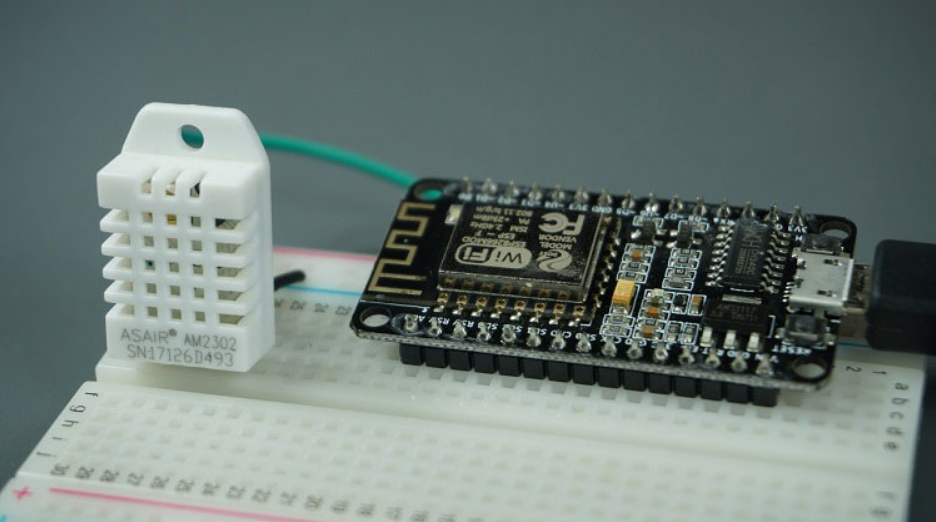
**Dunaújvárosi Egyetem Bánki Donát Technikum**   
**Projekt Feladat Dokumentáció**   
**Projekt tervezője:** Fésüs Martin   
**Projekt címe:** Iot  
**Osztály:** 13.C   
**Dátum:** 2024.10.16

**IoT Projekt – ESP8266 DHT11/DHT22 Hőmérséklet és Páratartalom Webszerver**



A projekt célja egy egyszerű IoT rendszer létrehozása, amely az ESP8266 mikrokontroller és a DHT11/DHT22 érzékelő segítségével mér hőmérsékletet és páratartalmat, és ezeket az adatokat egy weboldalon jeleníti meg. A rendszer Wi-Fi kapcsolaton keresztül működik, és bárhonnan elérhető a böngészőn keresztül.

**Szükséges eszközök**

A projekt elkészítéséhez egy ESP8266 Wi-Fi modulra (pl. NodeMCU vagy Wemos D1 Mini) és egy DHT11 vagy DHT22 érzékelőre van szükség a hőmérséklet és páratartalom mérésére. Az érzékelőt az ESP8266-hoz kell csatlakoztatni, amelyre az Arduino IDE segítségével írunk kódot. A DHT11 és DHT22 érzékelők közül a DHT22 pontosabb, de a DHT11 is megfelelő alapvető mérésekhez.

**Hardver Csatlakoztatás**

A DHT11/DHT22 érzékelőt a következő módon kell csatlakoztatni az ESP8266-hoz:

* A **VCC** pin-t az ESP8266 3V3 pin-jére,
* A **GND** pin-t az ESP8266 GND pin-jére,
* A **Data** pin-t a D2 (GPIO4) pin-re kell csatlakoztatni.

Ezután opcionálisan egy 4.7kΩ-10kΩ-os ellenállást érdemes elhelyezni a **Data** pin és a **VCC** között.

**Arduino IDE Beállítása**

Az Arduino IDE-ben először hozzá kell adni az ESP8266-ot a platformok listájához. Ehhez a **Preferences** menüpontban az "Additional Boards Manager URLs" mezőbe be kell írni az ESP8266 csomag

URL-jét:  
http://arduino.esp8266.com/stable/package\_esp8266com\_index.json  
Ezt követően a **Boards Manager**-ben telepíthetjük az ESP8266 csomagot, és a megfelelő ESP8266 modult (pl. NodeMCU vagy Wemos D1 Mini) választva programozhatjuk a mikrokontrollert.

**Kód Írása**

A következő kódot használjuk az ESP8266 programozásához, amely beolvassa a DHT11/DHT22 érzékelőtől a hőmérsékletet és a páratartalmat, majd megjeleníti ezeket egy HTML weboldalon.

#include <ESP8266WiFi.h>

#include <DHT.h>

// Wi-Fi hálózati beállítások

const char\* ssid = "SSID"; // Itt add meg a saját Wi-Fi hálózati neved

const char\* password = "PASSWORD"; // Itt add meg a Wi-Fi jelszavad

// DHT érzékelő beállítása

#define DHTPIN 4 // DHT11/DHT22 adatvonal (GPIO4 - D2)

#define DHTTYPE DHT11 // Választható DHT11 vagy DHT22

DHT dht(DHTPIN, DHTTYP E);

WiFiServer server(80); // Webszerver 80-as porton

void setup() {

// Kezdő beállítások

Serial.begin(115200);

delay(10);

// Csatlakozás a Wi-Fi hálózathoz

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

Serial.println("Kapcsolódás a Wi-Fi hálózathoz...");

}

Serial.println("Csatlakozva a Wi-Fi-hoz!");

Serial.println("IP cím: " + WiFi.localIP().toString());

// DHT érzékelő inicializálása

dht.begin();

// Webszerver elindítása

server.begin();

}

void loop() {

WiFiClient client = server.available(); // Várunk a bejövő kapcsolatra

if (client) {

String currentLine = "";

while (client.connected()) {

if (client.available()) {

char c = client.read();

Serial.write(c);

if (c == '\n') {

// Üdvözlő üzenet elküldése

if (currentLine.length() == 0) {

float h = dht.readHumidity(); // Páratartalom

float t = dht.readTemperature(); // Hőmérséklet

if (isnan(h) || isnan(t)) {

Serial.println("Hiba az érzékelésben!");

return;

}

String html = "<!DOCTYPE HTML><html><body><h1>Hőmérséklet és Páratartalom</h1>";

html += "<p>Hőmérséklet: " + String(t) + " &#8451;</p>";

html += "<p>Páratartalom: " + String(h) + " %</p>";

html += "</body></html>";

// HTTP fejléc elküldése

client.println("HTTP/1.1 200 OK");

client.println("Content-Type: text/html");

client.println("Connection: close");

client.println();

client.print(html);

break;

} else {

currentLine = "";

}

} else if (c != '\r') {

currentLine += c;

}

}

}

client.stop();

Serial.println("Kapcsolat megszakítva.");

}

}

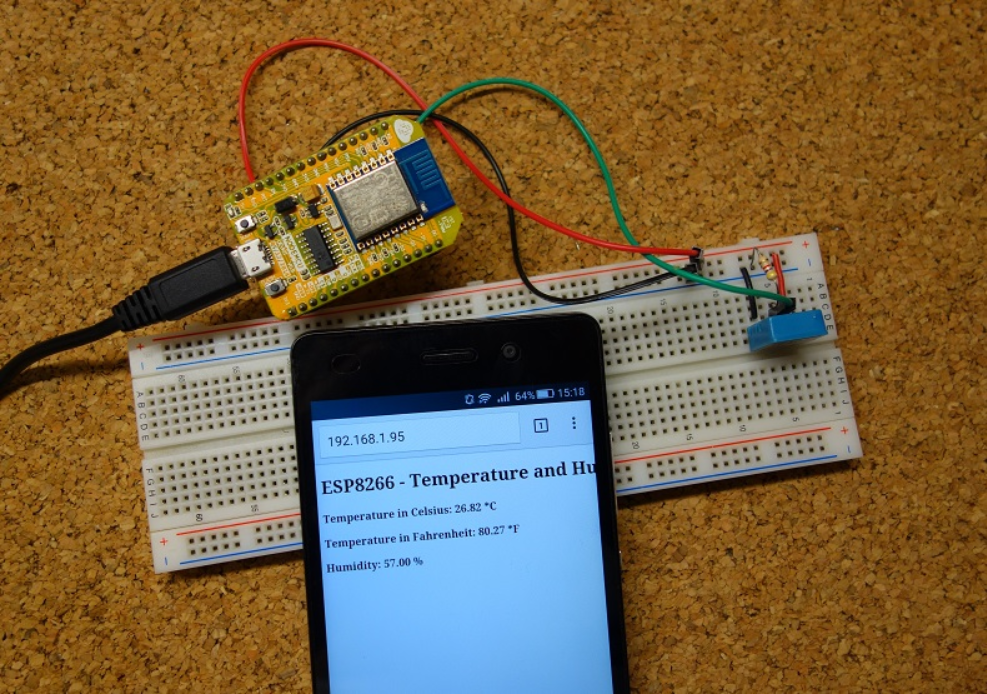
Ez a kód az ESP8266-ot arra programozza, hogy csatlakozzon a Wi-Fi hálózathoz, beolvassa a DHT érzékelőt, és a hőmérsékletet és páratartalmat egy HTML oldalon jeleníti meg. Az ESP8266 IP címére navigálva láthatjuk a mért adatokat.

**Weboldal**

A kód tartalmazza a HTML kódot, amely a weboldalon megjeleníti a hőmérsékletet és páratartalmat. Például, ha az ESP8266 IP címe 192.168.1.95, akkor a böngészőbe beírva:

http://192.168.1.95

A weboldal megjeleníti a hőmérséklet és páratartalom értékeket.

****

**Fejlesztési Lehetőségek**

A projekt később bővíthető: például a mért adatokat felhőszolgáltatásba (pl. ThingSpeak) küldhetjük, vagy akár mobilos alkalmazásban is megjeleníthetjük. Emellett érdemes lehet értesítéseket bevezetni, amelyek figyelmeztetnek, ha a hőmérséklet vagy páratartalom túl magas vagy túl alacsony.

**Önreflexió**

A projekt során sikerült létrehoznom egy alap IoT rendszert, amely képes mérni a hőmérsékletet és páratartalmat, majd ezeket egy weboldalon megjeleníteni. A rendszer működőképes, és az ESP8266 modul segítségével az adatok valós időben elérhetők Wi-Fi-n keresztül. Az alapvető funkciók jól működnek, és az Arduino IDE-ben való programozás során rengeteget tanultam az ESP8266-ról, a DHT érzékelőkről és az IoT rendszerek alapjairól.

Azonban úgy érzem, hogy még van mit tanulnom, különösen a stabilabb adatkommunikáció és a rendszer megbízhatóságának javítása terén. Az eszköz Wi-Fi kapcsolata néha instabil lehet, és a projekt bővítése (például felhőintegráció vagy értesítési rendszer) is új kihívásokat jelenthet. Összességében elégedett vagyok az elért eredménnyel, de biztos vagyok benne, hogy további fejlesztések és finomhangolások szükségesek a rendszer optimalizálásához.