



# CEIS 114 FINAL COURSE PROJECT

GABRIEL SANDOVAL

FEB 25, 2023

# INTRODUCTION

Arduino is an open-source software that is free; downloading it to my computer allowed me to write and execute codes. The project consisted of 6 steps leading up to simulating a traffic light system with an emergency signal. All components used in this project are from the IOT CORE KIT.



# MODULE 2

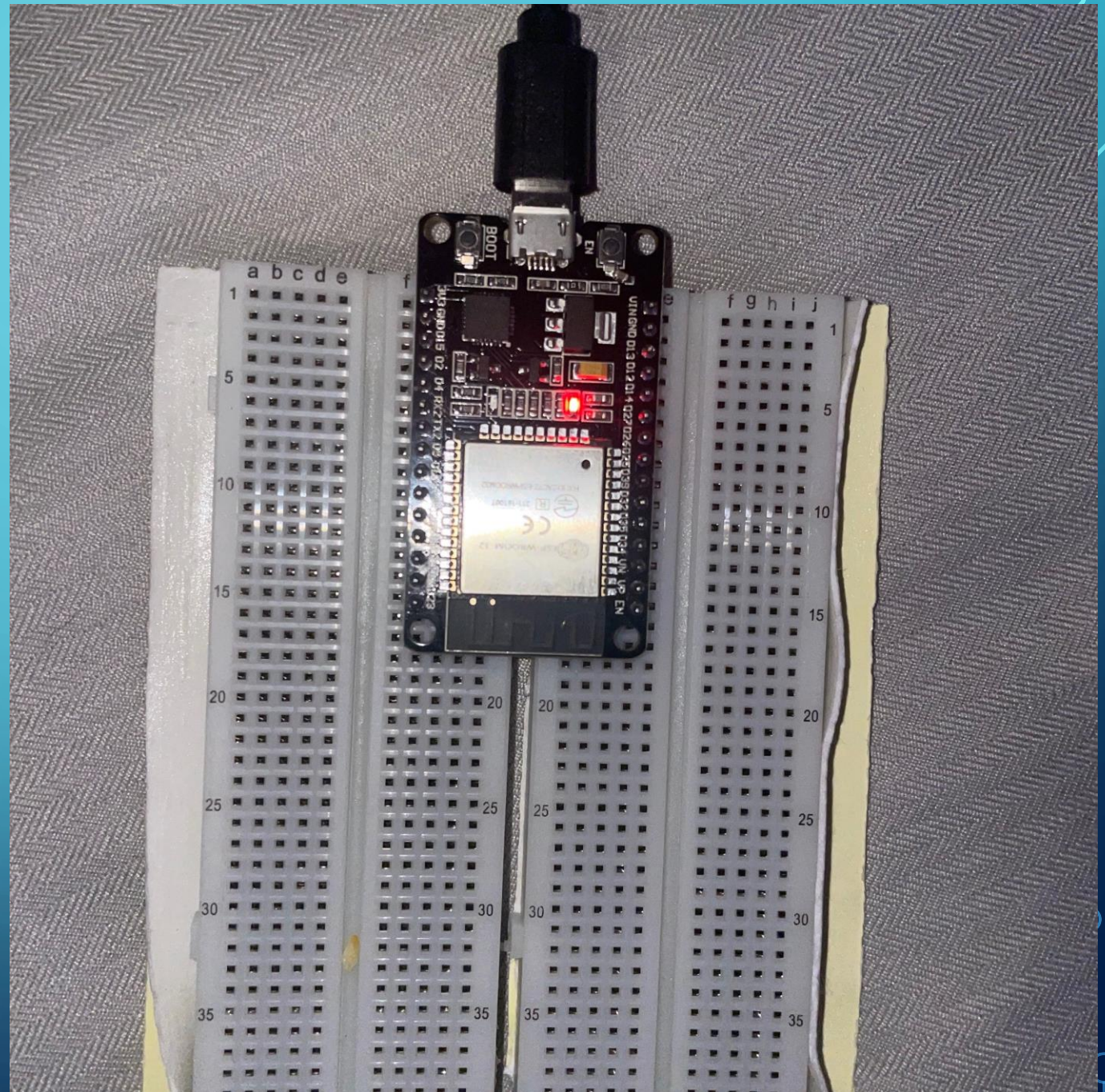
- DOWNLOADING ARDUINO WAS EASY HOWEVER I RAN INTO A PROBLEM WITH MY COMPUTER WHERE IT WOULD NOT ALLOW ME TO CONNECT TO THE ESP32.
- DOWNLOADING A COM3 DRIVER WAS NEEDED.
- DOWNLOADED DOIT\_DEVKIT
- GATHERING INVENTORY





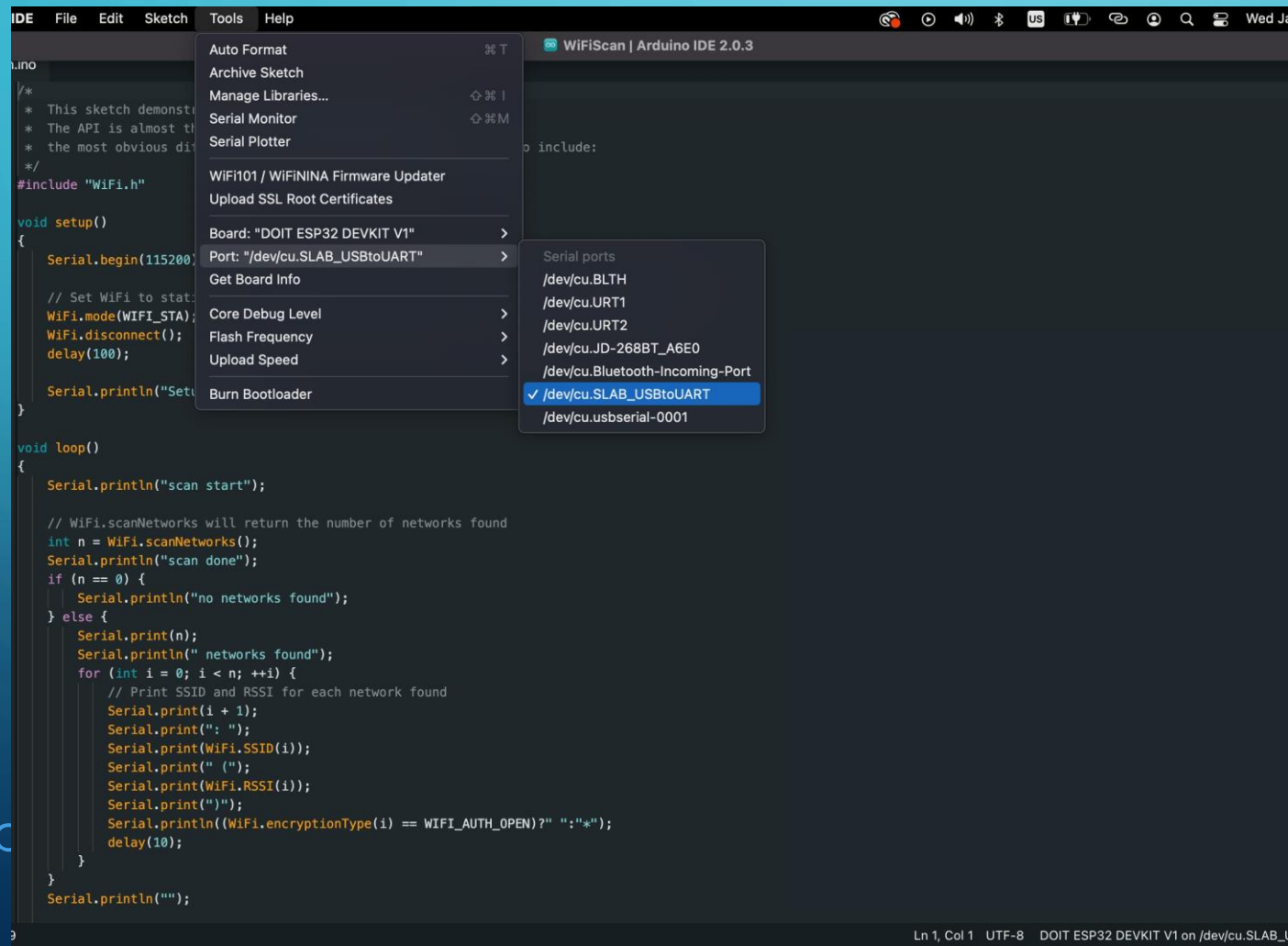
# ESP32 (PICTURE)

MICROCONTROLLER MOUNTED AND POWERED ON



# INSTALLATION OF ARDUINO IDE

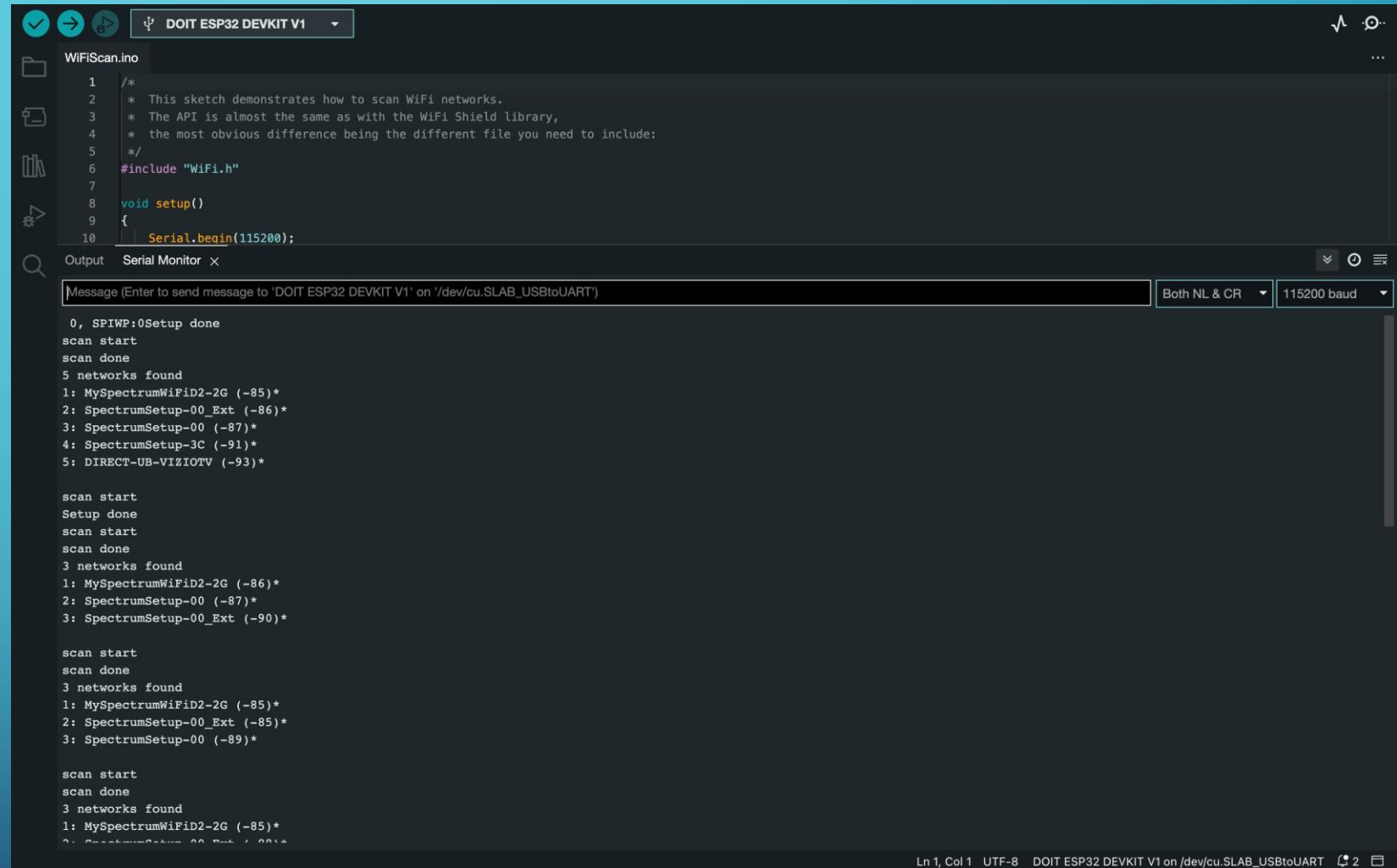
SCREENSHOT OF ARDUINO IDE WITH **PORT** SELECTED FROM TOOLS MENU.





# ESP32 WiFi Scan

Screenshot of **Serial Monitor** in Arduino IDE showing the available networks



The screenshot displays the Arduino IDE interface. The top toolbar shows the 'Serial Monitor' icon. The 'Serial Monitor' window is open, showing the output of the 'WiFiScan.ino' sketch. The sketch code is visible in the background, showing the setup and scan functions. The Serial Monitor output shows the results of the WiFi scan, including the number of networks found and their details (SSID, BSSID, and RSSI).

```
WiFiScan.ino
1  /*
2   * This sketch demonstrates how to scan WiFi networks.
3   * The API is almost the same as with the WiFi Shield library,
4   * the most obvious difference being the different file you need to include:
5   */
6  #include "WiFi.h"
7
8  void setup()
9  {
10     Serial.begin(115200);
```

Output Serial Monitor x

Message (Enter to send message to 'DOIT ESP32 DEVKIT V1' on '/dev/cu.SLAB\_USBtoUART') Both NL & CR 115200 baud

```
0, SPIWP:0Setup done
scan start
scan done
5 networks found
1: MySpectrumWiFD2-2G (-85)*
2: SpectrumSetup-00_Ext (-86)*
3: SpectrumSetup-00 (-87)*
4: SpectrumSetup-3C (-91)*
5: DIRECT-UB-VIZIOTV (-93)*

scan start
Setup done
scan start
scan done
3 networks found
1: MySpectrumWiFD2-2G (-86)*
2: SpectrumSetup-00 (-87)*
3: SpectrumSetup-00_Ext (-90)*

scan start
scan done
3 networks found
1: MySpectrumWiFD2-2G (-85)*
2: SpectrumSetup-00_Ext (-85)*
3: SpectrumSetup-00 (-89)*

scan start
scan done
3 networks found
1: MySpectrumWiFD2-2G (-85)*
2: SpectrumSetup-00_Ext (-85)*
```

Ln 1, Col 1 UTF-8 DOIT ESP32 DEVKIT V1 on /dev/cu.SLAB\_USBtoUART 2



# MODULE 3

- WIRING LED'S
- WIRING 2 BREADBOARDS TO FUNCTION TOGETHER



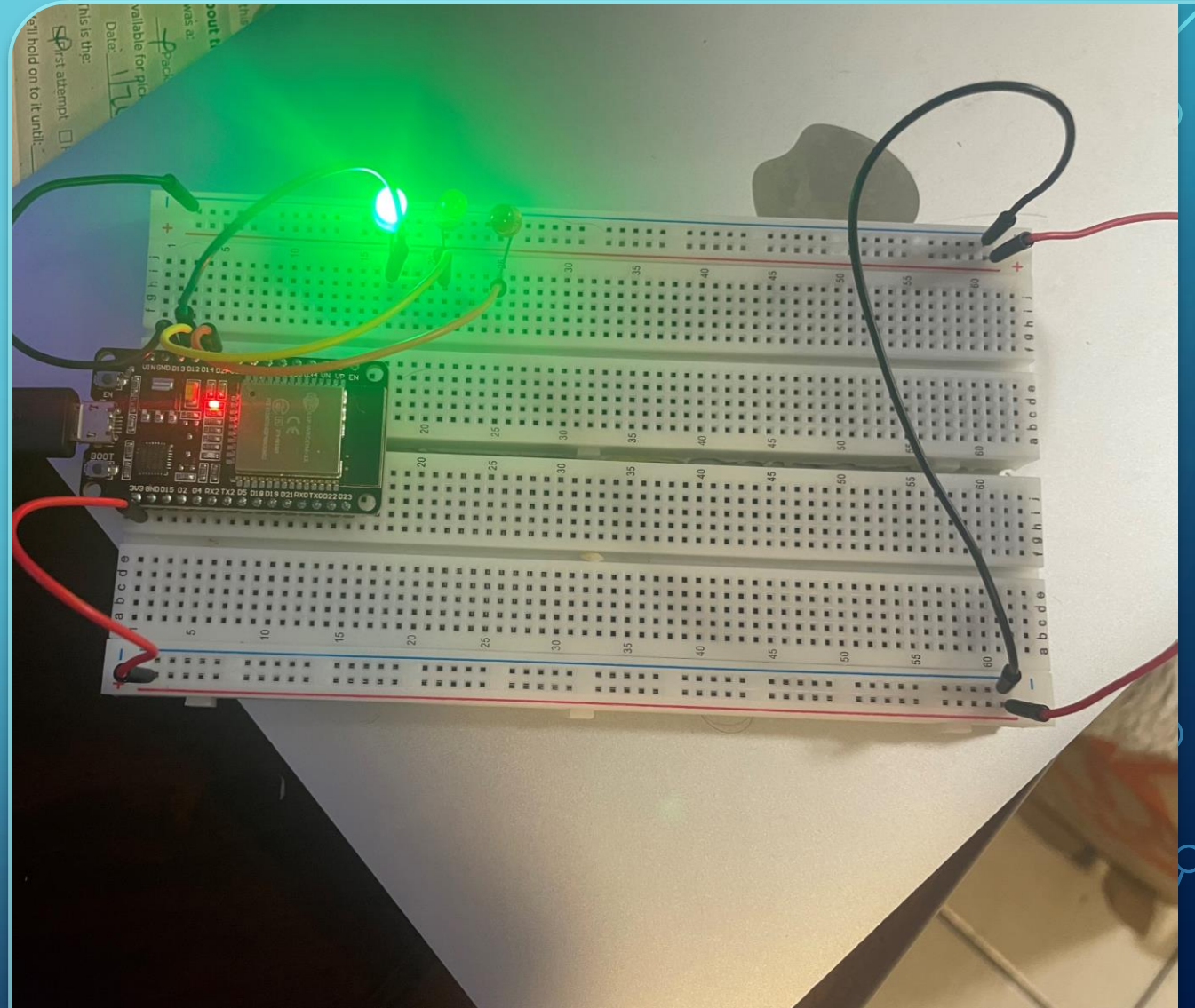
# PICTURE OF CIRCUIT WITH WORKING LEDs

ESP 32 Board

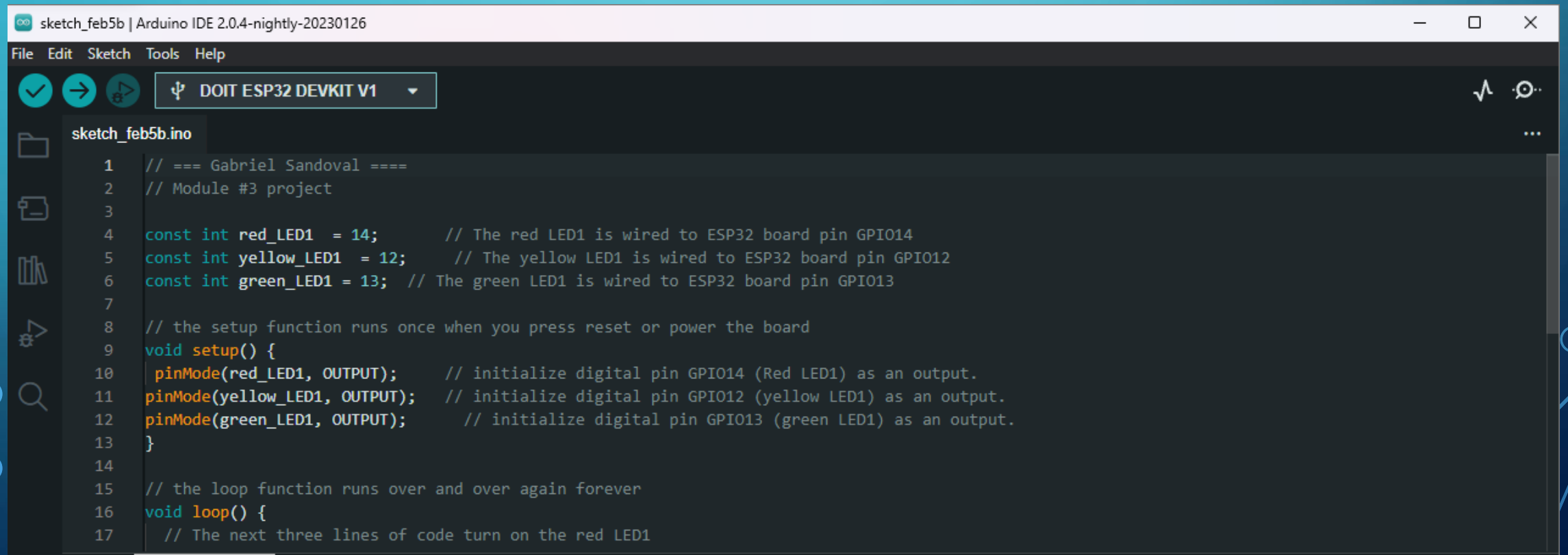
Colored LEDs: Red, Yellow and Green

Wires

Breadboard



# SCREENSHOT OF CODE IN ARDUINO IDE



```
sketch_feb5b | Arduino IDE 2.0.4-nightly-20230126
File Edit Sketch Tools Help
[Checkmark] [Next] [Previous] [USB] DOIT ESP32 DEVKIT V1 [Dropdown] [Waveform] [Microscope]
sketch_feb5b.ino
1 // === Gabriel Sandoval ===
2 // Module #3 project
3
4 const int red_LED1 = 14; // The red LED1 is wired to ESP32 board pin GPIO14
5 const int yellow_LED1 = 12; // The yellow LED1 is wired to ESP32 board pin GPIO12
6 const int green_LED1 = 13; // The green LED1 is wired to ESP32 board pin GPIO13
7
8 // the setup function runs once when you press reset or power the board
9 void setup() {
10   pinMode(red_LED1, OUTPUT); // initialize digital pin GPIO14 (Red LED1) as an output.
11   pinMode(yellow_LED1, OUTPUT); // initialize digital pin GPIO12 (yellow LED1) as an output.
12   pinMode(green_LED1, OUTPUT); // initialize digital pin GPIO13 (green LED1) as an output.
13 }
14
15 // the loop function runs over and over again forever
16 void loop() {
17   // The next three lines of code turn on the red LED1
```



## MODULE 4

- TESTING AND RUNNING CODE
- INSERTING 3 MORE LEDS
- WIRING NEW LED SET



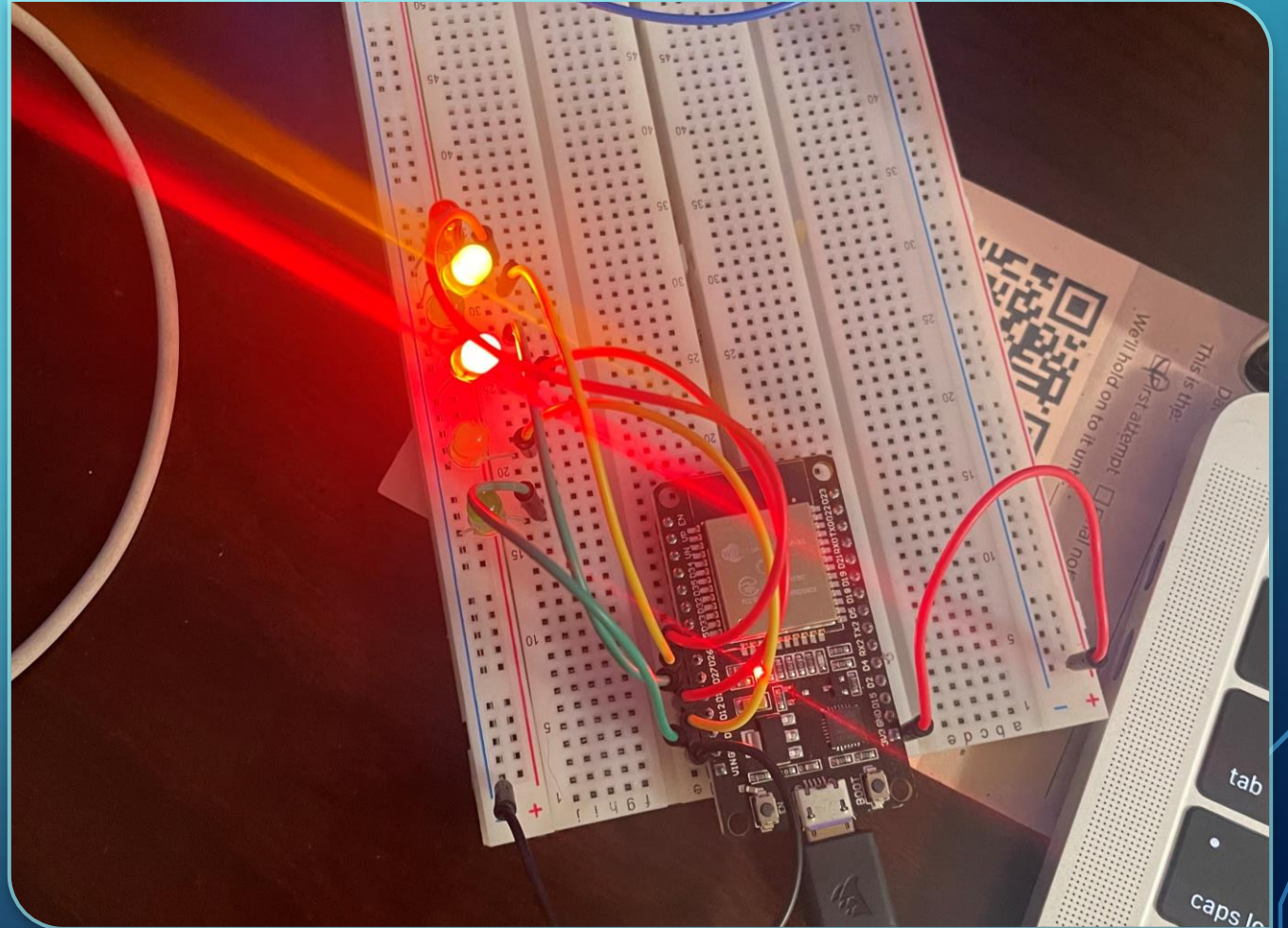
# PICTURE OF CIRCUIT WITH WORKING LEDs

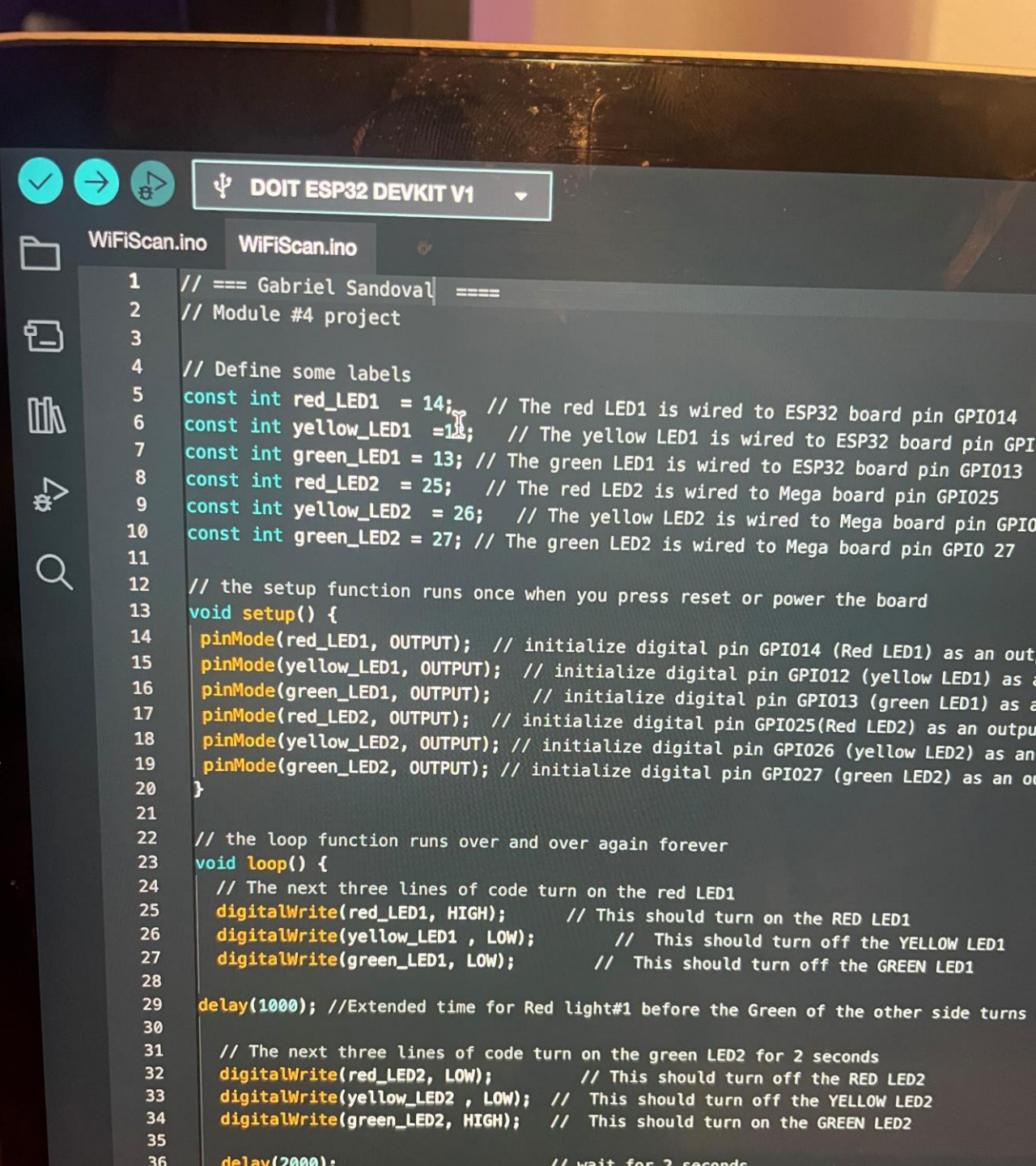
ESP 32 Board

Colored LEDs: Red, Yellow and Green (two sets)

Wires

Breadboard



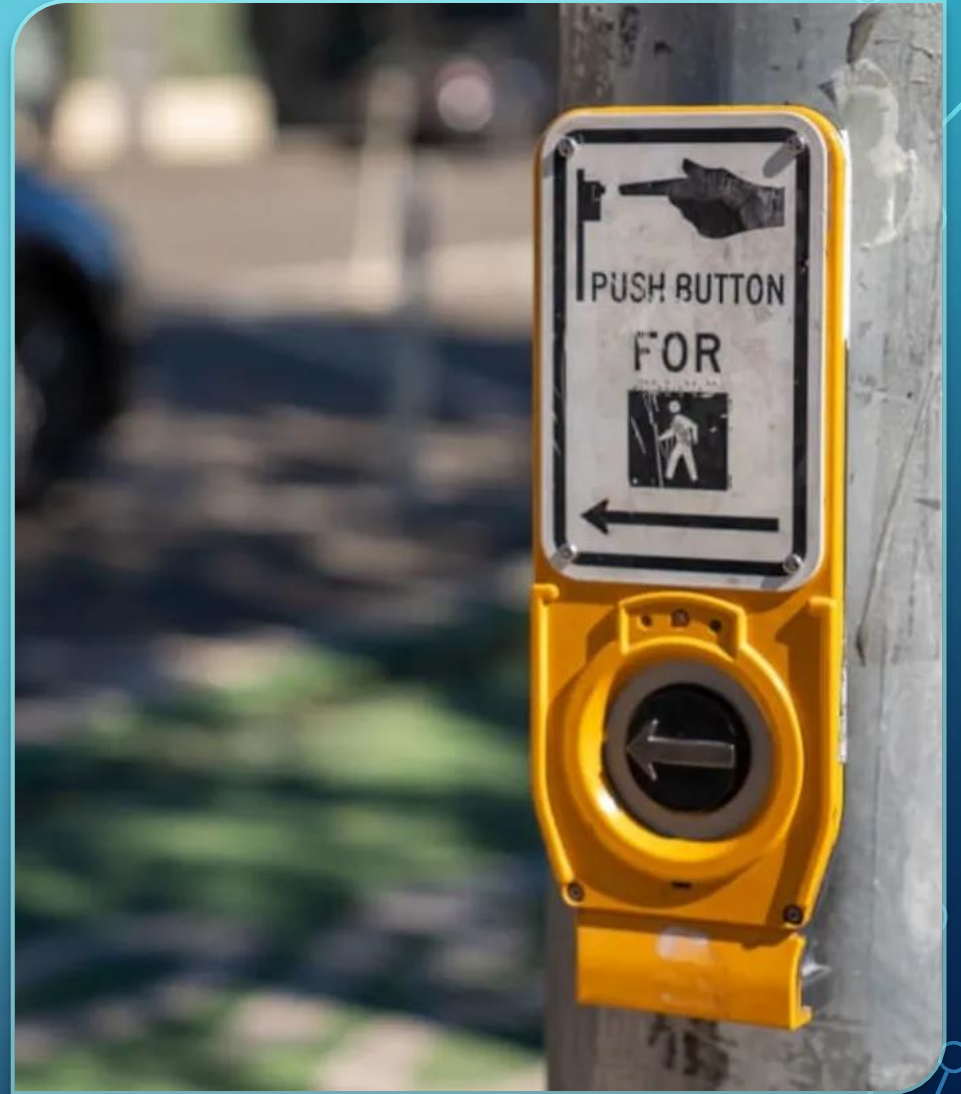


# SCREENSHOT OF CODE IN ARDUINO IDE



## MODULE 5

- INSERTING BOTTON SIMULATING CROSSWALK BUTTON
- WRITING CODE TO INITIATE CROSSWALK COUNT DOWN





# PICTURE OF THE CIRCUIT WITH WORKING LEDs

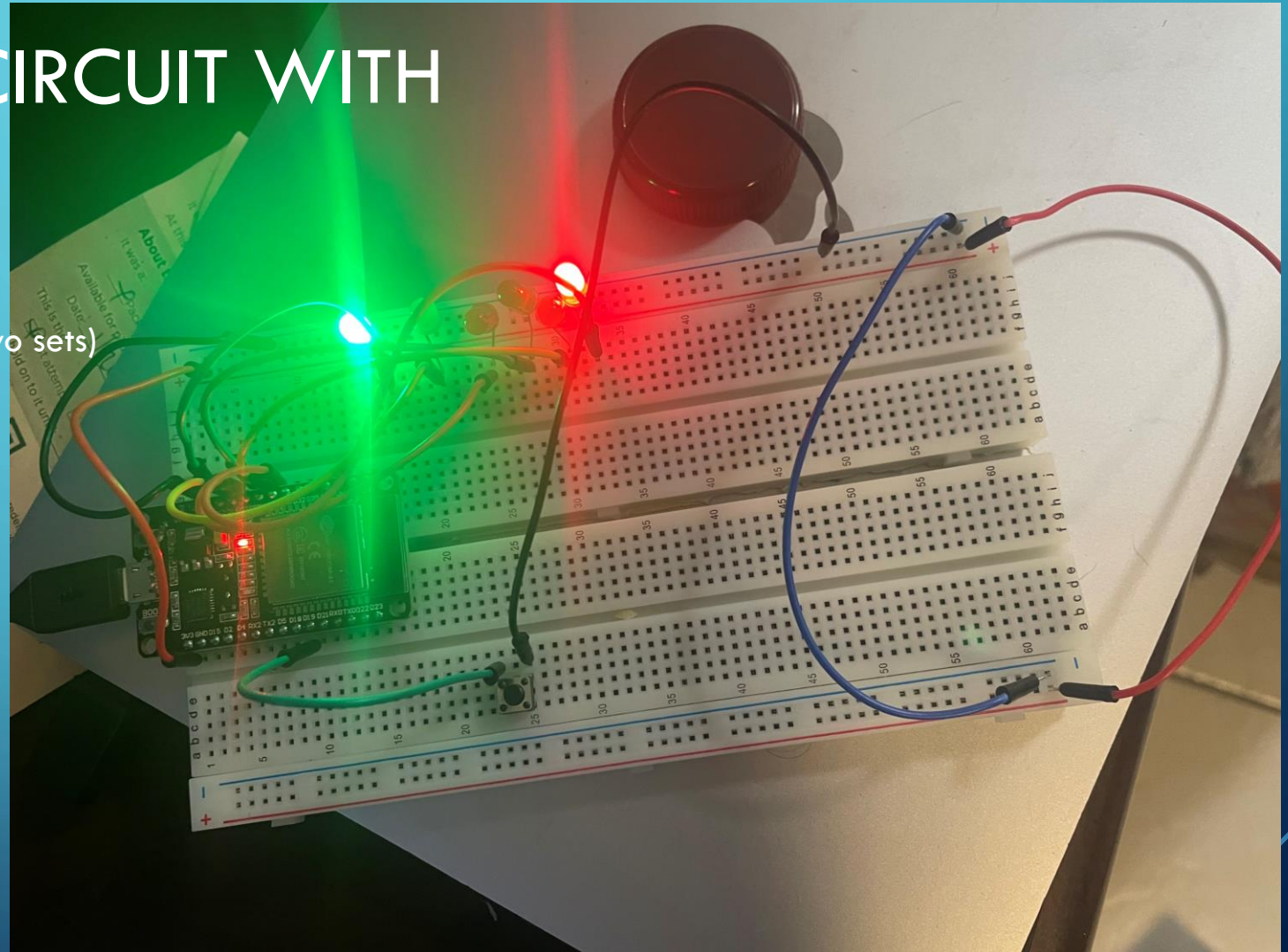
ESP 32 Board

Colored LEDs: Red, Yellow, and Green (two sets)

Push Button

Wires

Breadboard



DOIT ESP32 DEVKIT V1

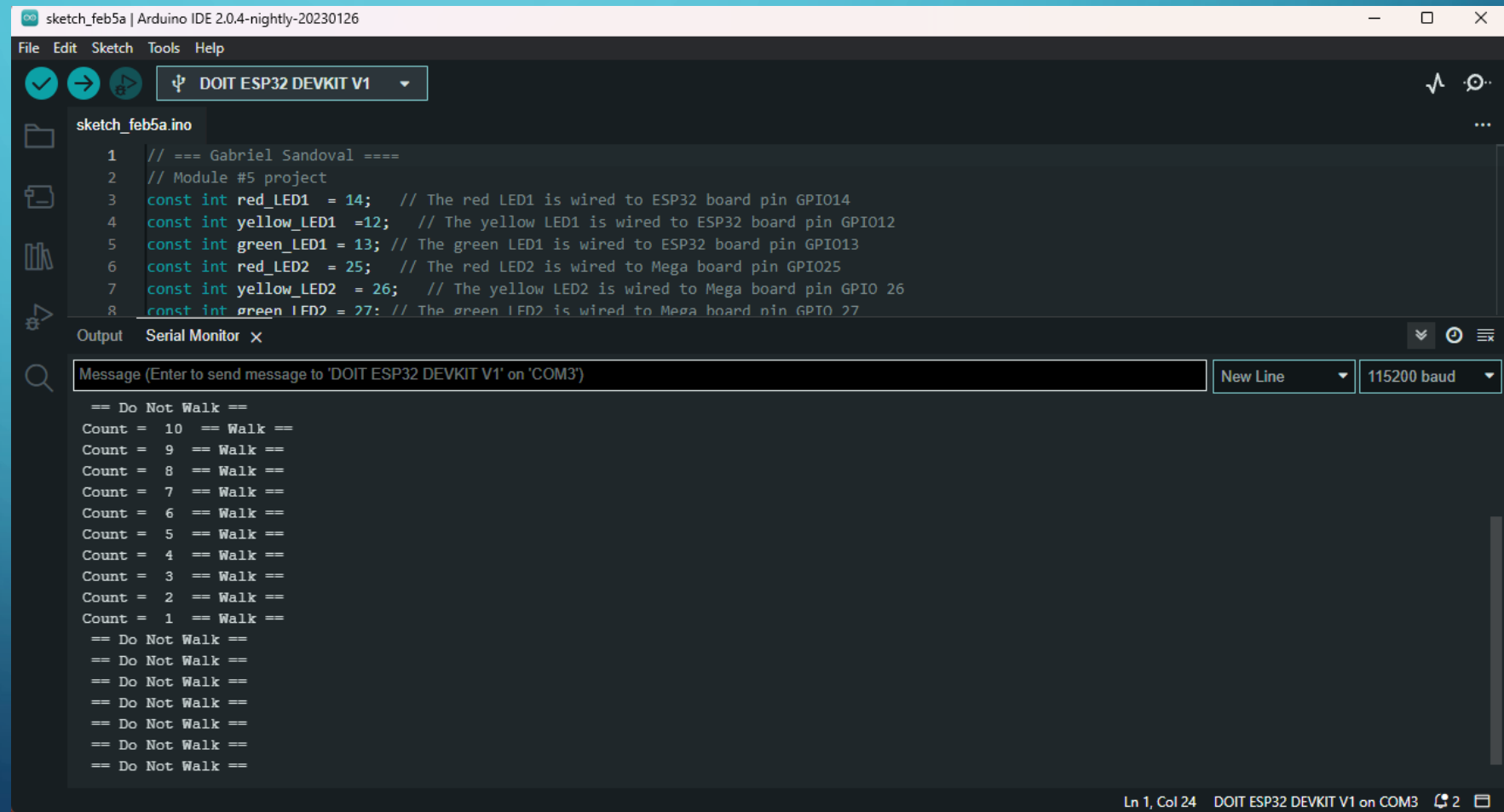
sketch\_feb5a.ino

```
1 // === Gabriel Sandoval ===
2 // Module #5 project
3 const int red_LED1 = 14; // The red LED1 is wired to ESP32 board pin GPIO14
4 const int yellow_LED1 = 12; // The yellow LED1 is wired to ESP32 board pin GPIO12
5 const int green_LED1 = 13; // The green LED1 is wired to ESP32 board pin GPIO13
6 const int red_LED2 = 25; // The red LED2 is wired to Mega board pin GPIO25
7 const int yellow_LED2 = 26; // The yellow LED2 is wired to Mega board pin GPIO 26
8 const int green_LED2 = 27; // The green LED2 is wired to Mega board pin GPIO27
9
10 int Xw value;
11 const int Xw_button = 19; //Cross Walk button
12
13 // the setup function runs once when you press reset or power the board
14 void setup() {
15
16   pinMode(Xw_button, INPUT_PULLUP); // 0=pressed, 1 = unpressed button
17   Serial.begin(115200);
18   pinMode(red_LED1, OUTPUT); // initialize digital pin 14 (Red LED1) as an output.
19   pinMode(yellow_LED1, OUTPUT); // initialize digital pin 12 (yellow LED1) as an output.
20   pinMode(green_LED1, OUTPUT); // initialize digital pin 13 (green LED1) as an output.
21
22   pinMode(red_LED2, OUTPUT); // initialize digital pin 25(Red LED2) as an output.
23   pinMode(yellow_LED2, OUTPUT); // initialize digital pin 26 (yellow LED2) as an output.
24   pinMode(green_LED2, OUTPUT); // initialize digital pin 27 (green LED2) as an output.
25 }
26
27 // the loop function runs over and over again forever
28 void loop() {
29
```

# SCREENSHOT OF CODE IN ARDUINO IDE

SCREENSHOT OF CODE IN ARDUINO IDE

# SCREENSHOT OF SERIAL MONITOR IN ARDUINO IDE



The screenshot displays the Arduino IDE interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for checking, running, and uploading code, along with a dropdown menu showing the selected board as 'DOIT ESP32 DEVKIT V1'. The main editor window shows a sketch named 'sketch\_feb5a.ino' with the following code:

```
1 // === Gabriel Sandoval ===  
2 // Module #5 project  
3 const int red_LED1 = 14; // The red LED1 is wired to ESP32 board pin GPIO14  
4 const int yellow_LED1 = 12; // The yellow LED1 is wired to ESP32 board pin GPIO12  
5 const int green_LED1 = 13; // The green LED1 is wired to ESP32 board pin GPIO13  
6 const int red_LED2 = 25; // The red LED2 is wired to Mega board pin GPIO25  
7 const int yellow_LED2 = 26; // The yellow LED2 is wired to Mega board pin GPIO 26  
8 const int green_LED2 = 27; // The green LED2 is wired to Mega board pin GPIO 27
```

Below the editor is the 'Serial Monitor' window, which is currently open. It has a search icon on the left and a 'x' icon on the right. The input field contains the text 'Message (Enter to send message to 'DOIT ESP32 DEVKIT V1' on 'COM3')'. To the right of the input field are two dropdown menus: 'New Line' and '115200 baud'. The output area shows the following text:

```
== Do Not Walk ==  
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 ==  
== Do Not Walk ==  
== Do Not Walk ==  
== Do Not Walk ==  
== Do Not Walk ==  
== Do Not Walk ==
```

The status bar at the bottom right indicates 'Ln 1, Col 24 DOIT ESP32 DEVKIT V1 on COM3' and shows a refresh icon and a tab icon.





# MODULE 6

- INSTALLING DRIVER FOR LCD SCREEN
- WIRING FOR LCD SCREEN
- BUZZER
- WIRING FOR THE BUZZER

# PICTURE OF CIRCUIT WITH WORKING LEDs AND LCD DISPLAY

ESP 32 Board

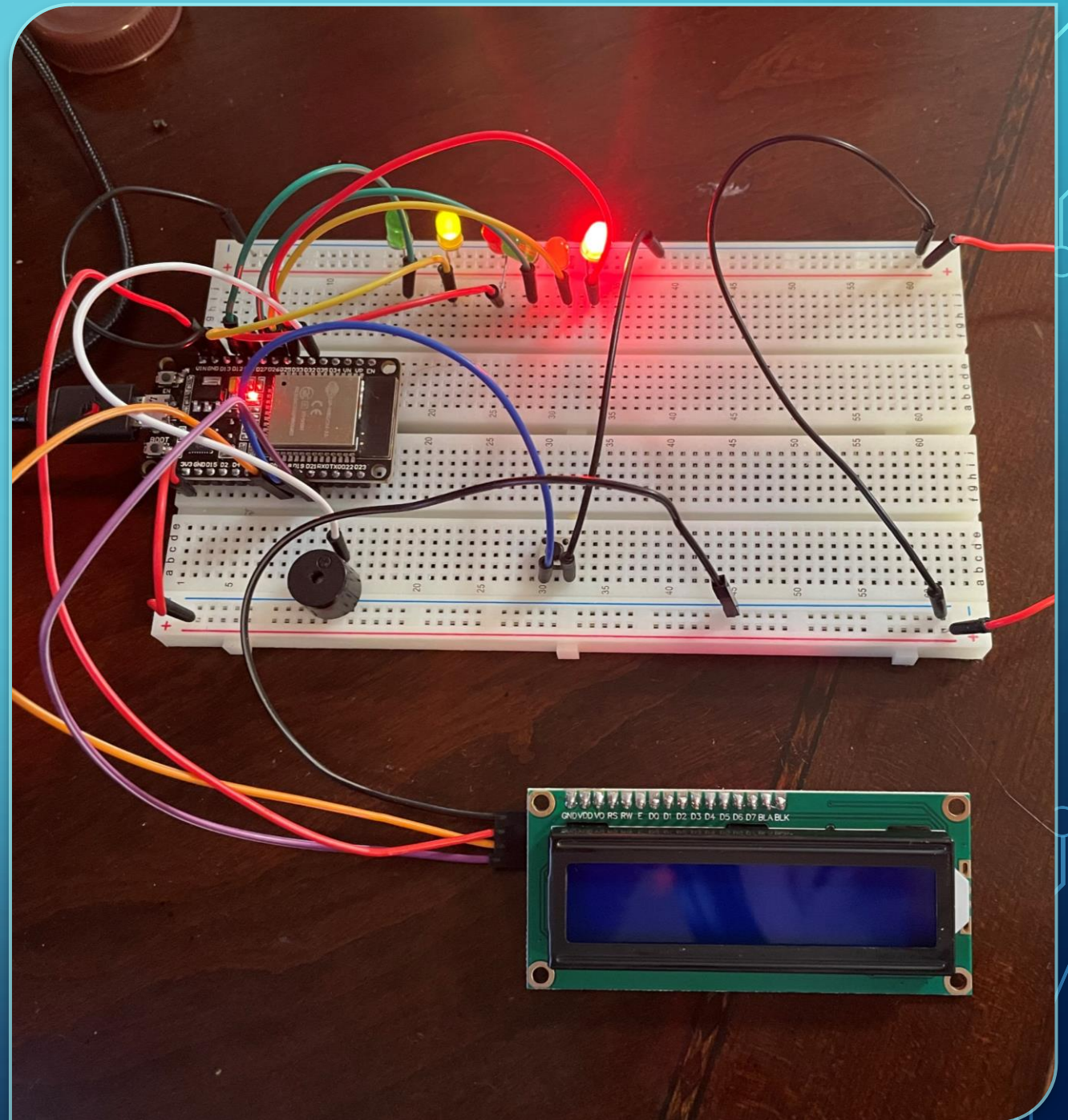
Colored LEDs: Red, Yellow and Green (two sets)

Push Button

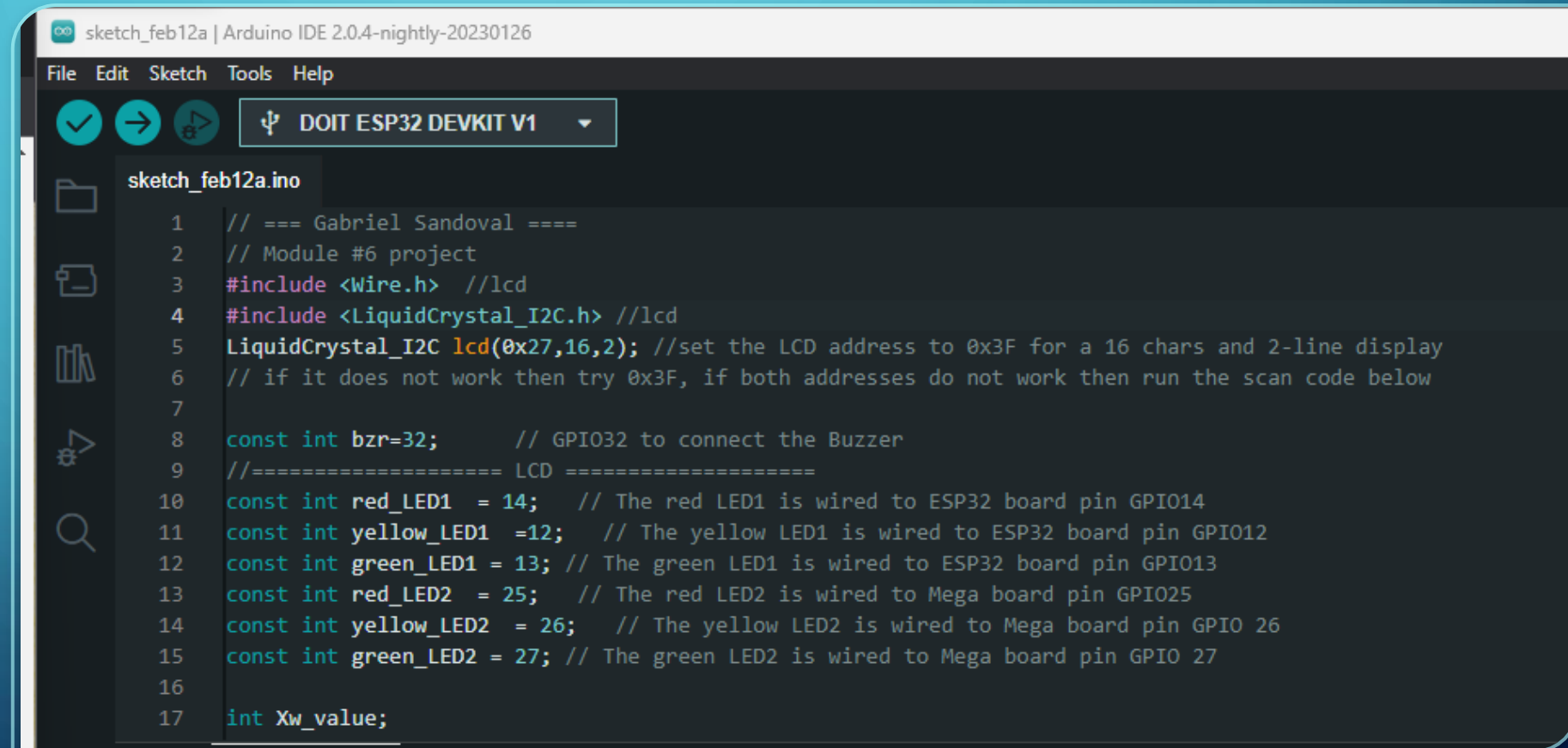
LCD Unit with Message Display (Defective)

Wires

Breadboard



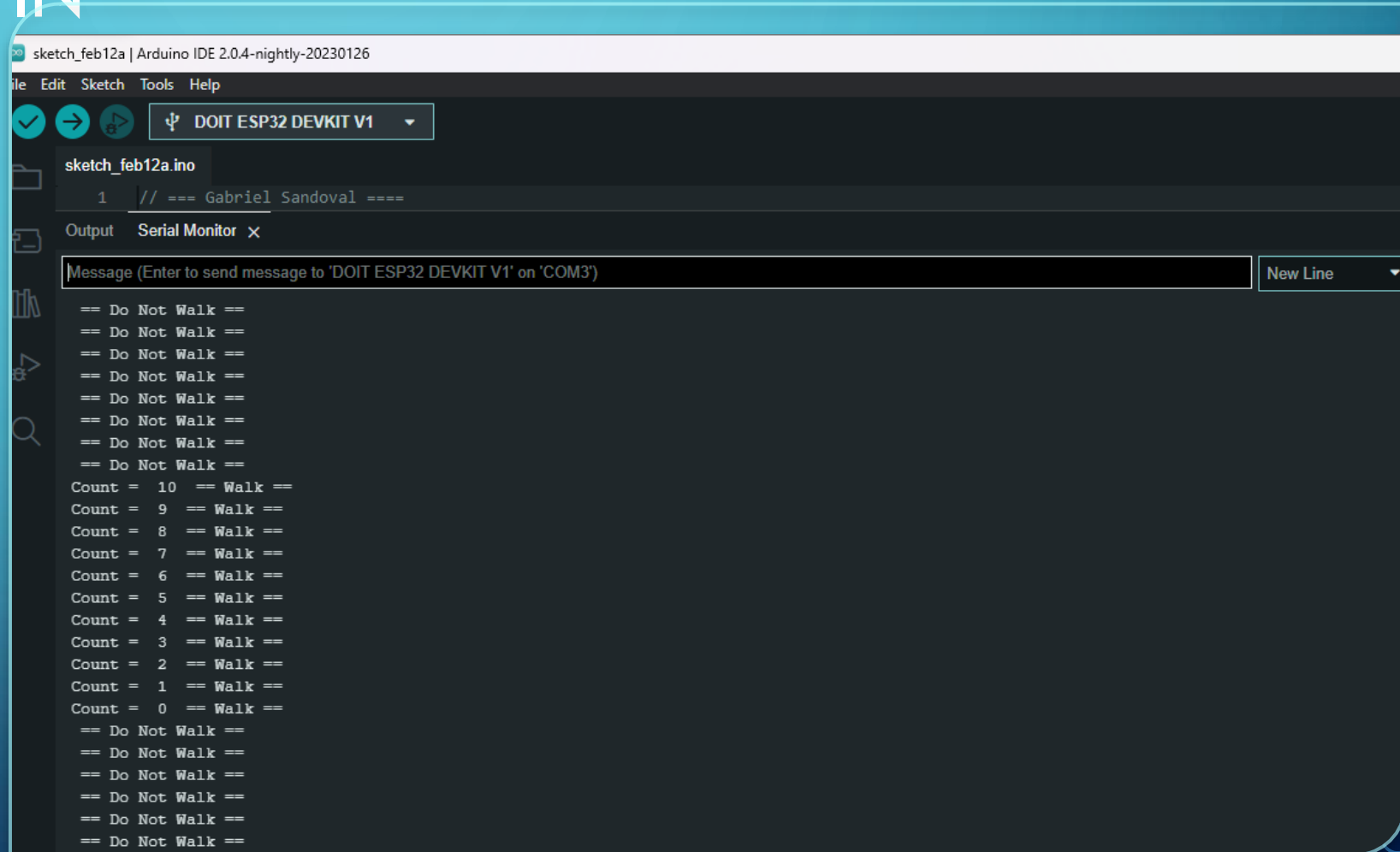
# SCREENSHOT OF CODE IN ARDUINO IDE



```
sketch_feb12a | Arduino IDE 2.0.4-nightly-20230126
File Edit Sketch Tools Help
[Checkmark] [Next] [Upload] [USB] DOIT ESP32 DEVKIT V1
sketch_feb12a.ino
1 // === Gabriel Sandoval ===
2 // Module #6 project
3 #include <Wire.h> //lcd
4 #include <LiquidCrystal_I2C.h> //lcd
5 LiquidCrystal_I2C lcd(0x27,16,2); //set the LCD address to 0x3F for a 16 chars and 2-line display
6 // if it does not work then try 0x3F, if both addresses do not work then run the scan code below
7
8 const int bzc=32; // GPIO32 to connect the Buzzer
9 //===== LCD =====
10 const int red_LED1 = 14; // The red LED1 is wired to ESP32 board pin GPIO14
11 const int yellow_LED1 = 12; // The yellow LED1 is wired to ESP32 board pin GPIO12
12 const int green_LED1 = 13; // The green LED1 is wired to ESP32 board pin GPIO13
13 const int red_LED2 = 25; // The red LED2 is wired to Mega board pin GPIO25
14 const int yellow_LED2 = 26; // The yellow LED2 is wired to Mega board pin GPIO 26
15 const int green_LED2 = 27; // The green LED2 is wired to Mega board pin GPIO 27
16
17 int Xw_value;
```



# SCREENSHOT OF SERIAL MONITOR IN ARDUINO IDE



The screenshot shows the Arduino IDE interface with the Serial Monitor window open. The top bar indicates the sketch is named 'sketch\_feb12a' and the IDE version is '2.0.4-nightly-20230126'. The menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. The toolbar shows icons for checking, running, and uploading, along with a dropdown menu for the selected board, 'DOIT ESP32 DEVKIT V1'. The editor window shows a single line of code: `1 // === Gabriel Sandoval ===`. The Serial Monitor window has tabs for 'Output' and 'Serial Monitor', with the latter being active. It features a text input field with the placeholder 'Message (Enter to send message to 'DOIT ESP32 DEVKIT V1' on 'COM3')' and a 'New Line' button. The output area displays the following text:

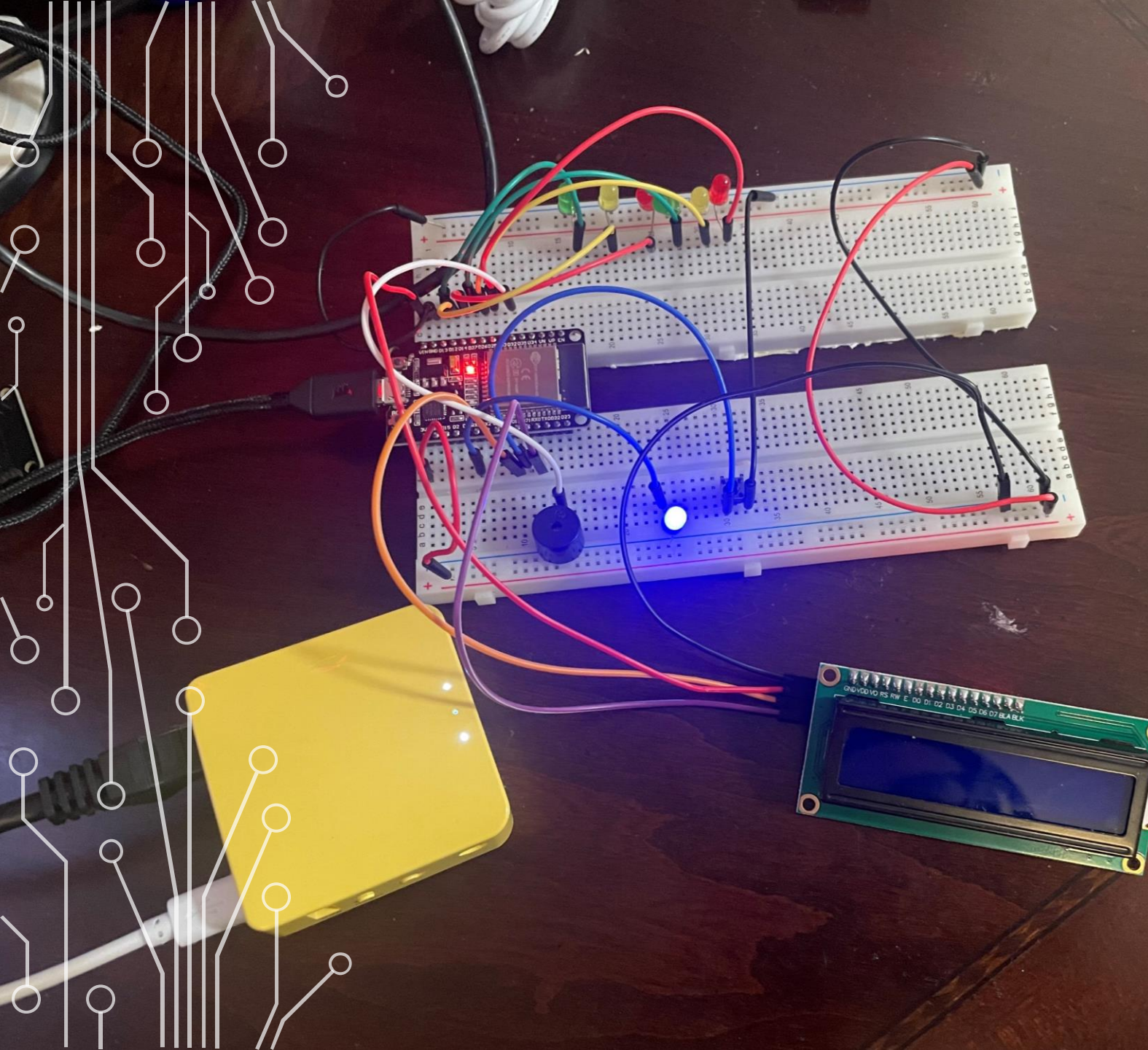
```
== Do Not Walk ==  
== Do Not Walk ==  
== Do Not Walk ==  
== Do Not Walk ==  
== Do Not Walk ==  
== Do Not Walk ==  
== Do Not Walk ==  
== Do Not Walk ==  
Count = 10 == Walk ==  
Count = 9 == Walk ==  
Count = 8 == Walk ==  
Count = 7 == Walk ==  
Count = 6 == Walk ==  
Count = 5 == Walk ==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



## MODULE 7

- SETTING UP WIRELESS ROUTER
- CONNECTING ESP32 TO WIFI ROUTER
- WRITING AND EXECUTING NEW CODE
- TRIGGERING EMERGENCY SINGAL



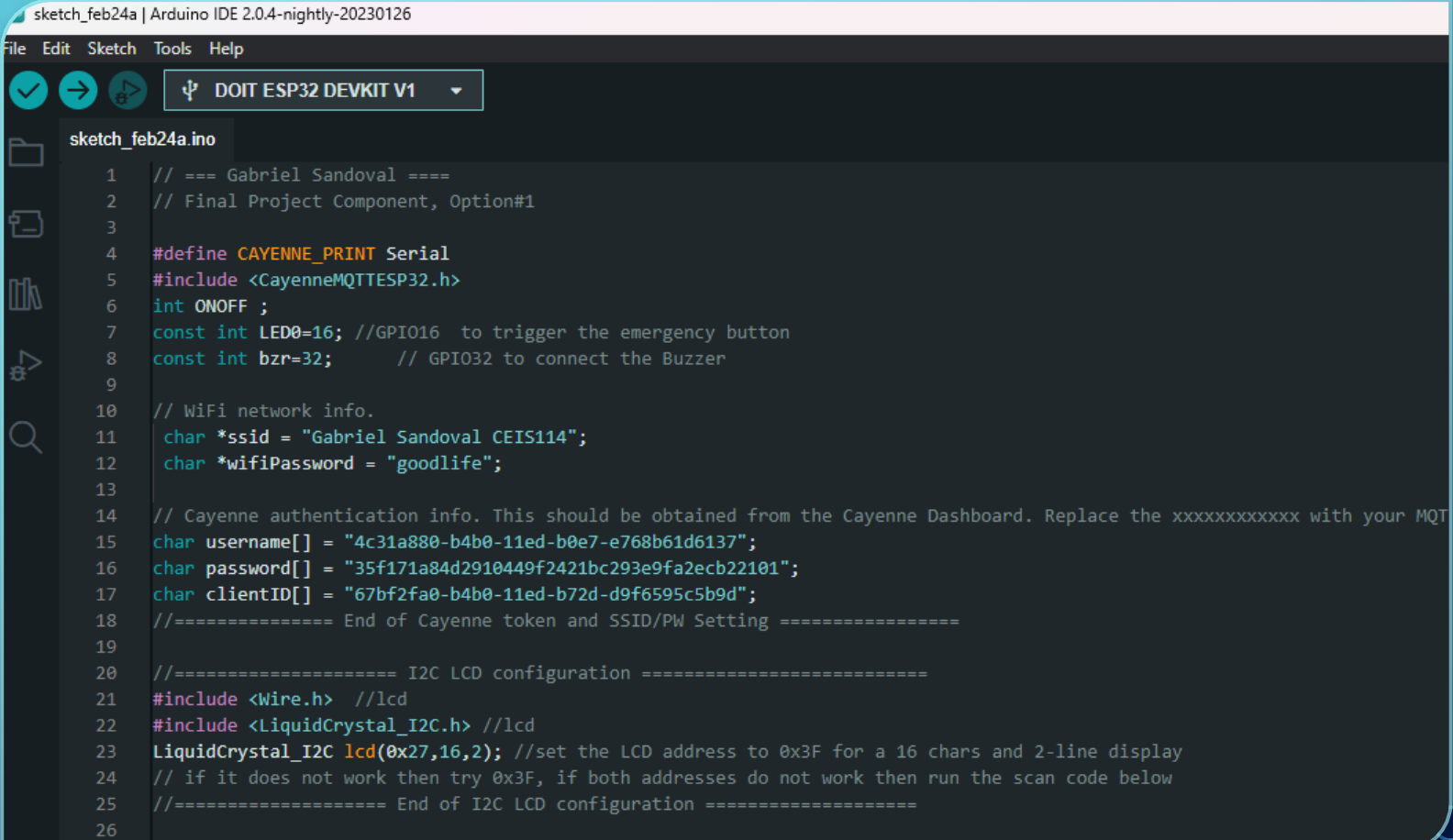


## PICTURE OF CIRCUIT WITH WORKING LEDS AND LCD DISPLAY

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

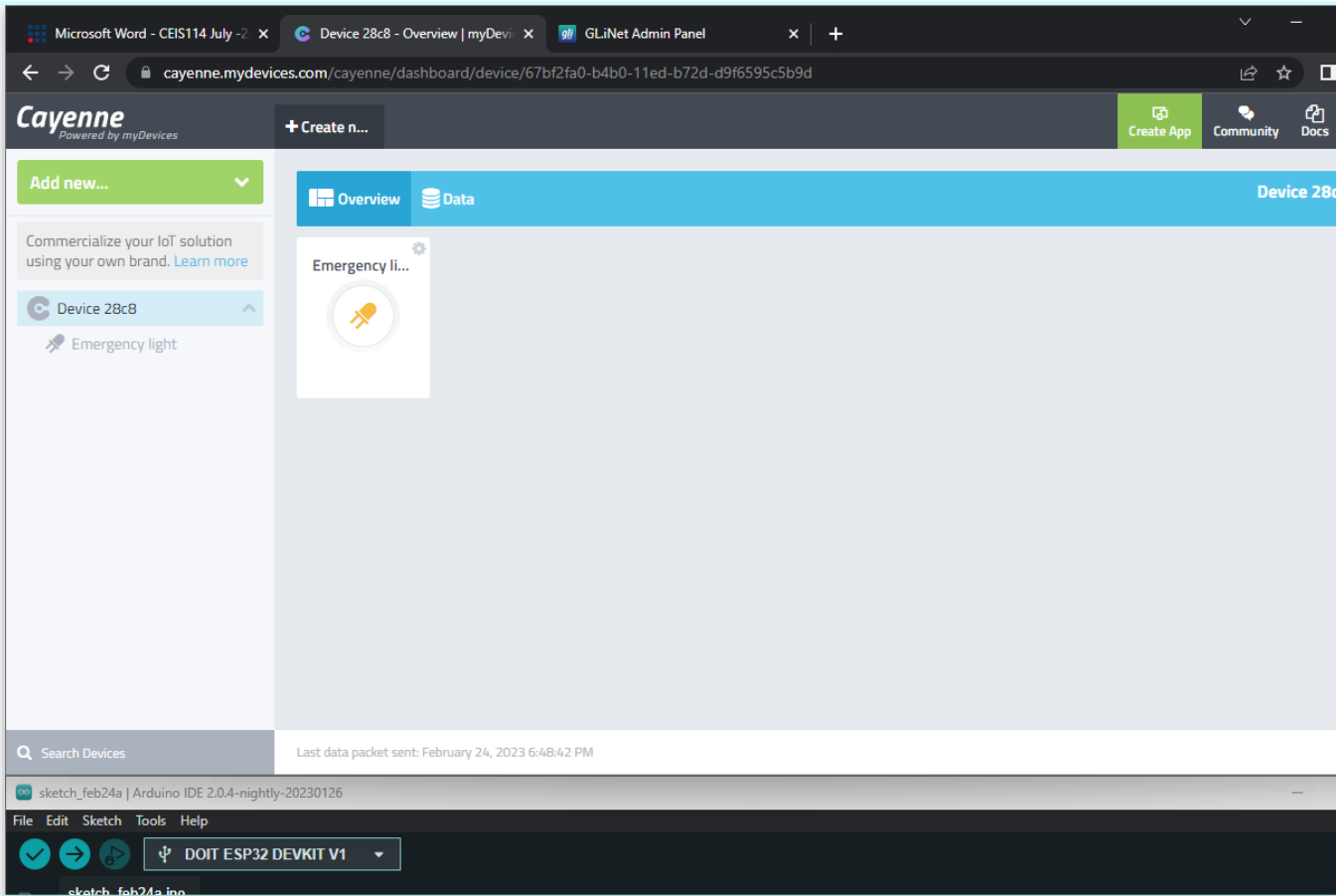


# SCREENSHOT OF CODE IN ARDUINO IDE



```
sketch_feb24a | Arduino IDE 2.0.4-nightly-20230126
File Edit Sketch Tools Help
ψ DOIT ESP32 DEVKIT V1
sketch_feb24a.ino
1 // === Gabriel Sandoval ===
2 // Final Project Component, Option#1
3
4 #define CAYENNE_PRINT Serial
5 #include <CayenneMQTTESP32.h>
6 int ONOFF ;
7 const int LED0=16; //GPIO16 to trigger the emergency button
8 const int bzt=32; // GPIO32 to connect the Buzzer
9
10 // WiFi network info.
11 char *ssid = "Gabriel Sandoval CEIS114";
12 char *wifiPassword = "goodlife";
13
14 // Cayenne authentication info. This should be obtained from the Cayenne Dashboard. Replace the xxxxxxxxxxxx with your MQTT
15 char username[] = "4c31a880-b4b0-11ed-b0e7-e768b61d6137";
16 char password[] = "35f171a84d2910449f2421bc293e9fa2ecb22101";
17 char clientId[] = "67bf2fa0-b4b0-11ed-b72d-d9f6595c5b9d";
18 //===== End of Cayenne token and SSID/PW Setting =====
19
20 //===== I2C LCD configuration =====
21 #include <Wire.h> //lcd
22 #include <LiquidCrystal_I2C.h> //lcd
23 LiquidCrystal_I2C lcd(0x27,16,2); //set the LCD address to 0x3F for a 16 chars and 2-line display
24 // if it does not work then try 0x3F, if both addresses do not work then run the scan code below
25 //===== End of I2C LCD configuration =====
26
```

# SCREENSHOT OF SERIAL MONITOR IN ARDUINO IDE





# CHALLENGES

- TRANSFERRING THE PROJECT ONTO A NEW COMPUTER
- WIRING NOT CONNECTING PROPERLY
- LCD SCREEN DEFECTIVE AND NOT ABLE TO WORK
- PROPERLY INPUTTING NEW CODES
- SETTING UP WI-FI ROUTER WITH HOME INTERNET



# SKILLS OBTAINED

System  
programming

Networking

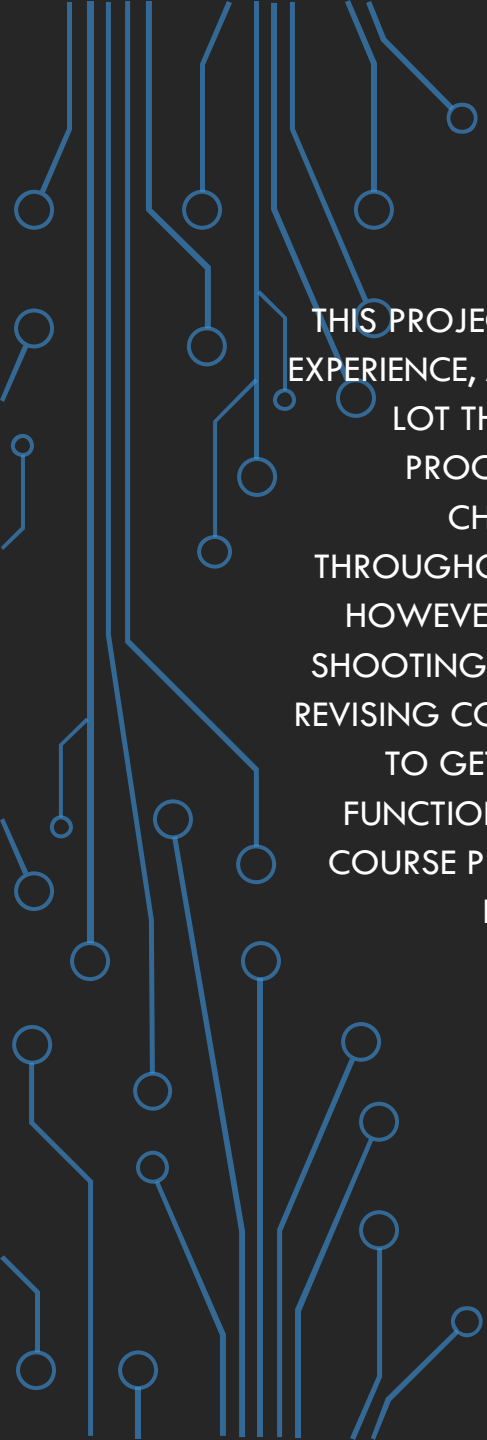
Organization

Coding

Wiring

Trouble  
shooting

Time  
management

A decorative graphic on the left side of the slide, consisting of a network of light blue lines and circles on a dark grey background, resembling a circuit board or a neural network.

THIS PROJECT WAS A GREAT  
EXPERIENCE, AND I LEARNED A  
LOT THROUGHOUT THE  
PROCESS. THERE WERE  
CHALLENGES I FACE  
THROUGHOUT THE PROJECT  
HOWEVER AFTER TROUBLE  
SHOOTING M SYSTEMS AND  
REVISING CODES, I WAS ABLE  
TO GET IT TO PROPERLY  
FUNCTION. I ENJOYED THE  
COURSE PROJECT, COURSE,  
PROFESSOR, AND  
CLASSMATES.

# CONCLUSION



# REFERENCES

- COURSE VIDEOS GUIDES
- ARDUINO HELP FORUMS
- YOUTUBE