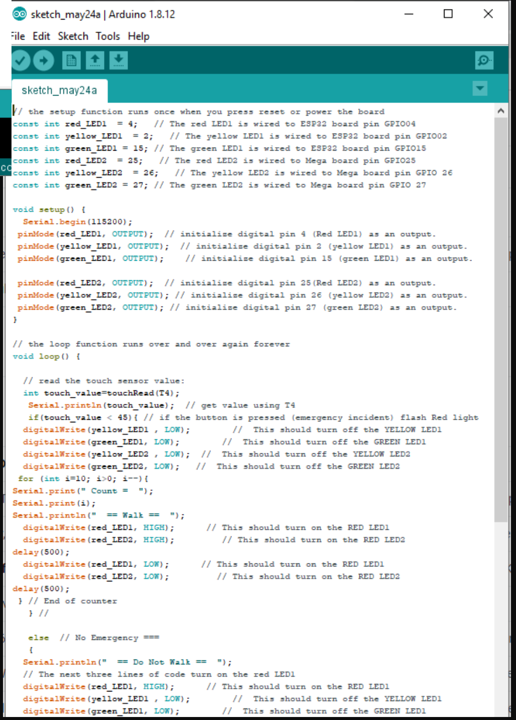
Code used on the Arduino IDE

Code showing 2 controllers with touch sensor



```
 // The next three lines of code turn on the red LED2
 digitalWrite(red_LED2, LOW);          // This should turn on the RED LED2
 digitalWrite(yellow_LED2 , LOW);  //  This should turn off the YELLOW LED2
 digitalWrite(green_LED2, HIGH);   //  This should turn off the GREEN LED2

 delay(2000);                      // wait for 1 second

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

// The next three lines of code turn on the red LED2
 digitalWrite(red_LED2, LOW);          // This should turn on the RED LED2
 digitalWrite(yellow_LED2 , HIGH);  //  This should turn off the YELLOW LED2
 digitalWrite(green_LED2, LOW);   //  This should turn off the GREEN LED2
 delay(2000);                     // wait for 1 second


// The next three lines of code turn on the red LED2
 digitalWrite(red_LED2, HIGH);          // This should turn on the RED LED2
 digitalWrite(yellow_LED2 , LOW);  //  This should turn off the YELLOW LED2
 digitalWrite(green_LED2, LOW);   //  This should turn off the GREEN LED2

 delay(1000); //Extended time for Red light#2 before the Green of the other side turns O

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

 delay(2000);                      // wait for 1 second

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

// The next three lines of code turn on the red LED2
 digitalWrite(red_LED2, HIGH);          // This should turn on the RED LED2
 digitalWrite(yellow_LED2 , LOW);  //  This should turn off the YELLOW LED2
 digitalWrite(green_LED2, LOW);   //  This should turn off the GREEN LED2
 delay(2000);                     // wait for 1 second
  }// Emergency Button closing ============
}
```

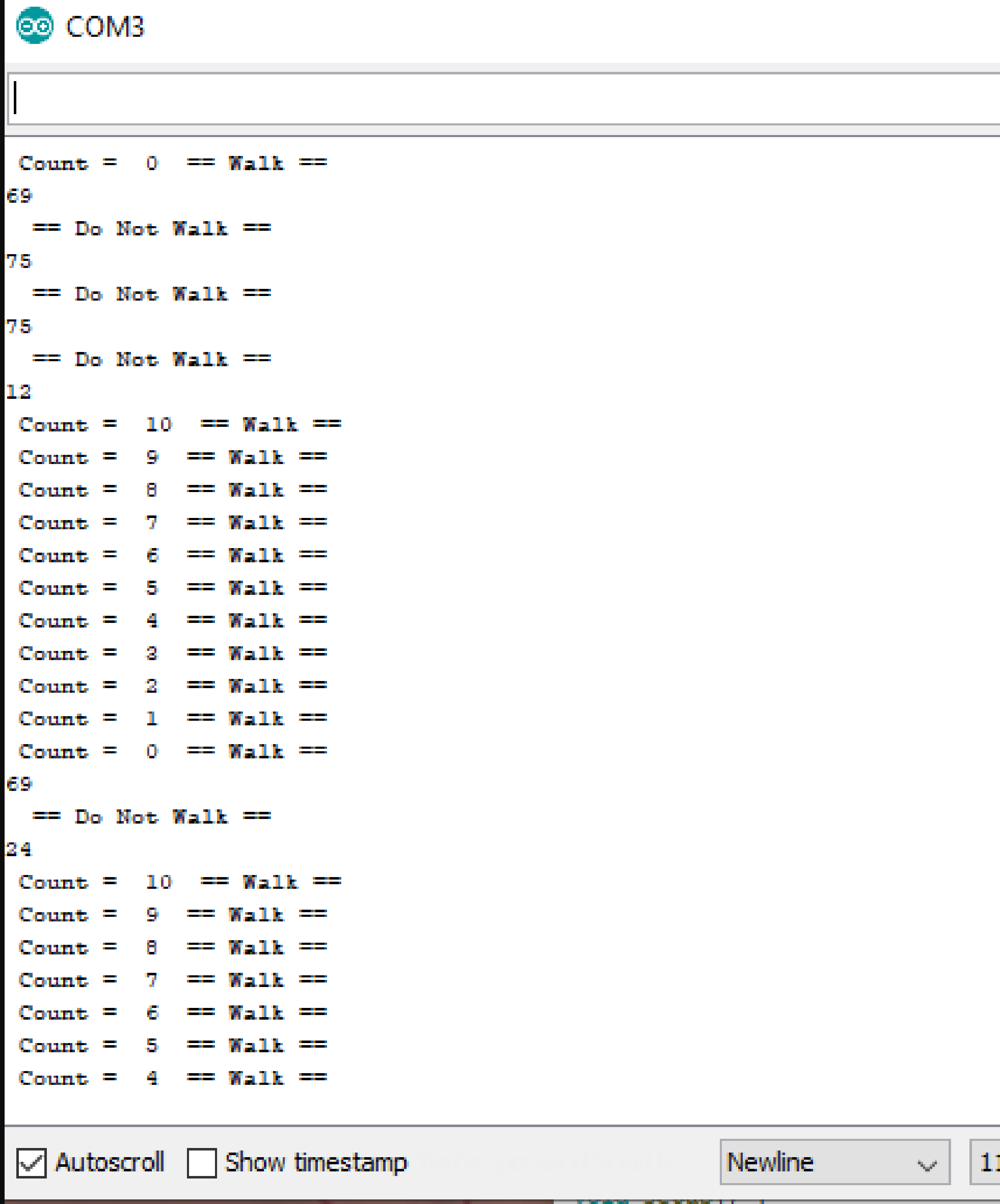
4MB with spiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 115200, None on CC



```
// the setup function runs once when you press reset or power the board
const int red_LED1  = 4;   // The red LED1 is wired to ESP32 board pin GPIO04
const int yellow_LED1  = 2;   // The yellow LED1 is wired to ESP32 board pin GPIO02
const int green_LED1 = 15; // The green LED1 is wired to ESP32 board pin GPIO15
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

void setup() {
  Serial.begin(115200);
 pinMode(red_LED1, OUTPUT);  // initialize digital pin 4 (Red LED1) as an output.
 pinMode(yellow_LED1, OUTPUT);  // initialize digital pin 2 (yellow LED1) as an output.
 pinMode(green_LED1, OUTPUT);    // initialize digital pin 15 (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() {

  // read the touch sensor value:
  int touch_value=touchRead(T4);
   Serial.println(touch_value);  // get value using T4
   if(touch_value < 45){ // if the button is pressed (emergency incident) flash Red light
  digitalWrite(yellow_LED1 , LOW);         //  This should turn off the YELLOW LED1
  digitalWrite(green_LED1, LOW);         //  This should turn off the GREEN LED1
  digitalWrite(yellow_LED2 , LOW);  //  This should turn off the YELLOW LED2
  digitalWrite(green_LED2, LOW);   //  This should turn off the GREEN LED2
 for (int i=10; i>0; i--){
Serial.print(" Count =  ");
Serial.print(i);
Serial.println("  == Walk ==  ");
  digitalWrite(red_LED1, HIGH);      // This should turn on the RED LED1
  digitalWrite(red_LED2, HIGH);          // This should turn on the RED LED2
delay(500);
  digitalWrite(red_LED1, LOW);      // This should turn on the RED LED1
  digitalWrite(red_LED2, LOW);          // This should turn on the RED LED2
delay(500);
 } // End of counter
   } //

   else  // No Emergency ===
   {
  Serial.println("  == Do Not Walk ==  ");
  // 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
```

Serial Monitor

- Output from touch sensor



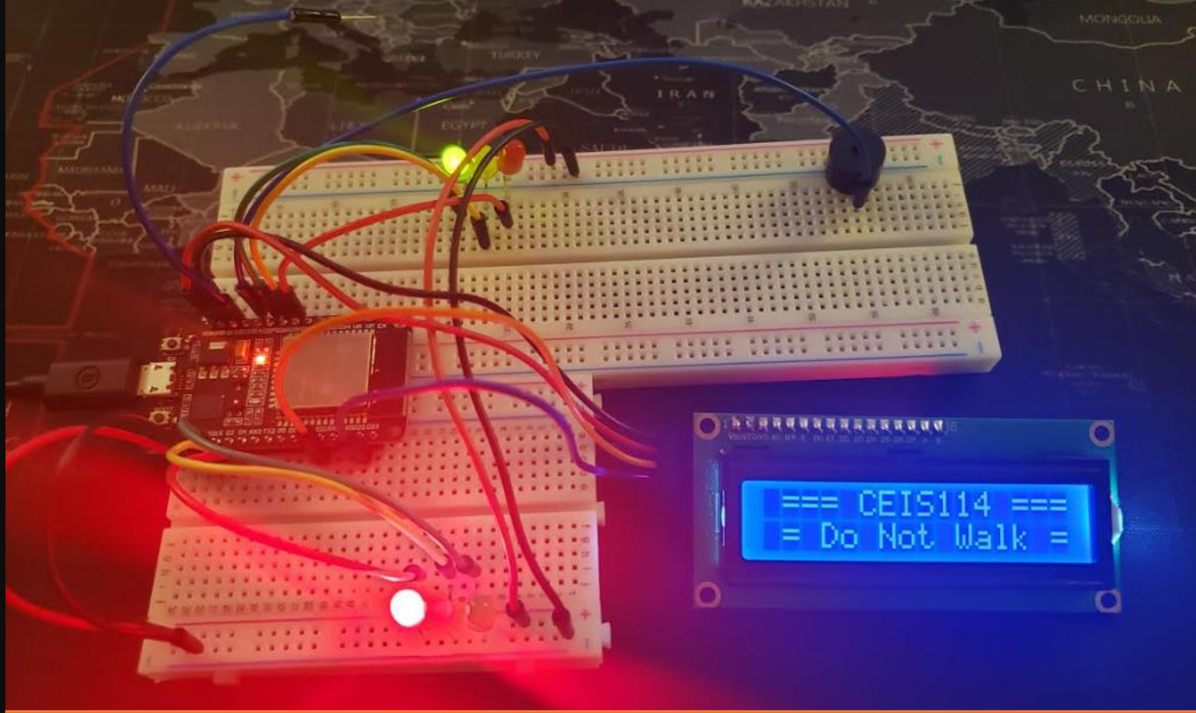
COM3

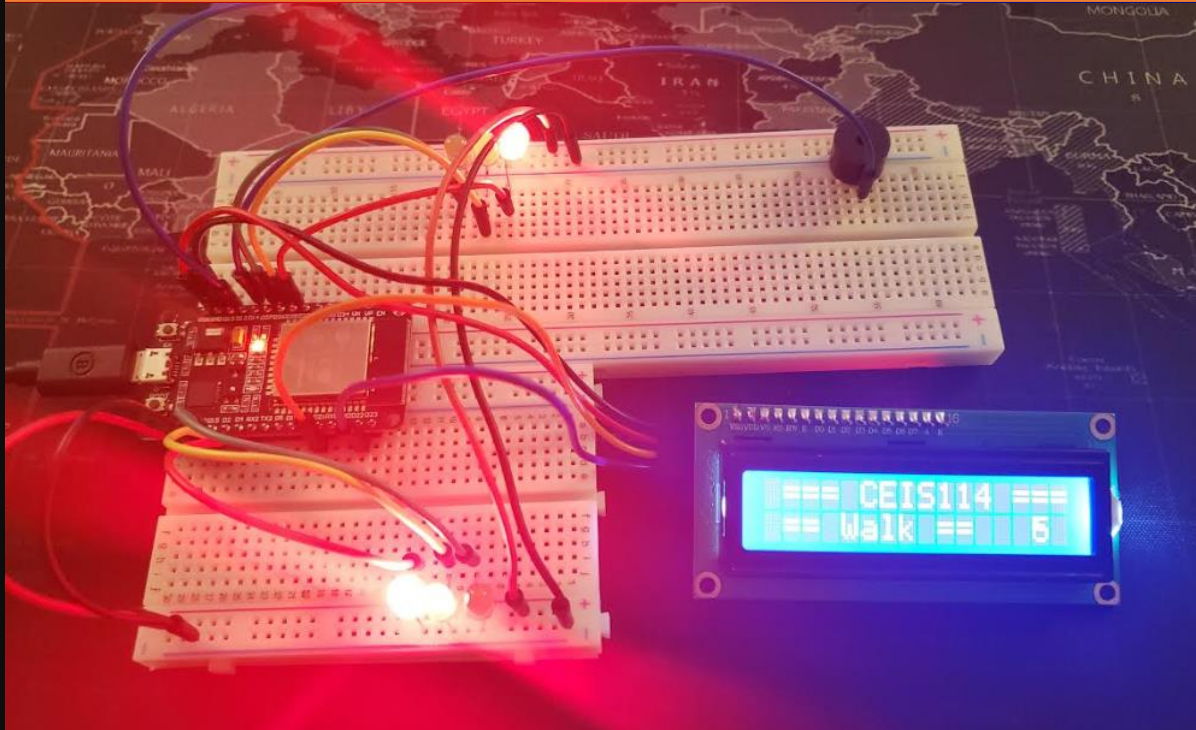
```
 Count =  0   == Walk ==
69
  == Do Not Walk ==
75
  == Do Not Walk ==
75
  == Do Not Walk ==
12
 Count =  10   == Walk ==
 Count =  9   == Walk ==
 Count =  8   == Walk ==
 Count =  7   == Walk ==
 Count =  6   == Walk ==
 Count =  5   == Walk ==
 Count =  4   == Walk ==
 Count =  3   == Walk ==
 Count =  2   == Walk ==
 Count =  1   == Walk ==
 Count =  0   == Walk ==
69
  == Do Not Walk ==
24
 Count =  10   == Walk ==
 Count =  9   == Walk ==
 Count =  8   == Walk ==
 Count =  7   == Walk ==
 Count =  6   == Walk ==
 Count =  5   == Walk ==
 Count =  4   == Walk ==
```

☑ Autoscroll □ Show timestamp Newline

Circuit with LCD

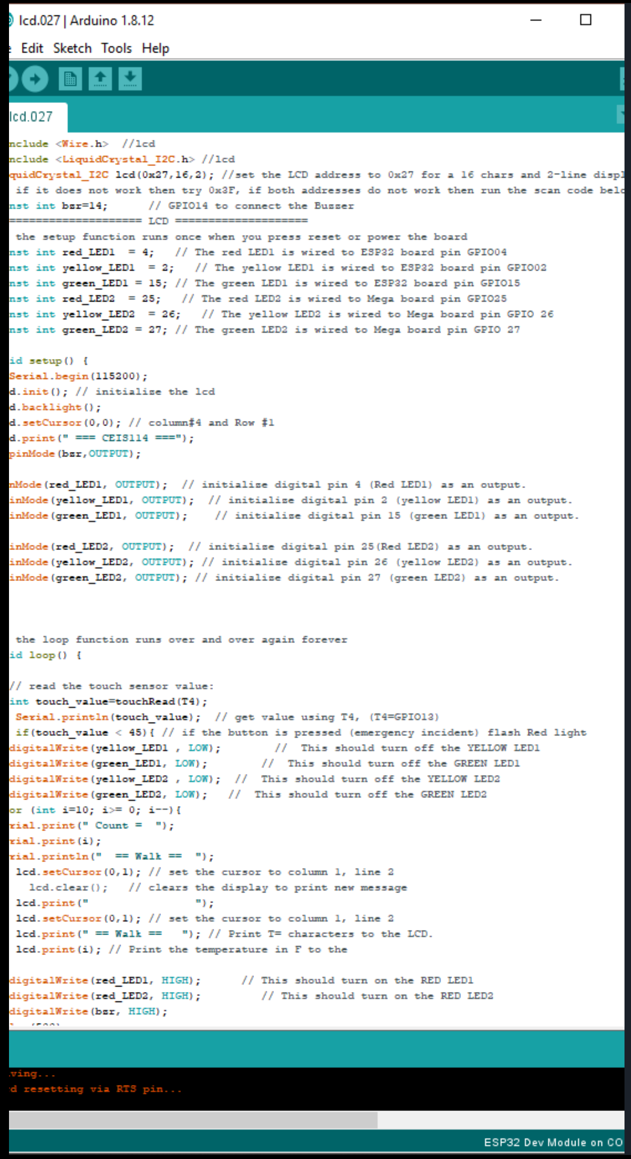
- ESP32 Board
- Two sets of Colored LEDs: Red, Yellow and Green
- Touch Sensor (Wire connected to GPIO 13)
- LCD Unit
- Buzzer
- Wires
- Breadboard





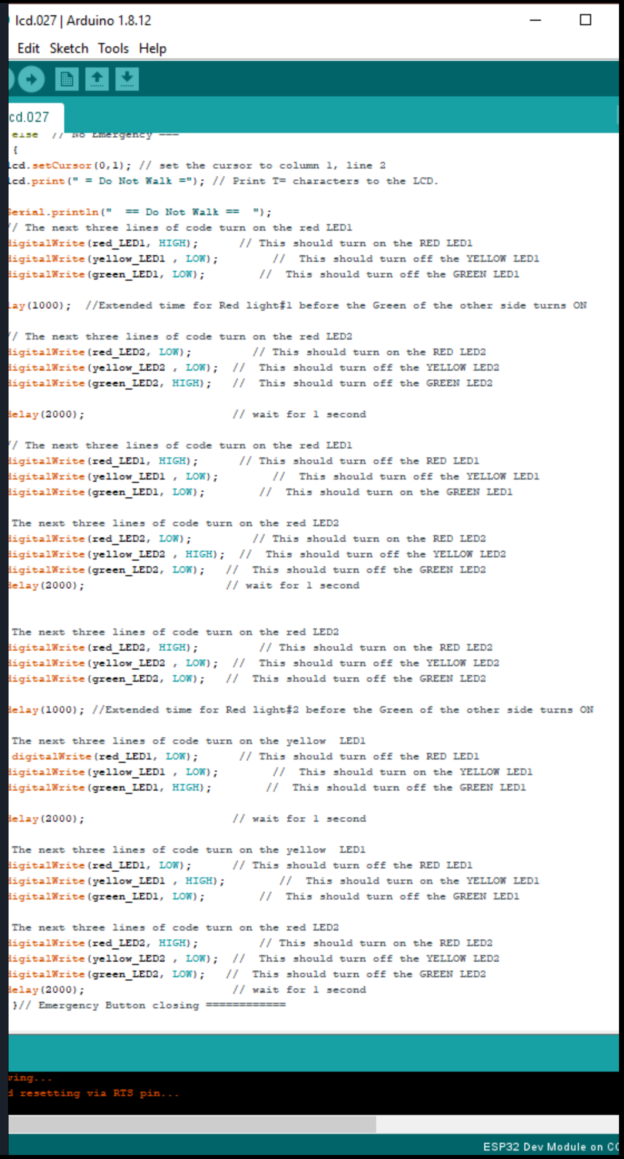
Arduino IDE

- Code to run LEDs, buzzer, touch sensor and LCD panel



```
nclude <Wire.h>  //lcd
nclude <LiquidCrystal_I2C.h> //lcd
quidCrystal_I2C lcd(0x27,16,2); //set the LCD address to 0x27 for a 16 chars and 2-line displ
 if it does not work then try 0x3F, if both addresses do not work then run the scan code belo
nst int bzr=14;      // GPIO14 to connect the Buzzer
==================== LCD ====================
 the setup function runs once when you press reset or power the board
nst int red_LED1  = 4;   // The red LED1 is wired to ESP32 board pin GPIO04
nst int yellow_LED1  = 2;   // The yellow LED1 is wired to ESP32 board pin GPIO02
nst int green_LED1 = 15; // The green LED1 is wired to ESP32 board pin GPIO15
nst int red_LED2  = 25;   // The red LED2 is wired to Mega board pin GPIO25
nst int yellow_LED2  = 26;   // The yellow LED2 is wired to Mega board pin GPIO 26
nst int green_LED2 = 27; // The green LED2 is wired to Mega board pin GPIO 27

id setup() {
Serial.begin(115200);
d.init(); // initialize the lcd
d.backlight();
d.setCursor(0,0); // column#4 and Row #1
d.print(" === CEIS114 ===");
pinMode(bzr,OUTPUT);

nMode(red_LED1, OUTPUT);  // initialize digital pin 4 (Red LED1) as an output.
inMode(yellow_LED1, OUTPUT);  // initialize digital pin 2 (yellow LED1) as an output.
inMode(green_LED1, OUTPUT);    // initialize digital pin 15 (green LED1) as an output.

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



 the loop function runs over and over again forever
id loop() {

// read the touch sensor value:
int touch_value=touchRead(T4);
 Serial.println(touch_value);  // get value using T4, (T4=GPIO13)
 if(touch_value < 45){ // if the button is pressed (emergency incident) flash Red light
digitalWrite(yellow_LED1 , LOW);        //  This should turn off the YELLOW LED1
digitalWrite(green_LED1, LOW);        //  This should turn off the GREEN LED1
digitalWrite(yellow_LED2 , LOW);  //  This should turn off the YELLOW LED2
digitalWrite(green_LED2, LOW);   //  This should turn off the GREEN LED2
or (int i=10; i>= 0; i--){
rial.print(" Count =  ");
rial.print(i);
rial.println("  == Walk ==  ");
 lcd.setCursor(0,1); // set the cursor to column 1, line 2
   lcd.clear();   // clears the display to print new message
 lcd.print("                ");
 lcd.setCursor(0,1); // set the cursor to column 1, line 2
 lcd.print(" == Walk ==   "); // Print T= characters to the LCD.
 lcd.print(i); // Print the temperature in F to the

digitalWrite(red_LED1, HIGH);      // This should turn on the RED LED1
digitalWrite(red_LED2, HIGH);         // This should turn on the RED LED2
digitalWrite(bzr, HIGH);
```

```
ving...
d resetting via RTS pin...
```



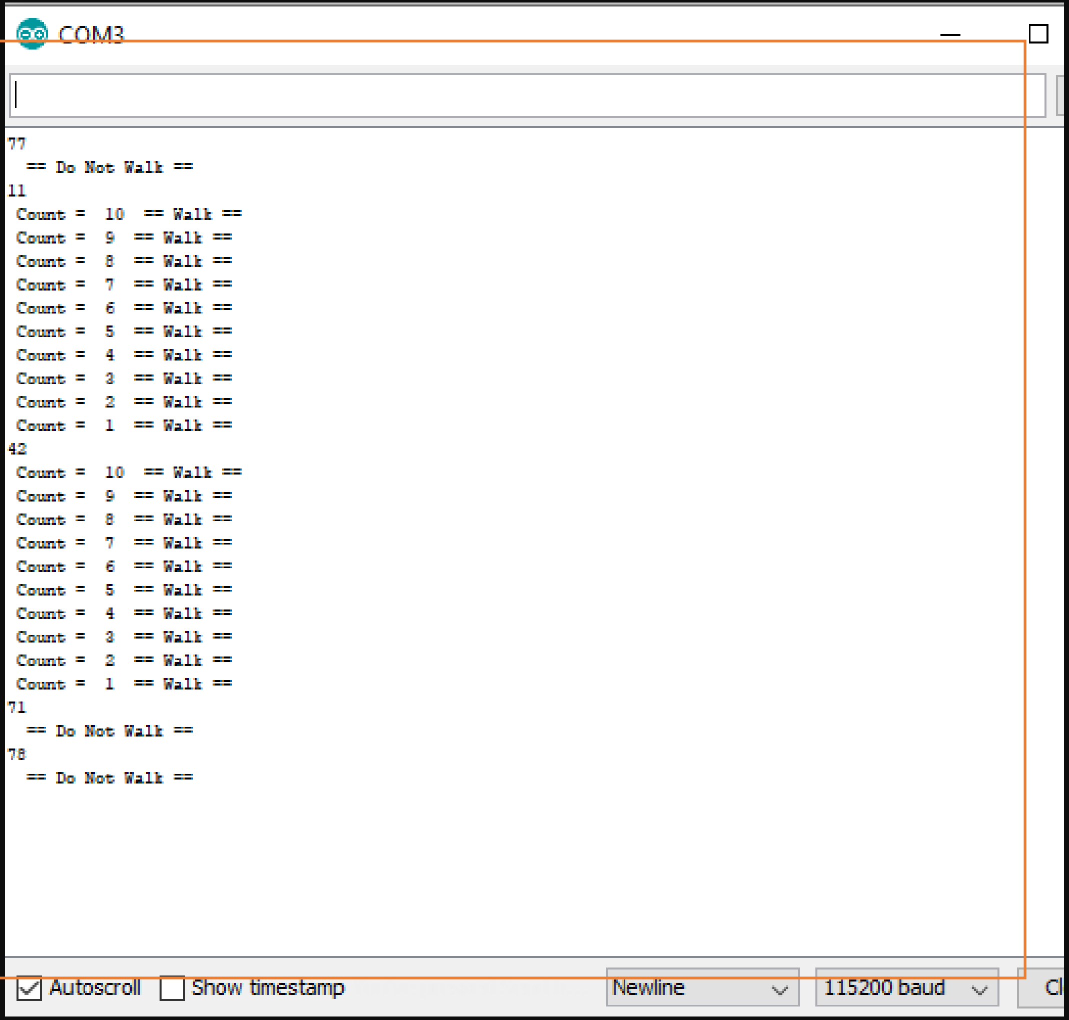
```
else  // No Emergency ---
{
lcd.setCursor(0,1); // set the cursor to column 1, line 2
lcd.print(" = Do Not Walk ="); // Print T= characters to the LCD.

Serial.println("  == Do Not Walk ==  ");
// 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

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

// The next three lines of code turn on the red LED2
digitalWrite(red_LED2, LOW);         // This should turn on the RED LED2
digitalWrite(yellow_LED2 , LOW);  //  This should turn off the YELLOW LED2
digitalWrite(green_LED2, HIGH);   //  This should turn off the GREEN LED2

delay(2000);                      // wait for 1 second

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

 The next three lines of code turn on the red LED2
digitalWrite(red_LED2, LOW);         // This should turn on the RED LED2
digitalWrite(yellow_LED2 , HIGH);  //  This should turn off the YELLOW LED2
digitalWrite(green_LED2, LOW);   //  This should turn off the GREEN LED2
delay(2000);                     // wait for 1 second


 The next three lines of code turn on the red LED2
digitalWrite(red_LED2, HIGH);         // This should turn on the RED LED2
digitalWrite(yellow_LED2 , LOW);  //  This should turn off the YELLOW LED2
digitalWrite(green_LED2, LOW);   //  This should turn off the GREEN LED2

delay(1000); //Extended time for Red light#2 before the Green of the other side turns ON

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

delay(2000);                      // wait for 1 second

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

 The next three lines of code turn on the red LED2
digitalWrite(red_LED2, HIGH);         // This should turn on the RED LED2
digitalWrite(yellow_LED2 , LOW);  //  This should turn off the YELLOW LED2
digitalWrite(green_LED2, LOW);   //  This should turn off the GREEN LED2
delay(2000);                      // wait for 1 second
 }// Emergency Button closing ============
```

```
ving...
d resetting via RTS pin...
```

Serial Monitor

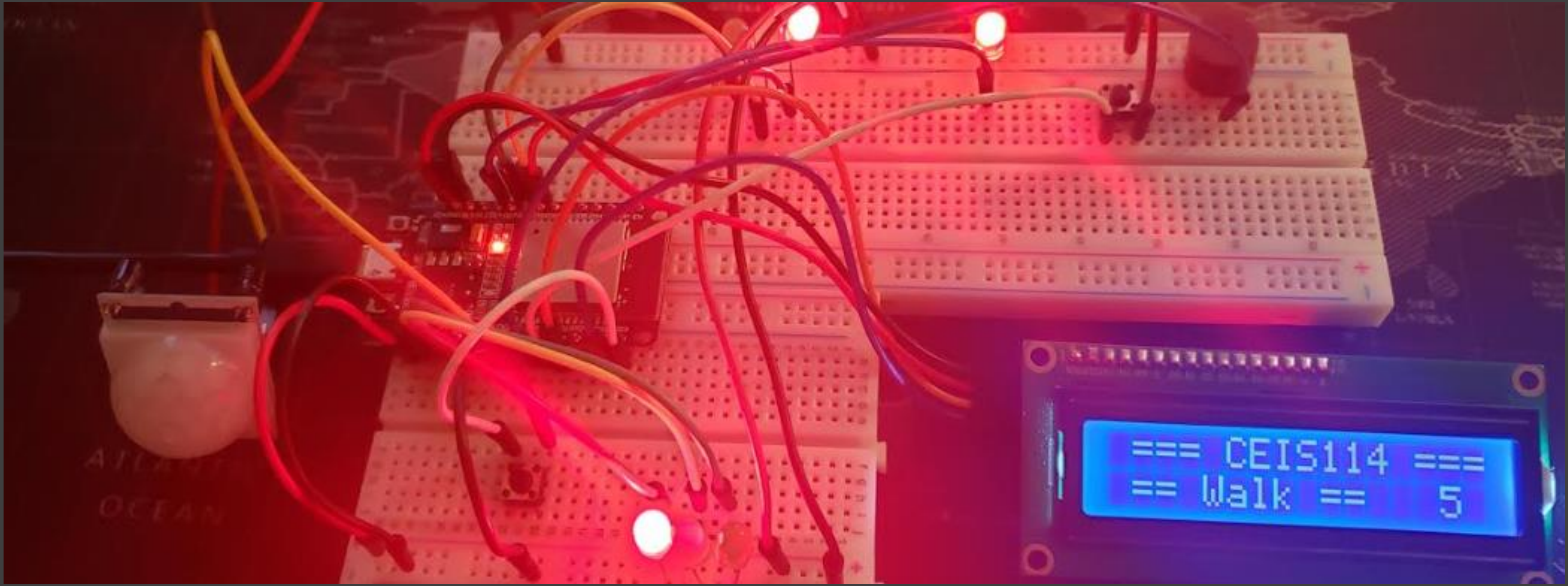
- Do Not Walk
- -When one light is green and no input to the sensor

- Walk with 10 second countdown
- -Triggered when input is given to the sensor



```
77
  == Do Not Walk ==
11
 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 ==
42
 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 ==
71
  == Do Not Walk ==
78
  == Do Not Walk ==
```

☑ Autoscroll □ Show timestamp Newline 115200 baud



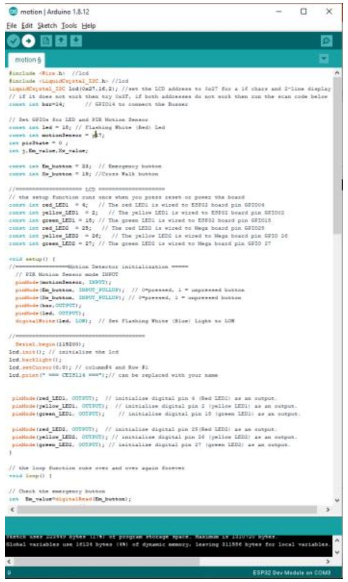
Circuit with 2 controllers, LCD, buzzer, 2 buttons and SMART sensor.

- ESP32 Board
- Two sets of Colored LEDs: Red, Yellow and Green
- One Blue LED
- Push Buttons
- LCD Unit

- Buzzer
- Motion Detector
- Wires
- Breadboard

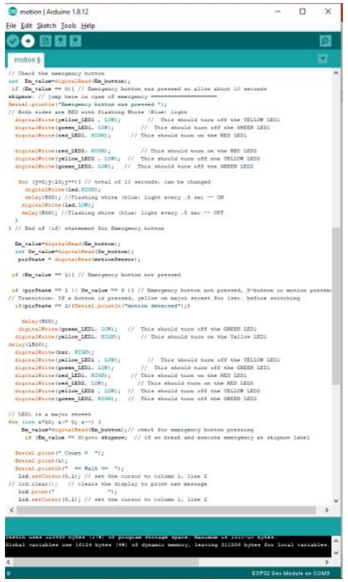
Arduino IDE

- code in Arduino IDE with motion sensor and 2 buttons



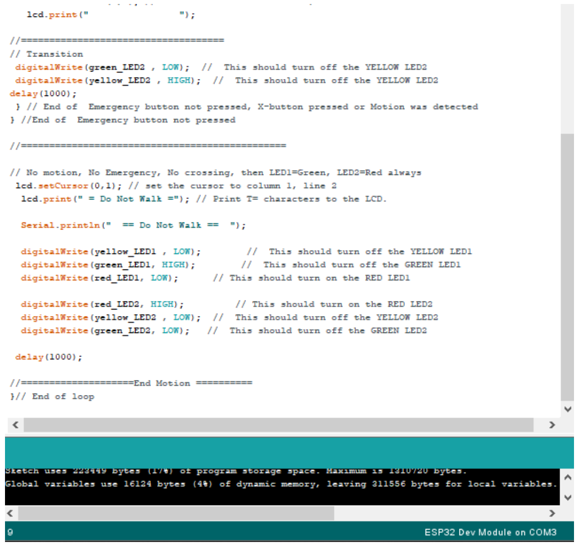
```
#include <Wire.h>  //lcd
#include <LiquidCrystal_I2C.h> //lcd
LiquidCrystal_I2C lcd(0x27,16,2); //set the LCD address to 0x27 for a 16 chars and 2-line display
// if it does not work then try 0x3F, if both addresses do not work then run the scan code below
const int bzr=14;      // GPIO14 to connect the Buzzer

// Set GPIOs for LED and PIR Motion Sensor
const int led = 18; // Flashing White (Red) Led
const int motionSensor = 27;
int pirState = 0 ;
int j,Em_value,Xw_value;

const int Em_button = 23;  // Emergency button
const int Xw_button = 19; //Cross Walk button

//==================== LCD ====================
// the setup function runs once when you press reset or power the board
const int red_LED1  = 4;    // The red LED1 is wired to ESP32 board pin GPIO04
const int yellow_LED1  = 2;    // The yellow LED1 is wired to ESP32 board pin GPIO02
const int green_LED1 = 15; // The green LED1 is wired to ESP32 board pin GPIO15
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

void setup() {
//================Motion Detector initialization =====
  // PIR Motion Sensor mode INPUT
  pinMode(motionSensor, INPUT);
  pinMode(Em_button, INPUT_PULLUP);  // 0=pressed, 1 = unpressed button
  pinMode(Xw_button, INPUT_PULLUP); // 0=pressed, 1 = unpressed button
  pinMode(bzr,OUTPUT);
  pinMode(led, OUTPUT);
  digitalWrite(led, LOW);  // Set Flashing White (Blue) Light to LOW

//======================================
  Serial.begin(115200);
lcd.init(); // initialize the lcd
lcd.backlight();
lcd.setCursor(0,0); // column#4 and Row #1
lcd.print(" === CEIS114 ===");// can be replaced with your name


 pinMode(red_LED1, OUTPUT);  // initialize digital pin 4 (Red LED1) as an output.
 pinMode(yellow_LED1, OUTPUT);  // initialize digital pin 2 (yellow LED1) as an output.
 pinMode(green_LED1, OUTPUT);    // initialize digital pin 15 (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() {

// Check the emergency button
int  Em_value=digitalRead(Em_button);
```

```
Sketch uses 223449 bytes (17%) of program storage space. Maximum is 1310720 bytes.
Global variables use 16124 bytes (4%) of dynamic memory, leaving 311556 bytes for local variables.
```



```
// Check the emergency button
int  Em_value=digitalRead(Em_button);
 if (Em_value == 0){ // Emergency button was pressed so allow about 10 seconds
skipnow: // jump here in case of emergency ====================
Serial.println("Emergency button was pressed ");
// Both sides are RED with flashing White (Blue) light
  digitalWrite(yellow_LED1 , LOW);         //  This should turn off the YELLOW LED1
  digitalWrite(green_LED1, LOW);         //  This should turn off the GREEN LED1
  digitalWrite(red_LED1, HIGH);      // This should turn on the RED LED1

  digitalWrite(red_LED2, HIGH);          // This should turn on the RED LED2
  digitalWrite(yellow_LED2 , LOW);  //  This should turn off the YELLOW LED2
  digitalWrite(green_LED2, LOW);   //  This should turn off the GREEN LED2

   for (j=0;j<10;j++){ // total of 10 seconds, can be changed
     digitalWrite(led,HIGH);
     delay(500); //Flashing white (blue) light every .5 sec -- ON
    digitalWrite(led,LOW);
    delay(500); //Flashing white (blue) light every .5 sec -- OFF
   }
} // End of (if) statement for Emergency button

  Em_value=digitalRead(Em_button);
  int Xw_value=digitalRead(Xw_button);
   pirState = digitalRead(motionSensor);

 if (Em_value == 1){ // Emergency button not pressed

 if (pirState == 1 || Xw_value == 0 ){ // Emergency button not pressed, X-button or motion presse
// Transition: If a button is pressed, yellow on major street for 1sec. before switching
  if(pirState == 1){Serial.println("motion detected");}

    delay(500);
   digitalWrite(green_LED1, LOW);   //  This should turn off the GREEN LED1
  digitalWrite(yellow_LED1, HIGH);      // This should turn on the Yellow LED1
delay(1500);
  digitalWrite(bzr, HIGH);
  digitalWrite(yellow_LED1 , LOW);         //  This should turn off the YELLOW LED1
  digitalWrite(green_LED1, LOW);         //  This should turn off the GREEN LED1
  digitalWrite(red_LED1, HIGH);      // This should turn on the RED LED1
  digitalWrite(red_LED2, LOW);          // This should turn on the RED LED2
  digitalWrite(yellow_LED2 , LOW);  //  This should turn off the YELLOW LED2
  digitalWrite(green_LED2, HIGH);   //  This should turn off the GREEN LED2

// LED1 is a major street
for (int i=10; i>= 0; i--) {
    Em_value=digitalRead(Em_button);// check for emergency button pressing
     if (Em_value == 0)goto skipnow;  // if so break and execute emergency at skipnow label

  Serial.print(" Count =  ");
  Serial.print(i);
  Serial.println("  == Walk ==  ");
   lcd.setCursor(0,1); // set the cursor to column 1, line 2
// lcd.clear();   // clears the display to print new message
   lcd.print("                ");
   lcd.setCursor(0,1); // set the cursor to column 1, line 2
```

```
Sketch uses 223449 bytes (17%) of program storage space. Maximum is 1310720 bytes.
Global variables use 16124 bytes (4%) of dynamic memory, leaving 311556 bytes for local variables.
```

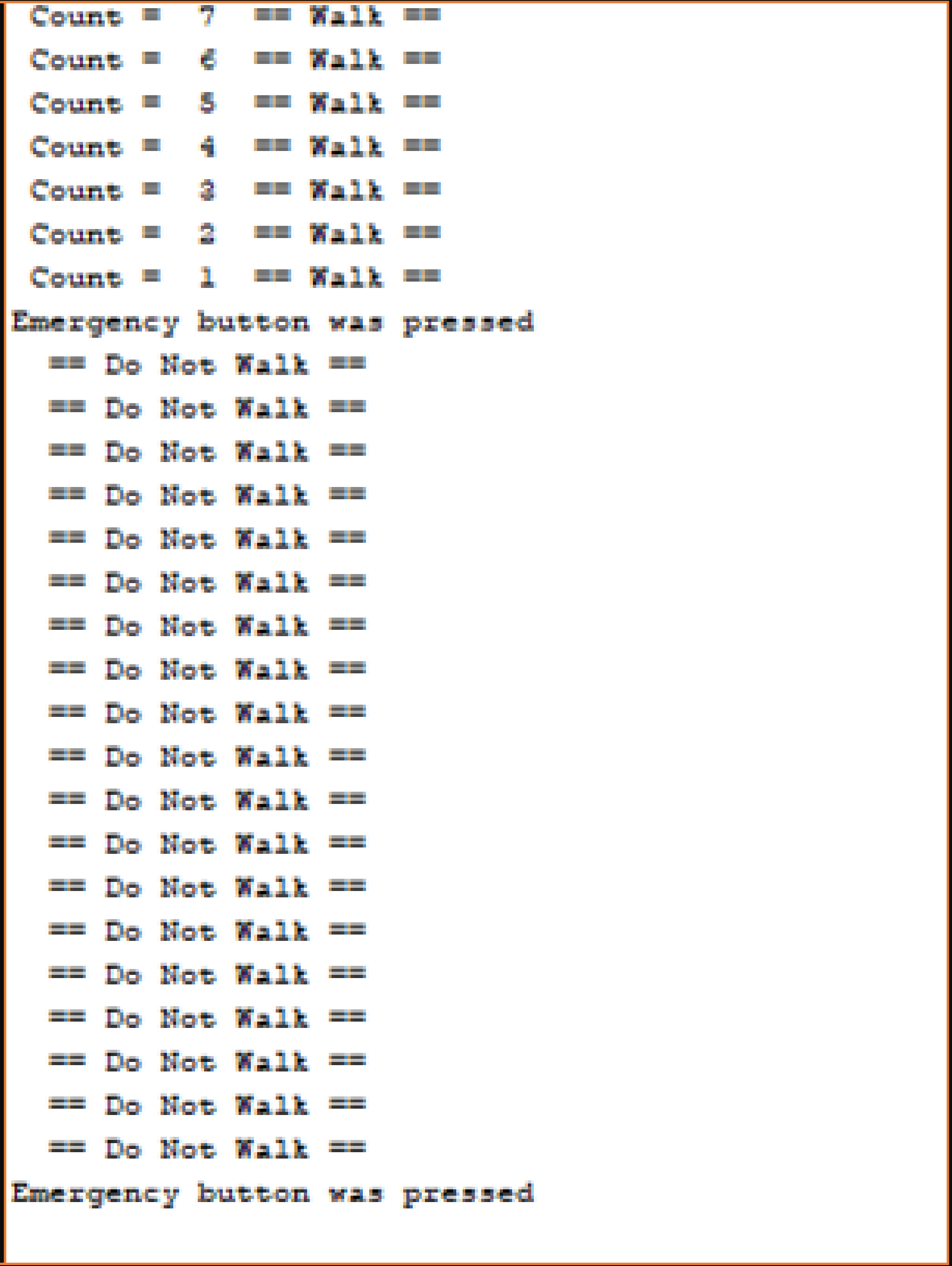


```
   lcd.print("                ");

//==================================
// Transition
 digitalWrite(green_LED2 , LOW);  //  This should turn off the YELLOW LED2
 digitalWrite(yellow_LED2 , HIGH);  //  This should turn off the YELLOW LED2
delay(1000);
 } // End of  Emergency button not pressed, X-button pressed or Motion was detected
} //End of  Emergency button not pressed

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

// No motion, No Emergency, No crossing, then LED1=Green, LED2=Red always
 lcd.setCursor(0,1); // set the cursor to column 1, line 2
  lcd.print(" = Do Not Walk ="); // Print T= characters to the LCD.

  Serial.println("  == Do Not Walk ==  ");

  digitalWrite(yellow_LED1 , LOW);        //  This should turn off the YELLOW LED1
  digitalWrite(green_LED1, HIGH);        //  This should turn off the GREEN LED1
  digitalWrite(red_LED1, LOW);      // This should turn on the RED LED1

  digitalWrite(red_LED2, HIGH);          // This should turn on the RED LED2
  digitalWrite(yellow_LED2 , LOW);  //  This should turn off the YELLOW LED2
  digitalWrite(green_LED2, LOW);   //  This should turn off the GREEN LED2

 delay(1000);

//===================End Motion =========
}// End of loop
```

```
Sketch uses 223449 bytes (17%) of program storage space. Maximum is 1310720 bytes.
Global variables use 16124 bytes (4%) of dynamic memory, leaving 311556 bytes for local variables.
```

Serial Monitor

Output in Serial Monitor



```
 Count =  7  == Walk ==
 Count =  6  == Walk ==
 Count =  5  == Walk ==
 Count =  4  == Walk ==
 Count =  3  == Walk ==
 Count =  2  == Walk ==
 Count =  1  == Walk ==
Emergency button was pressed
  == Do Not Walk ==
  == Do Not Walk ==
  == Do Not Walk ==
  == Do Not Walk ==
  == Do Not Walk ==
  == Do Not Walk ==
  == Do Not Walk ==
  == Do Not Walk ==
  == Do Not Walk ==
  == Do Not Walk ==
  == Do Not Walk ==
  == Do Not Walk ==
  == Do Not Walk ==
  == Do Not Walk ==
  == Do Not Walk ==
  == Do Not Walk ==
  == Do Not Walk ==
  == Do Not Walk ==
  == Do Not Walk ==
Emergency button was pressed
```

Conclusion

- In this project I demonstrated many different systems to observe traffic flow with the motion sensor. An LCD panel was added to show output to pedestrians when it is safe to walk and an audible tone with the addition of the buzzer.