

```

/*----- Simple captive portal for ESP8266 -----
A brief hard- and software description is found at: https://www.larsenhenneberg.dk
-----
Work scheme at reset or program restart:
.01: The wifi network ssid and password is read from the EEPROM
.02: The wifi connection is tried with the ssid and password, see .01
.03: In case of ok wifi connection, the program continues into the loop(), see .12
.04: In case of no wifi connection, the program waits for an active setup button.
.05: Upon an active setup button, an Access Point (AP) is started.
.06: The program continues into the loop(), see .12
.07: Use f.ex. a mobile phone to use the AP network.
.08: Browse to the captive portal at http://192.168.4.1
.09: Enter the wifi ssid and password at the captive portal web page and press Save.
.10: The ssid and password from the captive portal are saved into the EEprom.
.11: The ESP8266 is restarted, and starts from .01
.12: In the loop(): in case of no wifi connection, the captive portal web client is handled.
.13: Upon ok wifi connection, the AP server is closed.
.14: A while(1) loop inside loop() is now the main loop.
.15: The yield() function is called in while(1) to avoid a watch dog program reset.
.16: Upon an active setup button, the wifi is diconnected and the ESP8266 is restarted, see .01
-----
Hardware used: ESP-S01 board and a USB programmer that fits the ESP-S01.
At no wifi connection: the blue LED blinks with 0.5 sec. intervals, starts the AP to invoke
the captive portal when the setup input pin goes low.
At ok wifi connection: the blue LED blinks with 0.1 sec. intervals, starts the AP to invoke
the captive portal when the setup input pin goes low.
----- */
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <EEPROM.h>

#define setup_Pin    0      // GPIO0 as active low input to invoke the captive portal
#define Blue_LED_pin 2      // GPIO2 as test output , active high for blue LED light

// Soft AP ( Access Point) network setup for the captive portal
const char* APssid = "ESP8266 wifi setup";
const char* APpassword = ""; // might add AP password here !

ESP8266WebServer APserver(80); // this webservice is for AP only

// EEprom parameter savings are read at start, and on changes,
// the wifiSettings and the Hour_period are saved in EEprom
struct wifiSettings {
    char ssid[30];
    char password[30];
} user_wifi = {};

void setup() {
    pinMode(setup_Pin, INPUT);
    pinMode(Blue_LED_pin, OUTPUT); // Blue LED pin as output
    digitalWrite(Blue_LED_pin, 1); // set blue LED inactive
    EEPROM.begin(sizeof(struct wifiSettings));
    EEPROM.get(0, user_wifi);
    EEPROM.end();
    // blink LED while trying wifi connection with ssid,password
    // from EEprom until wifi connection OR the setup button is active
    WiFi.mode(WIFI_STA);
    WiFi.begin(user_wifi.ssid, user_wifi.password);
    while ((WiFi.status() != WL_CONNECTED) && (!setupButton())) {
        digitalWrite(Blue_LED_pin, 0);
        delay(500);
        digitalWrite(Blue_LED_pin, 1);
        delay(500);
    }
}

```

```

// go into AP mode on no wifi connection OR the setup button active
if ((WiFi.status() != WL_CONNECTED) || (setupButton())){
    WiFi.mode(WIFI_AP);
    WiFi.softAP(APssid, APpassword);
    APserver.on("/", handlePortal);    // start captive portal
    APserver.begin();
}
}

void loop() {
    while (WiFi.status() != WL_CONNECTED) { // waiting for AP client
        APserver.handleClient();           // to finish portal
    }
    APserver.close();    // Close the APserver

    while (1) { // going into the main loop !
        yield(); // disable wdt to avoid reset
        if (setupButton()) { // setup button pressed ?
            WiFi.disconnect(); // disconnect wifi
            ESP.restart();      // restart to continue with AP
        }
        // main loop goes here with a short LED blink
        digitalWrite(Blue_LED_pin, 0);
        delay(100);
        digitalWrite(Blue_LED_pin, 1);
        delay(900);
    } // while(1)
}

bool setupButton() {
    bool result = false;
    if (digitalRead(setup_Pin) == 0) {
        delay(20); // simple anti bounce delay
        if (digitalRead(setup_Pin) == 0) { result = true; }
    }
    return result;
}

void handlePortal() {
    if (APserver.method() == HTTP_POST) {
        struct wifiSettings temp_user_wifi;
        strncpy(temp_user_wifi.ssid, APserver.arg("ssid").c_str(), sizeof(user_wifi.ssid));
        strncpy(temp_user_wifi.password, APserver.arg("password").c_str(), sizeof(user_wifi.password));
    });
    temp_user_wifi.ssid[APserver.arg("ssid").length()] = temp_user_wifi.password[APserver.arg("password").length()] = '\0';
    if ((temp_user_wifi.ssid != user_wifi.ssid) || (temp_user_wifi.password != user_wifi.password)) {
        user_wifi = temp_user_wifi;
        EEPROM.begin(sizeof(struct wifiSettings));
        EEPROM.put(0, user_wifi);
        EEPROM.end();
    }
    APserver.send(200, "text/html", "<!doctype html><html lang='en'><head><meta charset='utf-8'><meta name='viewport' content='width=device-width, initial-scale=1'><title>Wifi Setup</title><style>*,::after,::before{box-sizing:border-box;}body{margin:0;font-family:'Segoe UI',Roboto,'Helvetica Neue',Arial,'Noto Sans','Liberation Sans';font-size:1rem;font-weight:400;line-height:1.5;color:#212529;background-color:#f5f5f5;}.form-control{display:block;width:100%;height:calc(1.5em + .75rem + 2px);border:1px solid #ced4da;}button{border:1px solid transparent;color:#fff;background-color:#007bff;border-color:#007bff;padding:.5rem 1rem;font-size:1.25rem;line-height:1.5;border-radius:.3rem;width:100%}.form-signin{width:100%;max-width:400px;padding:15px;margin:auto;}h1,p{text-align:center}</style></head><body><main class='form-signin'><h1>Wifi Setup</h1><br><br><p>The wifi settings have been saved.<br><br>The

```

```
device will restarted.</p></main></body></html>" );
    delay(200);      // delay to allow the response to be send to the client..
    ESP.restart();  // ..before resetting for setup() to try the ssid, password
  } else {
    APserver.send(200, "text/html", "<!doctype html><html lang='en'><head><meta
charset='utf-8'><meta name='viewport' content='width=device-width, initial-scale=1'><title>Wifi
Setup</title> <style>*,::after,::before{box-sizing:border-box;}body{margin:0;font-family:'Segoe
UI',Roboto,'Helvetica Neue',Arial,'Noto Sans','Liberation
Sans';font-size:1rem;font-weight:400;line-height:1.5;color:#212529;background-color:#f5f5f5;}.for
m-control{display:block;width:100%;height:calc(1.5em + .75rem + 2px);border:1px solid
#ced4da;}button{cursor: pointer;border:1px solid
transparent;color:#fff;background-color:#007bff;border-color:#007bff;padding:.5rem
1rem;font-size:1.25rem;line-height:1.5;border-radius:.3rem;width:100%}.form-signin{width:100%;max
-width:400px;padding:15px;margin:auto;}h1{text-align: center}</style></head><body><main
class='form-signin'><form action='/' method='post'><h1 class=''>Wifi Setup</h1><br><div
class='form-floating'><label>SSID</label><input type='text' class='form-control'
name='ssid'></div><div class='form-floating'><br><label>Password</label><input type='password'
class='form-control' name='password'></div><br><br><button
type='submit'>Save</button></form></main> </body></html>" );
  }
}
```