

CEIS 114

Module 6 – Option 1

Multiple Traffic Light Controller with Cross Walk and an Emergency
Buzzer with secured IoT Control via Web
Louie Venegas

What the System Does

Two traffic lights run a timed red, yellow, green sequence. A touch sensor on GPIO 13 registers a crosswalk request, the blue LED and LCD signal the walk state, and a buzzer provides an emergency alert. The ESP32 hosts a web interface behind a dedicated router so the controller can be operated remotely.

What I Took From It

Running several outputs on one timed state machine, reading a capacitive touch input, driving an LCD alongside other peripherals, and standing up a web server on the microcontroller itself. Putting that server behind its own router was the first time I treated a hobby build as something that needed securing.

Picture of Circuit

ESP32 Board

Two sets of Colored LEDs: Red, Yellow and Green

One Blue LED

Touch Sensor (Wire connected to GPIO 13)

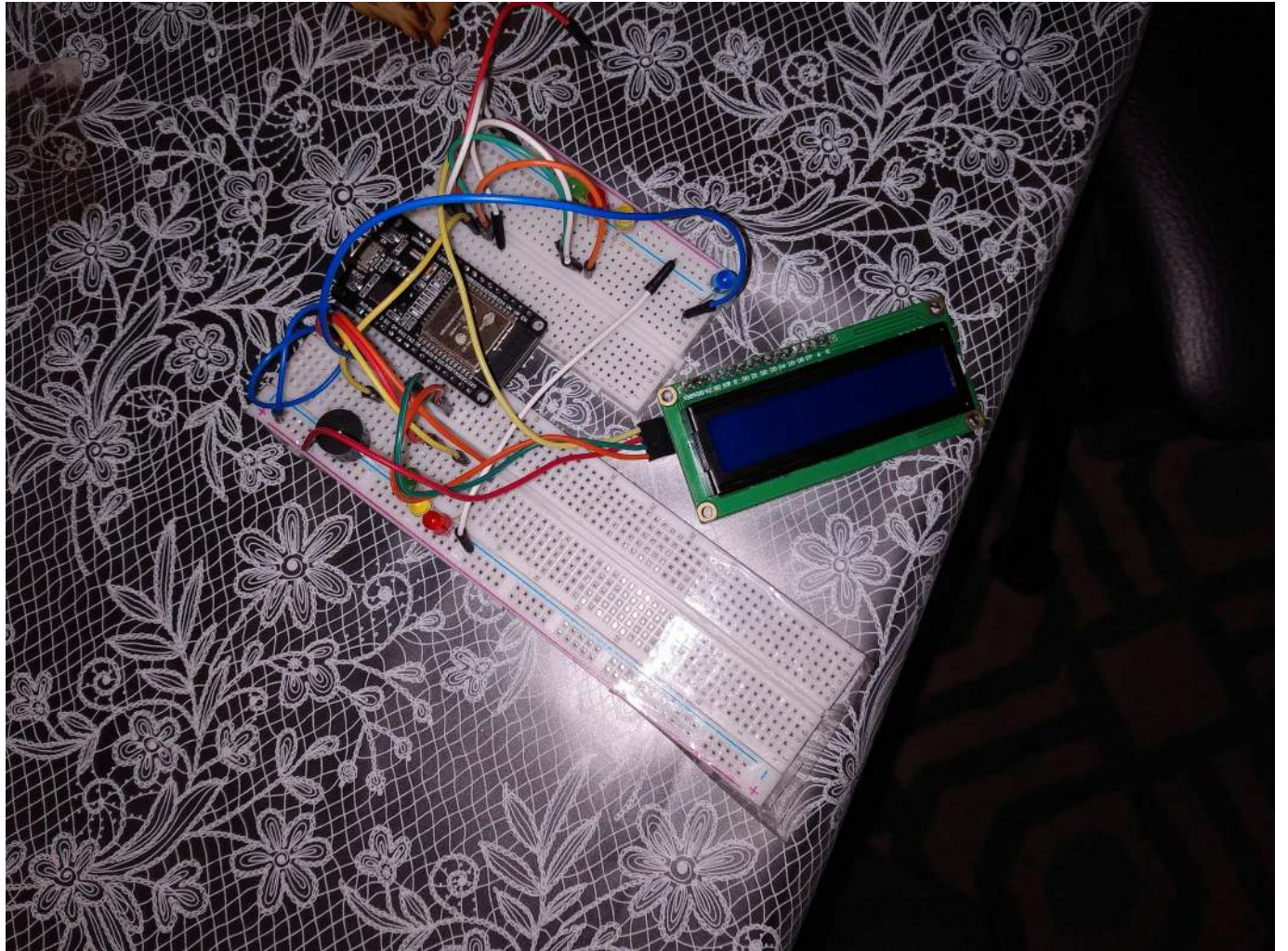
LCD Unit

Buzzer

Mini Router

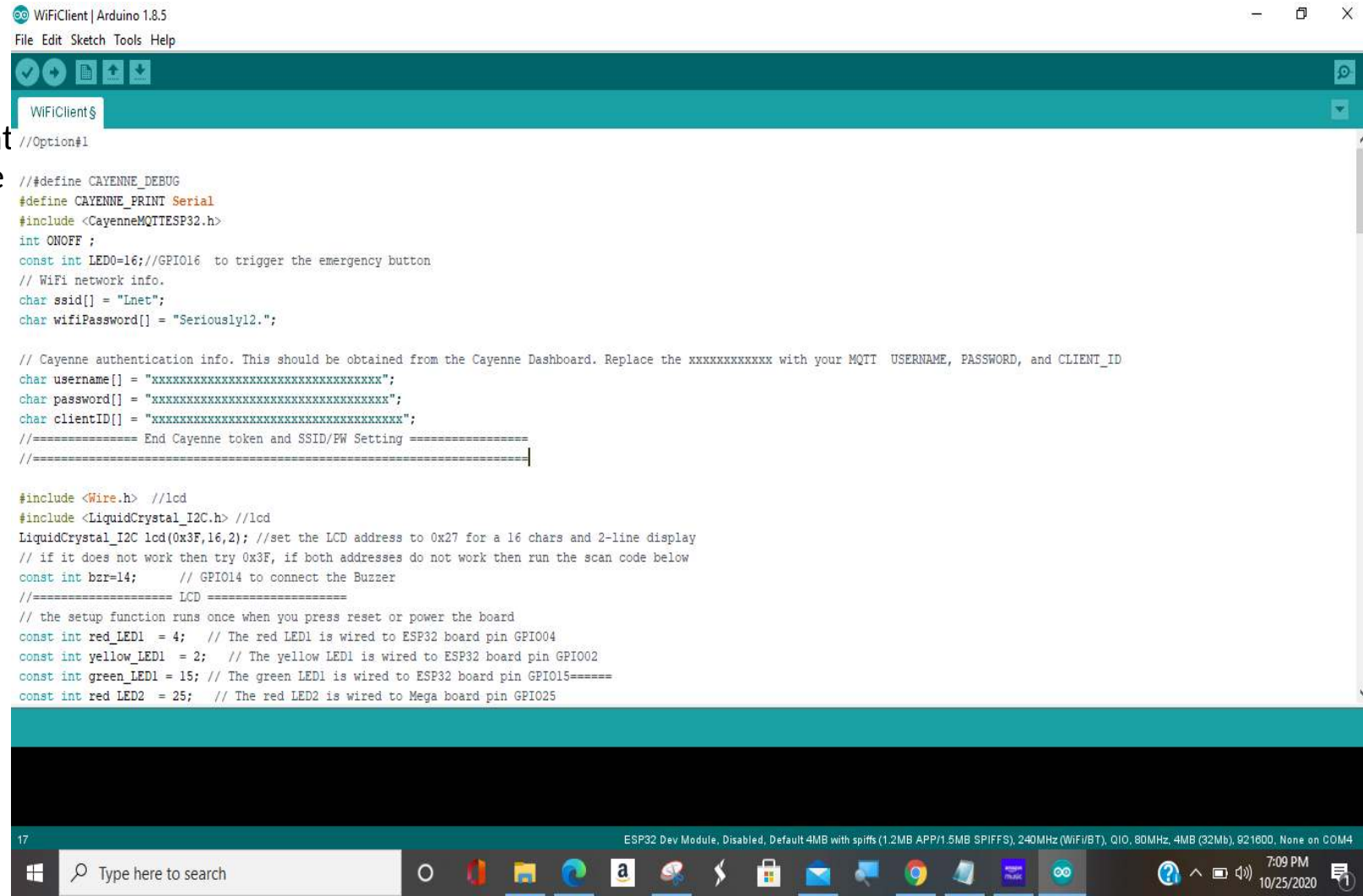
Wires

Breadboard



Screenshot of Arduino IDE

The controller code in the Arduino IDE: light sequencing, crosswalk touch input, and the emergency buzzer.



```
WiFiClient | Arduino 1.8.5
File Edit Sketch Tools Help

WiFiClient$
//Option#1

// #define CAYENNE_DEBUG
// #define CAYENNE_PRINT Serial
#include <CayenneMQTTESP32.h>
int ONOFF ;
const int LED0=16;//GPIO16 to trigger the emergency button
// WiFi network info.
char ssid[] = "Lnet";
char wifiPassword[] = "Seriously12.";

// Cayenne authentication info. This should be obtained from the Cayenne Dashboard. Replace the xxxxxxxxxxxx with your MQTT USERNAME, PASSWORD, and CLIENT_ID
char username[] = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx";
char password[] = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx";
char clientID[] = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx";
//===== End Cayenne token and SSID/PW Setting =====
//=====

#include <Wire.h> //I2C
#include <LiquidCrystal_I2C.h> //LCD
LiquidCrystal_I2C lcd(0x3F,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 buz=14; // GPIO14 to connect the Buzzer
//===== 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
```

Screenshot of Serial Monitor

Serial monitor output as the controller cycles
and a crosswalk request is triggered.