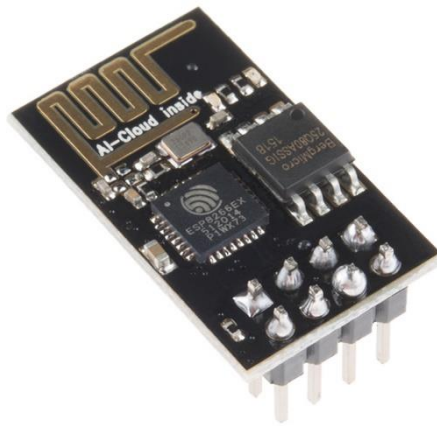


Hőmérsékletadat továbbítása Arduino-hoz illesztett wifi modulal

Ez a dokumentum egy leírás az Arduino Uno-hoz illesztett ESP8266-os wifi modul webszerverként való használatához. A projekt befejeztével egy böngészőben megjelenő hőmérséklet adathoz jut az érdeklődő.

Manapság egyre inkább kihasználjuk a vezeték nélküli eszközök nyújtotta előnyöket, elég csak a mobiltelefonunkban lévő wifi-re gondolni. A széles körben elterjedt Arduino alapú mikrovezérlőkhöz is találhatunk már vezeték nélküli modult. A népszerű ESP8266 nevű wifi modul is egy ilyen eszköz, ráadásul ezen a modulon is egy Arduino alapú, programozható mikrovezérlő található. Az ESP8266 rendkívül olcsó, akár 1000 Ft alatti áron is beszerezhető, viszont csak két szabad felhasználású lába van, ami csak kisebb feladatokhoz megvalósításához elegendő. Ezt a korlátot megszüntethetjük például úgy, hogy - egy már meglévő, több szabad felhasználású lábbal rendelkező - Arduino-hoz soros porton keresztül hozzákötjük az wifi modult.



ESP 8266 wifi module [1]

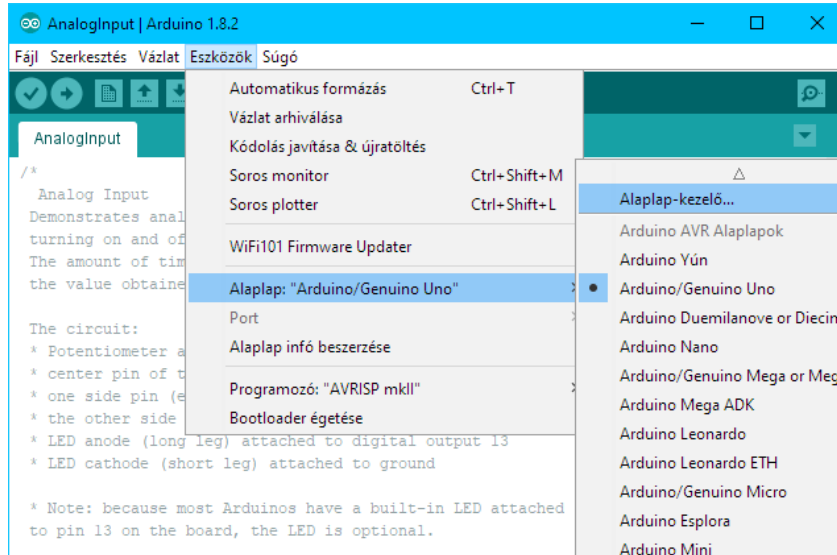
Alkatrészlista:

- Arduino Uno (vagy hasonló Arduino)
- ESP8266 wifi modul
- 3,3V-os USB to UART konverter (pl. FTDI kábel)
- DS18B20 hőmérő szenzor
- próba panel
- vezetékek
- ellenállások: 1kOhm, 2kOhm, 4,7kOhm, 10kOhm
- 2 db mikrokapcsoló
- 2 db ceruzaelem + elemtartó

Telepítés:

1. [az Arduino IDE telepítése](#)
2. ESP8266 board telepítése az IDE-be [9]:
 - a. Az Arduino IDE-ben kattintsunk a Fáj/Beállítások menüpontra

- b. a „További Alaplap-kezelő URL-ek” részhez illesztjük be a következő URL-t: http://arduino.esp8266.com/stable/package_esp8266com_index.json, majd kattintsunk az OK gombra
- c. ezután menjünk az Eszközök/Alaplap: „Arduino/Genuino Uno”/Alaplap-kezelő menübe

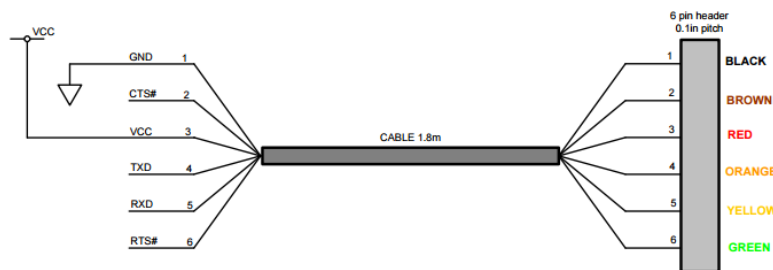


- d. válasszuk ki az „esp8266 by ESP8266 Community” elemet a listából és kattintsunk a telepítés gombra
- e. válasszuk ki a panelt az Eszközök/Alaplap: „Arduino/Genuino Uno”/Generic ESP8266 Module elemre kattintva
- f. végül indítsuk újra az Arduino IDE-t

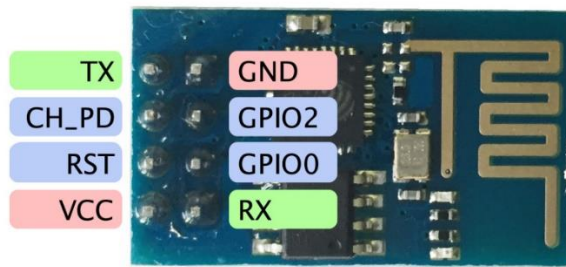
A wifi modul programozása:

Az ESP8266-on nem található USB-UART konverter, ezért a programozásához egy erre alkalmas eszközre is szükségünk lesz. Az FTDI által gyártott USB-soros konverter kábel például egy ilyen eszköz. A konverter kiválasztásánál arra kell nagyon figyelni, hogy 3,3V-os feszültségszintű legyen, mivel az ESP modul is ezen a feszültségen üzemel.

1. Az USB-soros konverter és az ESP modul összekötéséhez szükségünk lesz mindkét eszköz lábkiosztására.
 - a. FTDI kábel [2]:



b. ESP8266 (felülnézet) [3]:



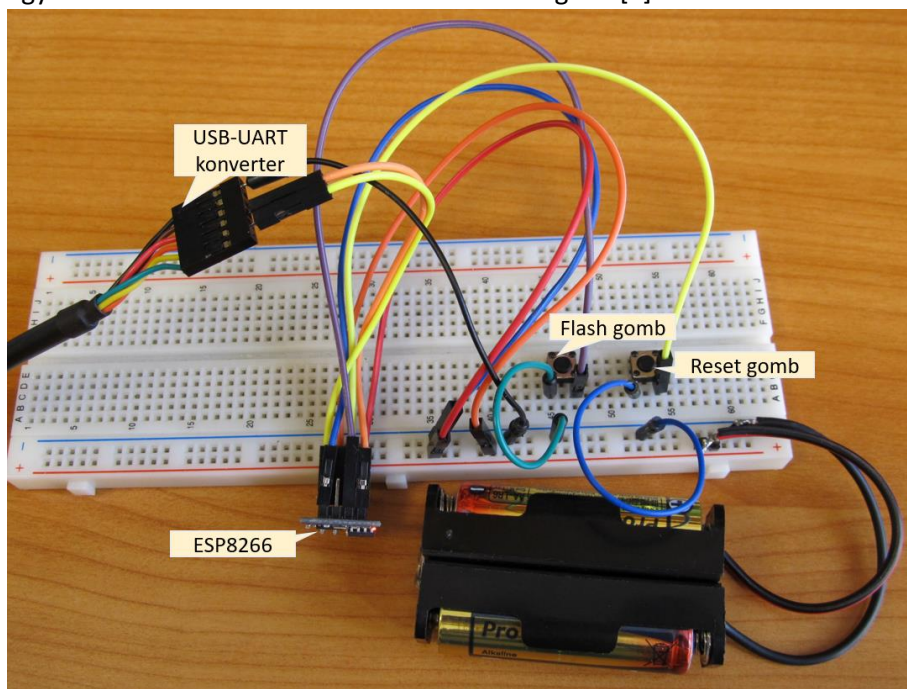
2. bekötés:

GND — GND

TX — RX

RX — TX

3. Ahhoz, hogy az ESP modul firmware-ét át tudjuk írni előbb át kell állítani flash módba. Ehhez egy kisebb áramkör összeállítására lesz szükségünk [4].



a. Kapcsolás:

+3.3V — CH_PD

flash gomb — GPIO

reset gomb — RST

elem — — GND

elem + — VCC

b. A flash mód aktiválása a fenti képen látható áramkörben lévő két mikrokapcsoló megfelelő idejű lenyomásával lehetséges. A 3-3,3V-os tápfeszültség bekötése előtt ellenőrizzük, hogy mindent jól kötöttünk be és különösen ügyeljünk a feszültség polaritására. Ezután először nyomjuk le a Flash gombot, tartsuk lenyomva és nyomjuk le a Reset gombot is, végül felengedhetjük mindkettőt.

4. A webszerver forráskódja [4]:

a. [letöltés](#)

b. A kód elején található "your-ssid", "your-password" értékek helyett a saját routerünk nevét és jelszavát kell megadnunk, hogy csatlakozni tudjon az ESP a router-hez.

- c. Az Arduino IDE-ben válasszuk ki a panelt az Eszközök/Alaplap: „Arduino/Genuino Uno”/Generic ESP8266 Module elemre kattintva, majd az Eszközök/Port menüben a legnagyobb számozású COM-portot (általában ez a módszer szokott működni, de ha több eszköz is csatlakoztatva van a számítógéphez, akkor megnézhetjük a Windows Eszközkezelőjében, hogy melyik COM-portot kapta az eszköz, amihez csatlakozni szeretnénk).
- d. Ezután már csak le kell fordítanunk a kódot és rátölteni az ESP-re.
- e. Ha minden jól sikerült és a konzol sem ír hibát, akkor az Eszközök menüből a Soros monitort megnyitva, a kommunikáció sebességét (baud rate) 115200-ra állítva, majd a Reset gomb megnyomása után a következő üzenet lesz látható:
 „Connecting to your-ssid
”
 (persze a saját SSID azonosítónkkal)
- f. A router által az ESP-nek kiosztott IP címet is megtudhatjuk a Reset gomb megnyomása után kiírt szövegből. Ezt jegyezzük meg mielőtt szétszedjük az ESP programozásához összeállított áramkört, mert később kelleni fog.

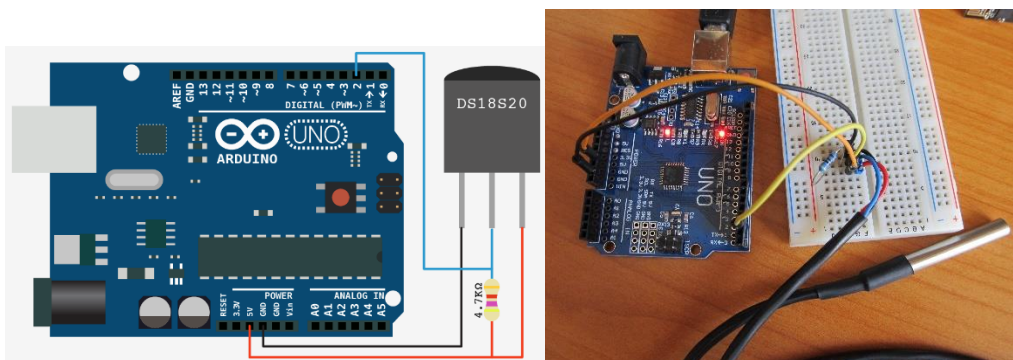
Az Arduino Uno programozása:

Következő lépésként, programozzuk fel úgy az Arduino Uno-t, hogy képes legyen beolvasni a DS18B20 nevű hőmérő szenzor adatát és kiküldeni azt az UART-on.

1. A forráskód [6, 7, 8] innen tölthető le: [ds18b20_code.zip](#)
2. Csomagoljuk ki a zip fájlt, majd nyissuk meg a ds18b20_code.ino fájlt
3. Az IDE-ben az alaplap menüben válasszuk ki a használandó Arduino-t, esetünkben az Arduino/Genuino Uno-t, válasszuk ki a számítógéphez csatlakoztatott eszköz COM port-ját a Port menüben, majd pedig töltsük fel a kódot az Arduino-ra.

Az hőmérő szenzor bekötése az Uno-ba.

1. A hőmérő szenzor bekötéséhez állítsuk össze az alábbi képen látható kapcsolást [6] (közben válasszuk le az Arduino-t a számítógépről az USB kábel kihúzásával, hogy ne szereljünk feszültség alatt). A jobb oldali képen lévő szenzor csak tokozásában tér el a bal oldali képen lévőttől.

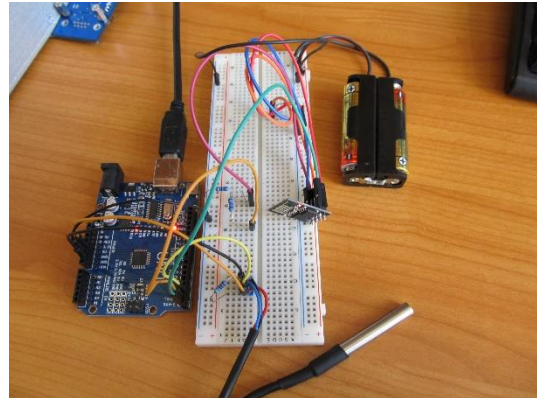
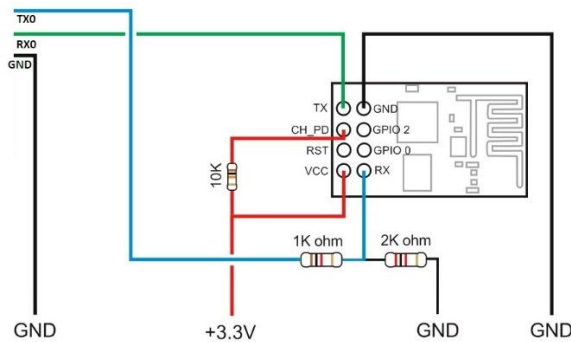


2. Ellenőrizzük a kapcsolást, majd csatlakoztassuk ismét az USB kábelt.
3. Ha mindent jól csináltunk, az IDE-ben a Soros monitort megnyitva, már látnunk is kell a másodpercenként frissülő hőmérsékletet.

A két eszköz összekapcsolása:

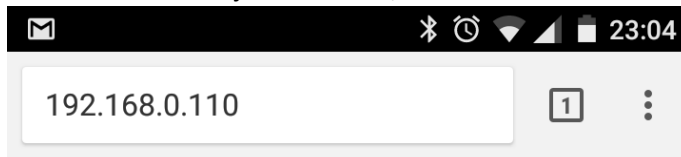
1. Már csak össze kell kötni az ESP-t az Arduino Uno-val. Mivel az ESP8266 wifi modul 3,3V-os az Arduino Uno pedig 5V-os, szükség van egy feszültség szintillesztésre. Ennek a problémának

a megoldására léteznek külön szintillesztő áramkörök, de a legegyszerűbb megoldást egy egyszerű feszültségosztó adja. Ezt egy 1 és egy 2kOhm-os ellenállással valósíthatjuk meg a következő ábrán látható módon [5].



Figyeljük meg, hogy a felső kapcsolási rajzon a GND (föld) nevű kivezetéseit is össze kell kötni a két eszköznek. Erre azért van szükség, mert az elemes tápfeszültség földfüggetlen, nincs egy fix potenciálon, így el is lebeghet a potenciálja a föld potenciálon lévő Arduino-hoz képest. Ezt úgy oldhatjuk meg, hogy egy vezetékkel összekötjük az Arduino föld pontját az elemes tápfeszültség negatív pontjával, így a telep negatív pontját föld potenciálra hoztuk.

2. Az áramkör tápfeszültség alá helyezése után jöhet a próba. Egy, a router-re csatlakozott okostelefon vagy számítógép böngészőjébe gépeljük be az ESP korábban megjegyzett IP címét. Ha mindent jól csináltunk, akkor a következő látható a böngészőben:



26

Az eddigiek során megvalósított rendszerrel számos hétköznapi és speciális feladat megvalósítását alakíthatjuk vezeték nélkülivé, valamint egy jó kiinduló pont lehet akár egy okos otthon vezérlő egységének megvalósításához is.

források:

- [1] <https://www.sparkfun.com/products/13678>
- [2] http://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_TTL-232R_CABLES.pdf
- [3] <http://fabacademy.org/archives/2015/doc/networking-esp8266.html>
- [4] <http://www.arduinesp.com/wifiwebserver>
- [5] <http://www.martyncurrey.com/arduino-to-esp8266-serial-commincation/>
- [6] <https://create.arduino.cc/projecthub/everth-villamil-ruiz/temperature-sensor-ds18b20-3decfc>

[7] <https://github.com/milesburton/Arduino-Temperature-Control-Library>

[8] <https://github.com/PaulStoffregen/OneWire>

[9] <http://randomnerdtutorials.com/how-to-install-esp8266-board-arduino-ide/>

kérdések:

- protoboard felépítéséről legyen egy kép?
- részletezzem, hogy miért csak a TX-re kell a szintillesztés?
- legyen plusz feladat, pl. LED vezérlése?
- ha tesztelve lesz az ESP Arduino Uno-val való programozása, akkor azt ide is be lehetne rakni
- a gyári firmware visszaállítását is be kéne rakni, csak tesztelni kell még (<http://iot-playground.com/blog/2-uncategorised/35-esp8266-firmware-update>)