單元名稱(LAB I2605)

Name : 劉泰佑 Student ID : C100E030

1.1 Objective

To be able to control LED color from our website through ThingSpeak by using its API.

1.2 Working theory

ESP8266 is a MCU which is mainly used in IoT appliances as the main board or the brain of the system. ESP8266 is equipped with wifi module which can be used to send data through the internet once the board is connected to the internet. In this case we tried to use ESP8266 to control each LED color where one LED has 4 pins, ground, red pin, green pin, and blue pin that are connected to the ESP8266. We use ThingSpeak as a bridge where we can control the LED color from the website. Where in the website there are 4 buttons the red button, green button, blue button, and turn off button, all of them button are use to change the color of the LED. So by using ThingSpeak we put the write API code in our website so that we can send data to the ESP8266 through the ThingSpeak, also we put our write API, read API, and channel id in the ESP8266 program files so that we can control and monitor the ESP8266 status.

1.3 Experimental device and components)

- ESP8266,
- DHT22 Sensor,
- Thonny IDE,
- ThingSpeak,
- Access Point.

1.4 Procedure

Step 1

Open ThinkSpeak.com.

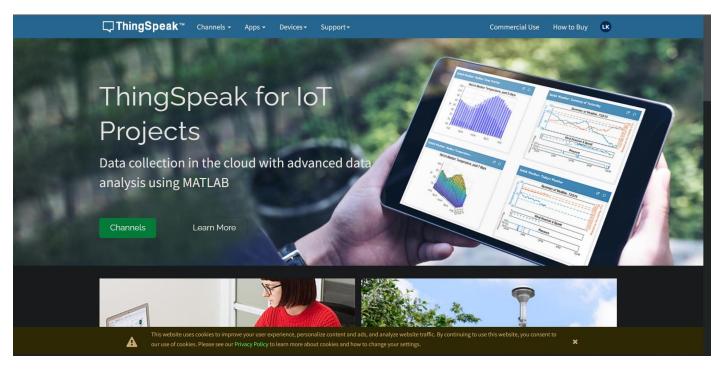


Fig. 1.4-1

Create new account.

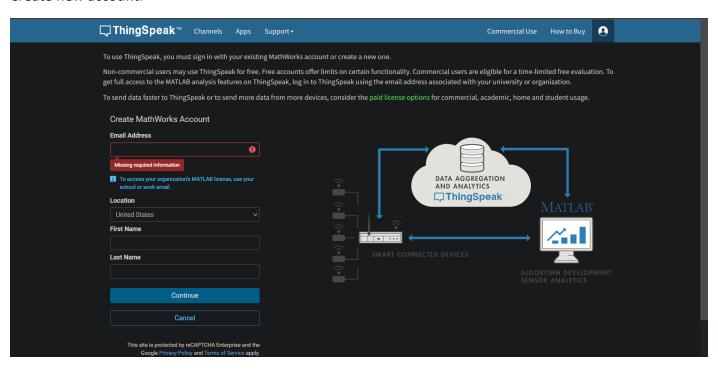


Fig. 1.4-2

Verify your account from your email inbox.

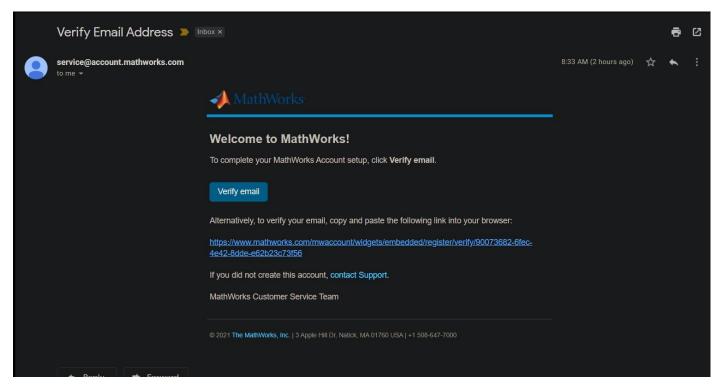


Fig. 1.4-3

Step 4

Create new channel. We use 4 field here, the first field we use to manage all the command to the ESP8266 from the website. And the last three field are for the red, green, and blue LED status which is send my the ESP8266. We add the last three field so that we can monitor our LED status through ThingSpeak so that we can integrate it to our website.

☐ ThingSpeak™	Channels - A	ops - Devices - Sup	port- Commercial Use How to Buy LK		
Channel ID: 1688112	Author: mwa0000026016481				
Private View Public View	w Channel Settir	ngs Sharing API Ke	ys Data Import / Export		
Channel Settings Help					
Percentage complete	30%		Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for		
Channel ID	1688112		status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.		
Name	ESP8266 - Web Contr	rolled LED	Channel Settings		
Description			Percentage complete: Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.		
Field 1	Command	2	Channel Name: Enter a unique name for the ThingSpeak channel.		
Field 2	Red LED		Description: Enter a description of the ThingSpeak channel.		
			 Field#: Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields. 		
Field 3			Metadata: Enter information about channel data, including JSON, XML, or CSV data.		
Field 4	Blue Led		Tags: Enter keywords that identify the channel. Separate tags with commas.		
			 Link to External Site: If you have a website that contains information about your ThingSpeak channel, specify the URL. 		
Field 5			Show Channel Location:		
Field 6			 Latitude: Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072. 		
Field 7			 Longitude: Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275. 		
Field 8			 Elevation: Specify the elevation position meters. For example, the elevation of the city of London is 35.052. 		

Fig. 1.4-4

Copy both your API keys and your channel ID to your program.

↓ThingSpeak™ Channels - Apps - Devices - Suppo	rt+ Commercial Use How to Buy LK
ESP8266 - Web Controlled LED Channel ID: 1688112 Author: mwa0000026016481 Access: Public	
Private View Public View Channel Settings Sharing API Keys	Data Import / Export
Write API Key Key W2P1P2DHF483WBVK Generate New Write API Key Read API Keys Key QN4C69GIT079VIXB	Help API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel. API Keys Settings • Write API Key. Use this key to write data to a channel, if you feel your key has been compromised, click Generate New Write API Key. • Read API Keys: Use this key to allow other people to view your private channel feeds and charts. Click Generate New Read API Key to generate an additional read key for the channel. • Note: Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel. API Requests
Note	Write a Channel Feed GET https://api.thingspeak.com/update?api_key-NZP1P2OHF483WBVX&field
Save Note Delete API Key Add New Read API Key	Read a Channel Feed GET https://api.thingspeak.com/channels/1688112/feeds.json?results=2 Read a Channel Field
	GET https://api.thingspeak.com/channels/1688112/fields/1.json?result Read Channel Status Updates

Fig. 1.4-5

Create a website to control and monitor the LED colors, in this case we use HTML language to make the website and use Visual Studio Code IDE to create and edit the website.

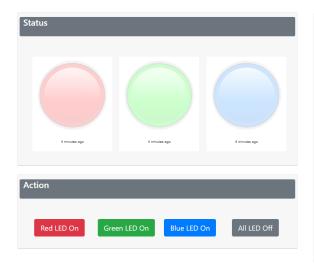
```
| Part | Section | Section
```

Fig. 1.4-6

Step 7

After some editing in Visual Studio Code IDE our website should look like this.

LED Control Panel



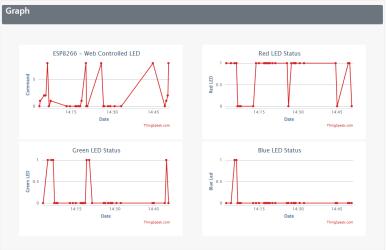


Fig. 1.4-7

Then what do next is to implemented code and API to our ESP8266 code.

```
1 import dht,machine
2 from machine import Pin
3 import urequests as requests
4 import ujson
5 import time, network
 7 SSID="Qing Shuo"
8 PASSWD="Staycation100%_"
9 W_APIKey='W2P1P2DHF483W8VK'
10 R_APIKey='QN4C69GIT079VIXB'
11 Channel_Id='1688112'
12 # Fill in your router's ssid and password here.codey.wifi.start('wifi_ssid', 'password')
14 sta_if = network.WLAN(network.STA_IF)
15
16 if not sta_if.isconnected():
17
      print('Connecting to network...')
18
      sta_if.active(True)
19
      sta_if.connect(SSID, PASSWD)
      # 等一下它連接
20
21
      while not sta_if.isconnected():
22
          pass
23 print('Network connected!')
24 print(sta_if.ifconfig())
25
26 ledR = Pin(0, Pin.OUT)
27 ledG = Pin(5, Pin.OUT)
28 ledB = Pin(4, Pin.OUT)
29
30 rstat = 0
31 gstat = 0
```

Fig. 1.4-8

1.5 Results

Step 1

After making sure everything are correct, then we just try it in real time. First open up your web control and assembly and run your ESP8266

LED Control Panel

劉泰佑 C100E030

Red LED Status

Blue LED Status



Fig. 1.5-1.1

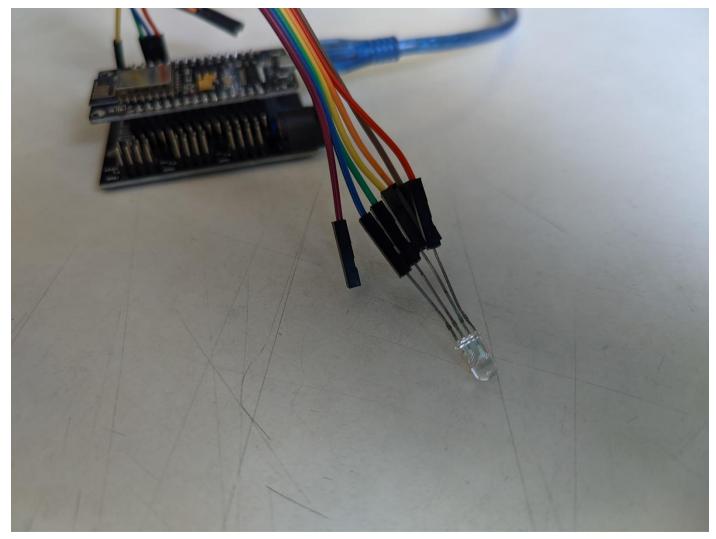


Fig. 1.5-1.2

Then we try to turn on the red LED using the button and we match the LED status on our web and real time. Also we can see the command input in the graph.

LED Control Panel



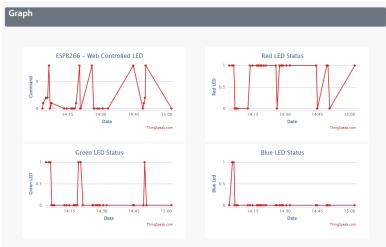


Fig. 1.5-2.1

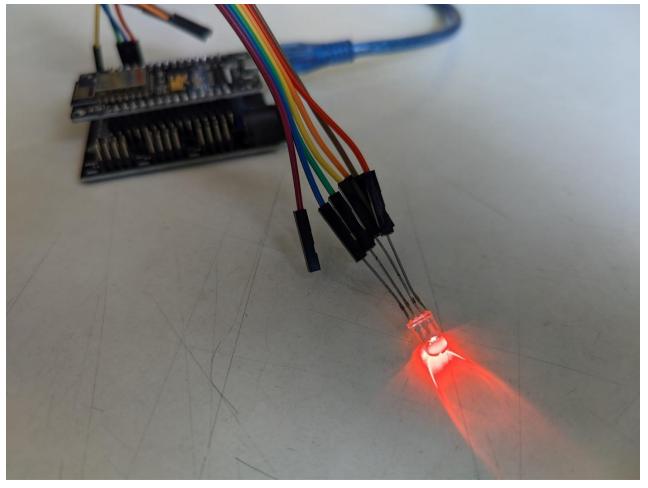


Fig. 1.5-2.2

Step 3

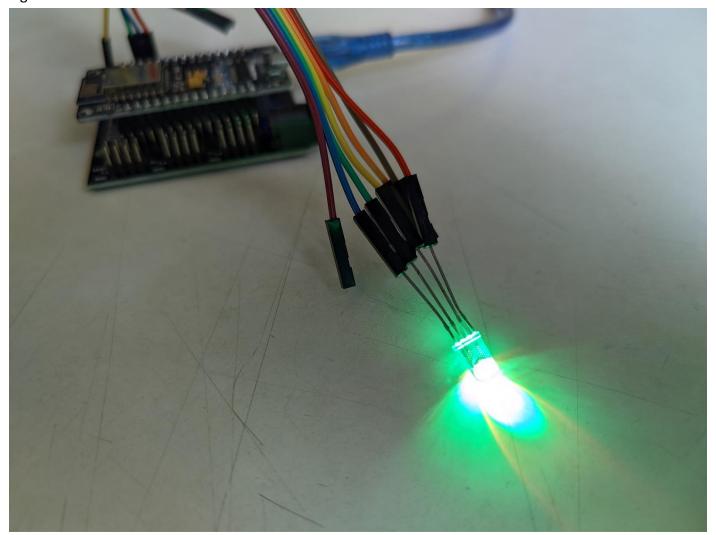
Then we try to turn on green LED

LED Control Panel





Fig. 1.5-3.1



Step 4

Next we try to turn on the blue LED

LED Control Panel



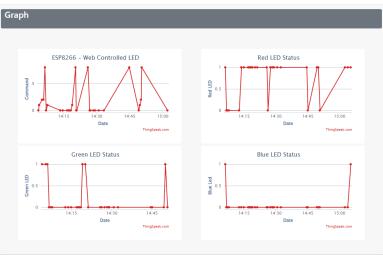


Fig. 1.5-4.1

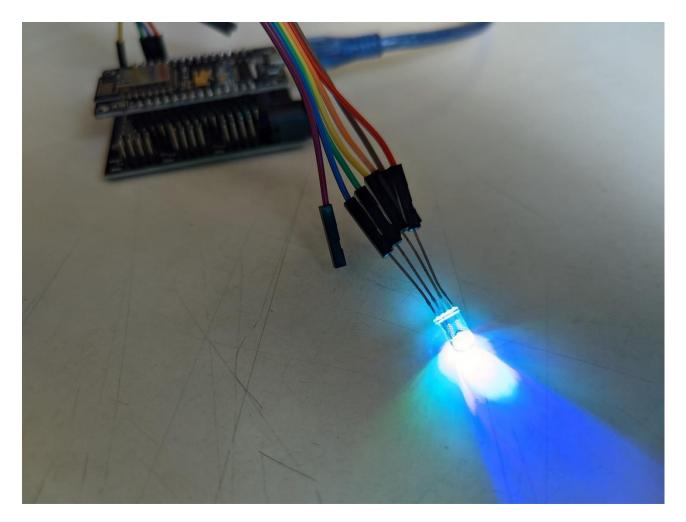


Fig. 1.5-4.2

Finally after we succeed to turn on each LED individually, we try to turn off all colors

LED Control Panel 劉泰佑 C100E030





Fig. 1.5-5.1

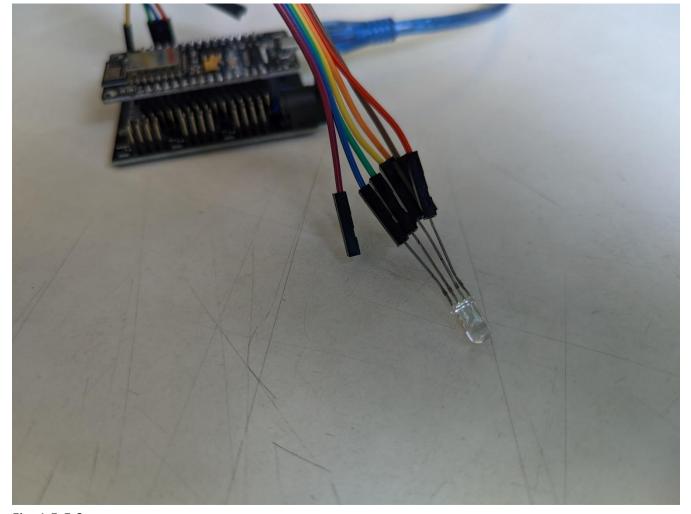


Fig. 1.5-5.2

1.6 Conclusion and Discussion

In conclusion, setting up web controlled LED using ESP8266 is pretty simple, in the process we find some error but we manage to fix it. All you have to do is to make the website to control the LED, ESP8266 with RGB LED attach to it, ThingSpeak, and internet access. We have to insert our read and write API and our channel ID into the ESP8266 program, and also we insert write API to our website. And we should be able to successfully control aur LED through website.