Adatgyűjtés és vezeték nélküli adattovábbítás

Önálló laboratórium dokumentáció
2022/23 II. Félév

Papp Dominik Edvárd

Konzulens: Scherer Balázs

Tartalomjegyzék

Feladat specifikáció

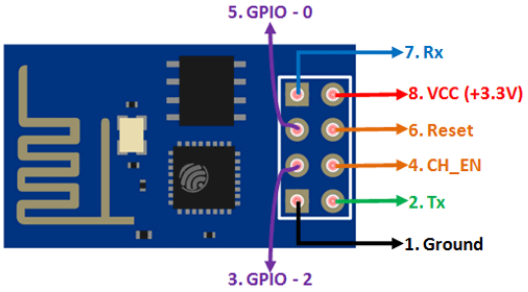
A feladat valamilyen kreatív mérnöki munka megvalósítása volt egy STM32H750B-DK felhasználásával. A specifikáció a félév során alakult ki. A cél egy olyan rendszer kialakítása lett, amely vezetéknélküli adattovábbítás segítségével valamely hőmérő hőmérsékletét bárhonnan meg- és vissza is lehet nézni log-szerűen. A rendszert egy mikrokontroller, egy WIFI modul, egy hőmérő és egy TCP (Python kód) - valamint PHP szerver futtatására alkalmas számítógép és egy weboldal alkotja.

Alkotóelemek

ESP8266-01

Ismertető

Az ESP8266 01-es verziója 8 kivezetéssel rendelkezik es közvetlenül illeszthető a mikrokontroller Fan-Out boardjára. A board rendelkezik külön dedikált foglalattal a modul számára. A modul UART protokollt használ kommunikációra 115200 baudrattel, így kezelése egyszerű. Képes WIFI hálózaton keresztül TCP szerverre csatlakozni, ami több mint alkalmassá teszi a projectben való használatára az alacsony ára és kis mérete mellett.



Bekötés

A bekötése a következő: PIN1: MCU földje, PIN2: MCU UART_Rx, PIN3: Logikai magas, PIN4: Logikai magas, PIN5: Logikai magas, PIN6: Logikai magas, PIN7: MCU UART_Tx, PIN8: MCU 3V3.

Programozás

A modul programozása ettől a bekötéstől eltérő huzalozást igényel. Úgy hozható programozható üzemmódba, hogy a PIN5 lábát földre kell kötni bootoláskor, azaz amikor tápot kap. A felhasznált programozó nem rendelkezett ilyen funkcióval, ezt manuálisan hozzá kellett forrasztani és így már egy gomb megnyomásával programozható állapotba hozhatóvá vált.

A feltöltentő firmware innen ingyenesen letölthető. A feltelepítéshez feltétlenül szükség van az esptool.py-ra. Ha ezek mind megvannak és a modul is csatlakoztatva van a számítógéphez programozói üzemmódban, akkor a következő paranccsal tölthető fel a firmware: esptool.py --chip auto --port /dev/tty.usbserialname --baud 115200 --before default_reset --after hard_reset write_flash -z --flash_mode dio --flash_freq 40m --flash_size 1MB 0x0 #PathToBinFile/Cytron_ESP-01S_AT_Firmware_V2.2.0.bin, ahol /dev/tty.usbserialname a csatlakoztatott programozó neve (cmd paranccsal való meghatározás Mac-en: ls /dev/tty.*) és #PathToBinFile a letöltött .bin file elérési útvonala. Ezek után nem programozói üzemmódban (PIN5: High) való rebootolás utan AT Command-ok segítségével vezérelhető a modul.

SHT31 SEN0385 Sensor

A hőmérő dokumentációja itt, részletesebb leírás róla pedig itt érhető el. A linkelt dokumentumok kellő részletességgel, kellő minőségben tárgyalják az egyszerű szenzor működését, így csak a project szempontjából a legfontosabbakat emelem ki.

A hőmérő hőmérséklet és páratartalom mérésére egyaránt alkalmas. A projectben csak a hőmérsékleti adatokra van szükség, amit I2C kommunikációval lehet lekérdezni. A maximális órajel frekvencia 1MHz, azonban mi csak 100kHz frekvenciával használjuk. Ez a sebesség bőven megfelel a céljainkra. A kezeléséhez Scherer Balázs konzulensem szolgáltatott drivert. Minimális módosítással tökéletesen alkalmazhatóvá vált az én projectemben is.

TCP szerver

Használat

A szerver Python programozási nyelven íródott, a file neve TCP_server.py. Algoritmusa és kezelése igen egyszerű, ugyanis csak el kell indítani. Futtatása a Python letöltése és a megfelelő alkönyvtárba való belépés után a „python TCP_server.py” paranccsal futtatható.

Folyamata

A szerver 1 klienst tud kezelni, 1 felhasználó tud csatlakozni. Ha csatlakozás után és bárhol a folyamat során a szerver 10 másodpercen keresztül nem kap adatot a klienstől, úgy veszi, hogy az lecsatlakozott. Ekkor visszaadja a socketet az operációs rendszernek és vár újabb 10 másodpercet míg az operációs rendszer újra engedélyezi (macOS Ventura 13.3.1(a) alatt) ugyan azon portnak a megnyitását. A szerver ’q’ lezáró karakter hatására formázás után a data.txt fileba azonnal kimenti az adatot. Formázás alatt azt kell érteni, hogy ha például a „21.9q” karaktersorozatot kapja a szerver akkor azt „YYYY-MM-DD HH:MM:SS 21.9 °C\n” karaktersorozattá alakítja át (YYYY-M stb. helyére az aktuális dátum és időpont kerül). A szerver továbbá még státusz információt, illetve a kapott adatot terminálba kiírja.

PHP szerver és weboldal

Szerver letelepítése

Az általam használt PHP szerver a következő terminál parancsokkal letölthető:

„ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)" ”, „ brew install php”. Az utóbbi parancs végrehajtásához elengedhetetlen, hogy a számítógép rendelkezzen a python Homebrew szolgáltatásával.

Weboldal

A felhasznált php file neve „site.php”. Ez egy igen egyszerű php oldal. Betölti a „data.txt” file adatait, majd másodpercenként frissíti magát. Így másodpercenként nyomon követhető a „data.txt” tartalma.

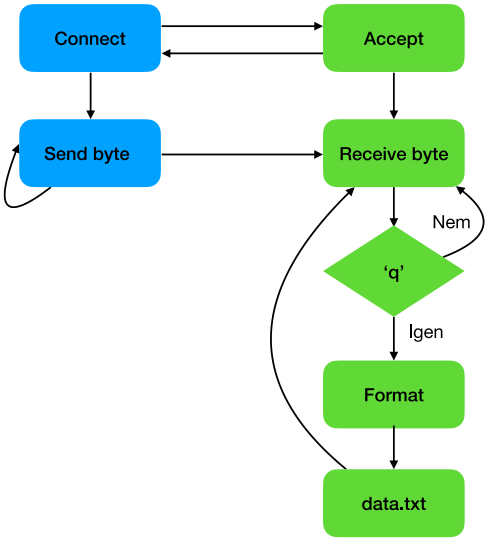
Használata

Letöltés és a „site.php” könyvtárába való benavigálás után a „php -S localhost:8000” paranccsal indítható is a szerver. Ezt a könyvtárat fogja localhostnak tekinteni. A böngészőbe a következő linket beírva: http://localhost:8000/site.php, már meg is tekinthető a „data.txt” tartalma.

TCP szerver és ESP/MCU kapcsolata

Kommunikáció

A mikrokontroller TCP üzenetek formájában továbbítja az adatokat byteonként az ESP segítségével. Az ehhez szükséges fejlécet az ESP önállóan, külön beavatkozás nélkül előállítja. A TCP szerver ilyen fejléceket fogad, innen is az elnevezés.



Algoritmus

A szerverre első sorban rá kell csatlakoznia az MCU-nak. Ezután a mikrokontroller folyamatosan byteokat küld ki magából, ennyi az ő feladata (ennek részleteiről később). A szerver folyamatosan figyeli hogy jött-e adat és a korábban leírtaknak megfelelően kezeli az adatot illetve a „data.txt” filet.

MCU algoritmusa

Bevezető

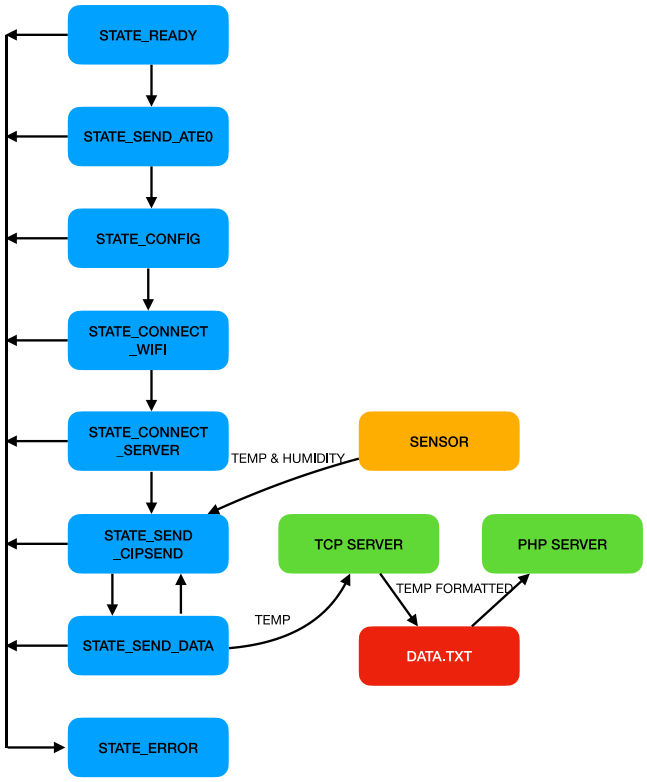
Az algoritmus FSM vezérelt és a különböző állapotokat tulajdonképpen az ESP állapotai jelentik. Ahogy korábban említésre került, az ESP-t AT commandok segítségével lehet vezérelni. Fontos megemlíteni, hogy az AT commandok úgy vannak megírva, hogy minden parancs hatására visszajelzést küldenek a végrehajtás állapotáról. Legtöbbször ez az „OK\r\n” stringet jelenti sikeres végrehajtás esetén. Bizonyos esetekben visszaküldheti az „ERROR\r\n”, vagy „busy p...\r\n” stringet. A válaszok parancsfüggőek. Az is megemlítendő, hogy alapesetben a kapott parancsokat az ESP visszaküldi a soros portján. Van AT parancs ami ezt kikapcsolja, erről később.

Hibakezelés

A szoftver rendelkezik hibakezelő képességgel. Hiba akkor keletkezik, ha rossz választ kapunk vissza, illetve ha 5 másodpercen keresztül nem kapunk választ. Az utóbbi esetben valószínűleg az ESP modul fizikai leválasztása történt, ugyanis 5 másodpercen belül minden általunk használt parancs valamilyen visszajelzést küld, legalább egy „ERROR\r\n”-t. A hibakezelés ilyenkor két módon történhet: 1,

Megpróbáljuk újraindítani az ESP-t és újrakezdeni az algoritmust, 2, Újraindítjuk a mikrokontrollert.

Folyamatábra



Az ábrán kék az MCU-t, sárga a hőmérőt, zöld a számítógépet és piros az adatbázist jelöli. Az MCU minden esetben a STATE_READY állapotból indul és minden (kék) állapotból el lehet jutni a STATE_ERROR, hibakezelő állapotba, ugyanis hiba bárhol, bármikor előfordulhat. A STATE_READY állapotban megvárjuk hogy az ESP bebootoljon, ilyenkor a sikeres bootolást a „ready\r\n” stringgel jelzi. Ezután egy „ATE0\r\n” paranccsal megkérjük, hogy a további parancsokat ne küldje vissza a soros porton. Ez megkönnyíti az ESP visszajelzéseinek a követését. A STATE_CONFIG állapotban beállítjuk, hogy majd WIFI-re csatlakozzon és hogy képes legyen az adatokat továbbítani is azon. Ezután csatlakozik a WIFI-re a STATE_CONNECT_WIFI állapotban. Wifire való csatlakozás után már képes (ha el van indítva a szerver) csatlakozni a szerverre. Sikeres csatlakozás után másodpercenként lekéri a hőmérsékletet a hőmérőtől az STATE_SEND_CIPSEND állapotban és továbbítja a STATE_SEND_DATA állapotban. Ez a két állapot váltogatja egymást egész addig, amíg hiba nem következik be. Az adatot ezután ugyebár a szerver lementi, a weboldal pedig megjeleníti.

Kikötés

Kiemelten fontos, hogy az algoritmus mindig 5 byte-ot továbbít. Ez azt jelenti, hogy a hőmérséklet legyen nagyobb mint 10°C és kisebb mint 100 °C. Az algoritmus levágja az első tizedesjegy utáni digiteket, így mindig 5 byteon továbbítható adatokat kapunk, hisz „xx.xq” formátumú minden továbbítandó adat az előbbi feltételezéssel élve.

Kimenet

Kimenet alatt jelenleg a weboldal megjelenített képét értjük. Látni, hogy a formázás és az adattovábbítás sikeres.

Contents of data.txt:

```
2023-05-29 17:03:19  25.1 ℃
2023-05-29 17:03:20  25.1 ℃
2023-05-29 17:03:21  25.1 ℃
2023-05-29 17:03:22  25.1 ℃
2023-05-29 17:03:23  25.1 ℃
2023-05-29 17:03:24  25.1 ℃
2023-05-29 17:03:25  25.1 ℃
2023-05-29 17:03:26  25.1 ℃
2023-05-29 17:03:27  25.1 ℃
2023-05-29 17:03:28  25.1 ℃
2023-05-29 17:03:29  25.1 ℃
2023-05-29 17:03:30  25.1 ℃
2023-05-29 17:03:31  25.1 ℃
2023-05-29 17:03:32  25.1 ℃
2023-05-29 17:03:33  25.1 ℃
2023-05-29 17:03:34  25.1 ℃
2023-05-29 17:03:36  25.1 ℃
2023-05-29 17:03:37  25.2 ℃
2023-05-29 17:03:37  25.2 ℃
2023-05-29 17:03:39  25.2 ℃
2023-05-29 17:03:40  25.2 ℃
2023-05-29 17:03:41  25.2 ℃
2023-05-29 17:03:42  25.2 ℃
2023-05-29 17:03:43  25.3 ℃
2023-05-29 17:03:44  25.3 ℃
2023-05-29 17:03:45  25.3 ℃
2023-05-29 17:03:46  25.3 ℃
2023-05-29 17:03:47  25.3 ℃
```

Továbbfejleszthetőség

Ez a project elég sok komponensből áll, így sok helyen lehet még jobbá tenni. A sok ötlet kitalálását az olvasóra bízom, de néhányat megemlítek.

Az első ötlet ami mindenkinek eszébe juthat, hogy egy txt file nem is igazi adatbázis. Ezt lehetne fejleszteni és igaza is van az olvasónak. A legnagyobb hiányt itt az adatbázis mivolta jelenti. A specifikáció említi a visszanézhetőséget. Ugyan a txt file ezt teljesíti, de a keresés benne kényelmetlen és nem mondható ergonomikusnak. Egy rendes adatbázis implementálása a legjobb ötlet e projekt esetében.

Egy másik ötlet a TCP szerver többfelhasználós csatlakozási képességének támogatása. Jelenleg csak 1 felhasználót tud kezelni és ez elég is, hisz egy hőmérő van. Egy kiterjesztett rendszerben több hőmérő is lehet, több forrásból is jöhet adat egyszerre. Ennek a támogatása egy logikus és hasznos kiegészítés lenne.

A többi ötlet kitalálását, mint említettem, az olvasóra bízom.