**Course Project – Module 3**

**College of Engineering and Information Sciences**

**Course Number: CEIS114**

# **Creating the Traffic Controller**

# PARTS LIST

* PC running Arduino IDE With ESP32 Add-on
* ESP32 Board
* Colored LEDs: Red, Yellow and Green
* 220 Ohm Resistors (optional)
* Wires
* Breadboard

# Deliverables

* Complete the Course Project PowerPoint Deliverable
* Include a picture of your circuit
* Screenshot of Arduino IDE code from your computer

# INTRODUCTION

## Traffic Light

The purpose of this part of the project is to execute a program to simulate a traffic light controller. In the first step, we will control a single LED light, then we will learn how to control three LEDs to simulate a single traffic light.

*Electricity Basics*

An electronic circuit is composed of components where electric current can flow. Some of the components might be resistors, transistors, capacitors, inductors, diodes, etc. A circuit is a closed loop through which current can flow. A simple electric circuit may contain a battery, LED, and wire to connect them. Voltage is the difference in charge between two points whereas current is the rate at which the charge flows. Resistance occurs when a material resists the flow of charge. Ohm’s law is V=IR or Voltage = Current\* Resistance. We will see this applied this Module in the lab. You can choose to use resistors in your circuit or not. If you do not use resistors the LED light will be brighter since there is nothing resisting the flow of electricity.

## Software Reuse

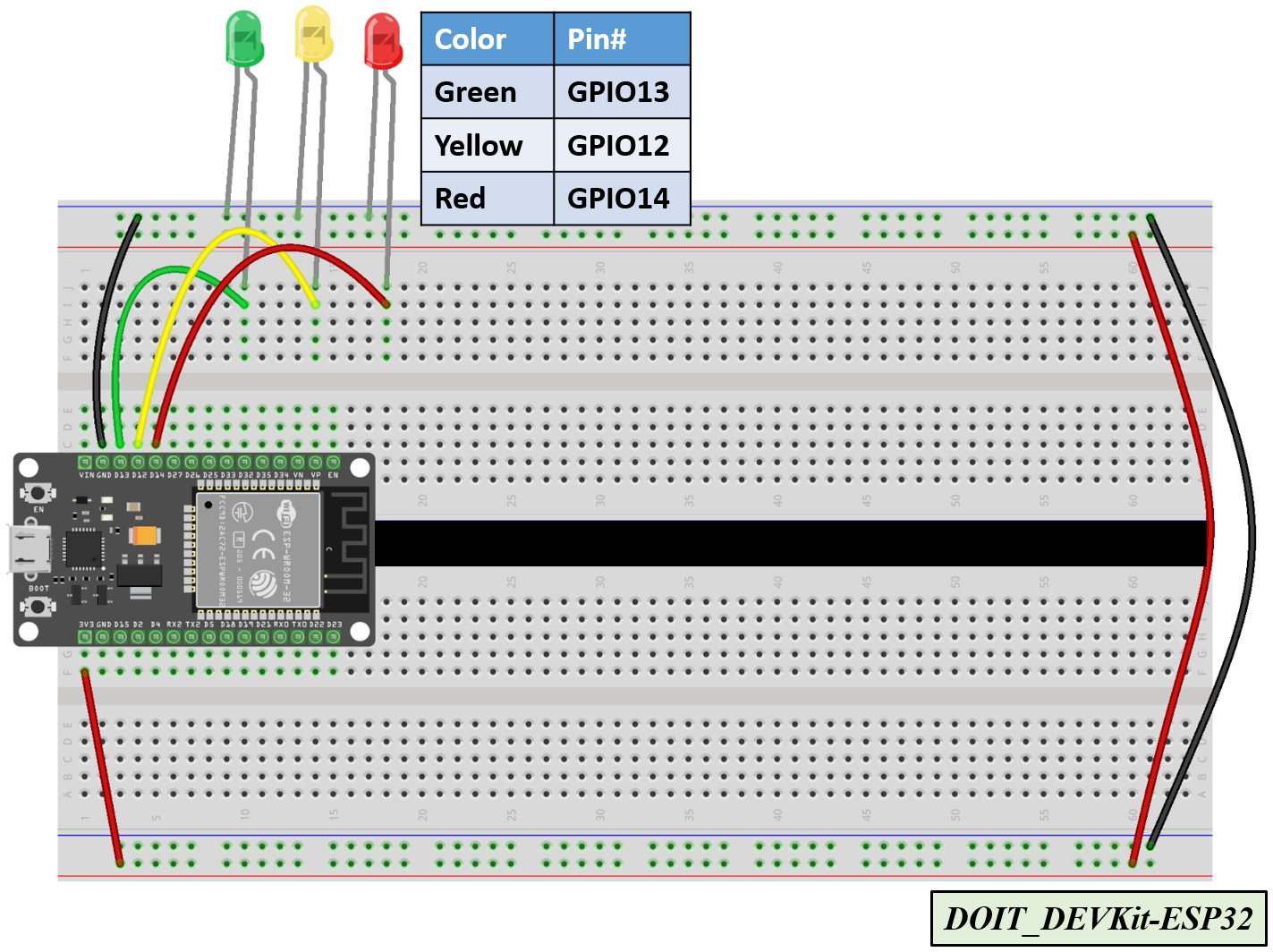
Most of the large software projects such as Linux, Windows have tens of millions of lines of source code. Rewriting every single line of the code for each new release is impossible. Reusing most of the existing code is necessary.

Code reuse does not mean simple copy and paste. To reuse the existing code correctly and efficiently, one must understand the code being reused. If the code is well documented, it will be much easier for the programmer to understand and reuse it properly. Once we are comfortable doing that, then we will add more complexity by controlling three LEDs at the same time. This in turn will allow us to simulate a traffic light controller.

# PROCEDURE

1. Insert the three LEDs (Red, Yellow, Green) in the breadboard as shown in Figure 1 (refer to the associated table below and the ESP32 pinout in the **Appendix**)
2. Connect all the wires as shown in Figure 1. Make sure to connect the Red, Yellow, and Green wires to the appropriate pins in the ESP32 Board.
3. Open the Arduino IDE and load the code shown in Figure 2.
4. Connect the ESP32 Board and check if the board is recognized by the Arduino IDE. If needed, please refer to Appendix A for connecting and configuring the ESP32 Board.

**Figure 1. Single Traffic Light**



**Figure 2. Code for One Traffic Light (3 LEDs: Green, Yellow, Red)**

*// === Replace this text with your Name ====*

*// 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*

*}*

Figure 2 – Code for Single Traffic Light

**Appendix A**

# **ESP32 WROOM32 DevKit Pinout**



# **Appendix B**

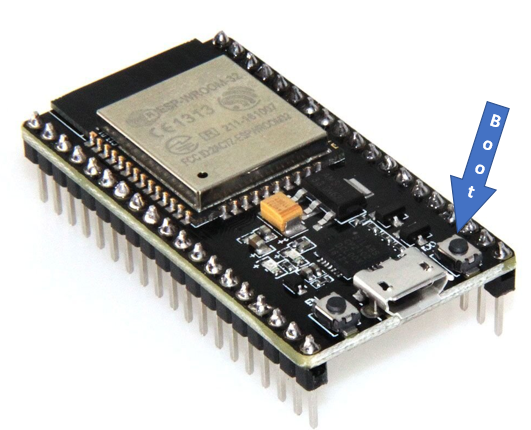
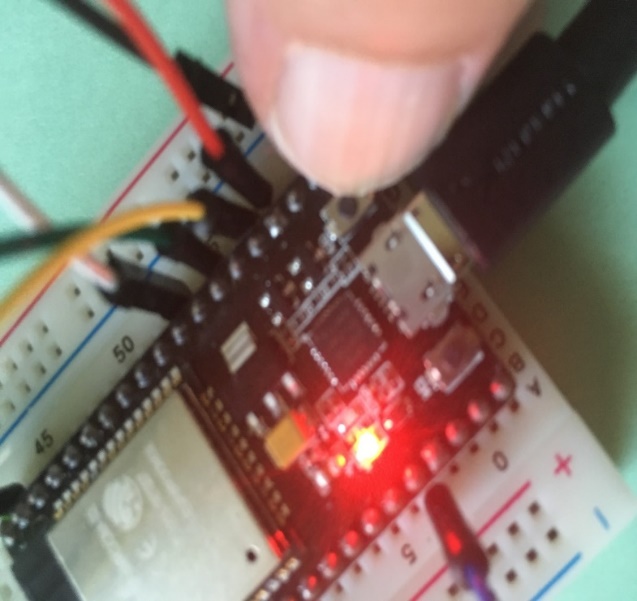
# **Troubleshooting sketch uploading**

**A fatal error occurred: “Failed to connect to ESP32: Timed out… Connecting…”**

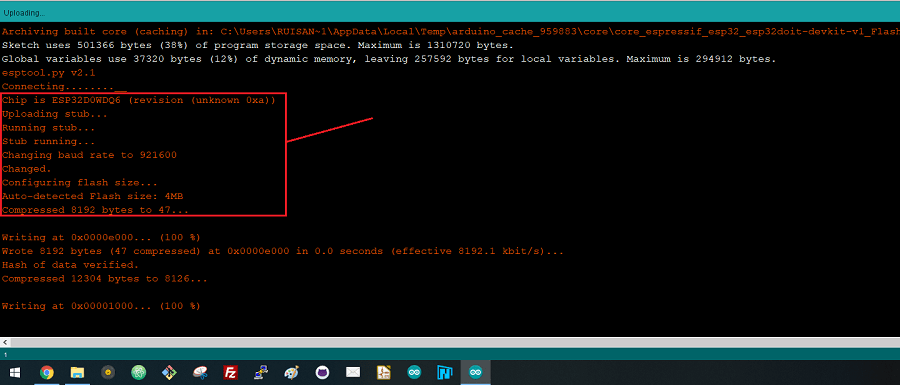
When you try to upload a new sketch to your ESP32 and it fails to connect to your board, it means that your ESP32 is not in flashing/uploading mode.

Given that you have already selected the right board name and COM port, follow these steps:

* Hold-down the “**BOOT**” button in your ESP32 board

* Press the “**Upload**” button https://i2.wp.com/randomnerdtutorials.com/wp-content/uploads/2016/12/arduino-ide-upload-button.png?resize=34%2C29 in the Arduino IDE to upload a new sketch:
* After you see the  “**Connecting….**” message in your Arduino IDE, release the finger from the “**BOOT**” button:



* After that, you should see the “**Done uploading**” message

# **Appendix C**

# **Connecting two breadboards together**

Some of the ESP32 boards may have a different size in width and would cover a larger area as shown in Figure 3. This would not allow you to make connections to both sides of the ESP board.

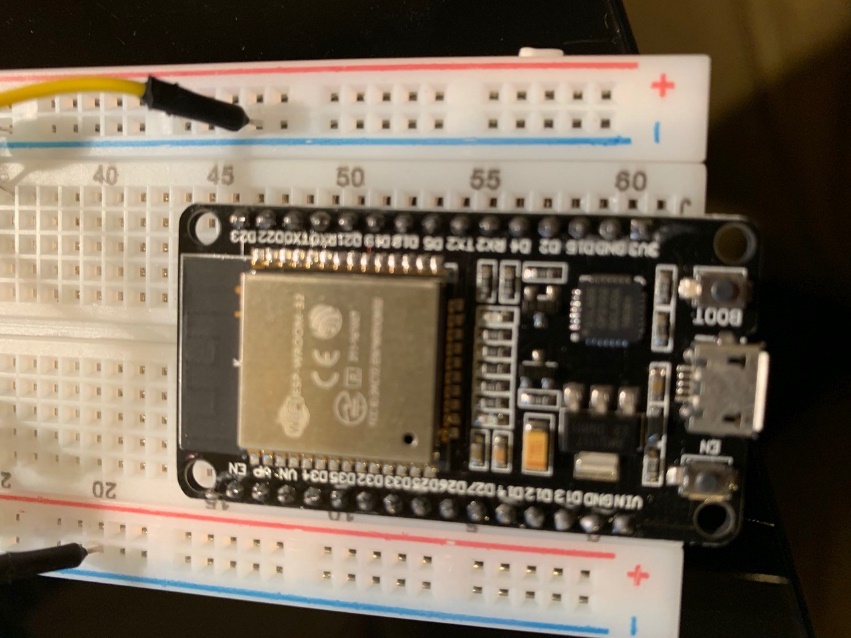


Figure 3

Please follow the following procedure for connecting two breadboards:

1. Remove one edge from both of the breadboards as shown in Figure 4.

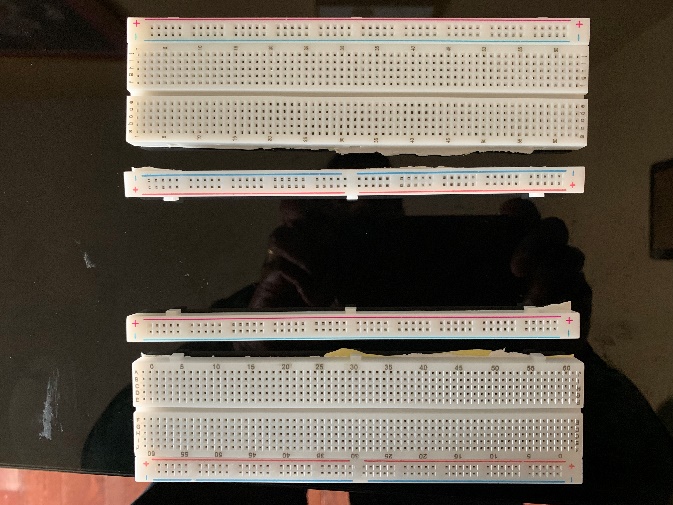


Figure 4

2. Bring the boards closer so that you could install the ESP32 board as shown in Figure 5.

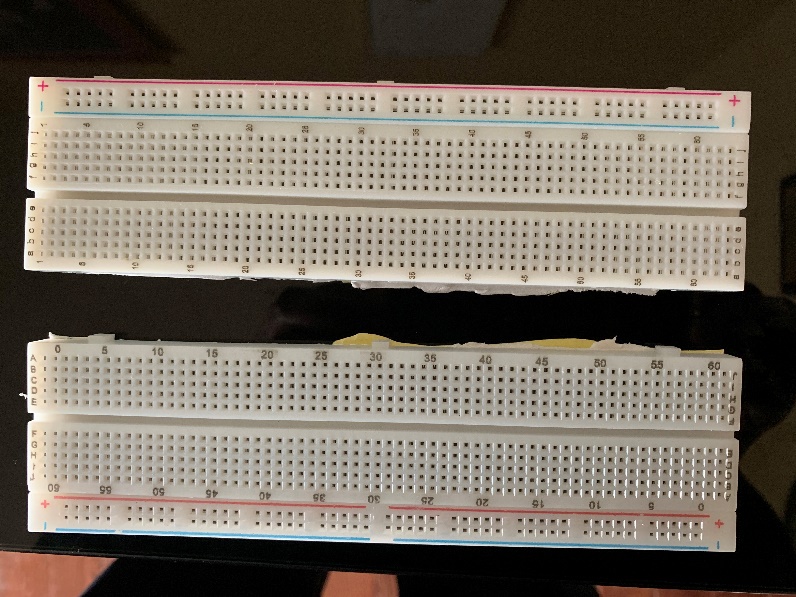


Figure 5

3. Install the ESP32 board as shown in Figure 6.

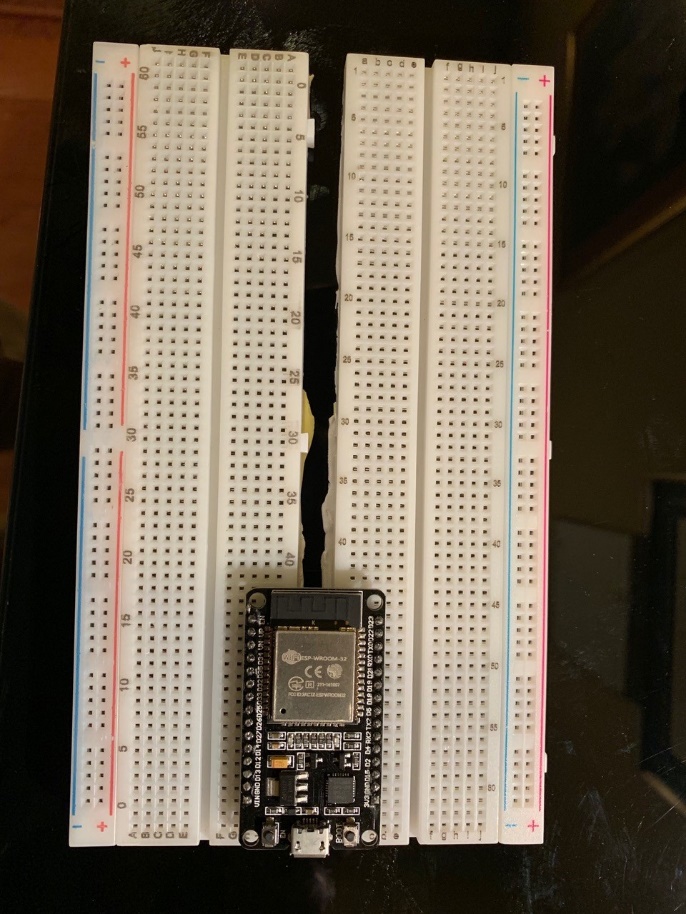


Figure 6