PSI3542 – 2023
SISTEMAS EMBARCADOS PARA IOT

AULA 11 – ATIVIDADE 11.1 WIFI MANAGER

SERGIO TAKEO KOFUJI

KOFUJI@USP.BR

Objetivos

- Resolver o problema:
  - Como configurar os parâmetros da rede WiFi sem gravar de forma fixa no firmware?
- Configurar e Implantar uma biblioteca de Gerenciamento de rede WiFi MicroPython em um dispositivo ESP32/8266
- Testar a biblioteca através de um programa simples de Servidor Web

- Referência:
  - MicroPython: Wi-Fi Manager with ESP32 (ESP8266 compatible). https://randomnerdtutorials.com/micropython-wi-fi-manager-esp32-esp8266/

Etapas da conexão WiFi

- ✓Quando o ESP32 inicializa pela primeira vez, ele é configurado como Ponto de Acesso;
- ✓Você pode se conectar a esse Ponto de Acesso estabelecendo uma conexão com a rede WiFiManager e acessando o endereço IP **192.164.4.1**;
- ✓É aberta uma página web que permite escolher e configurar uma rede;
- ✓O ESP32 salva essas credenciais de rede para que posteriormente possa se conectar a essa rede (modo Estação);
- ✓Assim que um novo SSID e senha forem definidos, o ESP32 reinicia, é colocado no modo Estação e tenta se conectar à rede salva anteriormente;
- ✓Se estabelecer uma conexão, o processo será concluído com sucesso. Caso contrário, ele será configurado como um Ponto de Acesso para você configurar novas credenciais de rede.

Biblioteca WiFi Manager

https://github.com/tayfunulu/WiFiManager

Description : WiFi manager for ESP8266 - ESP12 - ESP32 for micropython

Main Features:

- ✓ Web based connection manager
- ✓ Save wifi password in "wifi.dat" (csv format)
- ✓ Easy to apply

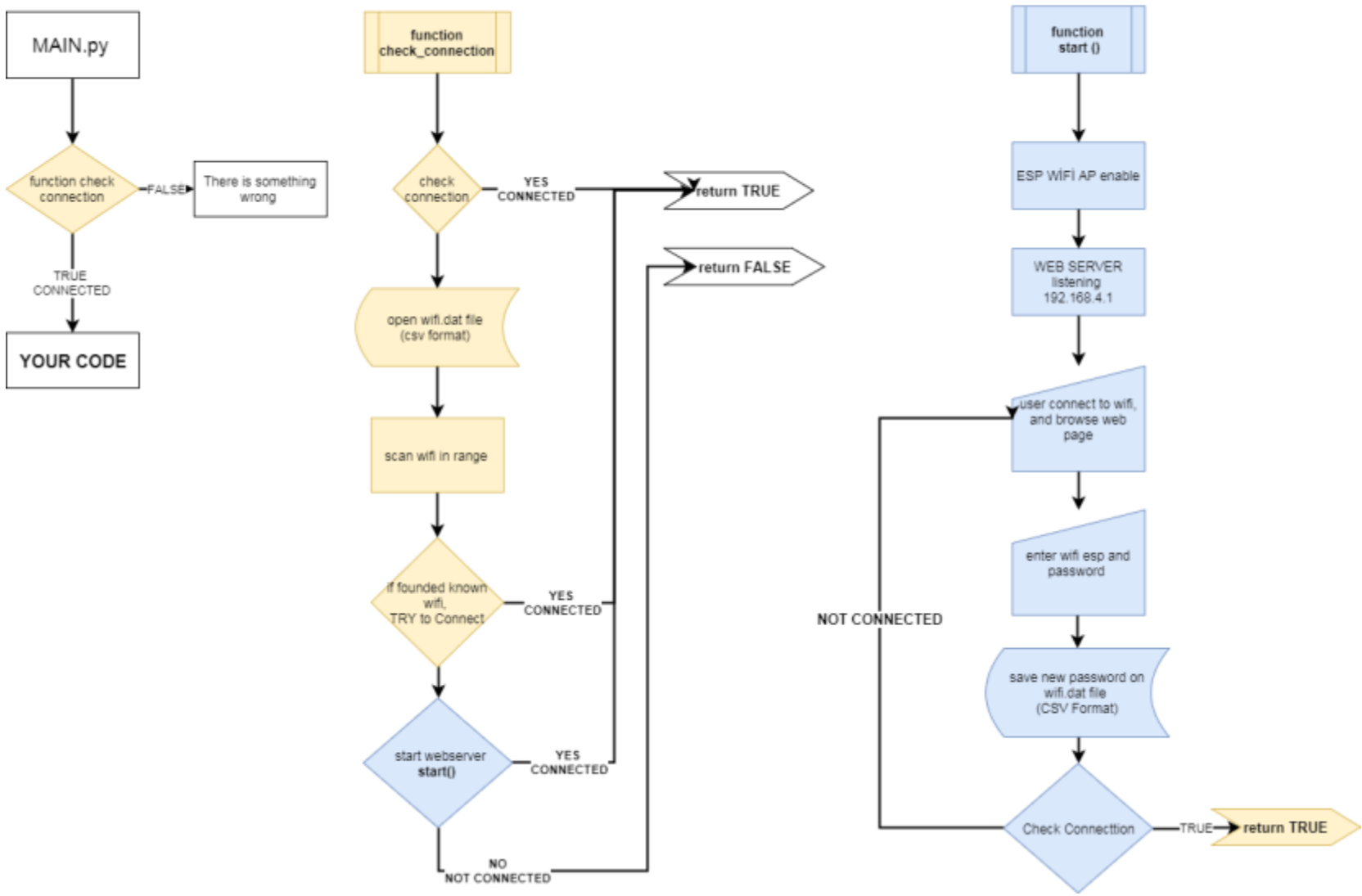
Usage:

- ✓ Upload main.py and wifimgr.py to ESP. Write your code into main.py or import it from main.py.

Logic:

- ✓ step: Check "wifi.dat" file and try saved networks/passwords.
- ✓ step: Publish web page to configure new wifi.
- ✓ step: Save network/password to "wifi.dat" file.
- ✓ step: Run user code.

WiFi Manager for ESP with Micropython



Roteiro

- Editar o código wifimgr.py
  - Nome do ssid do dispositivo no modo Access Point:
    - “wifimgr-<n. usp>”
  - Senha do ssid do dispositivo no modo Access Point:
    - “12345678”
- Fazer upload no dispositivo ESP32/8266 através do software Thonny
- Editar o Código main.py (servidor Web)
- Executar e testar o Código main.py no dispositivo ESP32/8266

Main.py: Web Server

```
# MicroPython: Wi-Fi Manager with ESP32 (ESP8266 compatible)
https://randomnerdtutorials.com/micropython-wi-fi-manager-esp32-esp8266/
#WEB SERVER https://randomnerdtutorials.com/esp32-esp8266-micropython-web-server/
#Complete project details at https://RandomNerdTutorials.com

import wifimgr
from time import sleep
import machine

try:
  import usocket as socket
except:
  import socket
```

Main.py, cont.

```
led = machine.Pin(2, machine.Pin.OUT)


wlan = wifimgr.get_connection()
if wlan is None:
    print("Could not initialize the network connection.")
    while True:
        pass  # you shall not pass :D
```

Main.py, cont.

```
# Main Code goes here, wlan is a working network.WLAN(STA_IF) instance.
print("ESP OK")

def web_page():
  if led.value() == 1:
    gpio_state="ON"
  else:
    gpio_state="OFF"

  html = """<html><head> <title>ESP Web Server</title> <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" href="data:,"> <style>html{font-family: Helvetica; display:inline-block; margin: 0px auto; text-align: center;}
  h1{color: #0F3376; padding: 2vh;}p{font-size: 1.5rem;}.button{display: inline-block; background-color: #e7bd3b; border: none;
  border-radius: 4px; color: white; padding: 16px 40px; text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;}
  .button2{background-color: #4286f4;}</style></head><body> <h1>ESP Web Server</h1>
  <p>GPIO state: <strong>""" + gpio_state + """</strong></p><p><a href="/?led=on"><button class="button">ON</button></a></p>
  <p><a href="/?led=off"><button class="button button2">OFF</button></a></p></body></html>"""
  return html
```

Main.py, cont.

```
try:
  s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
  s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
  s.bind(('', 80))
  s.listen(5)
except OSError as e:
  machine.reset()
```

Main.py, cont.

```
while True:
  try:
    if gc.mem_free() < 102000:
      gc.collect()
    conn, addr = s.accept()
    conn.settimeout(3.0)
    print('Got a connection from %s' % str(addr))
    request = conn.recv(1024)
    conn.settimeout(None)
    request = str(request)
    print('Content = %s' % request)
    led_on = request.find('/?led=on')
    led_off = request.find('/?led=off')
    if led_on == 6:
      print('LED ON')
      led.value(1)
    if led_off == 6:
      print('LED OFF')
      led.value(0)
    response = web_page()
    conn.send('HTTP/1.1 200 OK\n')
    conn.send('Content-Type: text/html\n')
    conn.send('Connection: close\n\n')
    conn.sendall(response)
    conn.close()
  except OSError as e:
    conn.close()
    print('Connection closed')
```

Bom Trabalho

kofuji@usp.br