IoT exercises – Week 5

This week, we will start using the Wi-Fi in NodeMCU/ESP8266 and use our devices as a Station or Access Point or both. And we will try all the tree basic modes to allow you to build the wireless sensor network after the unit. Particularly, as mentioned in the lecture, the key idea of IoT is to get things/sensors/controllers/systems online. The wifi module will be used in this practical and throughout the rest. Another module enduser setup will be introduced and practiced as well for being more user-friendly.

Besides, it is optional for you to implement the RSSI based localization algorithm when GPS module is not available (However, it is always highly recommended that you use the bespoke module and sensor dedicated for certain applications instead of using a replacement for better reliability).

The official documentation is provided here again for your reference. Please remember, when you are building your own IoT project in the future, always refer to the bespoke built-in modules and read their documentation first, which will be helpful!!!

https://nodemcu.readthedocs.io/en/master/

The details of wifi module and enduser setup can be found in

https://nodemcu.readthedocs.io/en/master/modules/wifi/

https://nodemcu.readthedocs.io/en/master/modules/enduser-setup/

Exercise 1:

In this exercise, you will need quickly go through the official documents of wifi module for NodeMCU/ESP8266 as a station and an ap and familiarize yourself with basic functions.

==========

wifi.sta.autoconnect(1)

--autoconnect

wifi.setmode(wifi.STATION)

```
station_cfg={}
station_cfg.ssid="uopiot2020"
station_cfg.pwd="2020a202"
station_cfg.save=true
wifi.sta.config(station_cfg)
--wifi.sta.connect()
--wifi.sta.disconnect()
--manual connect and disconnect
```

==========

==========

```
wifi.setmode(wifi.SOFTAP)
ap_cfg={}
ap_cfg.ssid="yourapname"
ap_cfg.pwd="2020a202"
--the rest have default value
--check them by yourself, which is helpful to know how it works - Dalin
wifi.ap.config(ap_cfg)
```

==========

Exercise 2:

Your NodeMCU/ESP8266 needs to be connected to Internet as a station to fully function as an IoT. It is not so convenient to keep all the Wi-Fi APs' name/ssid and password fixed in your codes. An alternative is to use the NodeMCU/ESP8266 as both a Station and an AP and pick the Wi-Fi and send the password to the system through your own mobile device.

In this exercise, you will need to check the combination of Station mode and AP mode, and use the enduser for a more user-friendly interface to get your device online through the APs available to be scanned. (enduser will require the STA+AP mode.)

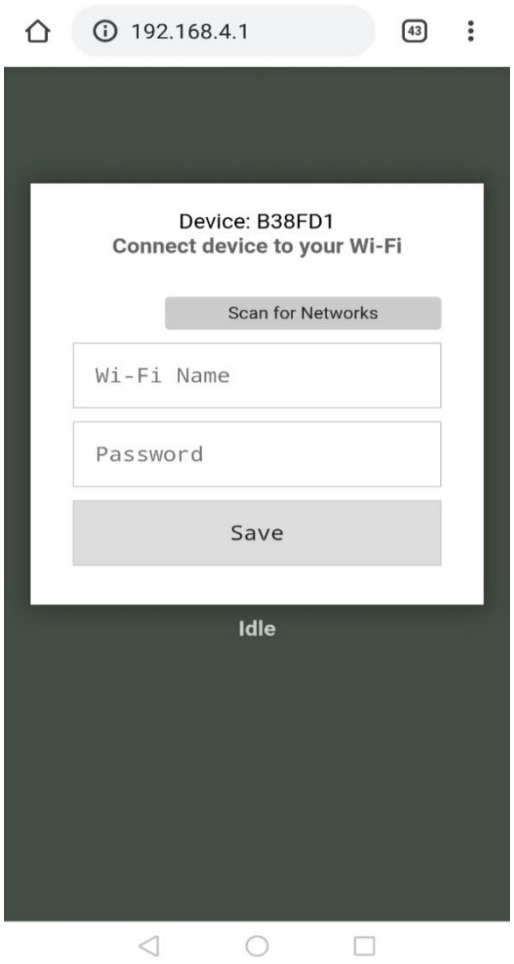
==========

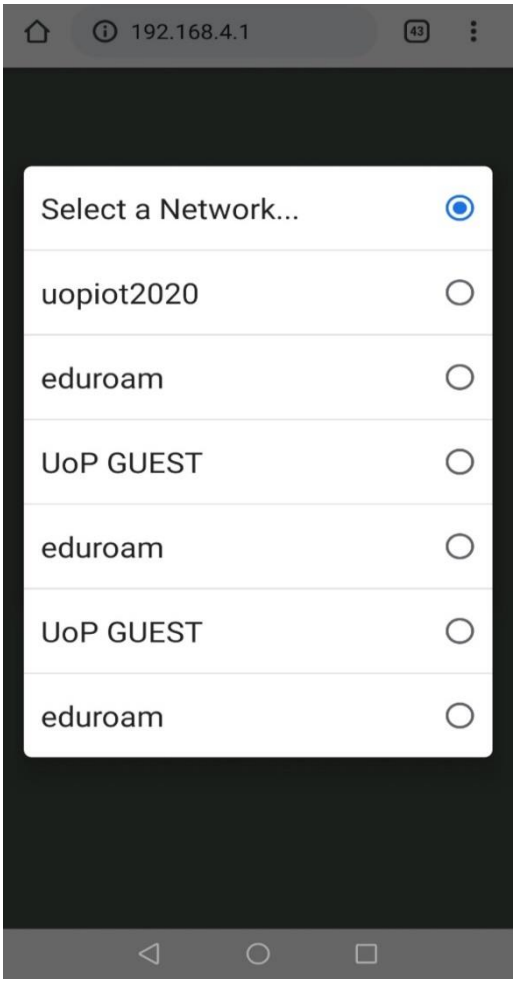
```
wifi.setmode(wifi.STATIONAP)
```

```
--you can do it manually with
--wifi.ap.config({ssid="nameyournodeMCU", pwd="yourpassword", auth=wifi.WPA2_PSK})
--enduser_setup.manual(true)
enduser_setup.start()
print("AP IP:"..wifi.ap.getip())
print("AP MAC:"..wifi.ap.getmac())
print("STA MAC:"..wifi.sta.getmac())
```

==========

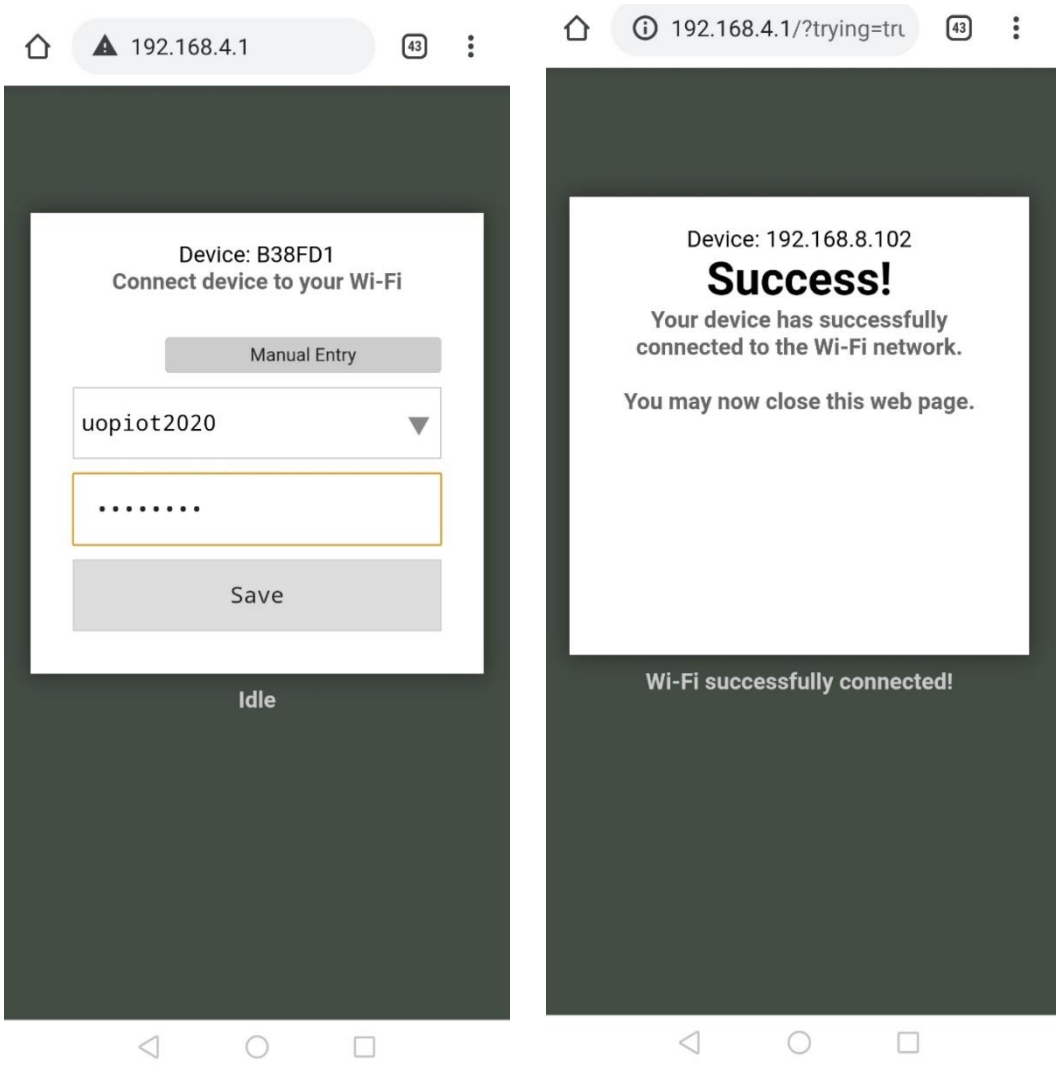
- After enduser_setup.start(), a wireless network named "SetupGadget_XXXXXX" will act as the AP for you to connect to.
- Connect to the AP with your mobile phone through Wi-Fi. In any browser, get access to the AP IP shown to you. (Should be 192.168.4.1)
- Now you can get your NodeMCU connectet to the Wi-Fi and online.





- (Optional) You can have your code next to the enduser_setup.start(), and use the wifi.sta.status() as a flag variable to control the running of your code.

https://nodemcu.readthedocs.io/en/master/modules/wifi/#wifistastatus



Exercise 3:

List all the available APs around NodeMCU/ESP8266 and get their information of

Authmode, RSSI, BSSID, Channel, where the RSSI could be used to localize our devie according to the algorithms we talked about during the lecture.

==========

--Please refer to the official documentation of wifi.sta.getap()

--The available APs are scanned and listed

```
--With their RSSI and location, we would localize our own device

--Accuracy depends on the perfornabce of the chip

function listAP(t)

-- (SSID : Authmode, RSSI, BSSID, Channel)

print("\n"..string.format("%32s","SSID").."\tBSSID\t\t\t\t\t    RSSI\t\tAUTHMODE\tCHANNEL")

      for ssid,v in pairs(t) do

            local authmode, rssi, bssid, channel = string.match(v, "([^,]+),([^,]+),([^,]+),([^,]+)")

--RSSI here will be picked

            print(string.format("%32s",ssid).."\t"..bssid.."\t    "..rssi.."\t\t"..authmode.."\t\t\t"..channel)

      end

end

wifi.sta.getap(listAP)
```

==========

(Optional) Exercise 4:

In this exercise, you can explore the RSSI based localization of your NodeMCU when 3 APs are available by using the algorithm we talked about in the lecture. Let me know it should you need any other resources.

(Optional Challenge) Exercise 5:

In this challenging exercise, you can explore the possibility of setting your NodeMCU/ESP8266 as a repeater. Currently from exercise 2, you can't use the chip as a repeater to transmit data between its AP and its Stations. Can you think of a solution? Or you can refer to the following link for some ideas. This is a challenging task and not covered in our module.

https://github.com/martin-ger/esp_wifi_repeater