**Course Project - Final Project Component- Option 1**

**College of Engineering and Information Sciences**

**Course Number: CEIS114**

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

# Objectives

1. To integrate IoT control to the traffic light project using cloud service.
2. To give emergency personal access to the traffic light control.
3. Upon triggering IoT signal, all traffic from all directions will see red light and a flashing blue light.

# PARTS LIST

* PC running Arduino IDE
* ESP32Board
* Two sets of Colored LEDs: Red, Yellow and Green
* One Blue LED
* Button
* LCD Unit
* Buzzer
* Mini Router
* Wires
* Breadboard

Deliverables

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

Introduction

**Traffic Light Emergency Button System Security**

Due to the traffic light emergency button importance, its activation requires three levels of security as follows:

1. The first level requires the knowledge of the router’s credentials (SSID and Password) that will allow an individual to connect to the Internet through the specific router
2. The second level, is the requirement for the user name and password needed to connect to the specific dashboard in the cloud service
3. And finally, in order to activate the emergency traffic light button, the software program and the Microcontroller specific token must be known. the requirement to
4. PROCEDURE

**Step 1: Setup the Mini Router by following the steps stated in Appendix A. Instead of the mini-router, you can use your home router to connect to the ESP32.**

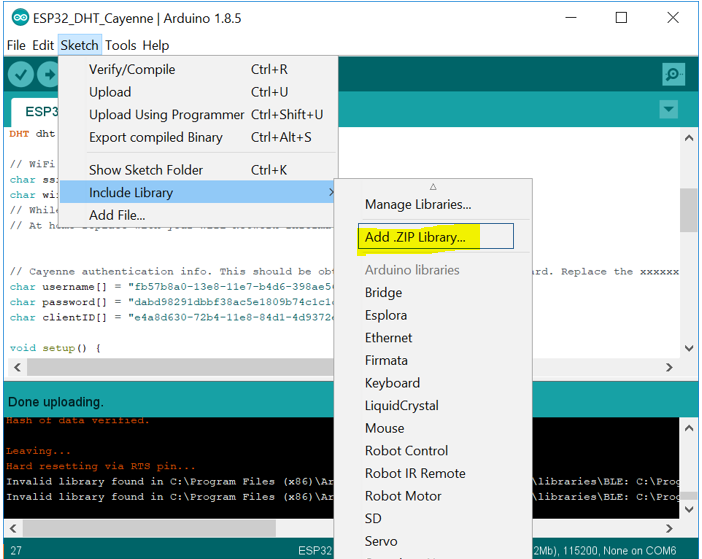
**Step 2: Cloud Connection Setup**

1. Download Cayenne library using the following link:

[**https://github.com/myDevicesIoT/Cayenne-MQTT-ESP**](https://github.com/myDevicesIoT/Cayenne-MQTT-ESP)

The library has been uploaded to the CEIS114 Module 6 project page. Assuming that the file was downloaded to the Downloads folder, use the following steps to add the Library.

1. Add the zipped library to your Arduino IDE as shown in Figure 1.



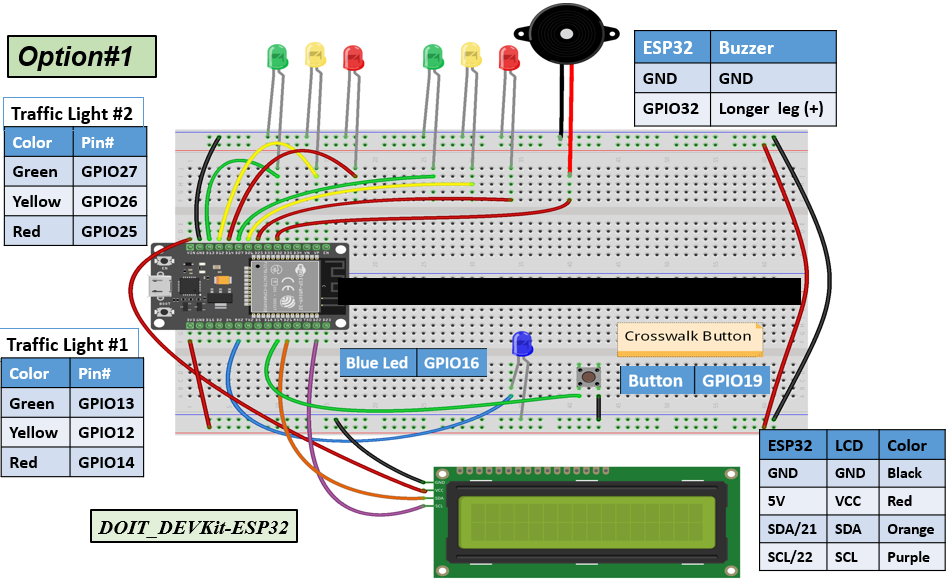
**Figure 1. Sketch > Include Library > Add .ZIP Library**

1. Now browse to the directory where you saved (Download folder) the zipped library (**Cayenne\_MQTT\_ESP\_master**) and select it. This will cause the ESP32 Cayenne library to be added.

**Step 3: Hardware circuit**

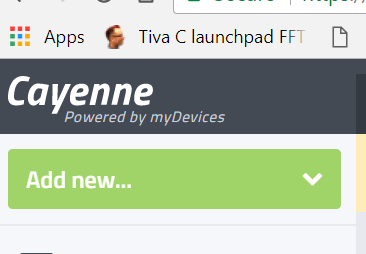
Use the same circuit we constructed in Module 5 with the addition of the blue flashing light as shown in the circuit shown in Figure 2.

**Figure 2: IoT Traffic Controller**

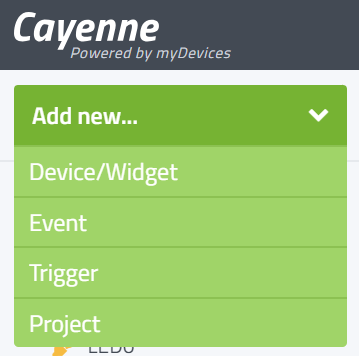


**Step 4: Follow the following steps to connect to the Cayenne cloud service**

1. Go to **MyDevices.com** and sign up for a free account (**https://Cayenne.mydevices.com/**)
2. Press Getting Started button
3. Add New ==> Device/Widget (1) and (2) in Figures 3 and 4
4. Select (**BYOT (Bring Your Own Thing**) as shown in Figure 5. This selection will generate your device special IDs



**1**



**2**

Figure 3 Figure 4

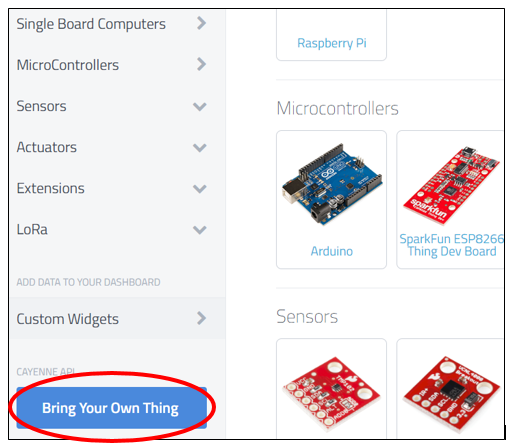


Figure 5

**Step 5:** This will generate your device special IDs of ( **MQTT USERNAME, MQTT PASSWORD, and MQTT CLIENT\_ID**). (**Notice: Do not copy the IDs in the figure 6 , you will have your own IDs generated**)

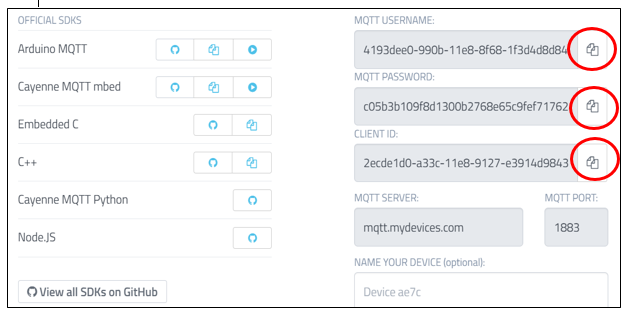


Figure 6

**Step 6:** Click on the circled icons one at a time **to copy and paste in the program shown in Figure 7.**

**Step 7**:Add the WiFi network info.(this is **your mini router (or home router) SSID and password, see Appendix A below for the steps)**

**char ssid[] = "xxxxxxxxx";**

**char wifiPassword[] = " xxxxxxxxx ";**

**Figure 7. Code for IoT Traffic Controller**

**// === John Francis CEIS114 ====**

**// Final Project Component, 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 = "**ATT8rSm3ru**";

char \*wifiPassword = "3ut7ty=fx%2p";

// Cayenne authentication info. This should be obtained from the Cayenne Dashboard. Replace the xxxxxxxxxxxx with your MQTT USERNAME, PASSWORD, and CLIENT\_ID

char username[] = " 71557690-55e9-11ec-9f5b-45181495093e";

char password[] = " 9eb8d76caa57ce1a804fd5723fcc27107343318f";

char clientID[] = " de41adf0-55e9-11ec-bbfc-979c23804144";

//=============== End Cayenne token and SSID/PW Setting =================

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

#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 ====================

// the setup function runs once when you press reset or power the board

onst 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

pinMode(LED0, OUTPUT); // For Cayenne Button

digitalWrite(LED0,LOW);

//================= DO NOT Change ===========

// This is here to force the ESP32 to reset the WiFi and initialize correctly.

Serial.print("WIFI status = ");

Serial.println(WiFi.getMode());

WiFi.disconnect(true);

delay(1000);

WiFi.mode(WIFI\_STA);

delay(1000);

Serial.print("WIFI status = ");

Serial.println(WiFi.getMode());

// End silly stuff !!! =======================

// Connect to provided SSID and PASSWORD

WiFi.begin(ssid, **password**);

//=============== Setting ESP32 for Cayenne ==============

Cayenne.begin(username, password, clientID, ssid, wifiPassword);

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 pin 12 (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() {

Cayenne.loop(); // Calling Cayenne =========

// read the cross walk button value:

**Xw\_value=digitalRead(Xw\_button);**

// if (Xw\_value == 0 ){ // if crosswalk button (X-button) pressed digitalWrite(yellow\_LED1 , LOW);

// This should turn off the YELLOW LED1

if(Xw\_value == 0 or ONOFF == 1){ // 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 == ");

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

//=================== can be RED only no flashing =============

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);

delay(500);

digitalWrite(red\_LED1, LOW); // This should turn off the RED LED1

digitalWrite(red\_LED2, LOW); // This should turn off the RED LED2

digitalWrite(bzr, LOW);

delay(500);

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

} // End of counter

// clears the display to print new message

lcd.setCursor(0,1); // set the cursor to column 1, line 2

lcd.print(" ");

} // === end of button pressed loop ==============

else // No Emergency === Normal traffic light operation ==

{

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

digitalWrite(red\_LED2, LOW); // This should turn off the RED LED2

digitalWrite(yellow\_LED2 , LOW); // This should turn off the YELLOW LED2

digitalWrite(green\_LED2, HIGH); // This should turn on the GREEN LED2

delay(2000); // wait for2 second

// 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

digitalWrite(red\_LED2, LOW); // This should turn off the RED LED2

digitalWrite(yellow\_LED2 , **HIGH**); // This should turn **on** the YELLOW LED2

digitalWrite(green\_LED2, LOW); // This should turn off the GREEN LED2

delay(2000); // wait for 1 second

// Both are RED for 1 second

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

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); // wait for 1 second

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

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 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

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 2 second

// Both are RED for 1 second

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

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); // wait for 1 second

}// No Emergency Button closing ============

}

CAYENNE\_IN(1)

// Using channel (1) (you can use any other channel but it must match the widget you will be creating for it. As shown in the example below for channel (1) widget

{

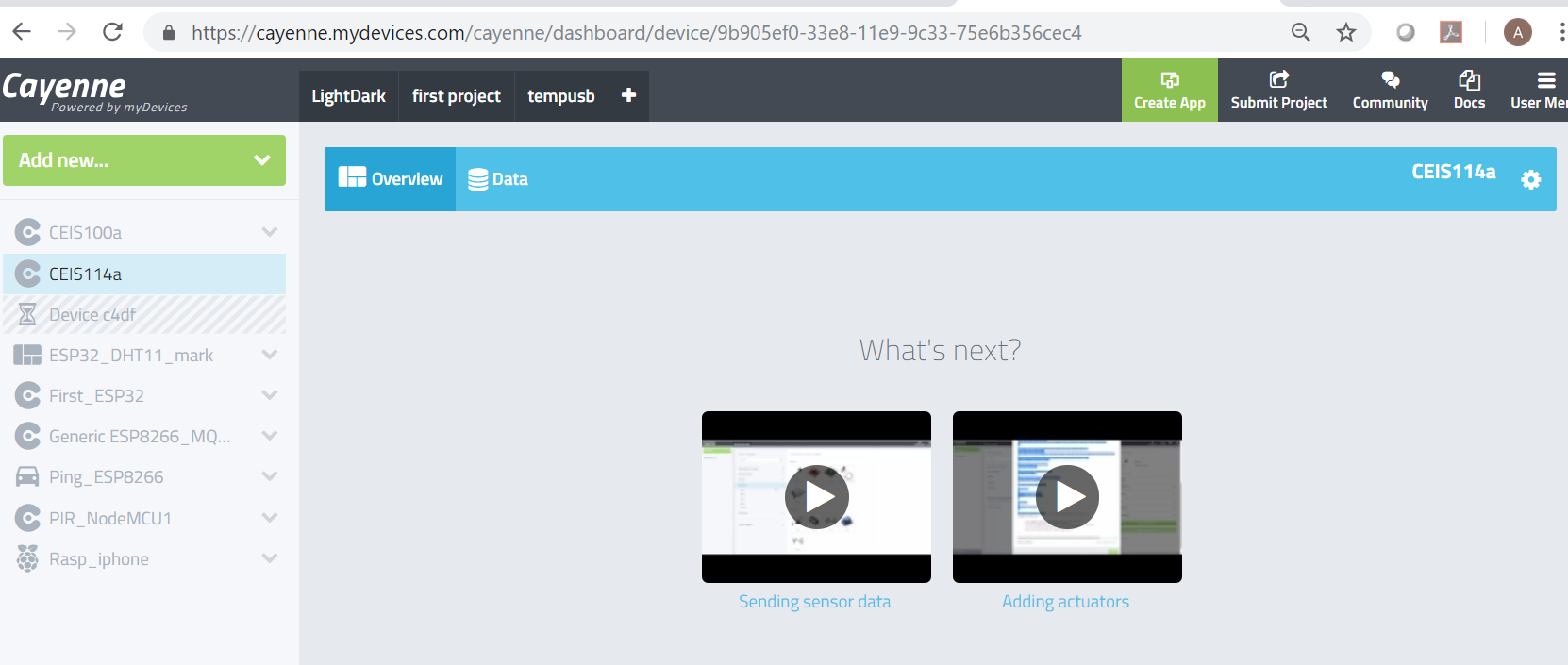
ONOFF = getValue.asInt();

digitalWrite(LED0, ONOFF);

//digitalWrite(LED0,getValue.asInt());

}

**Step 8:** Upload the code from Figure 7 to your ESP32 and when successfully finish uploading then Cayenne dashboard will be active showing the following window as shown in Figure 8.

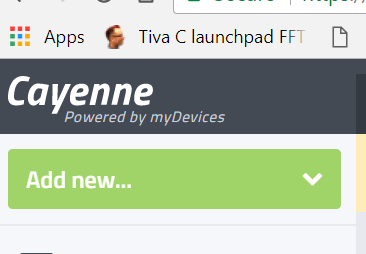


**Figure 8**

**Step 9:** Add your BLUE LED and other connected sensors/actuators. Follow the following example:

**Example of adding** **LED0** defined in our code and connected to our hardware board is as follows:

1. On Cayenne Dashboard Select: **Add new**



**Figure 9**

1. **Add New ==> Select Custom Widget** and fill in the information similar to your code. Most important is the **channel number** used for your actuator (for example I used **channel number 1** when creating a widget for LED0 to match the channel number used in the program for LED0 (as shown in Figure 10):

**CAYENNE\_IN(1)**

**{**

**digitalWrite(LED0,getValue.asInt());**

**}**

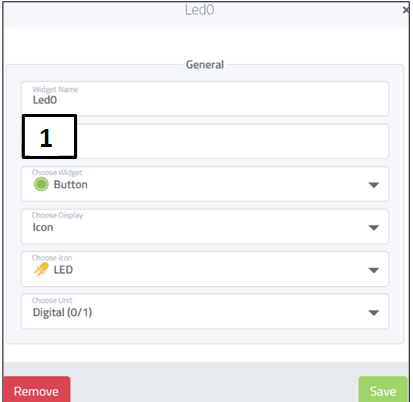
****

Figure 10

1. If everything goes well then you will see your active Cayenne Dashboard with the two LEDs as well as the temperature and humidity as shown below:

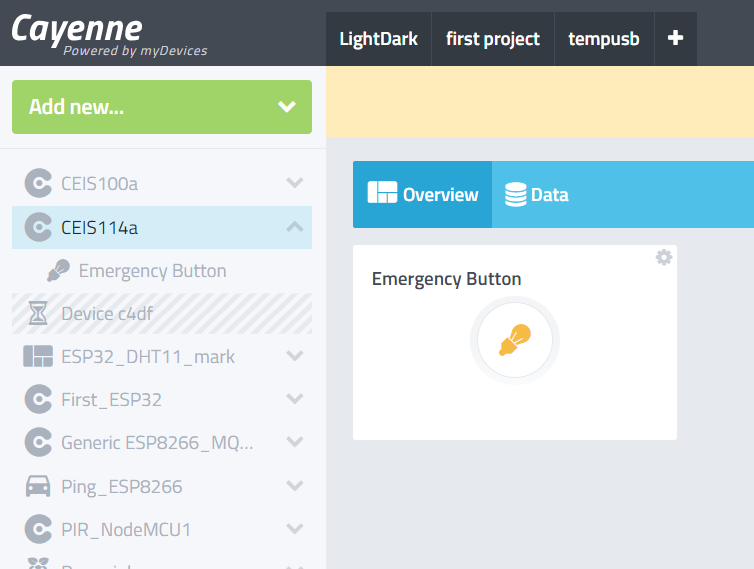


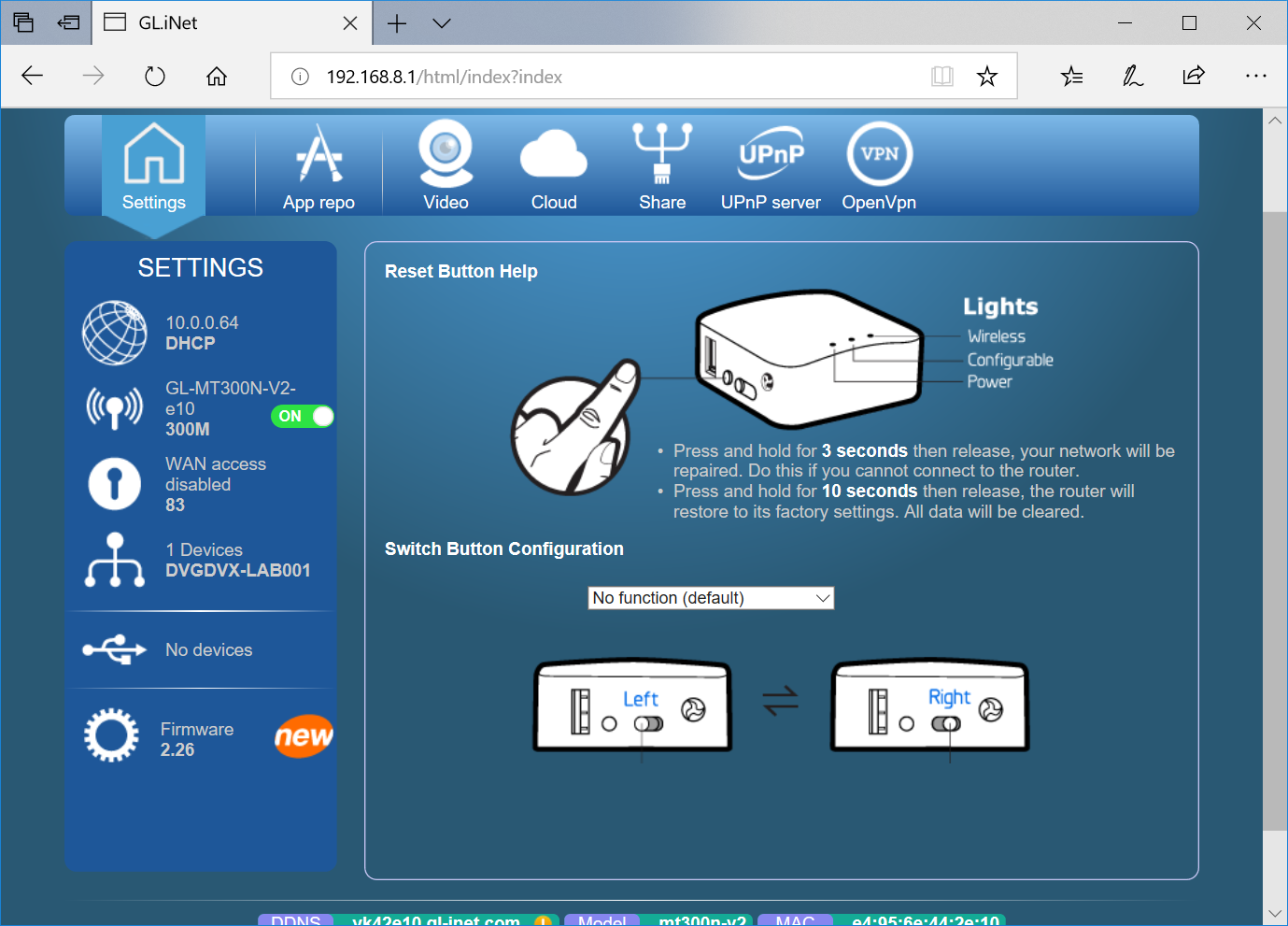
Figure 11

1. Test your LED by clicking on the dashboard buttons (Emergency Button).
2. Now you should see the flashing blue light and the two red lights are turned ON (or flashing)
3. If you have access to a smart phone you can download the MyDevices Cayenne app. and you can control your dashboard from your smart phone as well as from the **MyDevices.com** webpage in your laptop browser

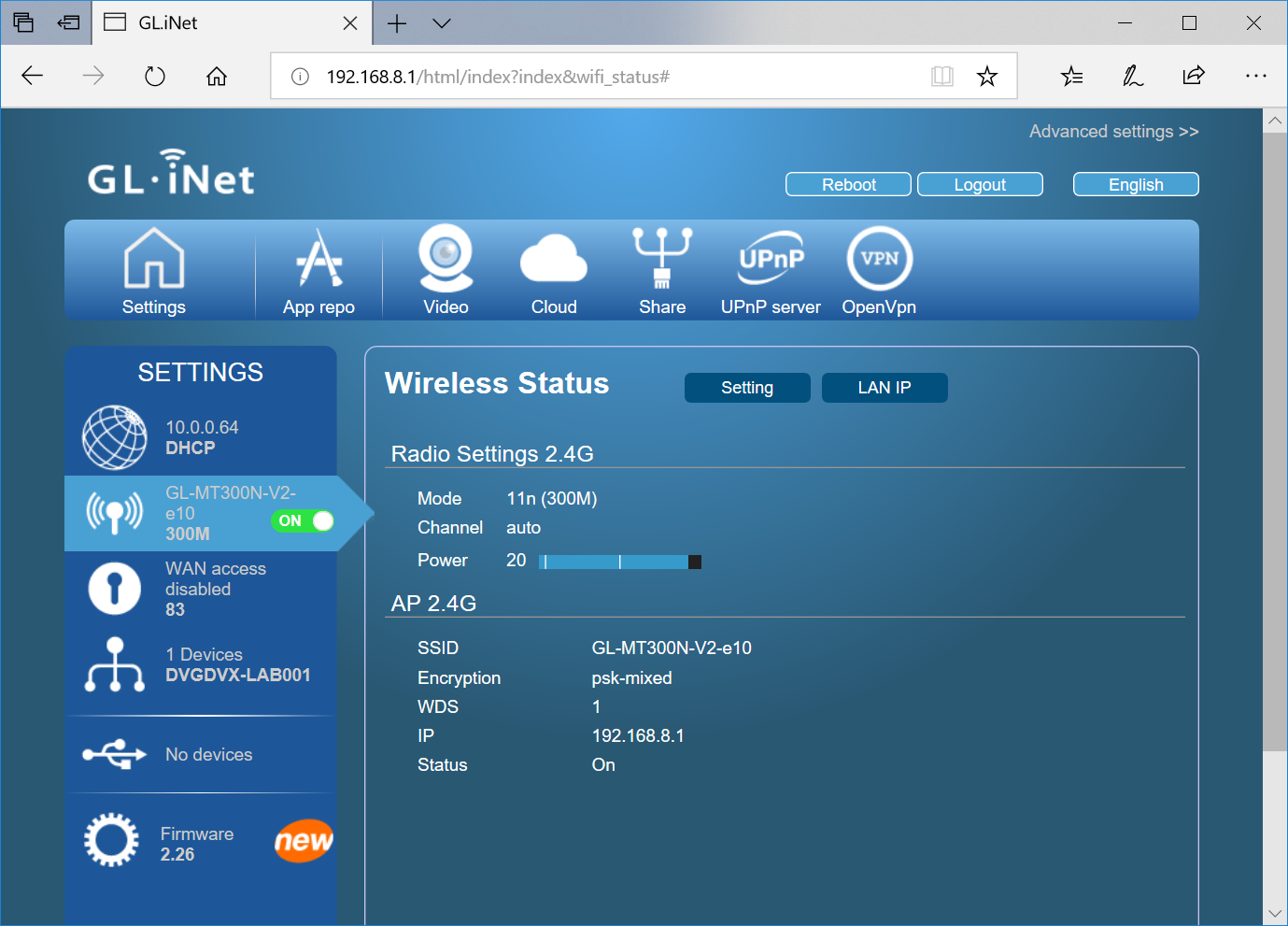
**Appendix A**

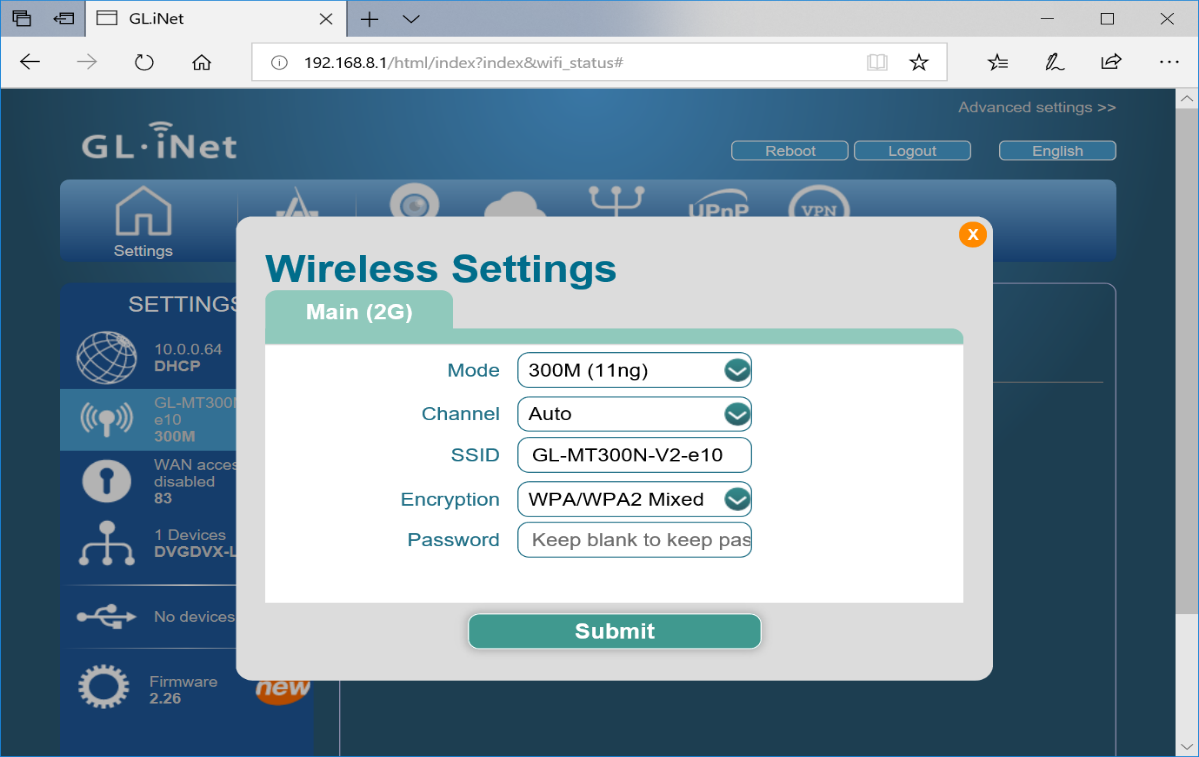
**Steps to setup your Mini Router**

1. Using an Ethernet cable connect your mini router WAN port to your home router and power your mini router up using the Micro USB cable provided.
2. Using your PC connect to your mini router using the default SSID (**GL-XXXXX-xxx**) written on the router box and the default password (**goodlife**)
3. Go to your browser (google Chrome did not work for me) e.g. explorer and type: **http://192.168.8.1**
4. Enter **a** password when asked
5. Choose language and time zone (in our case we use: ***English*** for language, and ***America/Chicago*** for time zone.)
6. Now you will be presented with the following window:

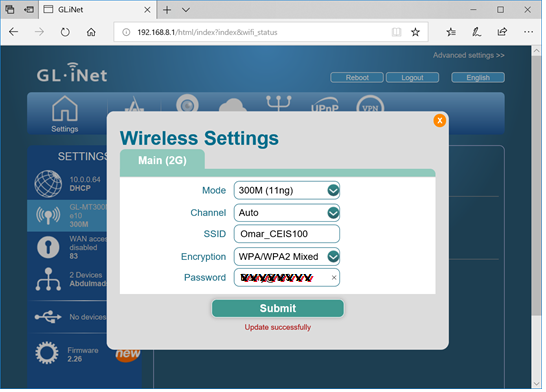


1. Click on wireless logo  then this window will open:



1. **Click on setting**: The following window will open where you can enter your own SSID and PW:
2. Change the SSID and Password to your own:

To be able to connect to your own mini router in class, I suggest that you use your name as part of the mini router’s SSID. In my case; I used **Omar\_CEIS100** for my SSID (see below), also choose a password (minimum of 8 characters)



1. Press submit. You should see Update successfully.
2. Testing your newly configure router network: use your laptop or your smart phone to check the new configuration. Here is a snap shot of my newly configured mini router.

