

Czujnik temperatury do akwarium, wyświetlający dane na dowolnym urządzeniu podłączonym do domowej sieci wifi

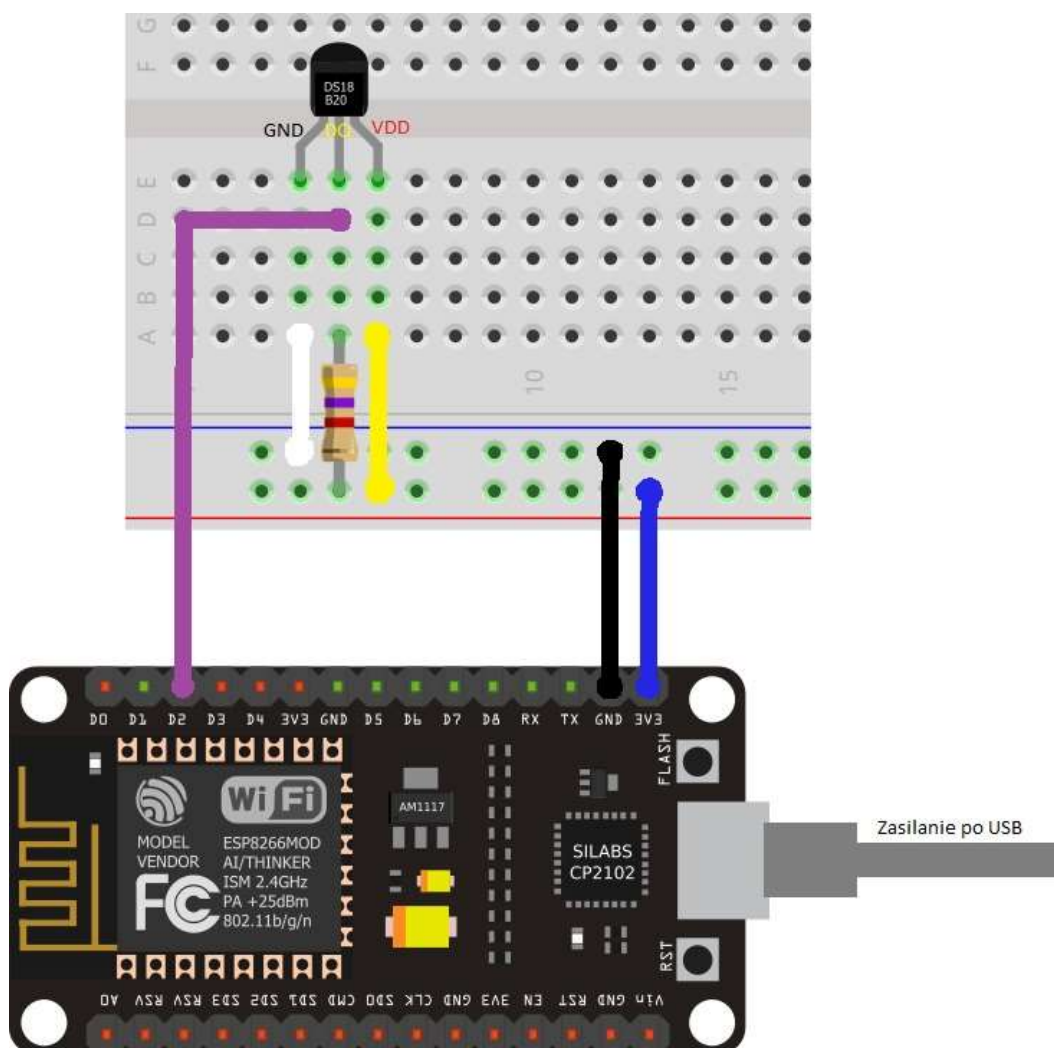
1. Opis

Celem projektu było stworzenie systemu, pozwalającego mojemu tacie (który jest zapalonym akwarystą) na podgląd w czasie rzeczywistym w jakiej temperaturze przebywają jego ukochane rybki. W tym celu została stworzona aplikacja, pozwalająca każdemu użytkownikowi podłączonemu do naszego domowego wifi zobaczyć, jaka temperatura panuje w akwarium. Celem tego na mikrokontrolerze, podłączonym do wodoodpornego czujnika temperatury, postawiono serwer http serwujący statyczne pliki (html, css, js, png itd.) oraz metoda GET (wykorzystywana przez aplikacje) pozwalająca na pobranie aktualnej temperatury.

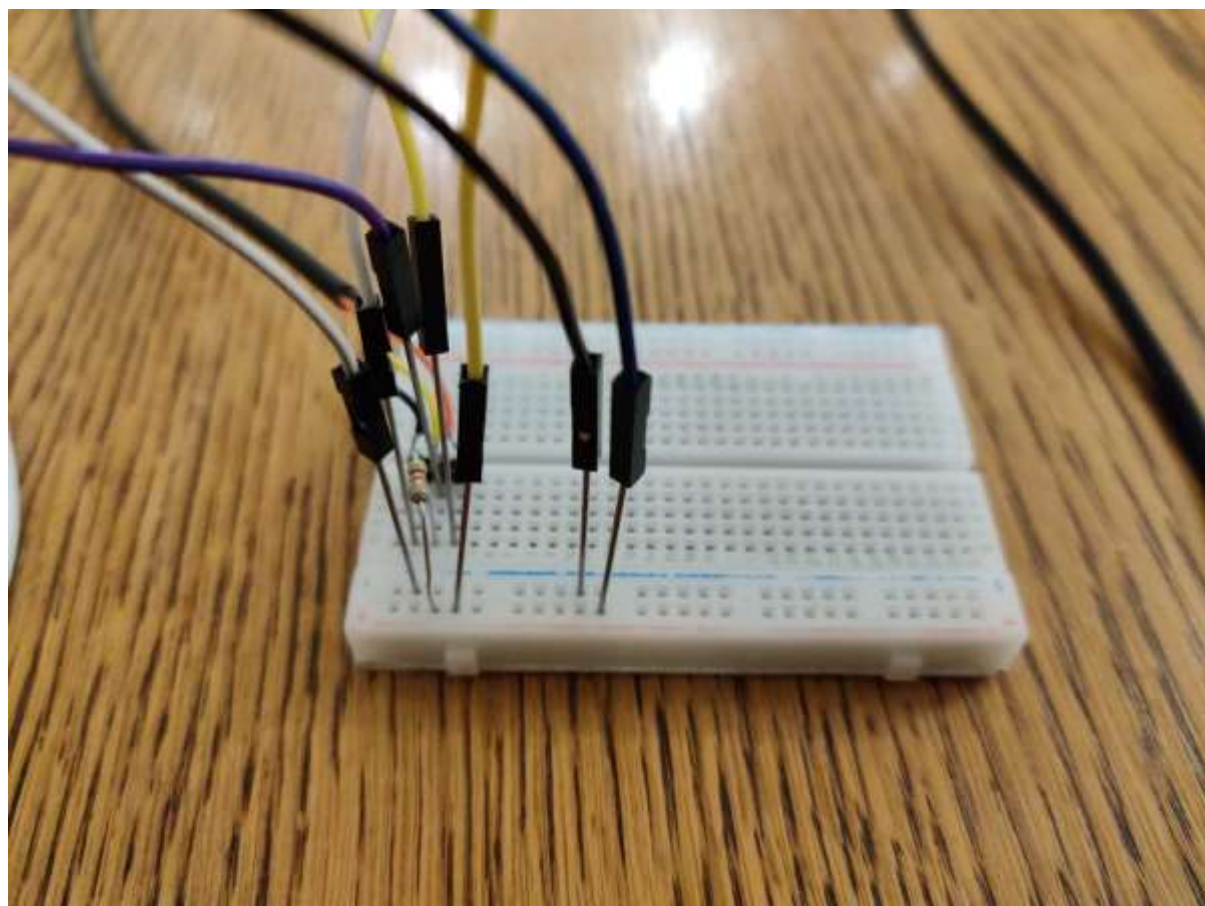
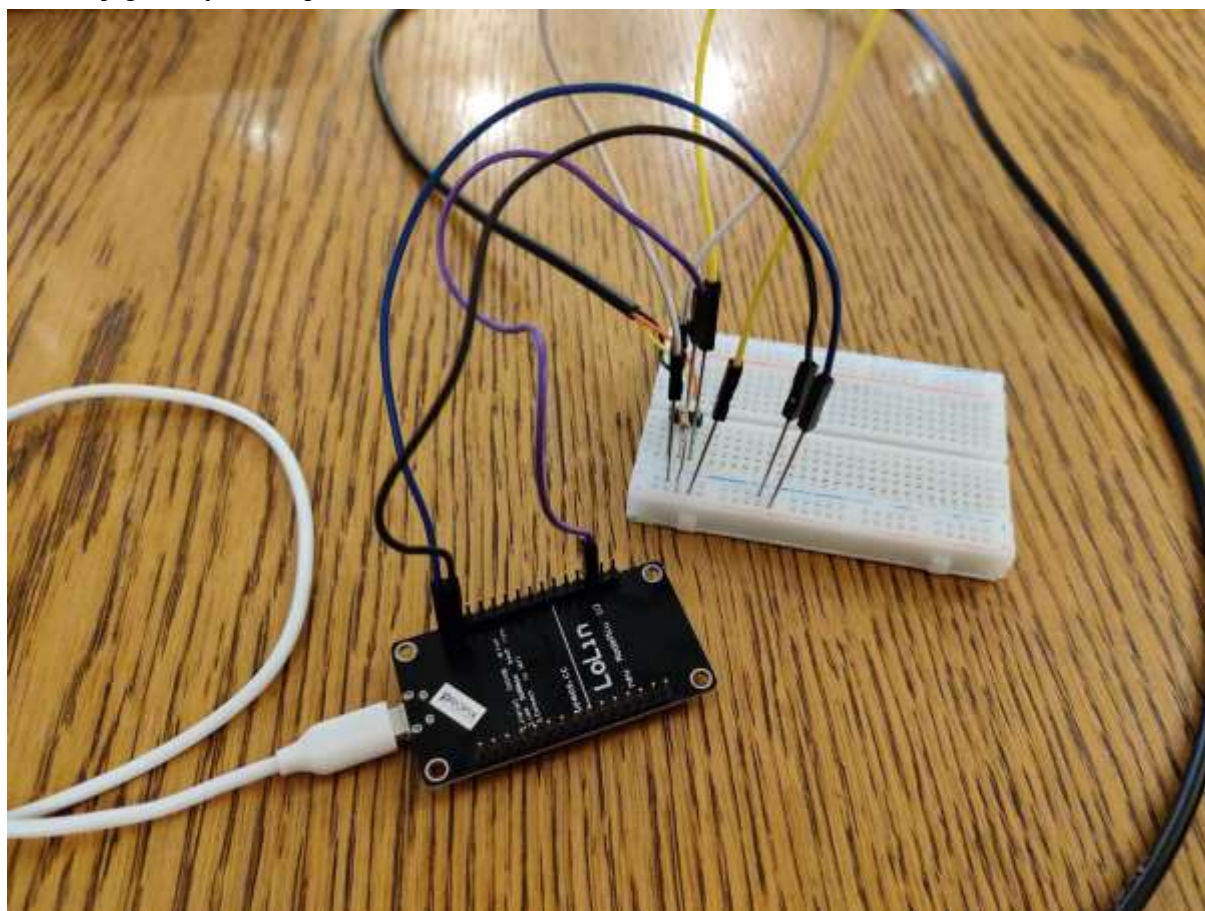
2. Użyte komponenty

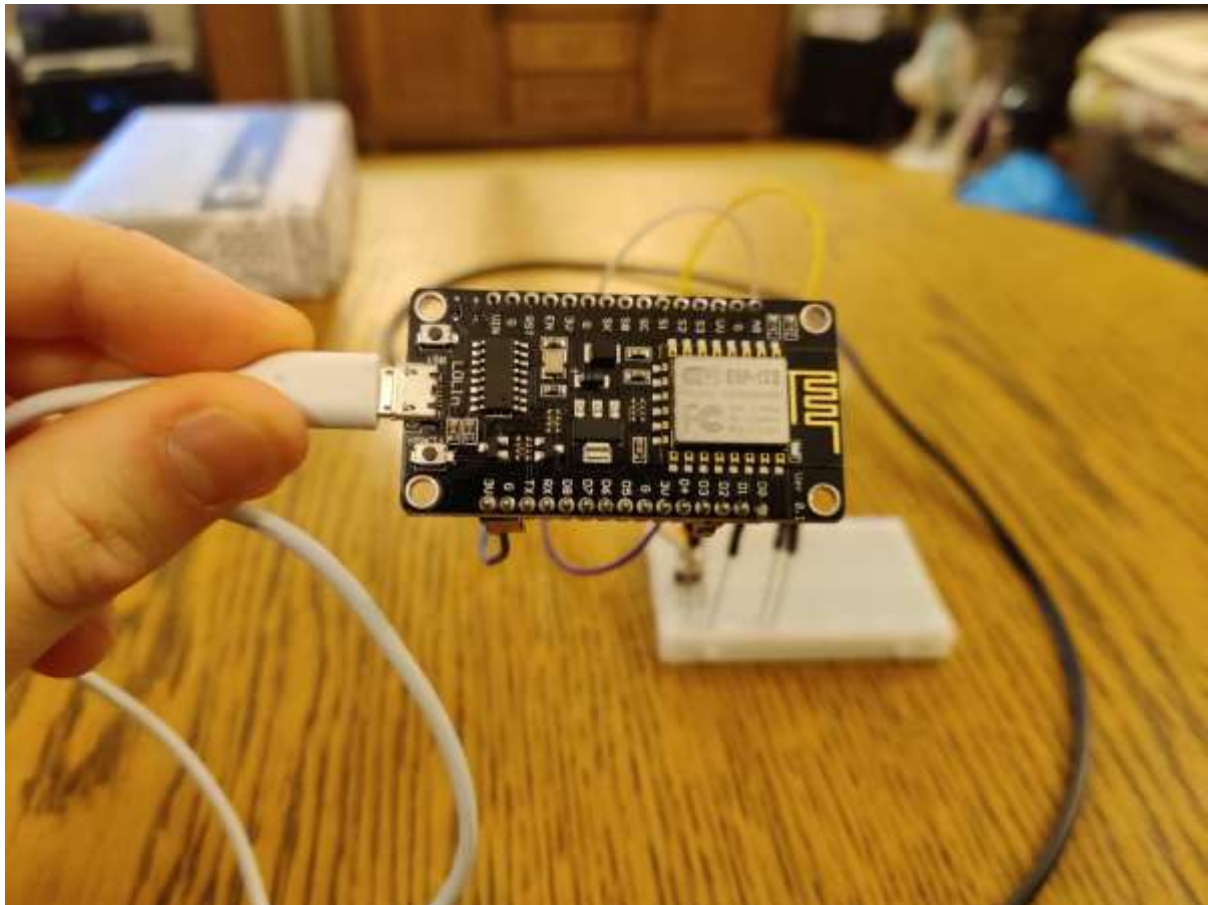
1. NodeMcu
2. Czujnik temperatury DS18B20
3. Rezystor - 4,7kΩ
4. Kable żeńsko-żeńskie
5. Szpilki
6. Ładowarka z microUSB

3. Schemat połączeń



4. Zdjęcia podłączenia





5. Screeny z aplikacji



5. Użyte biblioteki

ESP8266WiFiMulti – do połączenia z wifi

<https://github.com/esp8266/Arduino/blob/master/libraries/ESP8266WiFi/src/ESP8266WiFiMulti.h>

ESP8266mDNS – do uruchomienia serwera mDNS

<https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266mDNS>

ESP8266WebServer – do uruchomienia serwera http

<https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266WebServer>

DallasTemperature – do wygodnego pobrania temperatury w C

<https://www.arduino.cc/reference/en/libraries/dallastemperature/>

OneWire – do odczytania temperatury z czujnika DS18B20

<https://www.arduino.cc/reference/en/libraries/onewire/>

LittleFs – do odczytu plików statycznych

<https://github.com/littlefs-project/littlefs>

6. Kod mikrokontrolera

```
// Biblioteka do połączenia się z wifi
#include <ESP8266WiFiMulti.h>

// Biblioteka do wystawienia serwera mDNS
#include <ESP8266mDNS.h>

// Biblioteka do uruchomienia serwera http
#include <ESP8266WebServer.h>

// biblioteki do odczytu temperatury
#include <OneWire.h>
#include <DallasTemperature.h>

// biblioteka ułatwiająca serwoanie statycznych plików na serwerze
#include "LittleFS.h"

// Pin w który wpięliśmy czujnik temperatury
const int oneWireBus = 4;

// Klasa pozwalająca na odczyt temperatury z czujnika
OneWire oneWire(oneWireBus);
```

```

// Klasa ułatwiająca odczyt temperatury - możemy od razu odczytać wynik w
stopniach
// zamiast czytać pojedyncze bity
DallasTemperature sensors(&oneWire);

ESP8266WiFiMulti wifiMulti;    // Klasa pozwalająca na połączenie do wifi

ESP8266WebServer server(80);    // WebServer słuchający na porcie 80

void handleRoot();
void handleTemp();
void handleNotFound();
bool handleFileRead();

void setup(void){
    Serial.begin(115200);        // Ustaw Serial Monitor port 115200
    delay(10);
    Serial.println('\n');

    // dodawanie sieci wifi - połączy się z tą z najlepszym zasięgiem.
    // Mikrokontroler robiłem w domu, a jego docelowe miejsce jest u rodziców
    // więc to była całkiem wygodna opcja żeby nie musieć flashować kodu przy
zmianie miejsca
    wifiMulti.addAP("Orange_Swiatlowod_E6E0", "super_secret_password1");
    wifiMulti.addAP("OnePlus", "super_secret_password2");
    wifiMulti.addAP("domek", "super_secret_password3");

    Serial.println("Connecting ...");
    int i = 0;
    while (wifiMulti.run() != WL_CONNECTED) { // Czekamy na połączenie do WIFI
        delay(250);
        Serial.print('.');
    }
    Serial.println('\n');
    Serial.print("Connected to ");
    Serial.println(WiFi.SSID());                // Nazwa sieci do której się
połączyliśmy
    Serial.print("IP address:\t");
    Serial.println(WiFi.localIP());            // Adres IP naszego serwera http

    if (MDNS.begin("esp8266")) {                // Uruchomienie lokalnego serwera
MDNS o adresie esp8266.local
        Serial.println("mDNS responder started");
    } else {
        Serial.println("Error setting up MDNS responder!");
    }

    // rejestrowanie handlerów dla danych urli

```

```

server.on("/", HTTP_GET, handleRoot);
server.on("/temp", HTTP_GET, handleTemp);

// jeśli nie znalazło danego URL sprawdź czy jest plik - jeśli tak to go
zwróć, jeśli nie zwróć 404
server.onNotFound([]() {
    if (!handleFileRead(server.uri()))
        handleNotFound();
});

server.begin(); // rozpocznij nasłuch
Serial.println("HTTP server started");

// uruchomienie helpera dla czytania plików
if(!LittleFS.begin()){
    Serial.println("An Error has occurred while mounting LittleFS");
    return;
}
}
bool handleFileRead(String path) {
    Serial.println("handleFileRead: " + path);
    if (path.endsWith("/")) path += "index.html"; // If a folder is
requested, send the index file
    String contentType = getContentType(path); // Pobierze MME dla
pliku
    if (LittleFS.exists(path)) { // Jeśli istnieje
        File file = LittleFS.open(path, "r"); // otwórz plik
        size_t sent = server.streamFile(file, contentType); // zwróć jego
zawartość
        file.close(); // zamknij plik
        return true;
    }
    Serial.println("\tFile Not Found");
    return false;
}

void loop(void){
    // nasłuchujemy na requesty http
    server.handleClient();
}

void handleRoot() { // przekierowanie na index.html
    server.sendHeader("Location","index.html");
    server.send(303);
}

// pobierz temperature i wyślij ją jako tekst

```

```

void handleTemp() {
    sensors.requestTemperatures();
    float temperatureC = sensors.getTempCByIndex(0);
    server.send(200, "text/html", String(temperatureC));
}

void handleNotFound(){
    server.send(404, "text/plain", "404: Not found");
}

String getContentType(String filename){
    if(filename.endsWith(".htm")) return "text/html";
    else if(filename.endsWith(".html")) return "text/html";
    else if(filename.endsWith(".css")) return "text/css";
    else if(filename.endsWith(".js")) return "application/javascript";
    else if(filename.endsWith(".png")) return "image/png";
    else if(filename.endsWith(".jpg")) return "image/jpeg";
    else if(filename.endsWith(".ico")) return "image/x-icon";
    return "text/plain";
}

```

7. Video

W repozytorium znajduje się również video pokazujące działanie czujnika: [link](#). Na początku widzimy temperaturę panującą w akwarium, następnie czujnik przekładamy do szklanki z ciepłą wodą i po odświeżeniu wyświetlana jest nowa, wyższa temperatura.

8. Uwagi

W zamówieniu które złożyłem zaszła pomyłka i nie wysłali mi odpowiednich kabli – z tego powodu musiałem improwizować ze szpilkami (miałem jedynie kable żeńskie). Przez to nie mamy również docelowej obudowy na nasz mikrokontroler – zostanie ona zbudowana dopiero po dostaniu reszty części. Projekt jest jednak w pełni funkcjonalny i w tym polu nic nie będzie zmieniane.