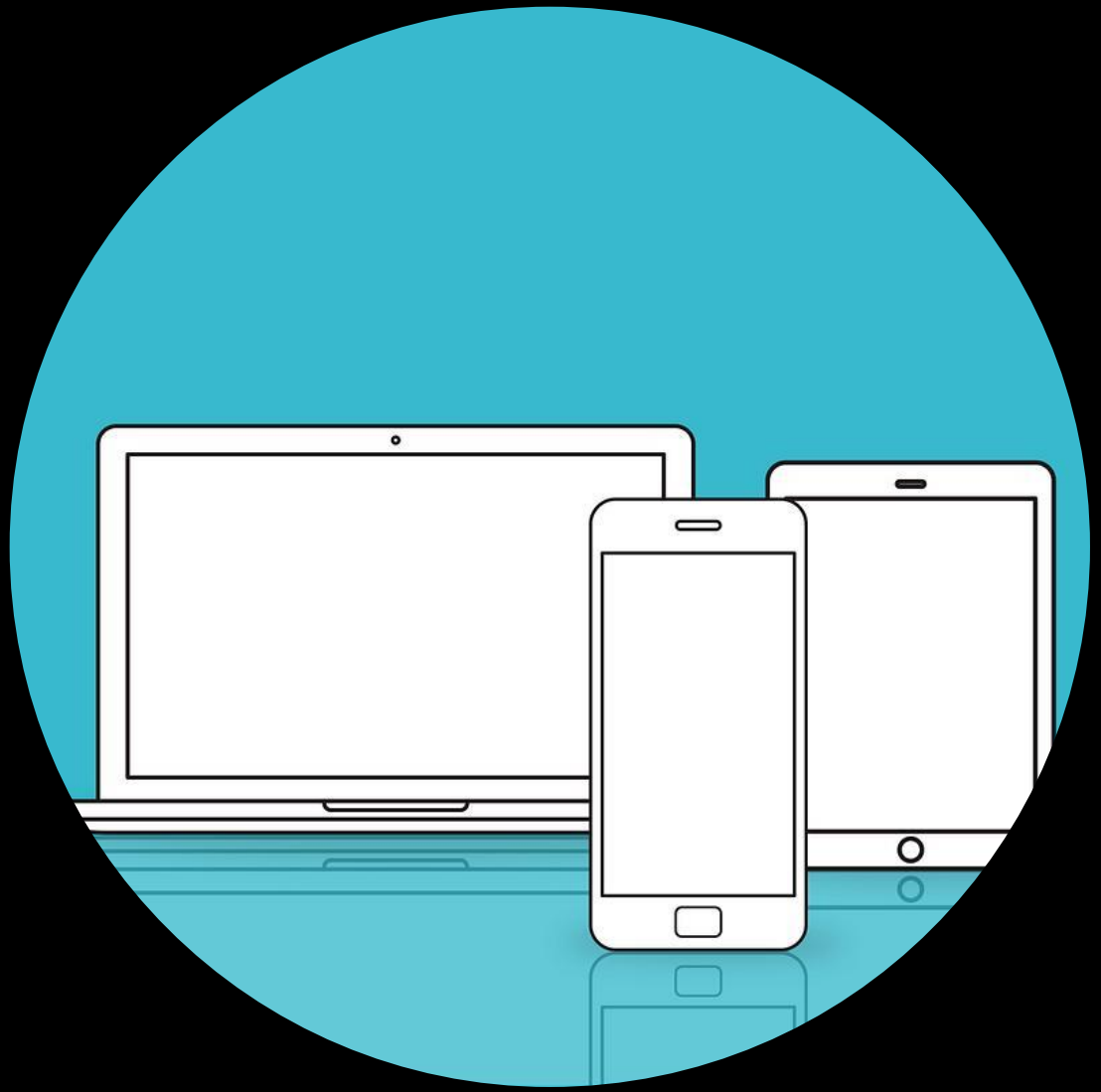


Introduction to Digital Devices (CEIS114)

Final Project

Jonathan Waugh

Professor Sean Caruthers

Oct. 2024

Introduction

This presentation covers my six-part project completed while taking Introduction to Digital Devices (CEIS114).

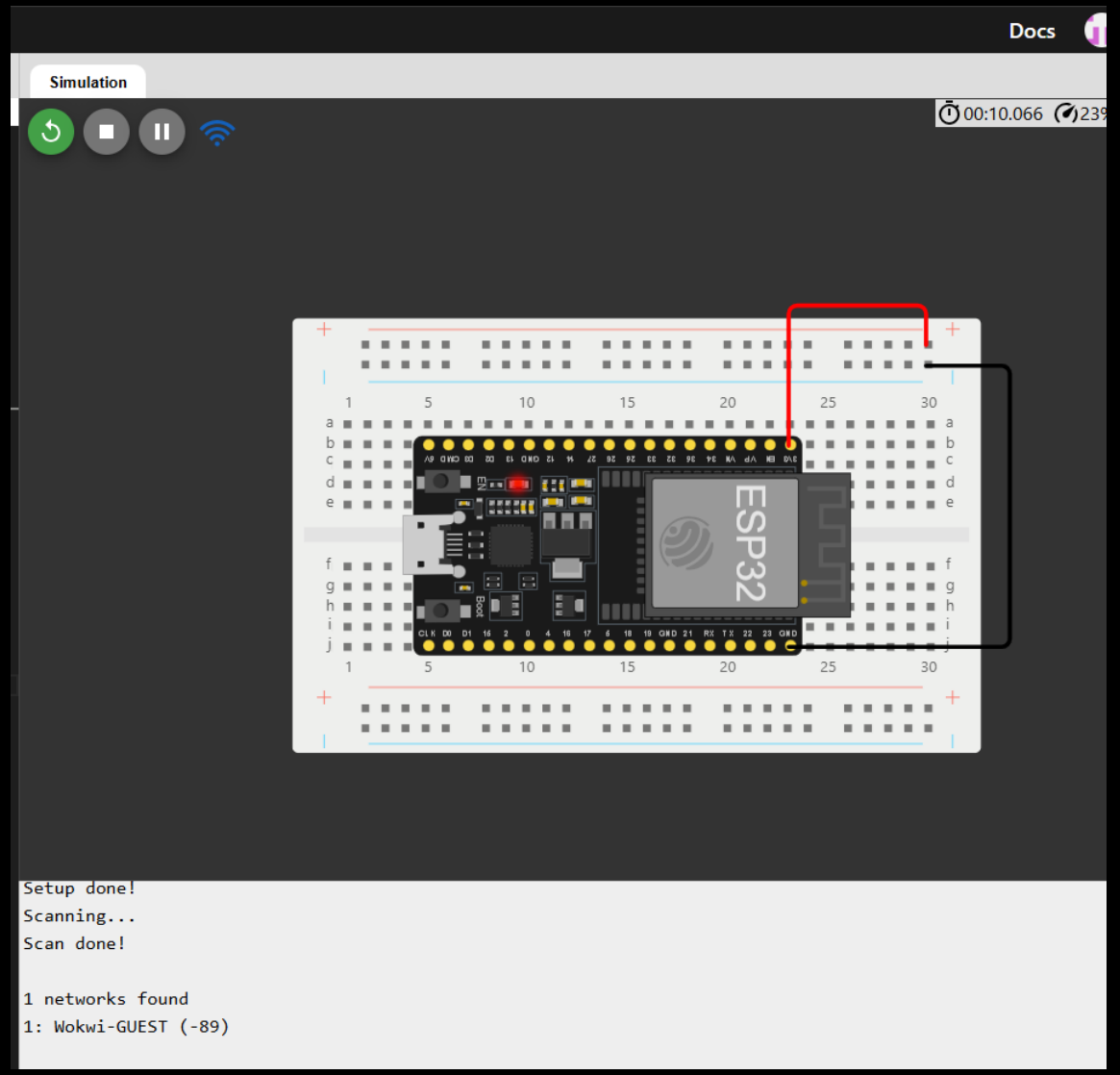
It includes examples of project planning for IoT traffic controller, creation of multi-light traffic controller, adding a crosswalk and emergency buzzer, secured IoT control via web

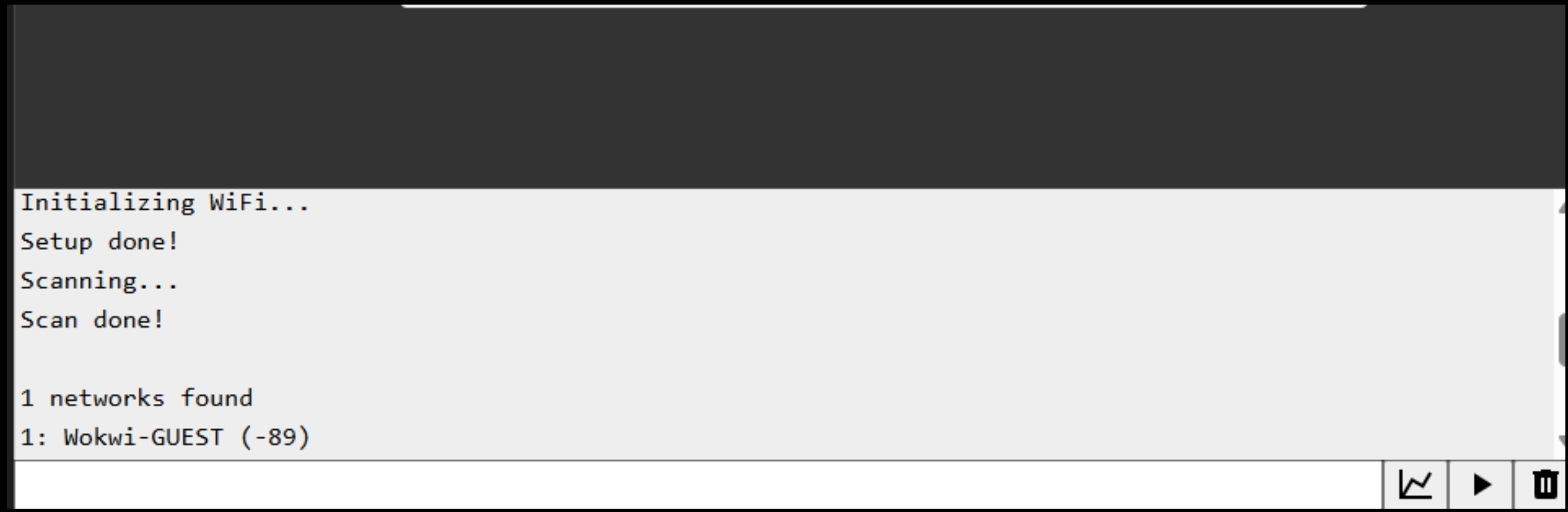
Project Plan for IoT Traffic Controller

- ESP32 installed and powered on
- WIFI scan

ESP32 (Screenshot)

- Microcontroller mounted and powered ON



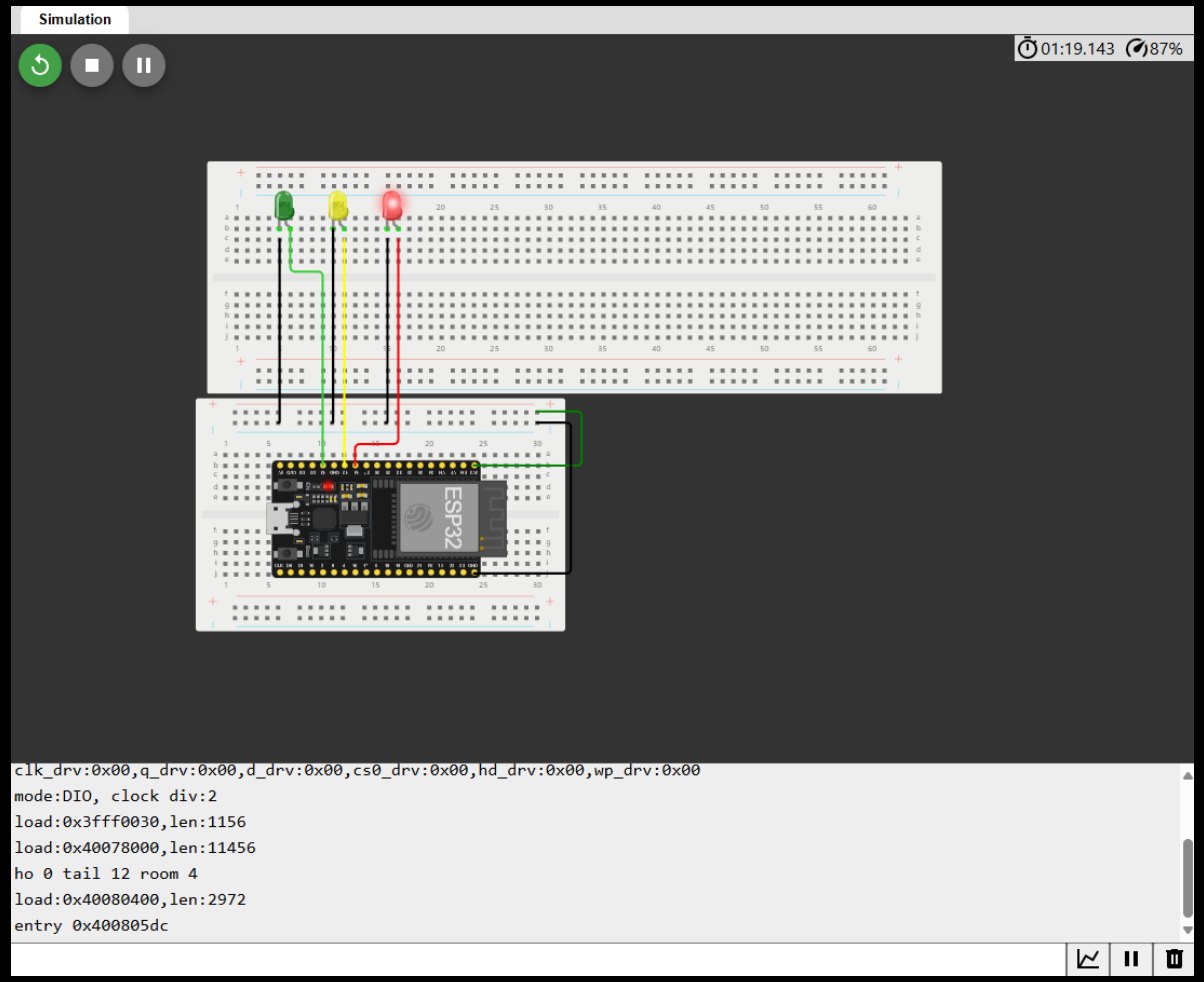


ESP32 WiFi Scan

Creating the Traffic Controller

Picture of circuit with working LEDs

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

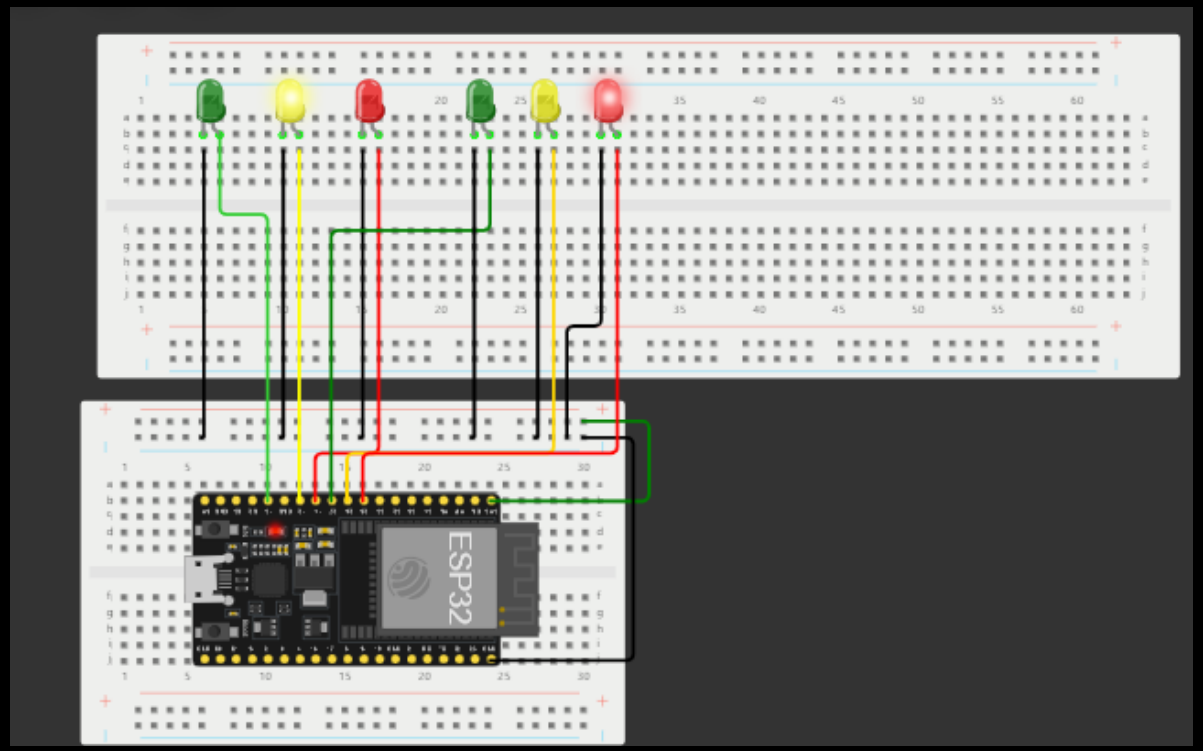


Screenshot of code in the Code Editor

```
// === Jonathan Waugh ====
// 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

// 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
}
```

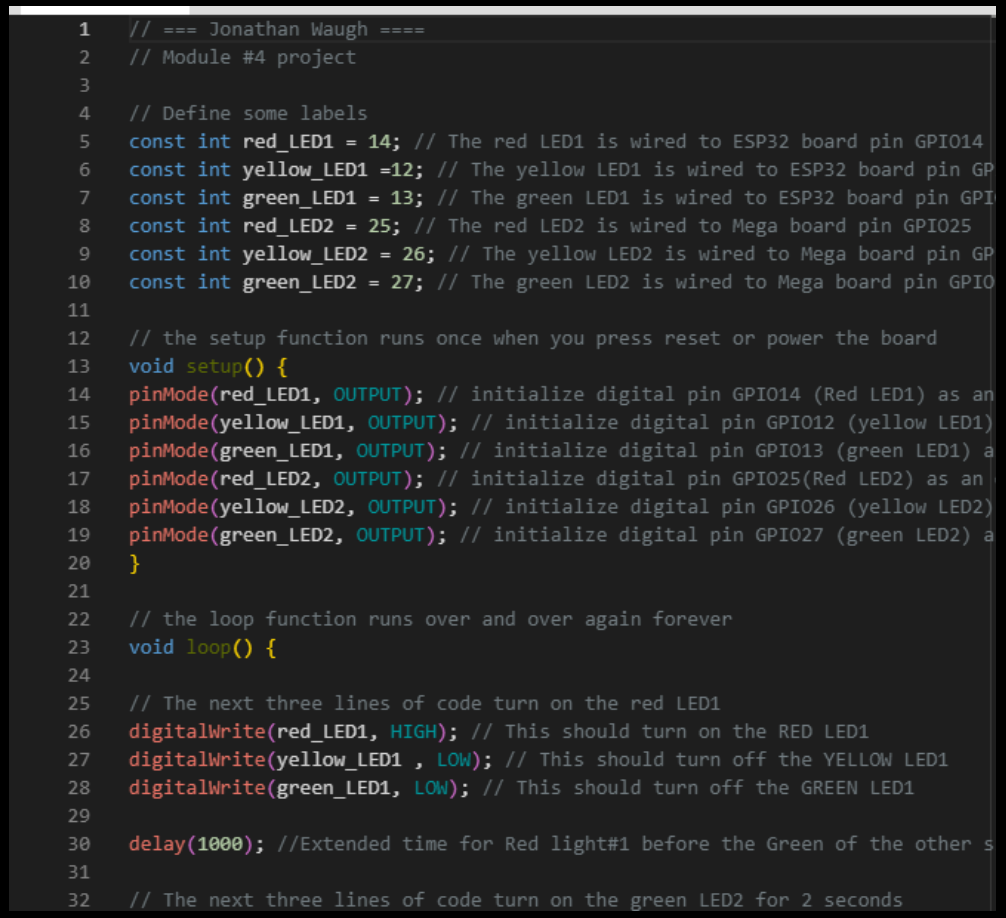
Creating a Multiple Traffic Light Controller

Picture of circuit with working LEDs



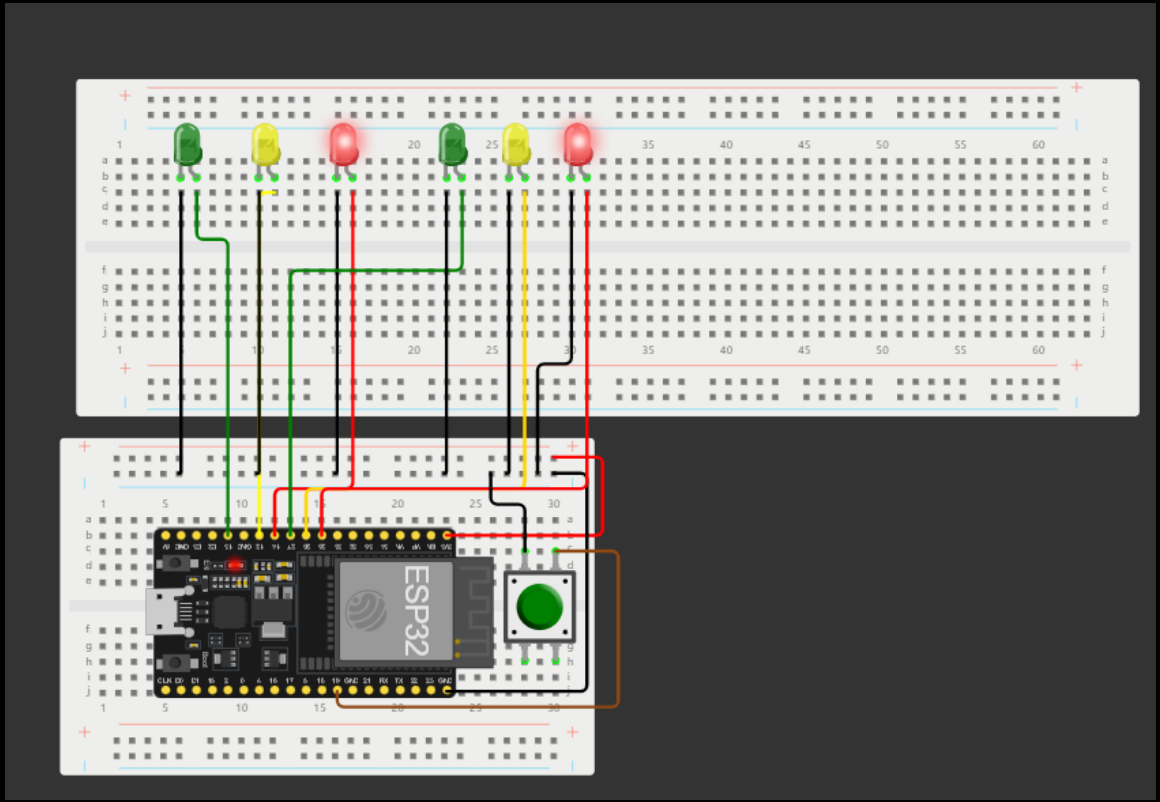
- ESP 32 Board
- Colored LEDs: Red, Yellow and Green (two sets)
- Wires
- Breadboard

Screenshot of code in Wokwi



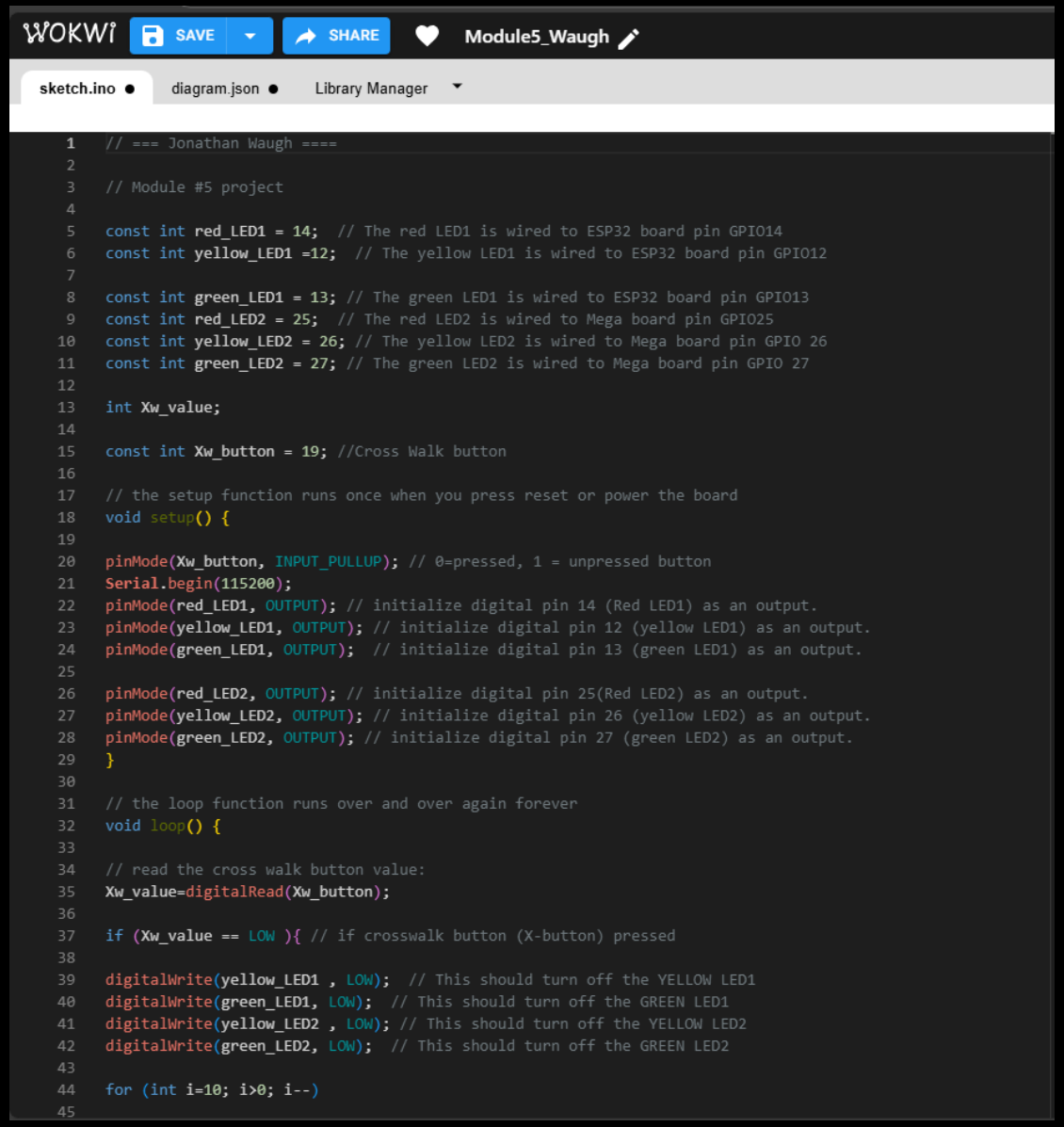
```
// === Jonathan Waugh ====
// Module #4 project

// 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 GP
const int green_LED1 = 13; // The green LED1 is wired to ESP32 board pin GPI
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 GP
const int green_LED2 = 27; // The green LED2 is wired to Mega board pin GPIO

// 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
pinMode(yellow_LED1, OUTPUT); // initialize digital pin GPIO12 (yellow LED1)
pinMode(green_LED1, OUTPUT); // initialize digital pin GPIO13 (green LED1) a
pinMode(red_LED2, OUTPUT); // initialize digital pin GPIO25(Red LED2) as an
pinMode(yellow_LED2, OUTPUT); // initialize digital pin GPIO26 (yellow LED2)
pinMode(green_LED2, OUTPUT); // initialize digital pin GPIO27 (green LED2) a
}

// 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 s

// The next three lines of code turn on the green LED2 for 2 seconds
```

Adding a Cross Walk



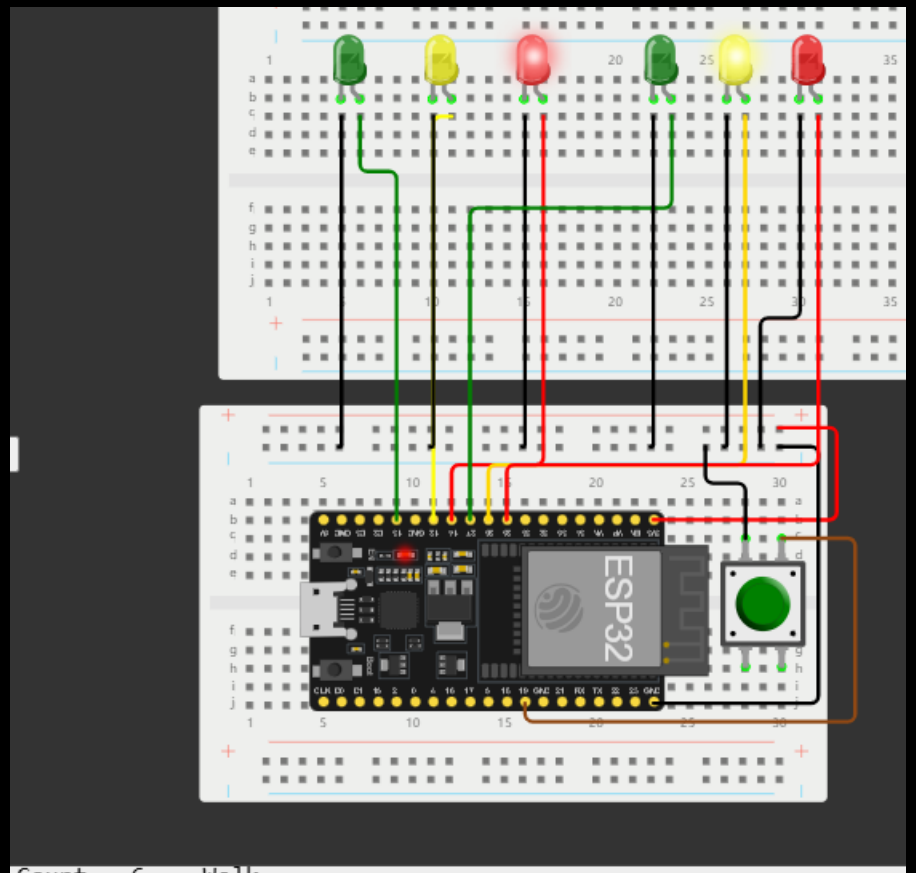
Screenshot of circuit with working LEDs

- ESP 32 Board
- Colored LEDs: Red, Yellow and Green (two sets)
- 220 Ohm Resistors (optional)
- Push Button
- Wires
- Breadboard



Screenshot of code in Wokwi

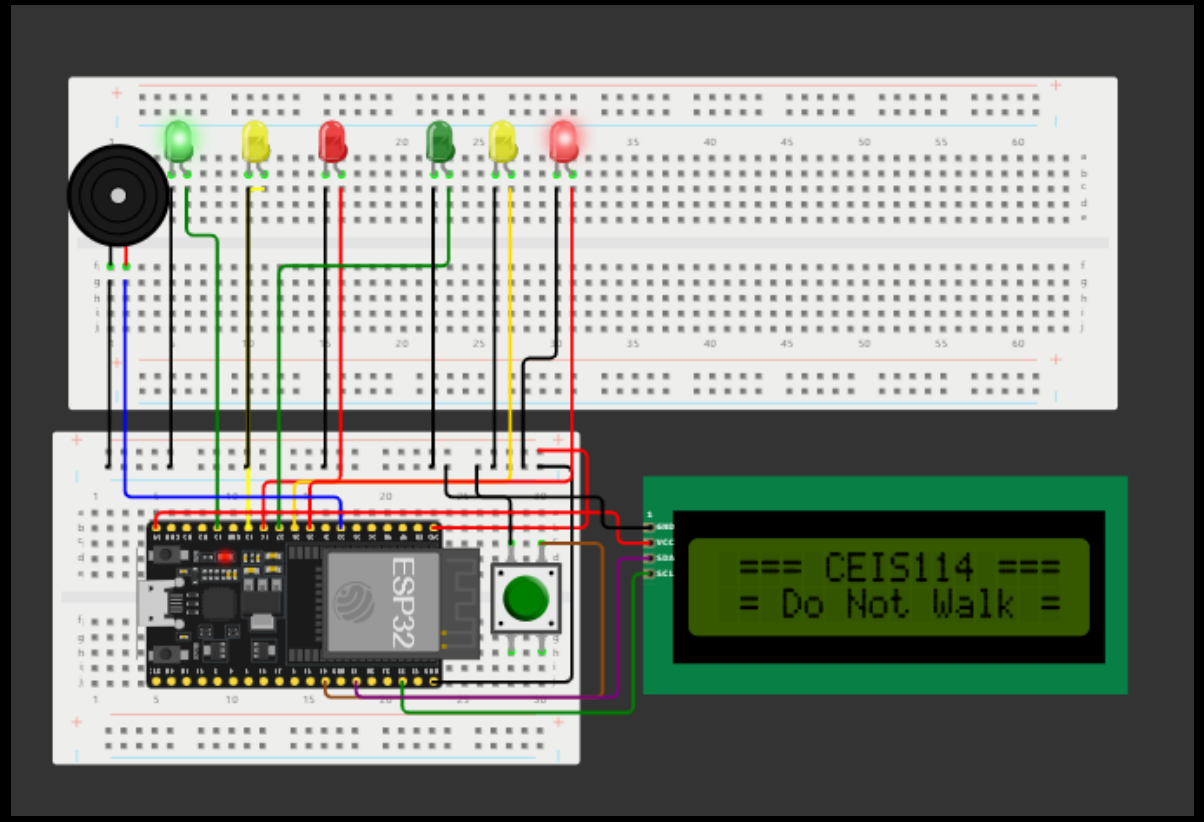
Screenshot of Serial Monitor in Wokwi

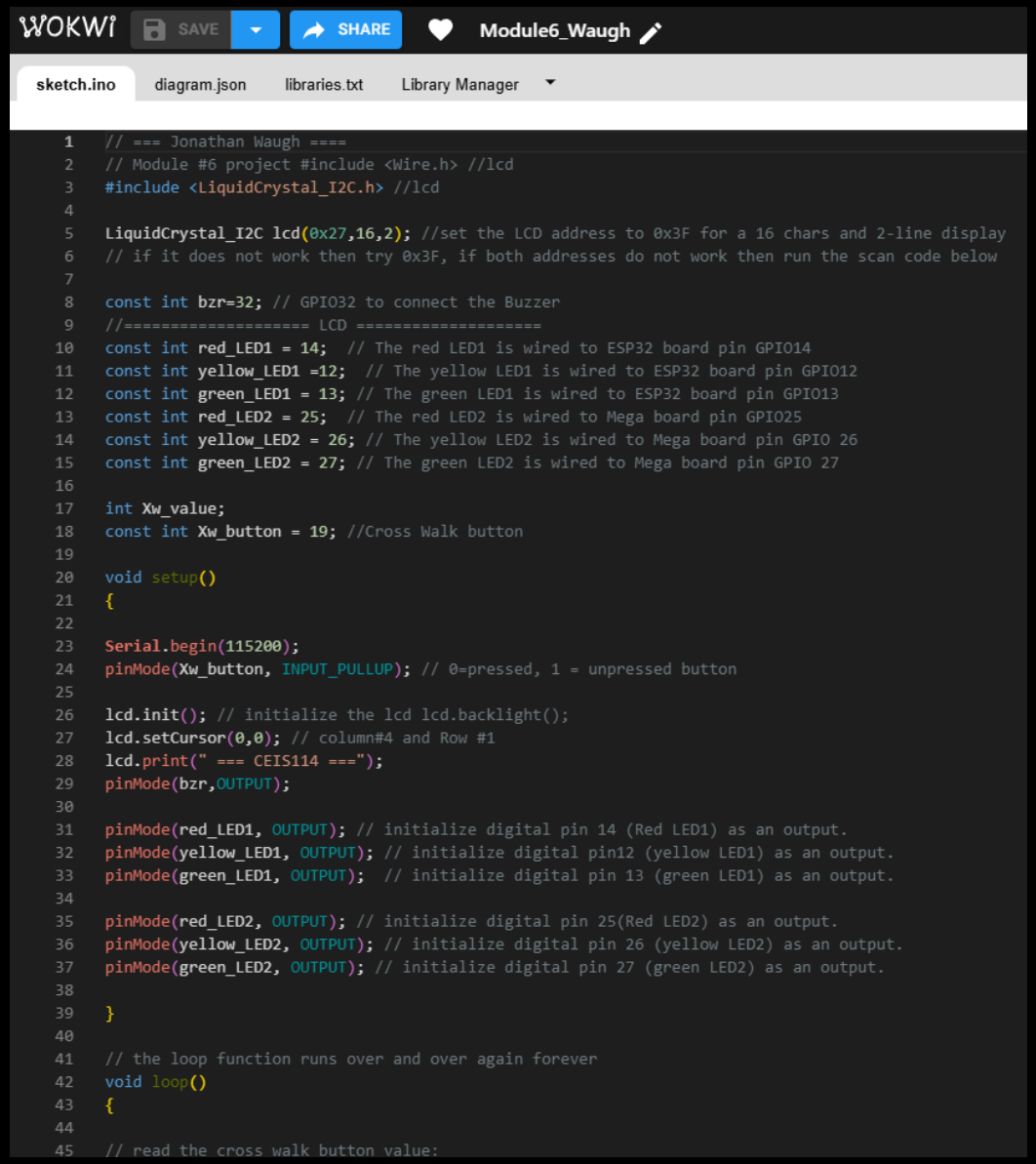



```
Count = 6 == Walk ==
Count = 5 == Walk ==
Count = 4 == Walk ==
Count = 3 == Walk ==
Count = 2 == Walk ==
Count = 1 == Walk ==
== Do Not Walk ==
```

Picture of circuit with working LEDs and LCD display

- ESP 32 Board
- Colored LEDs: Red, Yellow and Green (two sets)
- 220 Ohm Resistors (optional)
- Push Button
- LCD Unit with Message Display
- Wires
- Breadboard

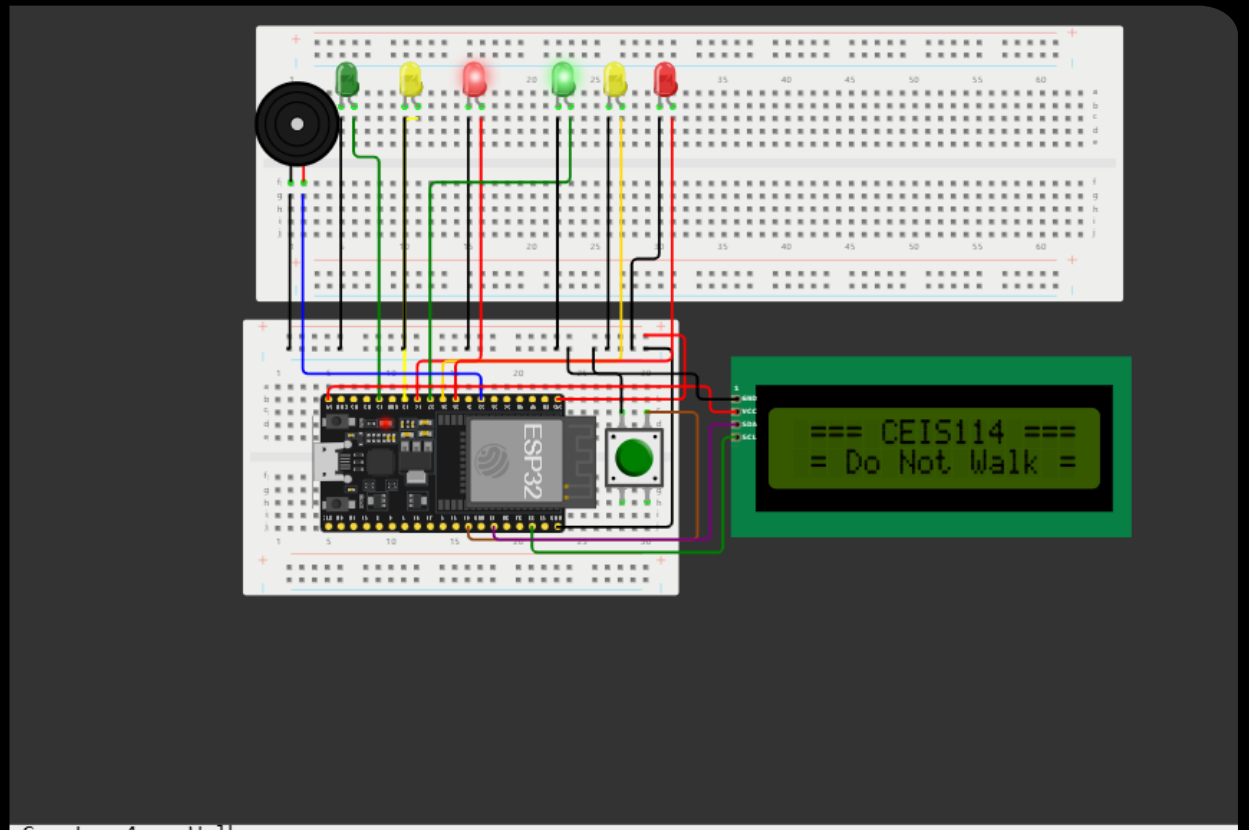




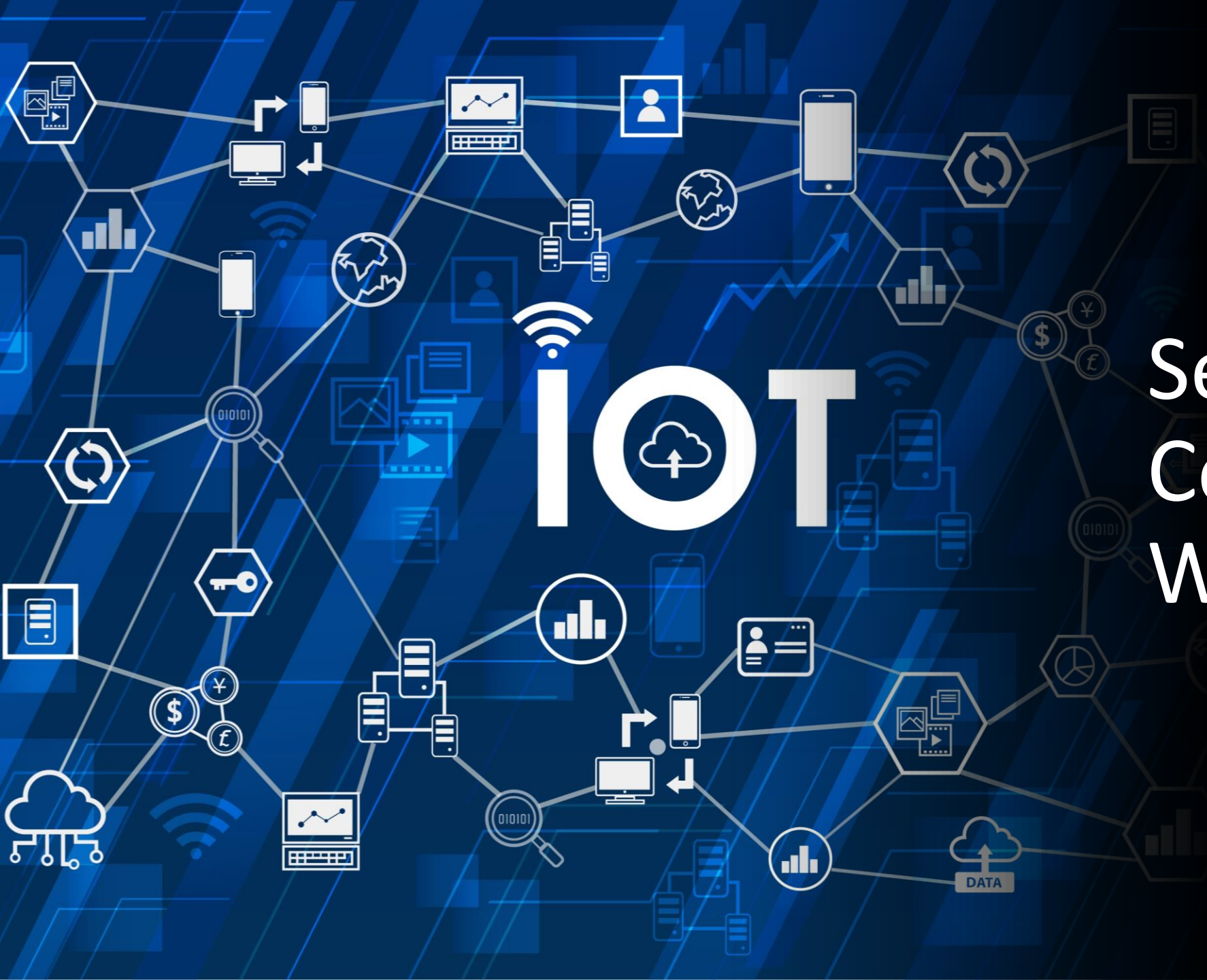

```ino
// === Jonathan Waugh ====
// Module #6 project #include <Wire.h> //lcd
#include <LiquidCrystal_I2C.h> //lcd

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 ===================
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

void setup()
{

Serial.begin(115200);
pinMode(Xw_button, INPUT_PULLUP); // 0=pressed, 1 = unpressed button

lcd.init(); // initialize the lcd lcd.backlight();
lcd.setCursor(0,0); // column#4 and Row #1
lcd.print(" === CEIS114 ===");
pinMode(bzr,OUTPUT);

pinMode(red_LED1, OUTPUT); // initialize digital pin 14 (Red LED1) as an output.
pinMode(yellow_LED1, OUTPUT); // initialize digital pin12 (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()
{

// read the cross walk button value:
```

Screenshot of code in Code Editor

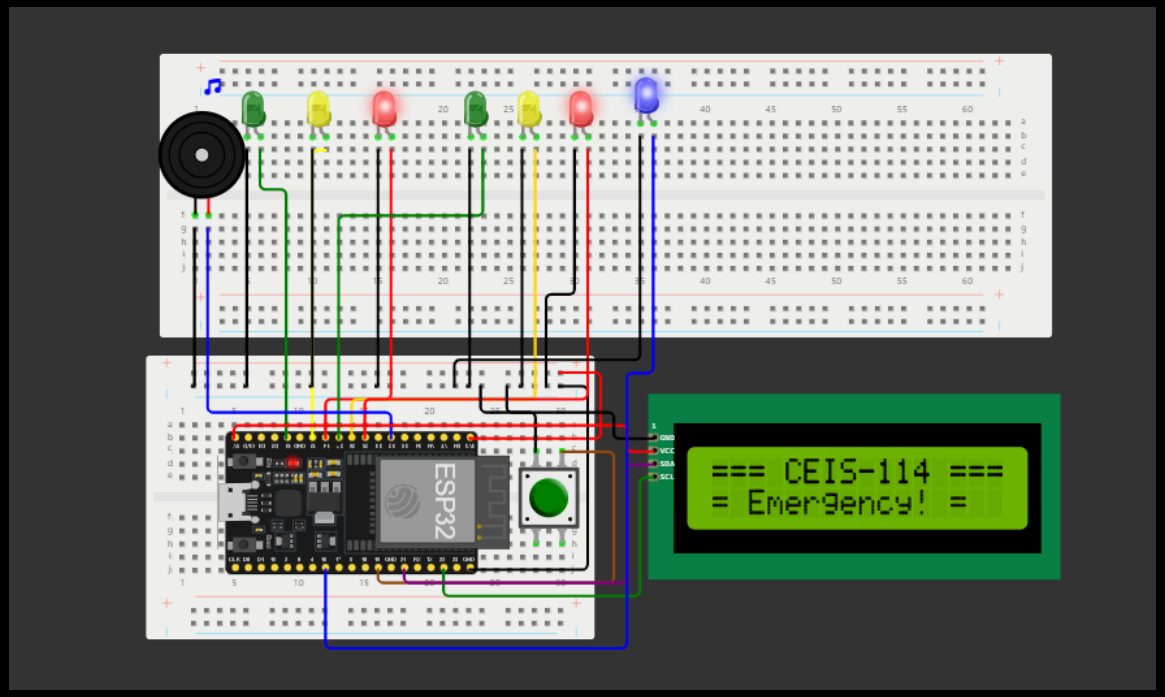
Screenshot of Serial Monitor




```
Count = 4 == Walk ==
Count = 3 == Walk ==
Count = 2 == Walk ==
Count = 1 == Walk ==
Count = 0 == Walk ==
== Do Not Walk ==
```



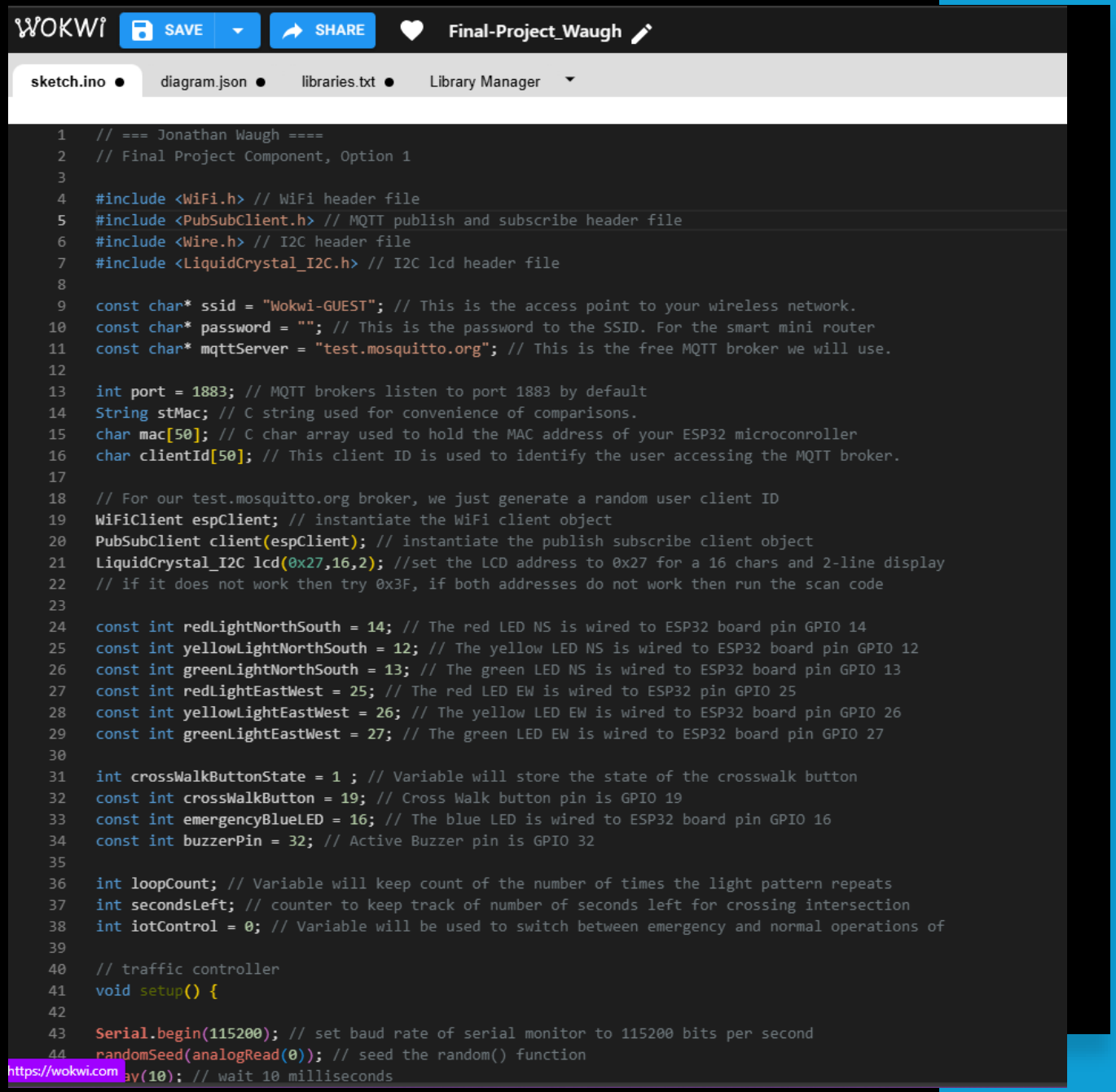
Secured IoT Control via WEB



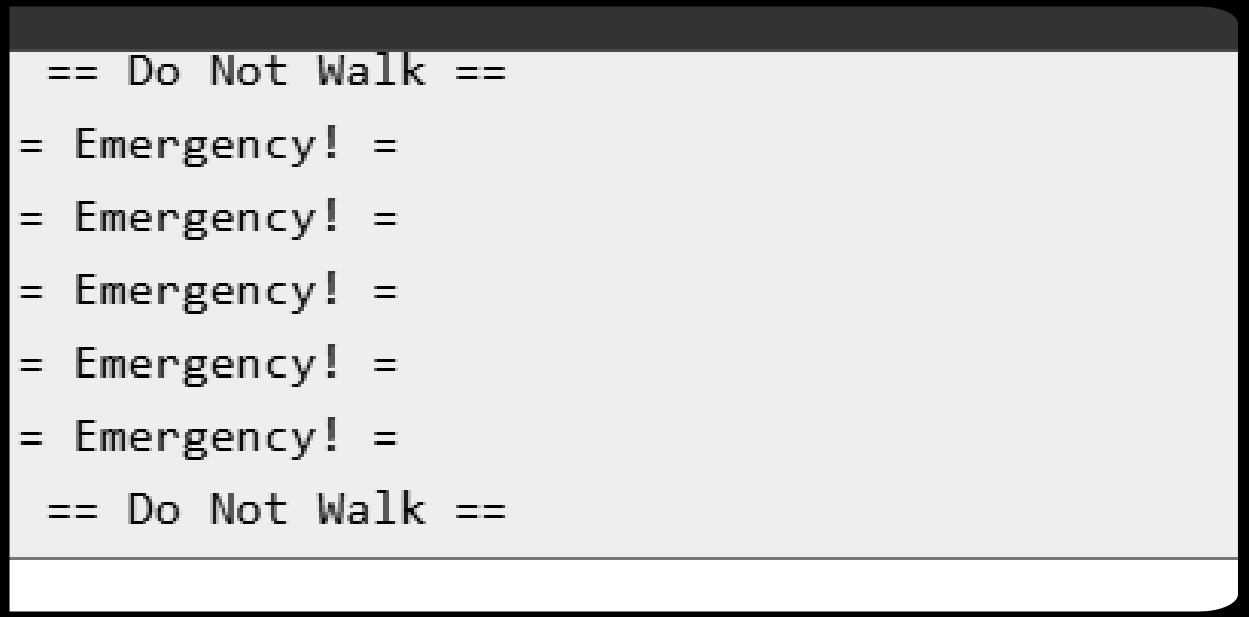
Screenshot of circuit **with working LEDs and LCD display** (Building/Operation)

- ESP 32 Board
- Colored LEDs: Red, Yellow and Green (two sets)
- One Blue LED – Emergency Light
- Push Button
- LCD Unit
- Buzzer
- Wires
- Breadboard

Screenshot of **code in Code Editor** (Testing)



Screenshot **of Serial Monitor** (Testing)



Challenges

- Challenges I faced while doing these projects were learning the code when not writing it myself
- I overcame this challenge by reading all the editor notes and making sure I understood each line and what its function is

Career Skills

- Diagram design
- Embedded systems knowledge
- Project Management
- Electronics and circuit design
- Networking and communications
- Programming
- Attention to Detail

Conclusion

This class provided a great teaching on the introduction to digital devices. With these teachings I now can design, code, troubleshoot, and manage different designs that include linking a secured IoT controller via web. Learning these skills will allow me to better understand device control, design, and troubleshooting and hone my skills to advance me in my career path. Learning about digital devices will allow me to design and bring to life products that could help people in their day to day lives.

References

- Live Lessons. Professor Muqri, Mohammad and Prof. Jellouli, Prof. Majumder, Prof. McKenzie, Prof. Mortezaie, Prof. Duclos, Prof. Ng, Prof. Qian, Prof. Sommer, Prof. St. John, Dr. Tilghman, Prof. Wang. CEIS114, 2024
- Project Guides

Wix Portfolio

Portfolio | Jonathan Waugh | IT services and logo/website design