

รายงาน

เรื่อง Web Embedded สำหรับ Setup/Configuration

จัดทำโดย

นาย ยศกร รัตนารักษ์พงษ์ รหัส 61027203

เสนอ

ดร. ฝิน จัตรแก้วมณี

คณะ เทคโนโลยีสารสนเทศ

สาขา วิศวกรรมคอมพิวเตอร์

วิชา CPE 405

มหาวิทยาลัยศรีปทุม

คู่มือติดตั้ง Web Embedded

- มีการเสียบจ่ายไฟให้กับอุปกรณ์
- ได้มีการที่เราได้เชื่อมต่อกับอุปกรณ์ผ่าน Wi-Fi ใน Smartphone หรือ Tablet ที่มี Internet



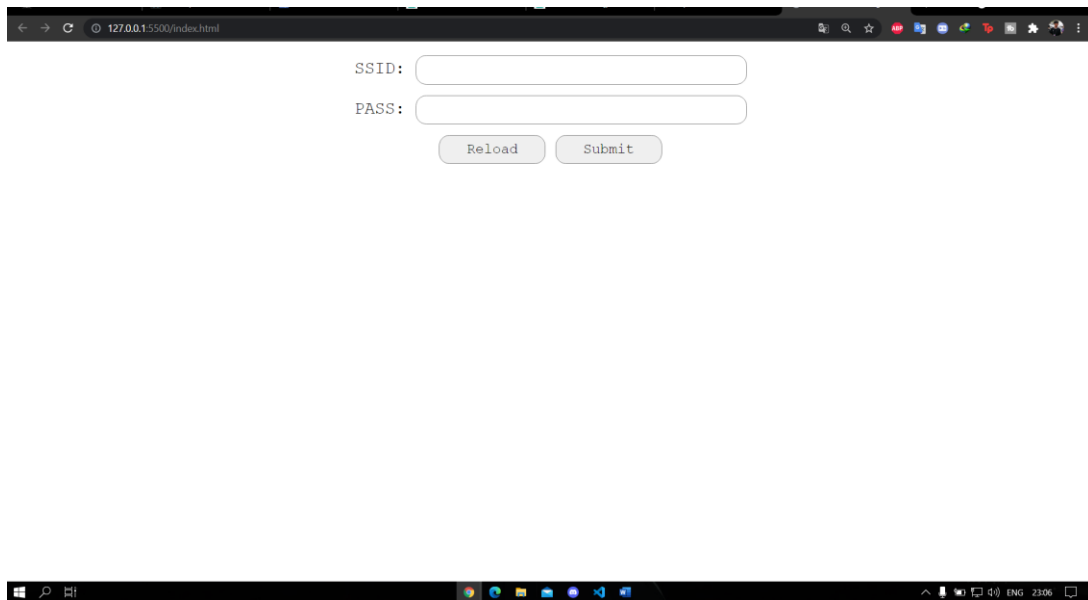
- ส่วนหน้าเข้าไปใน IP address /apSetup

ในส่วนหน้านั้นผมยังเข้าไปไม่ได้เพราะผมไม่มีพอร์ตในการเสถ้าเข้าได้มันจะประมาณนี้ครับ
ตัวอย่าง.

- เข้า IP address 192.168.4.1/apsetup

```
.....
Fail To Connect..
AccessPoint ssid: SPU_2G-[A4F544BF713C]
IP address (AccessPoint Mode): 192.168.4.1
HTTP server started
```

- อันนี้จะเป็นในส่วนหน้า SSID และ PASS ของ Wi-Fi ที่เราจะใช้



127.0.0.1:5500/index.html

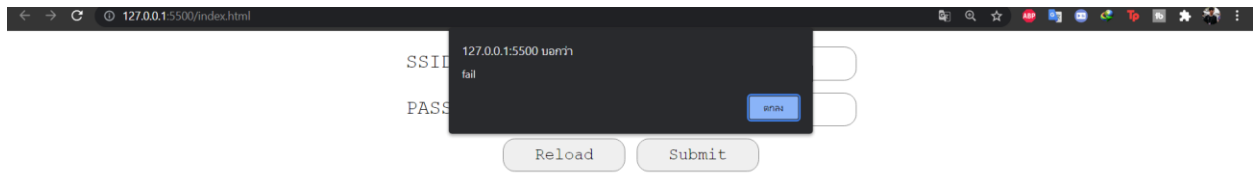
SSID:

PASS:

Reload Submit

- ถ้าเวลาเราจะกด Summit มันจะแจ้งเตือนมาว่า Sending Success

แต่ตอนนี้ของผมนั้นยังขึ้น Fail เพราะยังไม่ได้ทำการเชื่อมต่อกันในทางบอร์ด ESP32



- ในการ Setup ตัว Wi-Fi

SSID:

PASS:

- จะมีการ reboots borad 1 ครั้ง

ตัวอย่าง.

Connection Success..

IP address(STA mode): 192.168.1.51

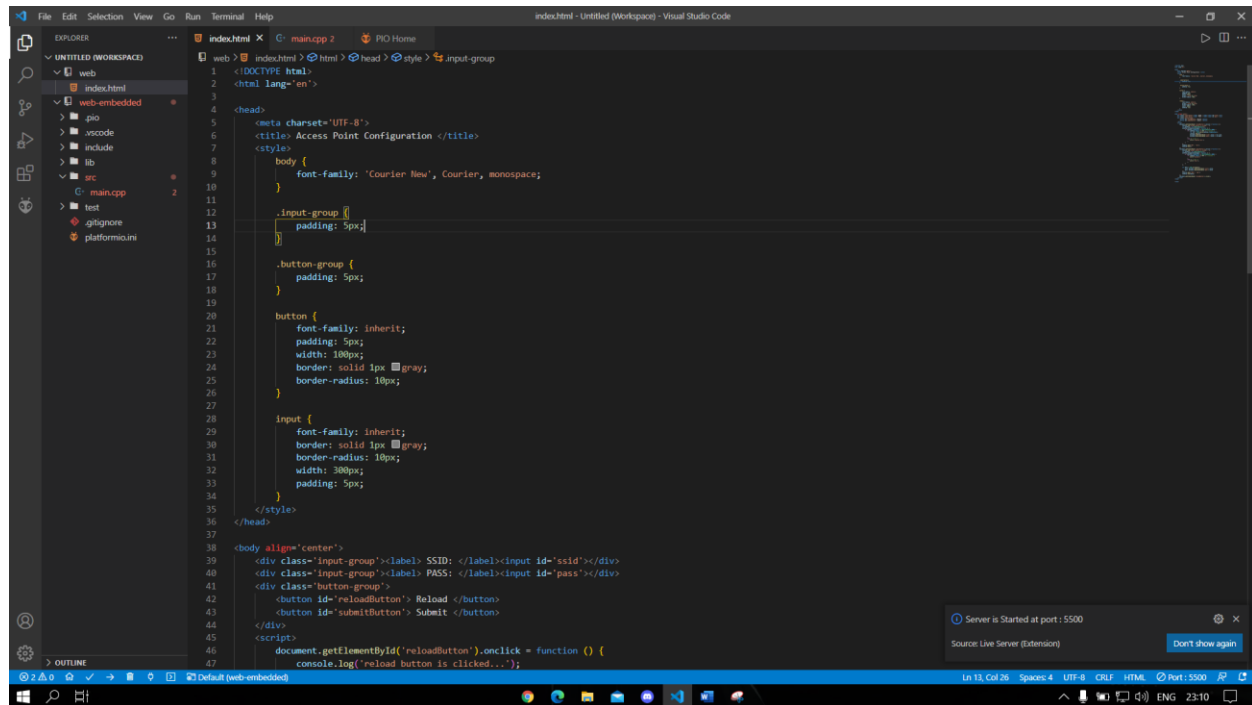
MDNS responder started

AccessPoint ssid: SPU_2G-[A4F544BF713C]

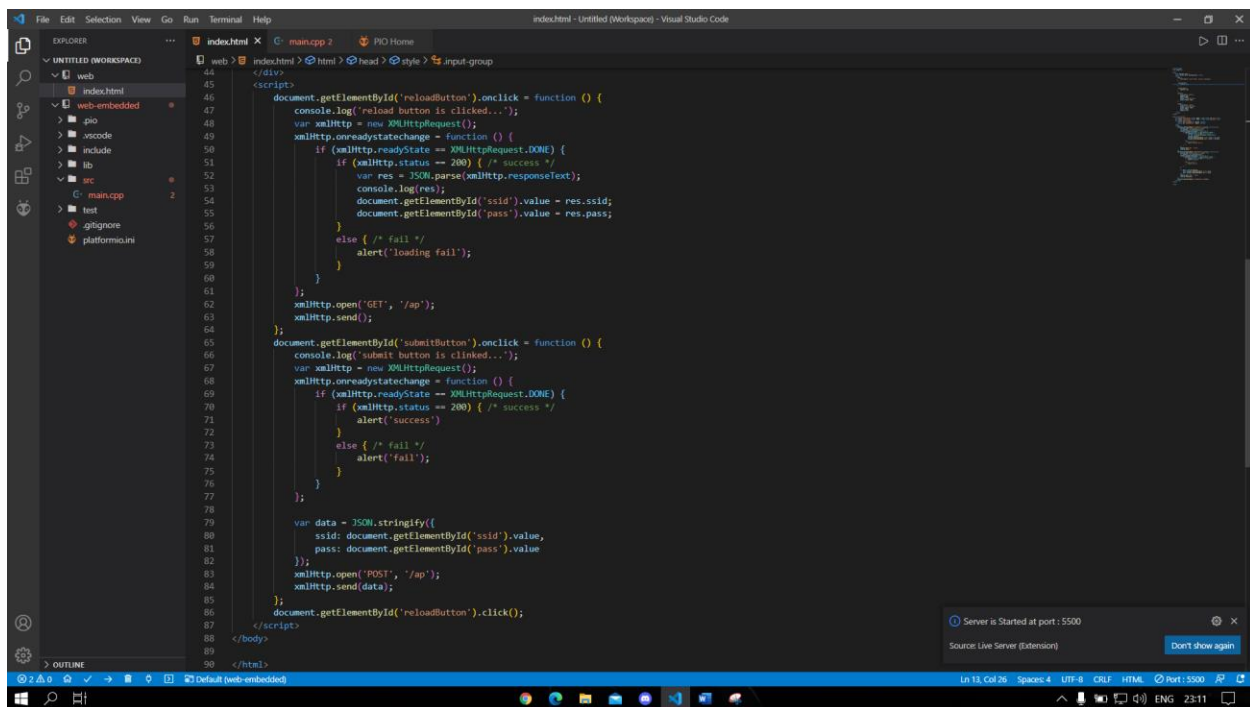
IP address(AccessPoint Mode): 192.168.4.1

HTTP server started

Code HTML

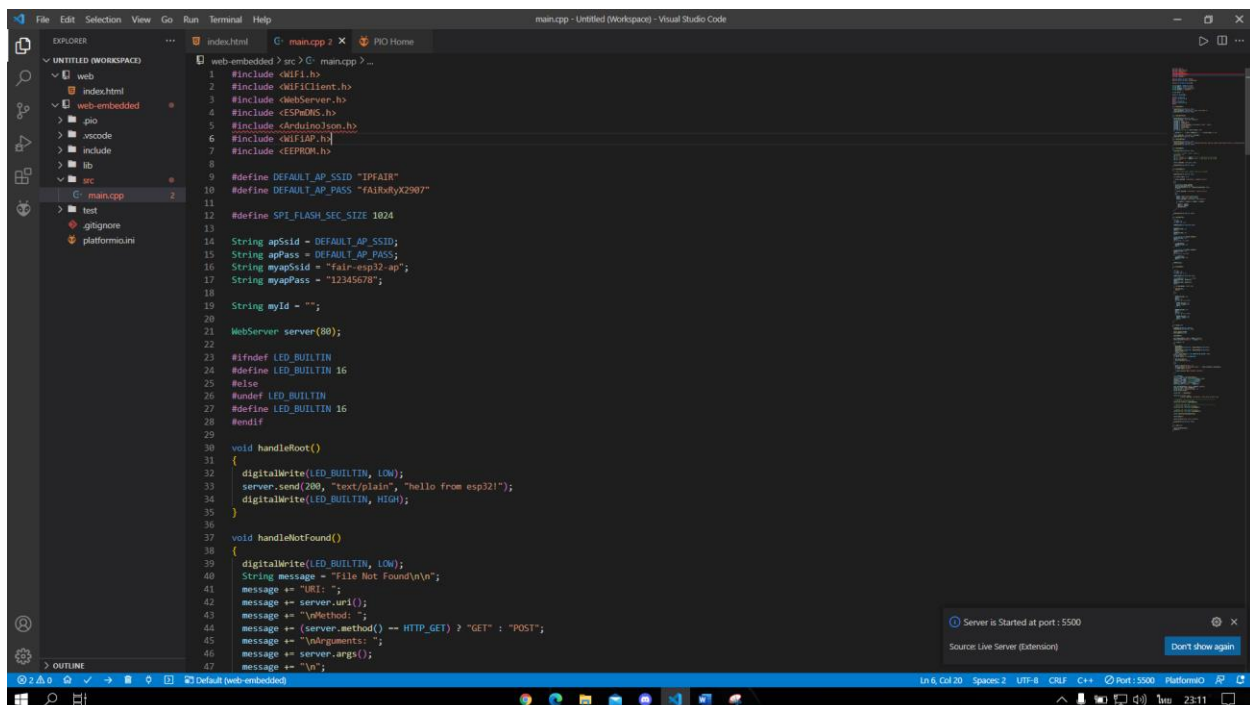


```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <title> Access Point Configuration </title>
7   <style>
8     body {
9       font-family: 'Courier New', Courier, monospace;
10    }
11
12    .input-group {
13      padding: 5px;
14    }
15
16    .button-group {
17      padding: 5px;
18    }
19
20    button {
21      font-family: inherit;
22      padding: 5px;
23      width: 100px;
24      border: solid 1px #gray;
25      border-radius: 10px;
26    }
27
28    input {
29      font-family: inherit;
30      border: solid 1px #gray;
31      border-radius: 10px;
32      width: 300px;
33      padding: 5px;
34    }
35  </style>
36 </head>
37
38 <body align="center">
39   <div class="input-group"><label> SSID: </label><input id="ssid"></div>
40   <div class="input-group"><label> PASS: </label><input id="pass"></div>
41   <div class="button-group">
42     <button id="reloadButton"> Reload </button>
43     <button id="submitButton"> Submit </button>
44   </div>
45   <script>
46     document.getElementById('reloadButton').onclick = function () {
47       console.log('reload button is clicked...');
```



```
44 </div>
45
46 document.getElementById('reloadButton').onclick = function () {
47     console.log('reload button is clicked...');
48     var xmlhttp = new XMLHttpRequest();
49     xmlhttp.onreadystatechange = function () {
50         if (xmlhttp.readyState == XMLHttpRequest.DONE) {
51             if (xmlhttp.status == 200) { /* success */
52                 var res = JSON.parse(xmlhttp.responseText);
53                 console.log(res);
54                 document.getElementById('ssid').value = res.ssid;
55                 document.getElementById('pass').value = res.pass;
56             }
57             else { /* fail */
58                 alert('loading fail');
59             }
60         }
61     };
62     xmlhttp.open('GET', '/ap');
63     xmlhttp.send();
64 }
65
66 document.getElementById('submitButton').onclick = function () {
67     console.log('submit button is clicked...');
68     var xmlhttp = new XMLHttpRequest();
69     xmlhttp.onreadystatechange = function () {
70         if (xmlhttp.readyState == XMLHttpRequest.DONE) {
71             if (xmlhttp.status == 200) { /* success */
72                 alert('success');
73             }
74             else { /* fail */
75                 alert('fail');
76             }
77         }
78     };
79
80     var data = JSON.stringify({
81         ssid: document.getElementById('ssid').value,
82         pass: document.getElementById('pass').value
83     });
84     xmlhttp.open('POST', '/ap');
85     xmlhttp.send(data);
86     document.getElementById('reloadButton').click();
87 }
88 </body>
89 </html>
```

Code Main ในส่วน IO

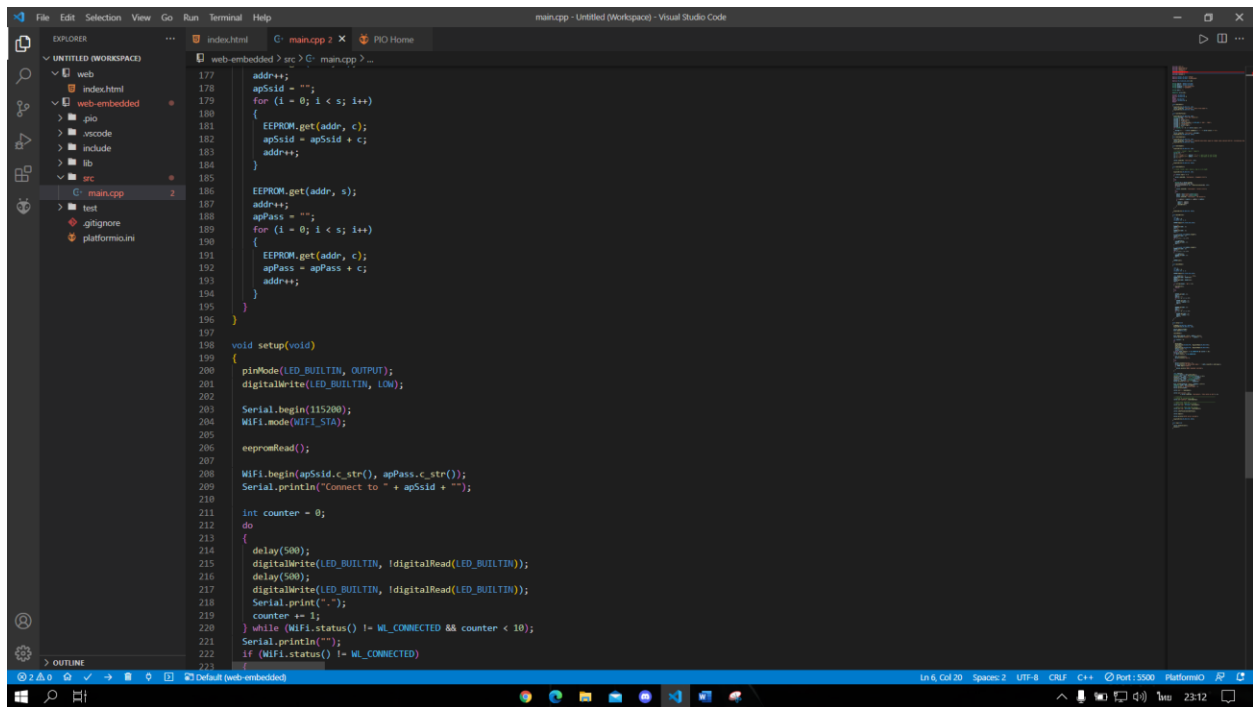


```
1 #include <stdio.h>
2 #include <WiFiClient.h>
3 #include <WebServer.h>
4 #include <ESPmDNS.h>
5 #include <ArduinoJson.h>
6 #include <WiFiAP.h>
7 #include <EEPROM.h>
8
9 #define DEFAULT_AP_SSID "IPFAIR"
10 #define DEFAULT_AP_PASS "fAIRbYx2907"
11
12 #define SPI_FLASH_SEC_SIZE 1024
13
14 String apSsid = DEFAULT_AP_SSID;
15 String apPass = DEFAULT_AP_PASS;
16 String myapSsid = "fAIR-esp32-ap";
17 String myapPass = "12345678";
18
19 String myId = "";
20
21 WebServer server(80);
22
23 #ifdef LED_BUILTIN
24 #define LED_BUILTIN 16
25 #else
26 #define LED_BUILTIN
27 #endif
28
29 void handleRoot()
30 {
31     digitalWrite(LED_BUILTIN, LOW);
32     server.send(200, "text/plain", "hallo from esp32!");
33     digitalWrite(LED_BUILTIN, HIGH);
34 }
35
36 void handleNotFound()
37 {
38     digitalWrite(LED_BUILTIN, LOW);
39     String message = "File Not Found\n\n";
40     message += "URI: ";
41     message += server.uri();
42     message += "\nMethod: ";
43     message += (server.method() == HTTP_GET) ? "GET" : "POST";
44     message += "\nArguments: ";
45     message += server.args();
46     message += "\n";
47 }
```

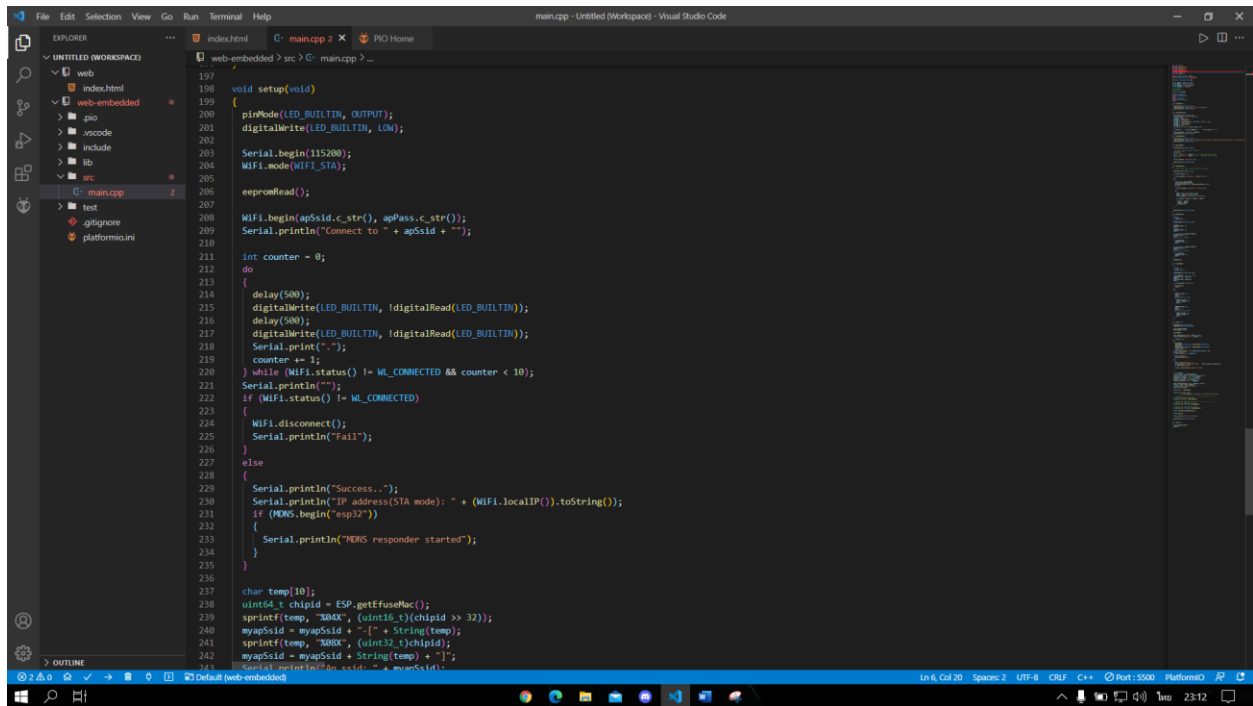


```
88 else
89 {
90     String str = server.arg(0);
91     StaticJsonDocument<100> doc;
92     DeserializationError err = deserializeJson(doc, str);
93     if (err)
94     {
95         server.send(500, "text/plain", "server error");
96     }
97     else
98     {
99         apSsid = doc["ssid"].as<String>();
100         apPass = doc["pass"].as<String>();
101         server.send(200, "text/plain", "OK Success");
102         if (_apSsid != apSsid || _apPass != apPass)
103         {
104             apSsid = _apSsid;
105             apPass = _apPass;
106             eepromWrite();
107         }
108     }
109 }
110 }
111 }
112 }
113 }
114 }
115 void eepromWrite()
116 {
117     char c;
118     int addr = 0;
119     unsigned char s, i;
120     EEPROM.begin(SPI_FLASH_SEC_SIZE);
121     for (i = 0; i < s; i++)
122     {
123         c = '0';
124         EEPROM.put(addr, c);
125         addr++;
126         c = '5';
127         EEPROM.put(addr, c);
128         addr++;
129     }
130     s = (unsigned char)apSsid.length();
131     EEPROM.put(addr, s);
132     addr++;
133     for (i = 0; i < s; i++)
```

```
131     s = (unsigned char)apPass.length();
132     EEPROM.put(addr, s);
133     addr++;
134     for (i = 0; i < s; i++)
135     {
136         c = apSsid[i];
137         EEPROM.put(addr, c);
138         addr++;
139     }
140     s = (unsigned char)apPass.length();
141     EEPROM.put(addr, s);
142     addr++;
143     for (i = 0; i < s; i++)
144     {
145         c = apPass[i];
146         EEPROM.put(addr, c);
147         addr++;
148     }
149     EEPROM.end();
150 }
151 }
152 }
153 void eepromRead()
154 {
155     char c;
156     int addr = 0;
157     unsigned char s, i;
158     EEPROM.begin(SPI_FLASH_SEC_SIZE);
159     char header[3] = { ' ', ' ', '\0' };
160     EEPROM.get(addr, header[0]);
161     addr++;
162     EEPROM.get(addr, header[1]);
163     addr++;
164     if (strcmp(header, "05") != 0)
165     {
166         eepromWrite();
167         return;
168     }
169     else
170     {
171         EEPROM.get(addr, s);
172     }
173 }
```

```
177     addr++;
178     apSsid = "";
179     for (i = 0; i < s; i++)
180     {
181         EEPROM.get(addr, c);
182         apSsid = apSsid + c;
183         addr++;
184     }
185
186     EEPROM.get(addr, s);
187     addr++;
188     apPass = "";
189     for (i = 0; i < s; i++)
190     {
191         EEPROM.get(addr, c);
192         apPass = apPass + c;
193         addr++;
194     }
195 }
196
197 void setup(void)
198 {
199     pinMode(LED_BUILTIN, OUTPUT);
200     digitalWrite(LED_BUILTIN, LOW);
201
202     Serial.begin(115200);
203     WiFi.mode(WIFI_STA);
204
205     eepromRead();
206
207     WiFi.begin(apSsid.c_str(), apPass.c_str());
208     Serial.println("Connect to " + apSsid + "");
209
210     int counter = 0;
211     do
212     {
213         delay(500);
214         digitalWrite(LED_BUILTIN, digitalRead(LED_BUILTIN));
215         delay(500);
216         digitalWrite(LED_BUILTIN, digitalRead(LED_BUILTIN));
217         Serial.print(".");
218         counter += 1;
219     } while (WiFi.status() != WL_CONNECTED && counter < 10);
220     Serial.println("");
221     if (WiFi.status() != WL_CONNECTED)
222     {
```



```
223     {
224         WiFi.disconnect();
225         Serial.println("Fail");
226     }
227     else
228     {
229         Serial.println("Success...");
230         Serial.println("IP address(STA mode): " + WiFi.localIP().toString());
231         if (MDNS.begin("esp32"))
232         {
233             Serial.println("MDNS responder started");
234         }
235     }
236
237     char temp[10];
238     uint64_t chipid = ESP.getFirmwareVer();
239     sprintf(temp, "%04X", (uint16_t)(chipid >> 32));
240     myapSsid = myapSsid + "-" + String(temp);
241     sprintf(temp, "%08X", (uint32_t)chipid);
242     myapSsid = myapSsid + String(temp) + "-";
243     Serial.println(myapSsid + " = myapSsid");
```

```
230 Serial.println("IP address(STA mode): " + (WiFi.localIP()).toString());
231 if (Mdns.begin("esp32"))
232 {
233     Serial.println("Mdns responder started");
234 }
235
236 char temp[10];
237 uint64_t chipid = ESP.getFirmwareVer();
238 sprintf(temp, "%04x", (uint16_t)(chipid >> 32));
239 myapSsid = myapSsid + "-" + String(temp);
240 sprintf(temp, "%04x", (uint16_t)chipid);
241 myapSsid = myapSsid + String(temp) + "-";
242 Serial.println("Ap ssid: " + myapSsid);
243
244 WiFi.softAP(myapSsid.c_str(), myapPass.c_str());
245 IPAddress myIP = WiFi.softAPIP();
246 Serial.print("IP address(AP Mode): ");
247 Serial.println(myIP);
248
249 server.on("/", handleRoot);
250
251 server.on("/inline", []()
252 {
253     { server.send(200, "text/plain", "this works as well"); });
254
255 // front-end -----
256 // display ap configuratio page
257 server.on("/apSetup", handleApSetup);
258
259 // webservices (back-end) -----
260 // return ssid, pass in json format
261 server.on("/ap", HTTP_GET, handleApGet);
262
263 // update ssid, pass from web browser
264 server.on("/ap", HTTP_POST, handleApPost);
265
266 server.onNotFound(handleNotFound);
267
268 server.begin();
269
270 Serial.println("HTTP server started");
271
272 digitalWrite(LED_BUILTIN, HIGH);
273
274
275 void loop(void)
```

ส่วนเสริมที่ต้องใช้คร่าว ๆ ที่ผมได้ลง

C/C++ 471ms
C/C++ IntelliSense, debugging, and code browsing.
Microsoft

HTML CSS Support 49ms
CSS Intellisense for HTML
ecmel

HTML Snippets
Full HTML tags including HTML5 Snippets
Mohamed Abusaid

HTML Snippets
Full HTML tags including HTML5 Snippets
Mohamed Abusaid



Live Server

🕒 5668ms

Launch a development local Server with live reload feature for st...
Ritwick Dey



PlatformIO IDE

🕒 4293ms

Professional development environment for Embedded, IoT, Ardui...
PlatformIO

