

IOT

Trembeczki Dávid

B58TC3

Programtervező Informatikus, Levelező tagozat

A beadandó dolgozatban megvalósított projekt egyszerű: egy ESP32 mikrokontroller segítségével és egy hő és páratartalom mérő szenzorral kinyerek adatot és egy webszerverre iratom ki.

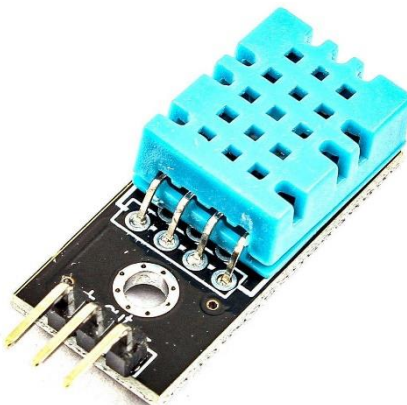
1. Eszközök

Ebben a projektben felhasznált eszközöket a modulshop.hu webshopból rendeltem, de ezeket be lehet szerezni más külföldi oldalról is.

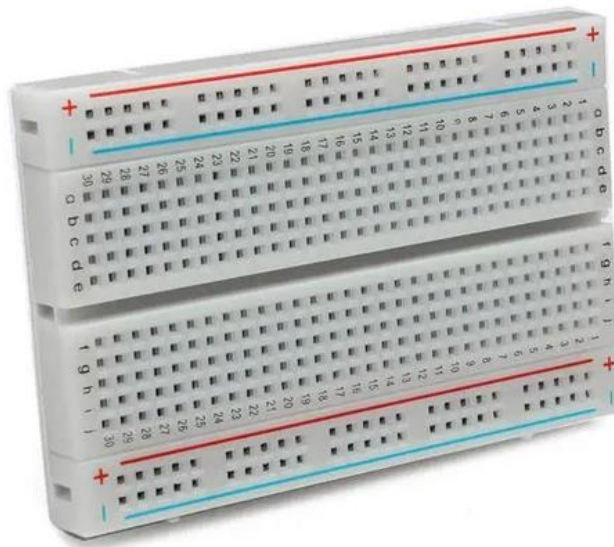
- ESP32WROOM32U mikrokontroller



- DHT11 Hőmérséklet és páratartalom érzékelő



- 400 lyukas breadboard, próbapanel

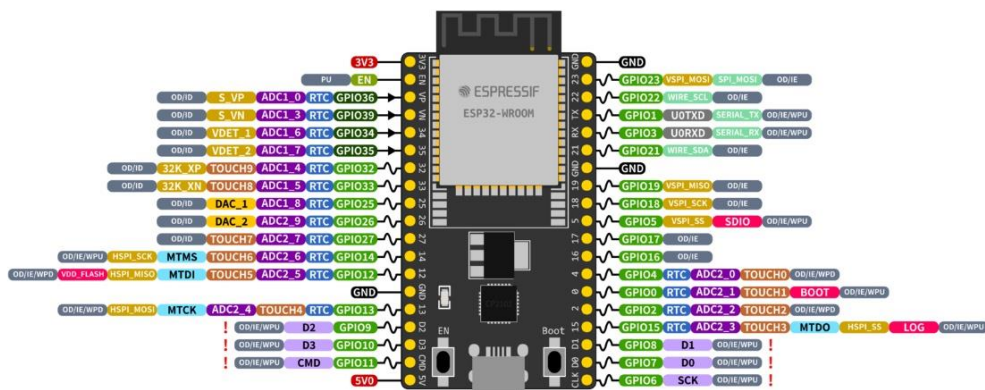


- jumper kábelek

2. Előkészület és összeállítás

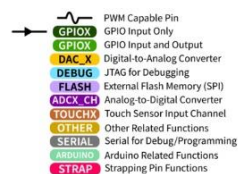
Az összeállítás nem egyszerű mivel figyelni kell az ESP32 pin kiosztására. Ha megtaláltuk a megfelelő pineket, akkor a szenzorral össze kell kötni a kábelek és a breadboard segítségével.

ESP32-DevKitC



ESP32 Specs

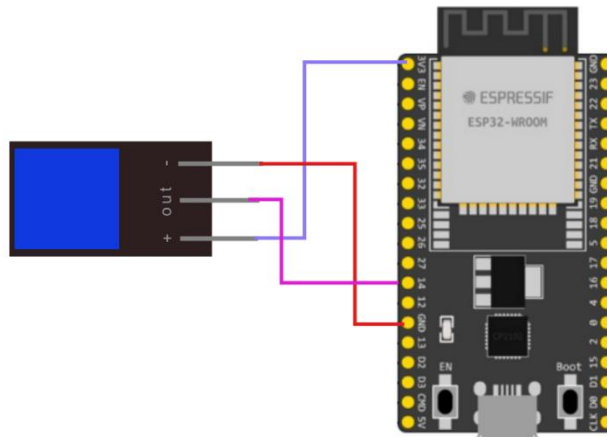
32-bit Xtensa® dual-core @240MHz
 Wi-Fi IEEE 802.11 b/g/n 2.4GHz
 Bluetooth 4.2 BR/EDR and BLE
 528 KB SRAM (16 KB for cache)
 448 KB ROM
 34 GPIOs, 4x SPI, 3x UART, 2x I2C,
 2x I2S, RMT, LED PWM, 1 host SD/eMMC/SDIO,
 1 slave SDIO/SPI, TWAI®, 12-bit ADC, Ethernet



RTC RTC Power Domain (VDD3P3_RTC)
 GND Ground
 PWD Power Rails (3V3 and 5V)
 ! Pin Shared with the Flash Memory
 Can't be used as regular GPIO

GPIO STATE
 WPU: Weak Pull-up (Internal)
 WPD: Weak Pull-down (Internal)
 PU: Pull-up (External)
 IE: Input Enable (After Reset)
 ID: Input Disabled (After Reset)
 OE: Output Enable (After Reset)
 OD: Output Disabled (After Reset)

Csatlakoztattam első körben anya-anya jumper kábelrel az ESP32-t és a DHT11 érzékelő modult: A 3v3 pinto a + megjelölésű pinre, a GND-t a – megjelölésű pinre és a GPIO14 pinto



pedig az „out” pinre.

3. Kódolás

A kódot az „Arduino IDE” nevezetű szoftverben lehet futtatni és rátölteni az ESP32-re. Ez viszont nem elég: ahhoz, hogy megfelelően funkcionáljon a mikrokontroller a CP210x drivert kellett telepíteni. Ezek után pedig a megfelelő sensor könyvtárát kell telepíteni, ami egyszerűsíti a programozó feladatát.

[Products](#)
[Applications](#)
[Ecosystems](#)
[Resources](#)
[Company](#)

[English](#)

[Developer Tools](#)
[USB to UART Bridge VCP Drivers](#)

CP210x USB to UART Bridge VCP Drivers

[OVERVIEW](#)
[DOWNLOADS](#)
[TECH DOCS](#)
[COMMUNITY & SUPPORT](#)

Download and Install VCP Drivers

Downloads for Windows, Macintosh, Linux and Android below.

*Note: The Linux 3.x.x and 4.x.x version of the driver is maintained in the current Linux 3.x.x and 4.x.x tree at www.kernel.org.

Software Downloads

[Software \(11\)](#)

Software	Version	Release Date
CP210x Universal Windows Driver	v11.3.0	8/9/2024
CP210x VCP Mac OSX Driver	v6.0.2	10/26/2021
CP210x VCP Windows	v6.7	9/3/2020
CP210x Windows Drivers	v6.7.5	9/3/2020
CP210x Windows Drivers with Serial Enumerator	v6.7.6	9/3/2020

[Show 6 more Software](#)

A(z) espressif.github.io soros porthoz szeretne csatlakozni

Kommunikációs port (COM1) – párosítva

CP2102 USB to UART Bridge Controller (COM3) – párosítva

Csatlakozás

Mégse

A kód egy része saját fejlesztés viszont a WiFi server létrehozására egy már meglévő, beépített, működő kód szolgált alapul.

A Wifi server kódját a File→Examples→WiFi→SimpleWiFiServer néven lehet elérni. Itt az ssid pointer-hez meg kell adni a WiFi hálózat nevét és a *password pointerhet pedig a jelszavát.

DHT dht(14, DHT11) függvényénél az első argumentum a felhasznált GPIO pin, a második pedig a használt szenzor pontos kódja. (Ebből a szenzor családból elérhető a DHT21 is)

```
float temp = dht.readTemperature();
float humidity = dht.readHumidity();
```

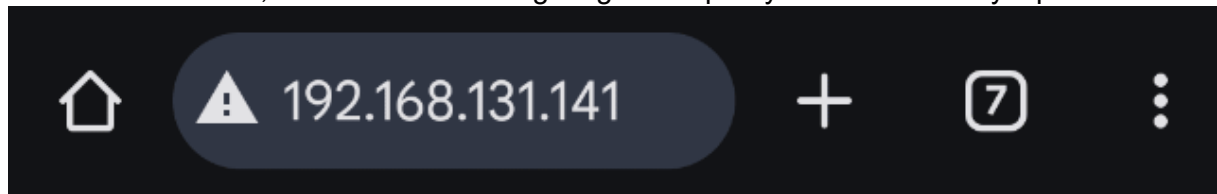
Itt megadjuk a temp és a humidity változók értékét, amit a dht könyvtárban lefejlesztett dht.readTemperature() és a dht.readHumidity() függvények olvasnak ki a szenzorból.

A hálózatra csatlakozás után a Serial Monitor-on kiírja a lokális IP címet, amit beírva a böngészőbe megnyílik az oldal.

A HTTP headerök használata hasznos lehet ebben az esetben: itt nem alapértelmezett az UTF-8 kódolás, így az a "Content-type:text/html; charset=utf-8" kerül beállításra. "Refresh: 2" HTTP headerrel az oldal 2 másodpercenkénti frissítését lehet elérni, így nem kell nekünk frissítgetni az oldalt, viszonylag friss adatokat kapunk.

4. Eredmény és Összegzés

A fejlesztés alatt több verzió és prototípus is született, majd pedig a kész kódot WiFi szerveren futtattam, amit a telefonom segítségével képernyőfotóval le is fényképeztem.



Hőmérséklet:

24.80 °C

Páratartalom:

55.00 %