

Laboratory 6: Using the M5Stack as WiFi Client to Request a Public API Service

Objectives

1. Be able to write and understand a sketch that will make the M5Stack connect to a WiFi network.
2. Be able to write and understand a sketch that will make the M5Stack request a service from a public Application Programming Interface (API) service via the Hypertext Transfer Protocol (HTTP).

The HTTP Protocol and REST APIs

APIs stands for Application Programming Interface. It is a way that two software pieces can use to communicate over the Internet. A public API's service is available for every other software on the Internet. In this lab, we will use an API that conforms to the Representational State Transfer (REST) standard. REST is an architectural style that dictates the constraints used to create the API services. Any software can request for a service from a REST API using a request message according to the Hypertext Transfer Protocol (HTTP). The latter is an important protocol that is also used for web-browsing. When an API server (endpoint) obtains a request message from another program, it will check the request message to see if it is feasible or not. If it is, the server will form a corresponding respond message (also according to the HTTP) and send it back to the requesting program, which can further use the result in its own codebase. An endpoint is specified by a URL.

Common HTTP requests:

- **GET** - The HTTP **GET** method requests data from a specified resource, which is indicated by a URL.
- **POST** - The HTTP **POST** method sends data to be processed on a specified resource on the web server.

In this lab, we will focus on the **GET** method of the HTTP protocol.

Some common HTTP code results:

200: OK
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
500: Internal Server Error
503: Service Unavailable

WiFi access point

In this lab, if no WiFi access point is available, you might have to set your mobile phone as a WiFi hotspot. Note that, your mobile phone default SSID name will be set to your phone name which might include some special characters. This will cause M5Stack not be able to connect to your mobile hotspot. Rename your mobile hotspot SSID to a new simple name. You can add as many access points as you like. M5Stack will try to connect to the strongest signal access point. For iPhone, you might have to enable “Maximize Compatibility” in Personal Hotspot setting (see picture).



```
void setup() {
    ...
    // replace SSID_x and PASSWORD_x with your SSID and PASSWORD
    wifiMulti.addAP("SSID_1", "PASSWORD_1");
    wifiMulti.addAP("SSID_2", "PASSWORD_2");
    wifiMulti.addAP("SSID_3", "PASSWORD_3");
    ...
}
```

Connect to WiFi

wifiMulti.run() will try to connect to the access point. If the connection is success, it will return WL_CONNECTED. This method should put inside loop() and should wait for some delay before try to reconnect.

```
void loop() {
    // wait for WiFi connection
    if((wifiMulti.run() == WL_CONNECTED)) {
        HTTPClient http;
        http.begin("http://example.com/index.html"); //HTTP
        ...
    } else {
        USE_SERIAL.printf("connecting... to WiFi\n");
    }
    delay(5000);
}
```

HTTP GET and result

Use HTTPClient object to connect to the server URL. Then, call the GET() method. If the result code from GET() is non-negative integer, the URL is connected. In this lab, we only concern when the result is HTTP_CODE_OK. To get the result from GET() method, use getString().

```
void loop() {
    // wait for WiFi connection
    if((wifiMulti.run() == WL_CONNECTED)) {
        HTTPClient http;
        ...
    }
}
```

```
http.begin("http://example.com/index.html"); //HTTP
int httpCode = http.GET();
if (httpCode > 0) {
    if(httpCode == HTTP_CODE_OK) {
        String payload = http.getString();
        USE_SERIAL.println(payload);
    }
} else {
    USE_SERIAL.printf("[HTTP] GET... failed, error: %s\n",
        http.errorToString(httpCode).c_str());
}
http.end();
} else {
    USE_SERIAL.printf("connecting... to WiFi\n");
}
delay(5000);
}
```

Lab Exercises

In this laboratory, you will first set up the Arduino IDE on your laptops to properly connect to the M5Stack development kit from an example sketch ([remove unnecessary comments for readability](#)).

Task 1: M5Stack as WIFI client.

- 1.1 Connect your M5Stack to your laptop and open Arduino IDE.
- 1.2 Write a sketch to make the M5Stack kit connect to a WiFi network and request a service from a public API service via HTTP protocol. Use the example code for M5Stack from the menu item: [File > Examples > Examples for M5Core > HTTPClient > BasicHTTPClient](#) as a starting point.
- 1.3 Modify the code to connect the M5Stack to a suitable WiFi network, such as your mobile hotspot or your home WiFi. **Disable WiFi for your laptop because it might interfere with M5Stack's connection** and open the Serial Monitor the Arduino IDE for output. Verify that the M5Stack can connect to the Internet via the WiFi interface. If the connection is successful, the Serial Monitor should display the message: `"[HTTP] GET... code: "` followed by a valid HTTP response code.

Tip: You might add additional access points and may need to try toggling your laptop WiFi on/off as it could interfere with the M5stack's WiFi connection.

Task 2: Connect M5Stack to public API Service

Modify the code to make the M5Stack request a service that find a location of a host using a specified IP address from a public API. The API located at <http://ip-api.com/json/>. To find a location of the host with IP address **161.200.192.243**, you can make an HTTP request using **GET** method to the URL: `"http://ip-api.com/json/161.200.192.243"`. Observe detailed in the result, such as `"zip"`, `"lat"`, `"long"`, and `"org"` (organization).

Task 3: Press button A, B, C for different IP addresses

Depending on which button (of the three buttons) is pressed on the M5Stack, show the information of three different IP addresses to serial monitor:

- a. Button A (left-most button): **161.200.192.243**
- b. Button B (middle button): **202.47.249.7**
- c. Button C (right-most button): **203.131.212.198**

Fill out detail and answer the questions.

Lab 6: WiFi Client

Section: _____ Date: _____

Members

Name: _____ Student ID: _____

Name: _____ Student ID: _____

Name: _____ Student ID: _____

Task			Graded by
1. M5Stack as WiFi Client			
2. Connect M5Stack to public API Service			
3. Press button A, B, C for different IP addresses			
4. IP address information			
	161.200.192.243	202.47.249.7	203.131.212.198
Zip			
Lat			
Long			
Org			

5. Reflection about this lab. See assignment in **MycourseVille**