

Beléptető rendszer

Önálló laboratórium téma – Dokumentáció

Szerző: Varga Péter

Konzulens: Dr. Iváncsy Szabolcs

2023. 05. 29.

TARTALOM

I. Specifikáció	2
a) Feladat megfogalmazása	2
b) Megoldás felépítése.....	2
c) Egységek közötti kommunikáció megvalósítása	2
d) Rendszer felépítésének blokkvázlata.....	3
II. Megoldás hardver és szoftver egységeinek bemutatása	5
a) Központi egység	5
Hardver	5
Szoftver	9
Képek	11
b) Távoli egység.....	12
Hardver	12
Szoftver	18
Képek	20
c) Központi és távoli egység közötti kapcsolat.....	21
d) Számítógépes felület.....	22
Cél	22
Utasítások szintaktikája, leírása.....	22
III. Használat bemutatása, tesztelés	24
IV. Tapasztalatok.....	29

I. SPECIFIKÁCIÓ

a) Feladat megfogalmazása

A feladat egy mikrokontroller alapú beléptető rendszer tervezése, amelynek egységei RFID technológia segítségével azonosítják a használóikat. A rendszer azonosítást és kapu kezelést végző egységei, ún. távoli egységek, vezeték nélkül, rádiós kapcsolaton keresztül kommunikálnak egy központi egységgel, amely kezeli a rendszerben tárolt adatok szétosztását az egyes távoli egységek között.

b) Megoldás felépítése

A rendszer 3 fő egységből áll: távoli egységek, központi egység, számítógépen elérhető kezelő felület. A távoli egységek WiFi hálózaton keresztül kommunikálnak a központi egységgel, a WiFi hálózatot a központi egység hozza létre. A központi egység USB-n keresztül kapcsolódik a számítógéphez.

A távoli egység rendelkezik RFID olvasóval és egy galvanikusan leválasztott, vezérelt kapcsolóval, amely 12V feszültséget képes kapcsolni. Az előző két eszköz valósítja meg a beléptető funkciót. Az egység akkumulátorról üzemel, továbbá képes a tárolt adatok és idő megőrzésére akkor is, ha a tápfeszültség megszűnik. Képes továbbá tárolni a belépési engedélyeket és feljegyzést a használatáról.

A központi egység kezelő felületet biztosít a rendszer számára. Fel van szerelve nyomógombokkal, LED-ekkel és egy kétsoros LCD kijelzővel. A működéséhez 5V DC tápot igényel, de képes a tárolt adatait tápfeszültség nélkül is megőrizni. A működése során tárolja a belépés engedélyezéséhez szükséges adatokat, továbbá a távoli egységektől begyűjtött feljegyzéseket. A USB interfészen keresztül virtuális soros porton kommunikál a számítógéppel.

A számítógépes felület hatékony kezelést biztosít a beléptető rendszer adatbázisához, lehetőséget biztosít a beléptetési adatok szerkesztésére, továbbá képes a központi egységen gyűjtött és tárolt működési feljegyzések olvasására és megjelenítésére.

c) Egységek közötti kommunikáció megvalósítása

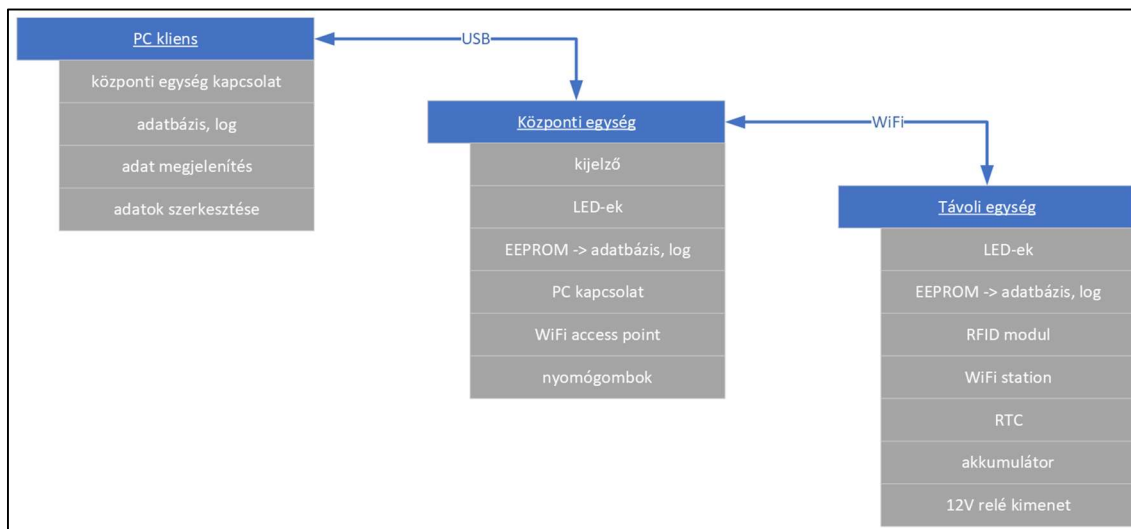
A távoli egység minden központi egységgel történő kapcsolatfelvétel alkalmával frissíti adatbázisát, és átküldi a működési feljegyzéseit. A távoli egység adott frekvenciával (pl. 24 óra)

megkísérli felvenni a kapcsolatot a központi egységgel, ezen túl csak akkor kommunikál, amikor újraindul.

Az távoli egységek belső órájukat a központi egységben tárolt idő szerint frissítik, amennyiben az be lett helyesen állítva egy kezelő személy vagy a számítógépes program által.

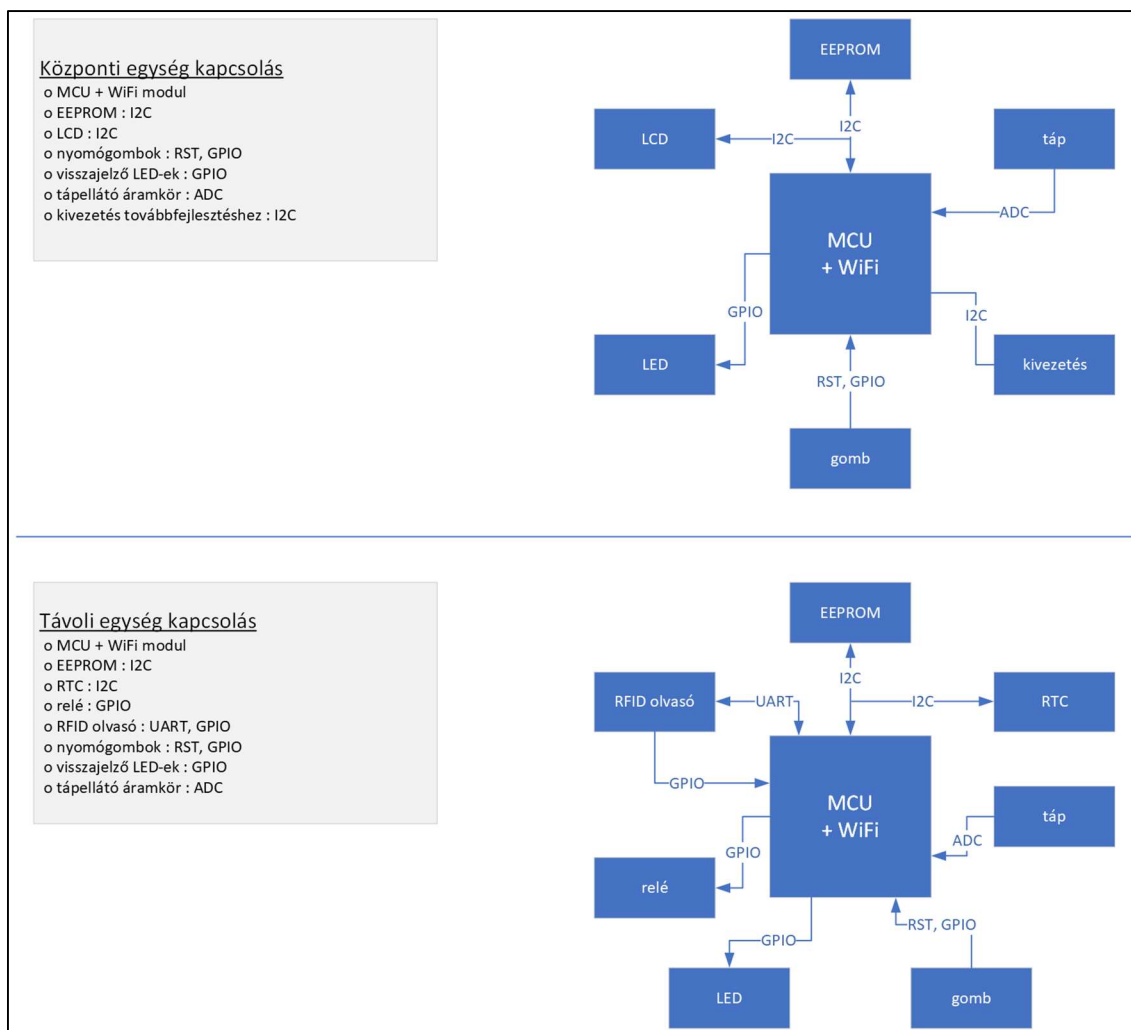
d) Rendszer felépítésének blokkvázlata

Vázlat a modulok alegységeiről és egymással való kapcsolatáról:



Ábra 1: Modulok és kapcsolataik

Vázlat a modulok hardveres felépítéséről:



Ábra 2: Modulok felépítése

II. MEGOLDÁS HARDVER ÉS SZOFTVER EGYSÉGEINEK BEMUTATÁSA

a) Központi egység

HARDVER

A központi egység főbb hardver részei:

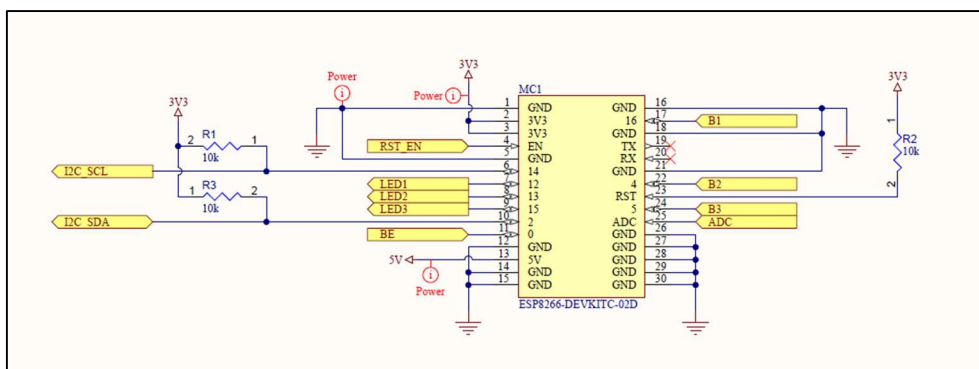
- Mikrokontroller és WiFi modul (ESP8266 DevKitC: ESP-Wroom-02D alapú fejlesztői kártya): Ez az egység felel a modul vezérléséért, és biztosítja a kommunikációt WiFi-n és USB virtuális soros porton keresztül.
- EEPROM (24LC64: 64 kbit EEPROM, I2C kommunikáció): Ez az egység tárolja a belépési adatokat és használati feljegyzéseket táp kimaradás esetén is. A kiválasztott EEPROM-ba 145 belépési engedélyt és 273 használati feljegyzést lehetséges eltárolni. (A tárolási kapacitás a szoftver fejlesztése közben lett meghatározva.)
- 1602-es, 32 karakteres LCD kijelző (LCM-1602 LCD és PCF8574 I2C busz illesztő modul): Ez az egység teszi lehetővé szöveges adatok megjelenítését a helyi felhasználói felület számára.
- Tápellátó áramkör (tolókapcsoló táp megszakításához, biztosíték, védő dióda): Ez az egység felel a megfelelő tápellátásért és az áramkör védelméért.
- I2C busz (egy-egy 10 k Ω -os felhúzó ellenállás az SCL és SDA vonalakon).

A modulhoz tartozik még néhány nyomógomb és LED a felhasználóval való kommunikáció érdekében. A nyomógombokon RC szűrőt alkalmaztam pergésmentesítésre. Beépítésre került még egy csatlakozási pont, amely lehetővé teszi a modul bővítését (táp és I2C csatlakozás).

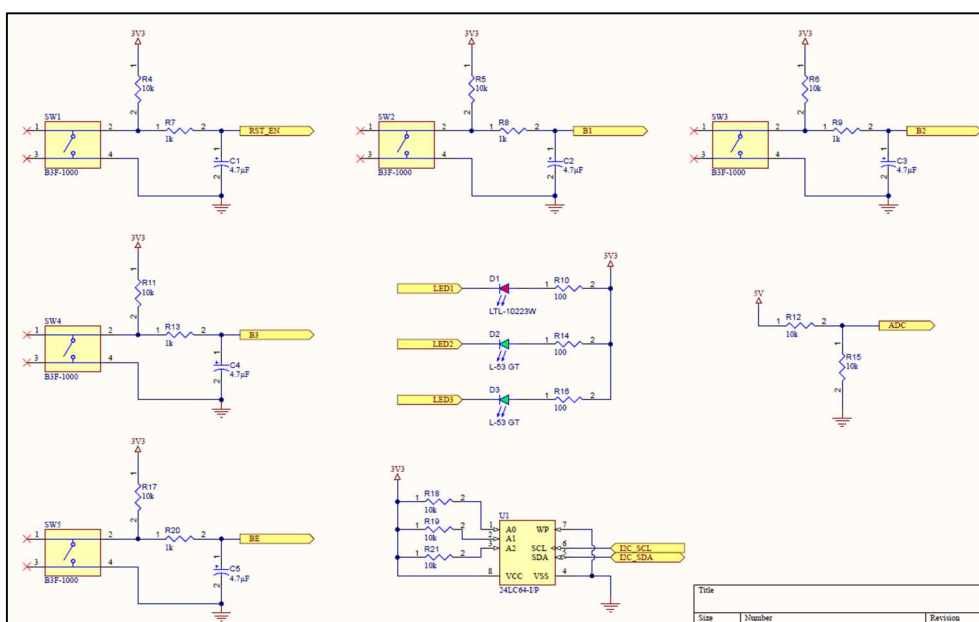
A komponensek kiválasztásánál fontos szempontok voltak az egyszerű illesztés és kezelés, továbbá a könnyű forraszthatóság. Ezen szempontok miatt használtam összetett modulokat, és furatszerelt alkatrészeket.

A kapcsolási rajzot és a nyomtatott áramkört az Altium Designer szoftver segítségével terveztem meg.

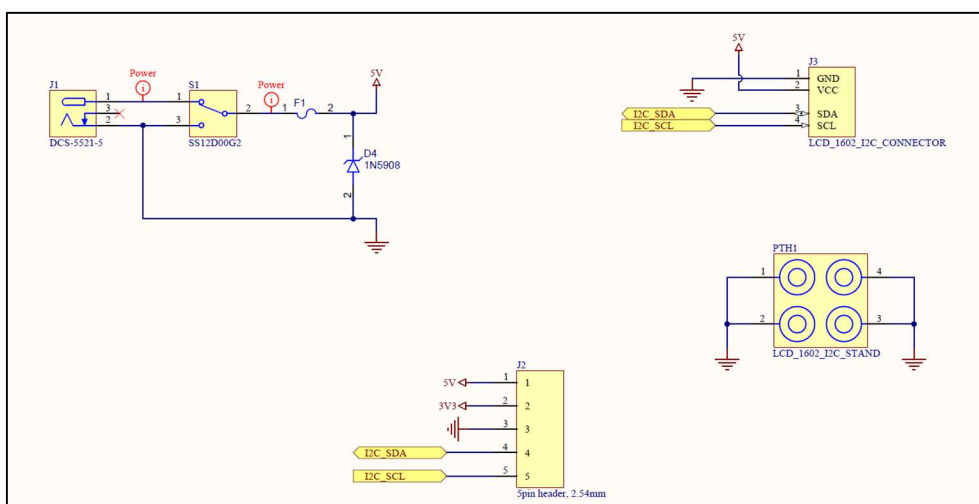
Kapcsolási rajz:



Ábra 3: Központi modul kapcsolási rajz: 1. oldal



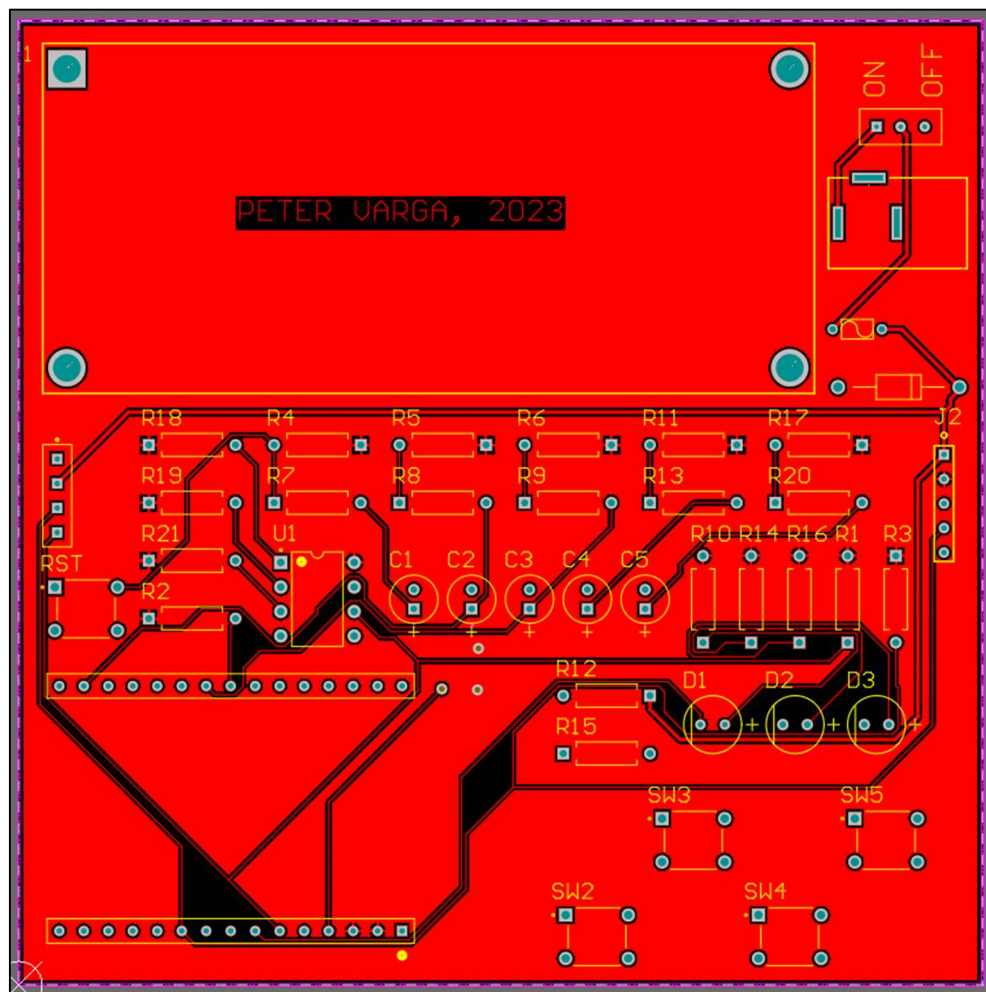
Ábra 4: Központi modul kapcsolási rajz: 2. oldal



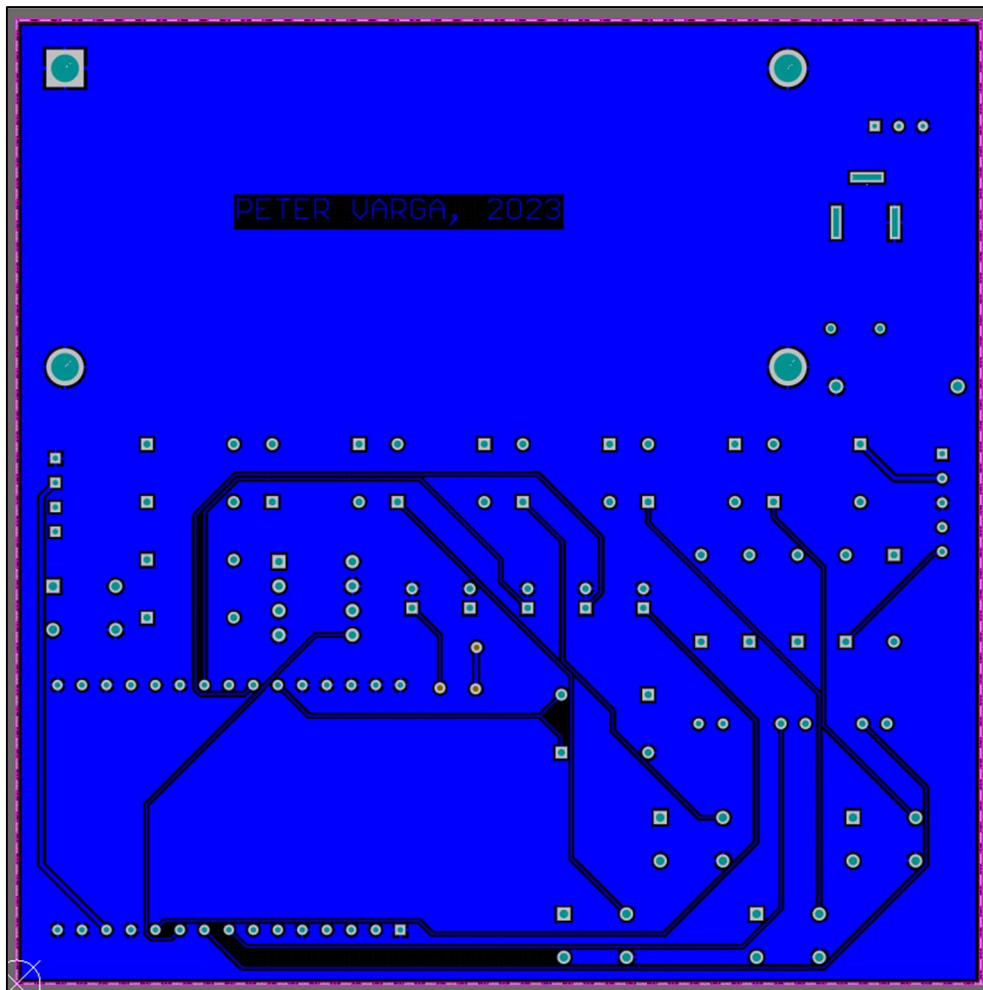
Ábra 5: Központi modul kapcsolási rajz: 3. oldal

A felhasznált komponensek teljes listája és a kapcsolási rajz PDF formátumban mellékletként elérhető: `alkatreszek_listaja.txt`, `belepteto_rendszer_kozponti_schematic.PDF`.

Nyomtatott áramkör tervek:



Ábra 6: Központi modul NYÁK terv: TOP



Ábra 7: Központi modul NYÁK terv: BOTTOM

A nyomtatott áramkörön a komponensek elhelyezésénél a fő irányelvek a következők voltak: a csatlakozók legyenek a lemez széléhez közel, nyomógombok (reset kivételével) jól különüljenek el a többi komponenstől, a tápot megszakító tolókapcsoló legyen az egyik sarokban.

Tapasztalatok az összeszerelés után:

- A vezérlő modulhoz rossz footprint-et használtam, ezért a modul nem illeszkedik közvetlenül a foglalatba. Ezt a hibát egyszerűen, kábelek közbeiktatásával meg lehetett oldani. A további hibák javítását is megkönnyíti ez a megoldás.
- TX és RX pinek nem lehetnek bekötve.
- A RST gomb által előállított reset jelet a vezérlő EN pin-je helyett az RST pin-re kell kötni.
- Az USB nem lehet bekötve amikor a táp tolókapcsolója OFF állásban van.
- A vezérlő 0 és 15 pin-jeit fel kell cserélni.

A fentebb felsorolt hibák többségét az okozza, hogy a fejlesztőkártyán megtalálható áramkörben felhasználják a vezérlő bizonyos lábait, ezért azok csak korlátozott funkcionalitással

vagy egyáltalán nem érhetőek el a fejlesztőkártyán kívül. Az, hogy ezeket a lábakat eredeti elnevezésükkel vezették ki, a hardver tervezése során megtévesztett.

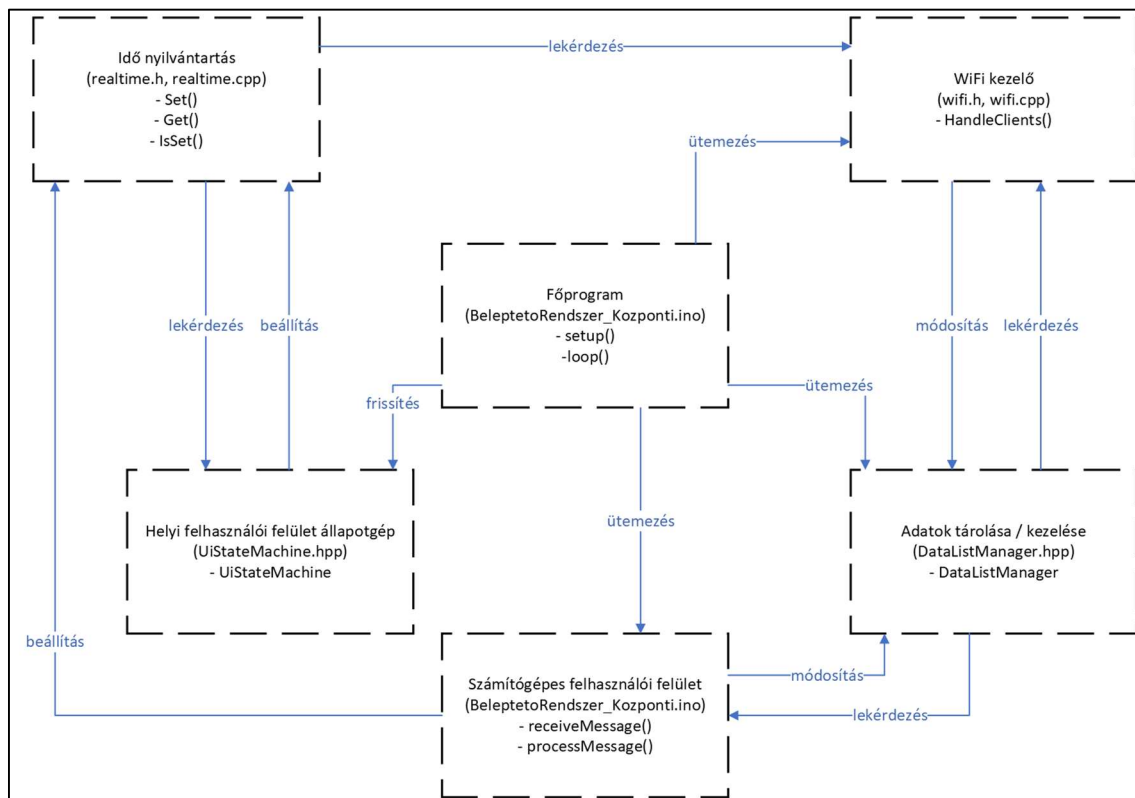
SZOFTVER

A központi modul szoftvere C++ nyelven, Arduino környezetben készült.

Szoftver által megvalósítandó feladatok:

- WiFi hálózat biztosítása a távoli modulok számára
- Távoli modulok kéréseinek megválaszolása
- Távoli moduloktól kapott adatok feldolgozása és mentése
- Belépési adatok és működési feljegyzések tárolása
- Számítógép számára hozzáférés és szerkesztési lehetőség biztosítása az eltárolt adatokhoz
- Helyi felhasználói felület működtetése
- Idő nyilvántartása

Blokkvázlat a szoftver rendszer felépítéséről:



Ábra 8: Központi modul szoftver felépítése

A szoftver rendszer főbb részei a következők:

- **Idő nyilvántartás:** A rendszer többi része számára szükséges lehet a pontos idő ismerete. A központi modul nem rendelkezik külön hardveres RTC-vel, ezért az idő nyilvántartásához egyedi megoldást alkalmaztam.
- **Helyi felhasználói felület állapotgépe:** A helyi felhasználói felület bemenetei gombok, kimenete az LCD-n megjelenített szöveg. A különböző felhasználói tevékenységek menüjét egy állapotgéppel kezelem. Ez az állapotgép felel a kijelzőre való írásért is.
- **Számítógépes felhasználói felület:** A számítógép és a modul egymással speciális felépítésű üzenetek formájában kommunikál. Ezen programrész feladata az üzenetek fogadása, az információ kicsomagolása és az információ alapján a megfelelő tevékenység végrehajtása.
- **WiFi kezelő:** A távoli egységek bármikor felvehetik a kapcsolatot a központi modullal, ekkor ennek a programrésznek a feladata, hogy válaszoljon nekik és továbbítsa a kapott adatokat a rendszer megfelelő része felé.
- **Adatok tárolása / kezelése:** Az adatokat listák formájában a RAM-ban és bináris formában az EEPROM-ban is tárolni kell. Ennek a helyzetnek a kezelésére szolgál ez a programrész.
- **Főprogram:** Inicializálja a perifériákat és a többi programrészt, kezeli a modul GPIO lábait, ütemezi a többi programrész futását.

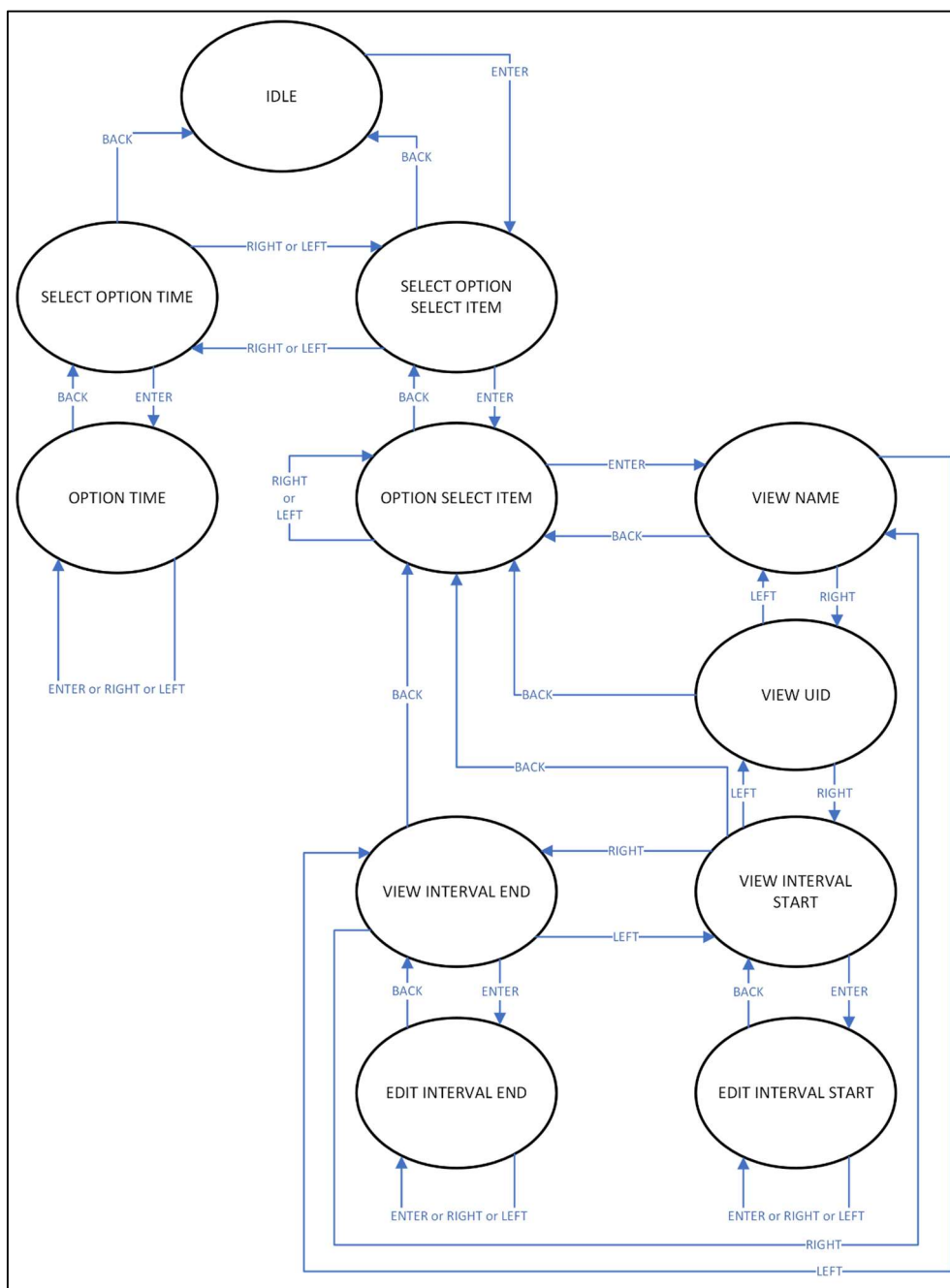
A programkód dokumentációját a forráskód kommentek formájában tartalmazza. A forráskódot a melléklet tartalmazza: „BeleptetoRendszer_Kozponti” mappa.

A használt vezérlő korlátjai miatt, ahol csak tehettem, statikusan foglaltam memóriát az adatoknak. A belépési adatokat és a használati feljegyzéseket körbufferekben tárolom. A `DataListManager` osztály felel az adatok bináris formátumra történő alakításáért és EEPROM-ba való mentéséért, továbbá a bináris formából való átalakításért is.

Az EEPROM memóriáját 3 részre osztottam:

- **Fejléc:** A memória legelején található, tartalmazza a többi rész méretét és kezdőcímét, továbbá a fejléc méretét hibakeresési okokból kifolyólag.
- **Belépési adatok:** A fejléc után kezdődő régió, a beléptetési adatok felsorolását tartalmazza.
- **Működési feljegyzések:** A memória második felét elfoglaló rész, tartalmazza a működési feljegyzések felsorolását.

A helyi felhasználói felület állapotgépe a következő módon épül fel:

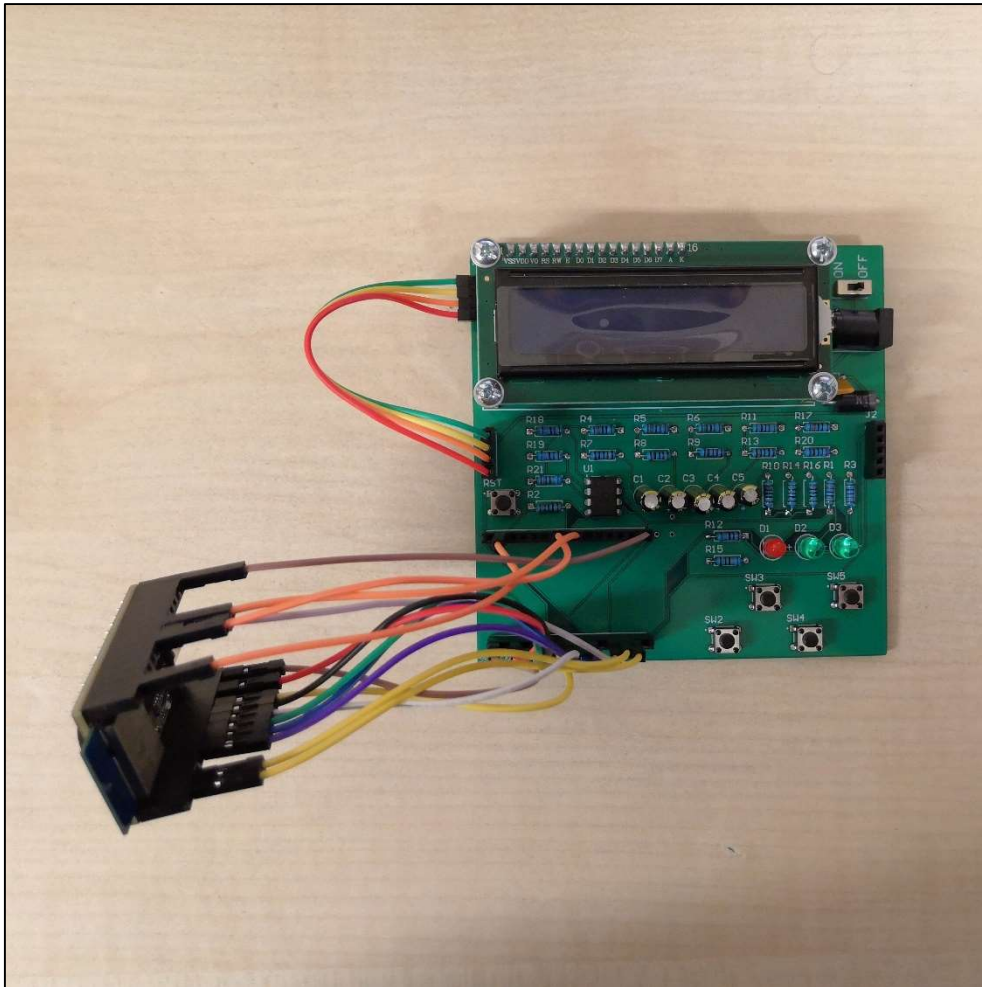


Ábra 9: Helyi felhasználói felület állapotgépe

A helyi felhasználói felület gombjait a következő módon definiáltam: ENTER – SW5, BACK – SW3, LEFT – SW2, RIGHT – SW4.

KÉPEK

Elkészült modul:



Ábra 10: Központi modul megépítve

b) Távoli egység

HARDVER

A távoli egység főbb hardver részei:

- Mikrokontroller és WiFi modul (ESP8266 DevKitC: ESP-Wroom-02D alapú fejlesztői kártya): Ez az egység felel a modul vezérléséért, és biztosítja a kommunikációt WiFi-n és USB virtuális soros porton keresztül.
- EEPROM (24LC64: 64 kbit EEPROM, I2C kommunikáció): Ez az egység tárolja a belépési adatokat és használati feljegyzéseket táp kimaradás esetén is. A kiválasztott EEPROM-ba 145 belépési engedélyt és 273 használati feljegyzést lehetséges eltárolni. (A szoftverkészítés közben lett a tárolási kapacitás meghatározva.)
- Tápellátó áramkör (tolókapcsoló táp megszakításához, biztosíték, védő dióda): Ez az egység felel a megfelelő tápellátásért és az áramkör védelméért.

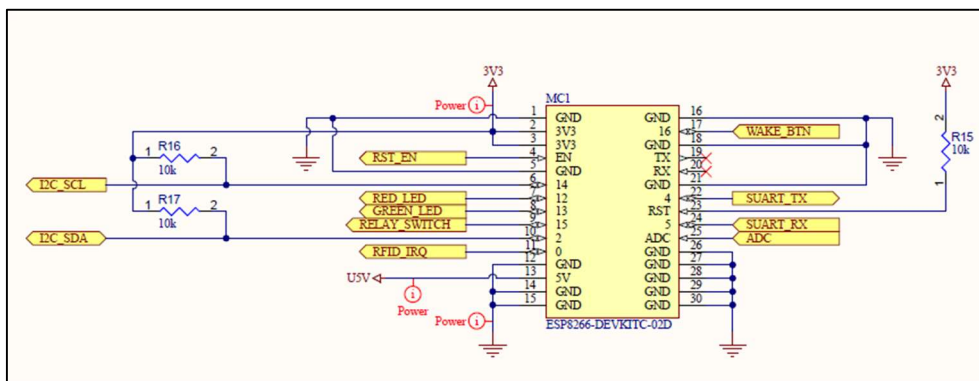
- I2C busz (egy-egy 10 kΩ-os felhúzó ellenállás az SCL és SDA vonalakon).
- RTC modul (Adafruit PCF8523): Ez az egység tápkimaradás esetén is tárolja a pontos időt.
- Relé (Omron G5V-1, tranzisztor): Ennek az elemnek a segítségével egy külső áramköri elemet lehet kapcsolni beléptetési céllal.
- RFID modul (PN532 alapú modul, a schematic-on MFRC522-ként jelenik meg): Érzékeli és olvassa az RFID tag-eket, amelyekkel azonosítja a rendszer a felhasználóit.

A rendszerhez tartozik még egy gomb a modul alvó állapotból való felébresztéséhez, továbbá néhány LED a felhasználónak való visszajelzés érdekében. A gombot a központi moduléhoz hasonlóan valósítottam meg.

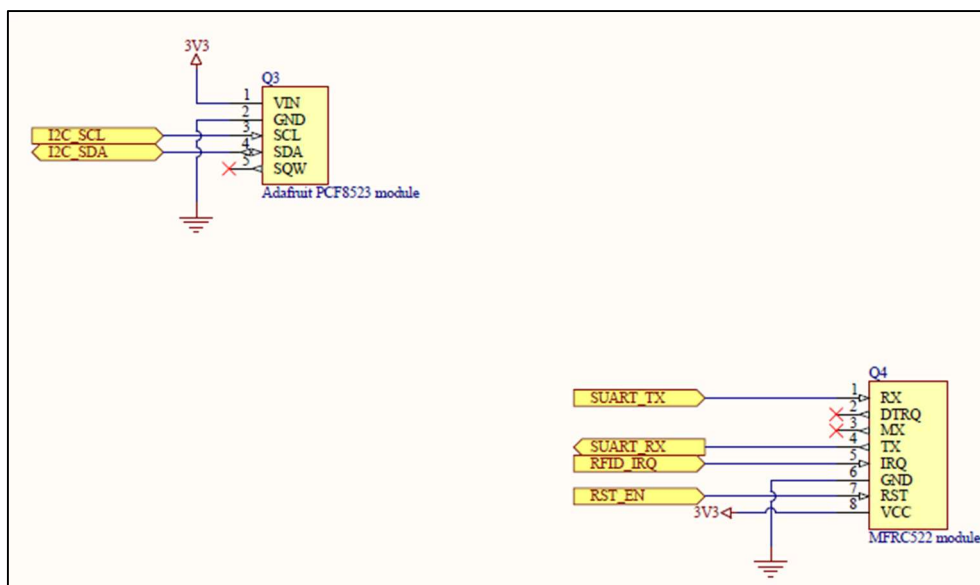
A komponensek kiválasztásánál fontos szempontok voltak az egyszerű illesztés és kezelés, továbbá a könnyű forraszthatóság. Ezen szempontok miatt használtam összetett modulokat, és furatszerelt alkatrészeket.

A kapcsolási rajzot és a nyomtatott áramkört az Altium Designer szoftver segítségével terveztem meg.

Kapcsolási rajz:



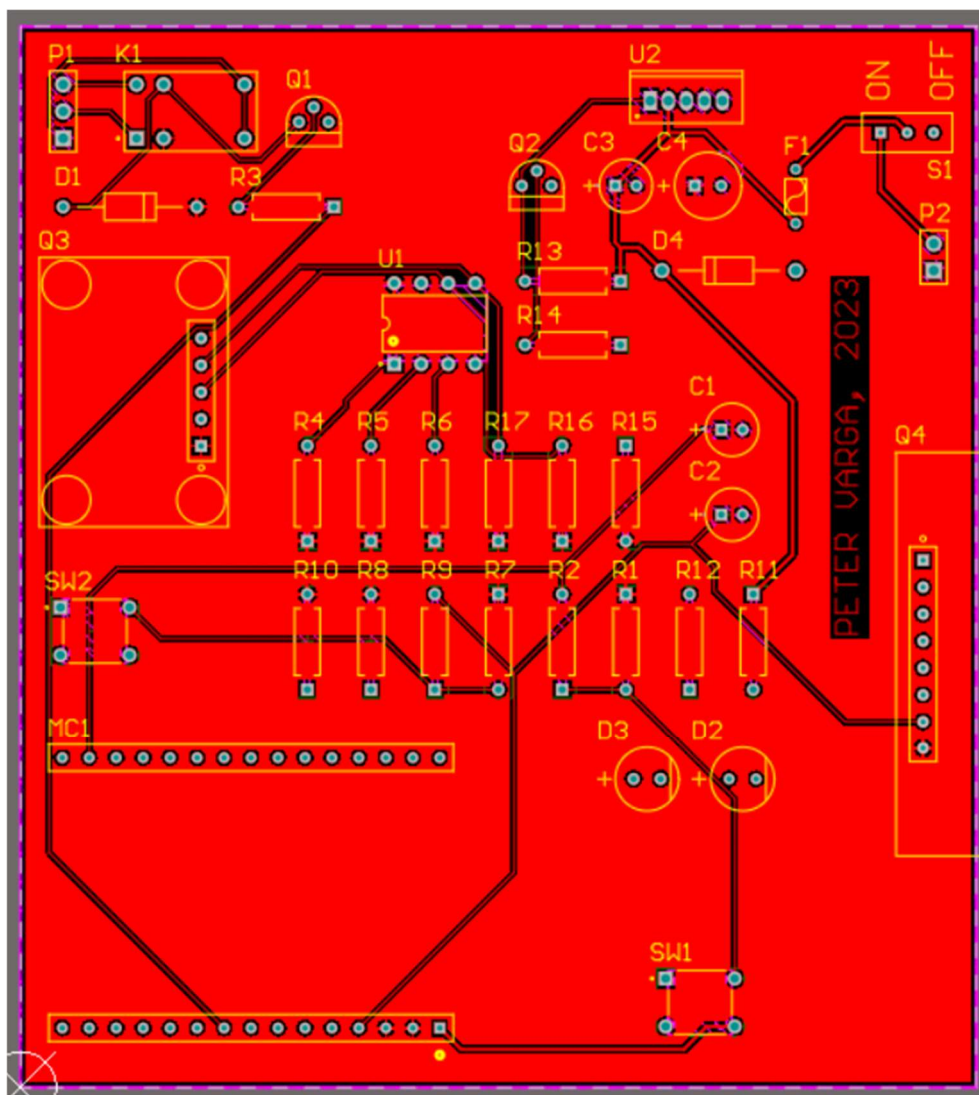
Ábra 11: Távoli modul kapcsolási rajz: 1. oldal



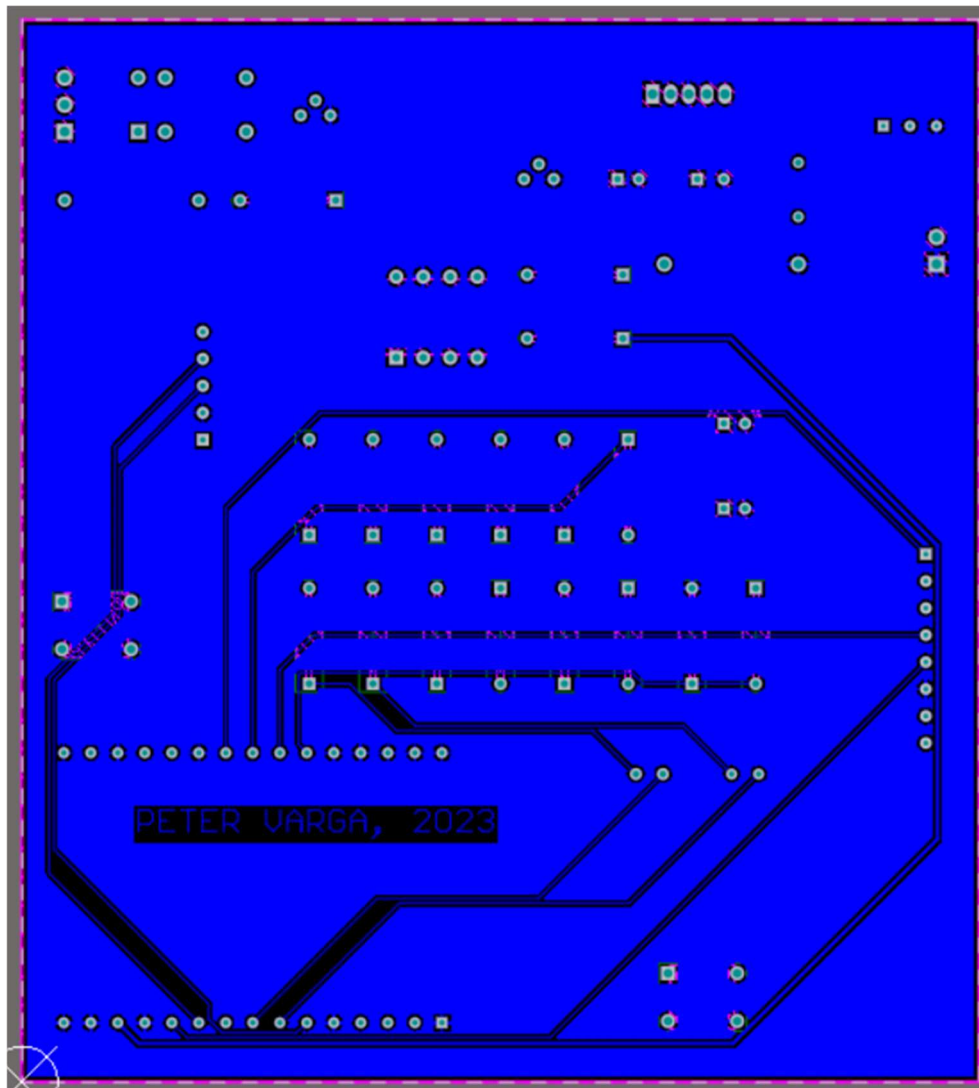
Ábra 14: Távoli modul kapcsolási rajz: 4. oldal

A felhasznált komponensek teljes listája és a kapcsolási rajz PDF formátumban mellékletként elérhető: alkatreszek_listaja.txt, belepteto_rendszer_tavoli_schematic.PDF.

Nyomtatott áramkör tervek:



Ábra 15: Távoli modul NYÁK terv: TOP



Ábra 16: Távoli modul NYÁK terv: BOTTOM

A nyomtatott áramkörön a komponensek elhelyezésénél a fő irányelvek a következők voltak: csatlakozók legyenek a lemez széléhez közel, nyomógombok (reset kivételével) jól különüljenek el a többi komponenstől, a tápot megszakító tolókapcsoló legyen az egyik sarokban.

Tapasztalatok az összeszerelés után:

- A vezérlő modulhoz rossz footprint-et használtam, ezért a modul nem illeszkedik közvetlenül a foglalatba. Ezt a hibát egyszerűen, kábelek közbeiktatásával meg lehetett oldani. A további hibák javítását is megkönnyíti ez a megoldás.
- TX és RX pinek nem lehetnek bekötve.
- A RST gomb által előállított reset jelet a vezérlő EN pin-je helyett az RST pin-re kell kötni.
- 5V pin nem lehet bekötve.
- Az eredeti tervekben szereplő MFRC522 modul helyett egy PN532 alapú modult használtam a végső megoldáshoz.

A fentebb felsorolt hibák többségét az okozza, hogy a fejlesztőkártyán megtalálható áramkörben felhasználják a vezérlő bizonyos lábait, ezért azok csak korlátozott funkcionalitással vagy egyáltalán nem érhetőek el a fejlesztőkártyán kívül. Az, hogy ezeket a lábakat eredeti elnevezésükkel vezették ki, a hardver tervezése során megtévesztett.

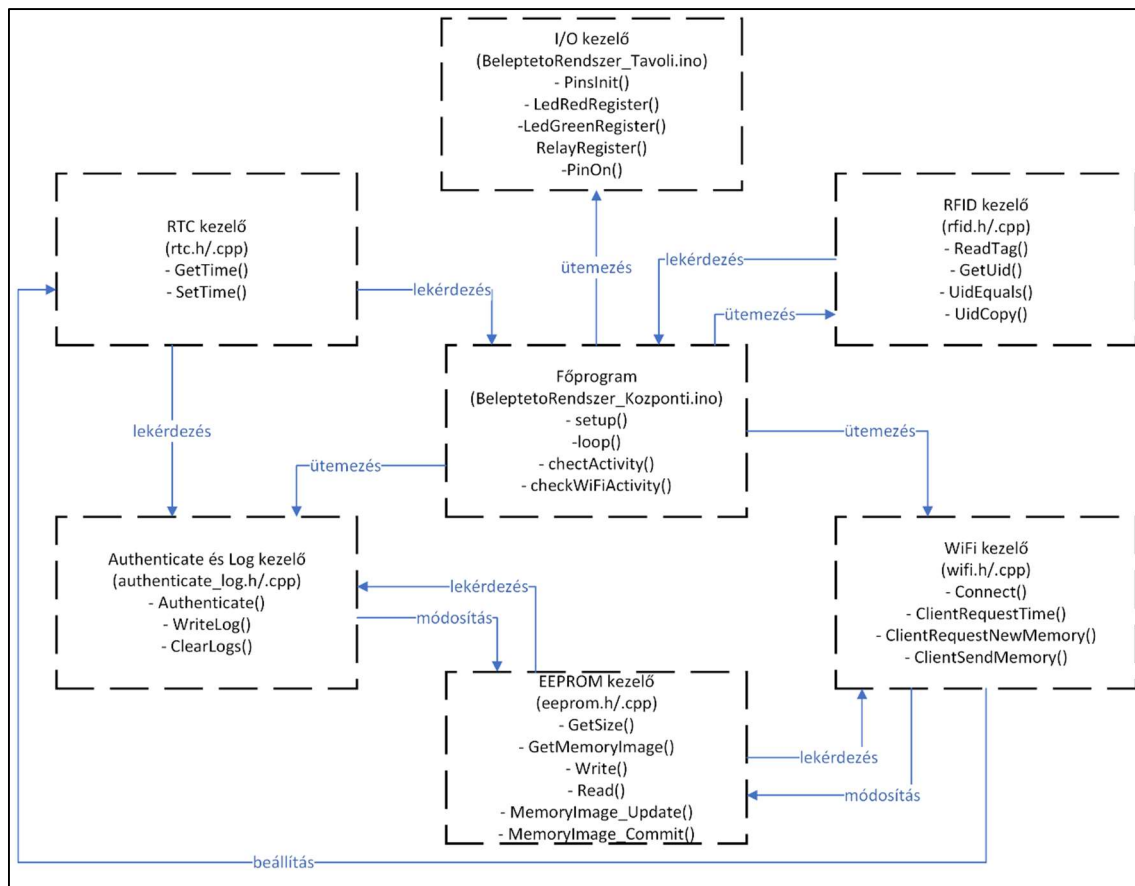
SZOFTVER

A távoli egység szoftvere C++ nyelven, Arduino környezetben készült.

Szoftver által megvalósítandó feladatok:

- RFID tag-ek olvasása.
- Felhasználó azonosítása az RFID tag-ből kiolvasott adatok alapján.
- Belépés engedélyezése a felhasználó számára, amennyiben erre jogosult a tárolt adatok alapján.
- Felhasználási feljegyzések tárolása RFID tag olvasása esetén.
- Adatok tárolása tápkimaradás esetén is.
- Pontos idő nyilvántartása.
- WiFi-n keresztüli kommunikáció a központi modullal.

Blokkvázlat a szoftver rendszer felépítéséről:



Ábra 17: Távoli egység szoftver felépítése

A szoftver rendszer főbb részei a következők:

- Idő nyilvántartás: A rendszer többi része számára szükséges lehet a pontos idő ismerete. A távoli egység rendelkezik RTC-vel, ezért a pontos időt ennek lekérdezésével állapítja meg.
- Aktivitást figyelő rész: A távoli modul az idő nagy részét alvó állapotban tölti, csak felhasználói beavatkozás (gombnyomás), vagy WiFi kommunikáció miatt ébred fel.
- a megfelelő tevékenység végrehajtása.
- WiFi kezelő: A távoli egységek egy előre meghatározott időpontban megkísérlik felvenni a kapcsolatot a központi modullal. Kapcsolatfelvétel alkalmával szinkronizálják a tárolt adatokat és az időt.
- Adatok tárolása / kezelése: Az belépési adatok és felhasználási feljegyzések az EEPROM-ban vannak tárolva. A tárolt adatokhoz való hozzáféréshez biztosít interfészt ez a programrész.
- Főprogram: Inicializálja a perifériákat és a többi programrészt, kezeli a modul GPIO lábait, ütemezi a többi programrész futását.

A programkód dokumentációját a forráskód kommentek formájában tartalmazza. A forráskódot a melléklet tartalmazza: „BeleptetoRendszer_Tavoli” mappa.

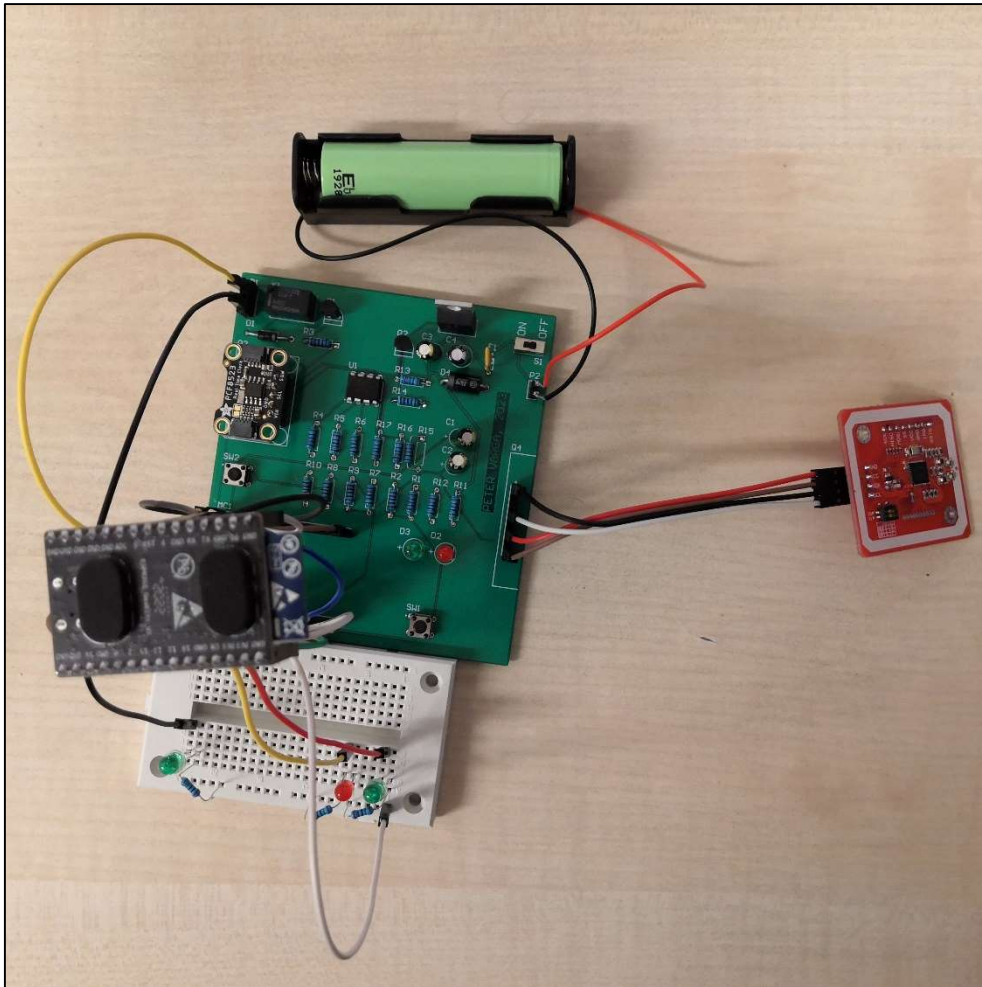
A használt vezérlő korlátjai miatt, ahol csak tehettem, statikusan foglaltam memóriát az adatoknak. A nem blokkoló időzítésekhez (pl. LED kapcsolás) implementáltam egy egyszerű szoftveres timert.

Az EEPROM memóriáját 3 részre osztottam:

- Fejléc: A memória legelején található, tartalmazza a többi rész méretét és kezdőcímét, továbbá a fejléc méretét hibakeresési okokból kifolyólag.
- Belépési adatok: A fejléc után kezdődő régió, a beléptetési adatok felsorolását tartalmazza.
- Működési feljegyzések: A memória második felét elfoglaló rész, tartalmazza a működési feljegyzések felsorolását.

KÉPEK

Elkészült modul:



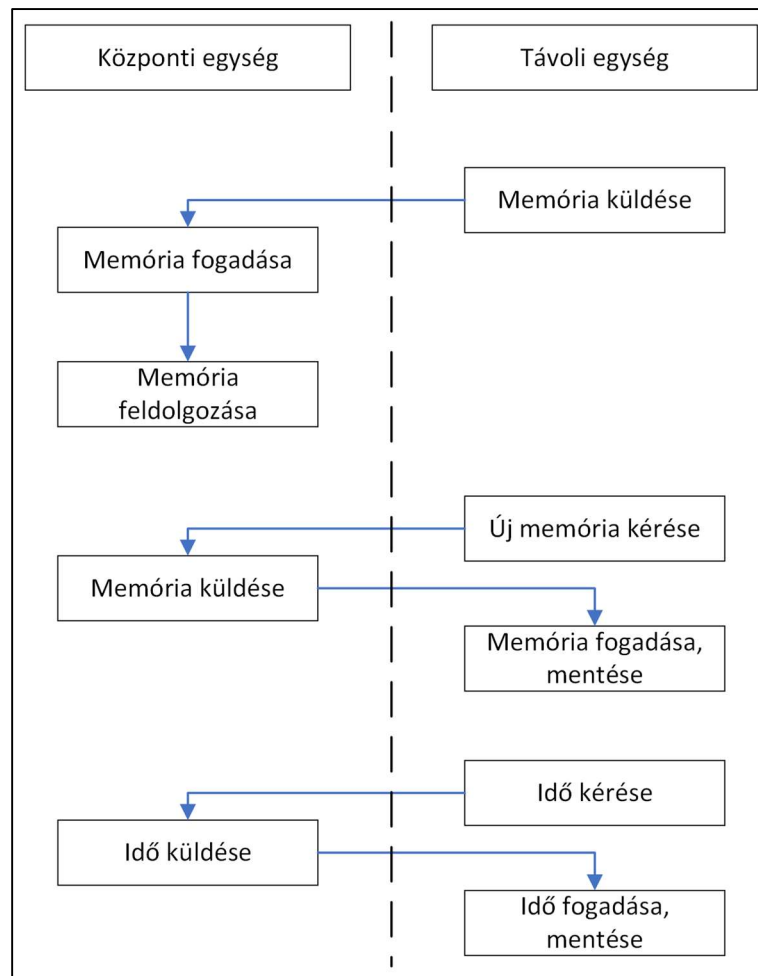
Ábra 18: Távoli modul megépítve

c) Központi és távoli egység közötti kapcsolat

A központi egység WiFi-n keresztül kommunikál a távoli egységekkel. A kommunikációt mindig a távoli egységek kezdeményezik.

A távoli egységek kéréseket küldenek a központi egység számára, amely ezekre válaszol. A következő kérések lehetségesek:

- Memória küldés: A távoli modul elküldi az EEPROM tartalmát a központi modulnak. Ebből lehet kinyerni a használati feljegyzéseket.
- Új memória kérés: A központi modul elküldi a távoli modul EEPROM memóriájába betöltetendő adatokat.
- Idő kérés: A központi modul elküldi a benne beállított időt, amellyel a távoli modul beállítja a saját óráját.



Ábra 19: Egységek közti kommunikáció folyamata

d) Számítógépes felület

CÉL

A számítógépes felület célja hozzáférés biztosítása a központi modulban tárolt adatokhoz, továbbá a rendszer konfigurálásának lehetővé tétele. A felület virtuális soros port kezelő terminál segítségével érhető el a számítógépről.

UTASÍTÁSOK SZINTAKTIKÁJA, LEÍRÁSA

Az utasításokat egy üzenet keretbe kell foglalni. A keret egy kezdő karakterből ('<') és egy záró karakterből ('>') áll. A következő felsorolásban példákkal mutatom be az utasítások típusait és használatát. A példák tartalmazzák az üzenet keretet.

- Belépési adat hozzáadása: <A RFID_UID{20} NÉV{16} KEZDŐ_IDŐ{10} ZÁRÓ_IDŐ{10}>. Az RFID_UID a felhasználó tag-jének uid-ja 10 bájton, hexadecimális formátumban. A NÉV a felhasználóhoz rendelt név. A KEZDŐ_IDŐ a felhasználó belépésre engedélyezett

idejének kezdete, Unix idő formátumban. A ZÁRÓ_IDŐ a felhasználó belépésre engedélyezett idejének vége. A kezdő és záró időknél csak az óra és perc információ kerül eltárolásra. A {}-ben lévő számérték az adott mező szélességét fejezi ki karakterekben (bájtokban). Ha a megadott UID-vel már van bejegyezve felhasználó, akkor az új adat felülírja a régit.

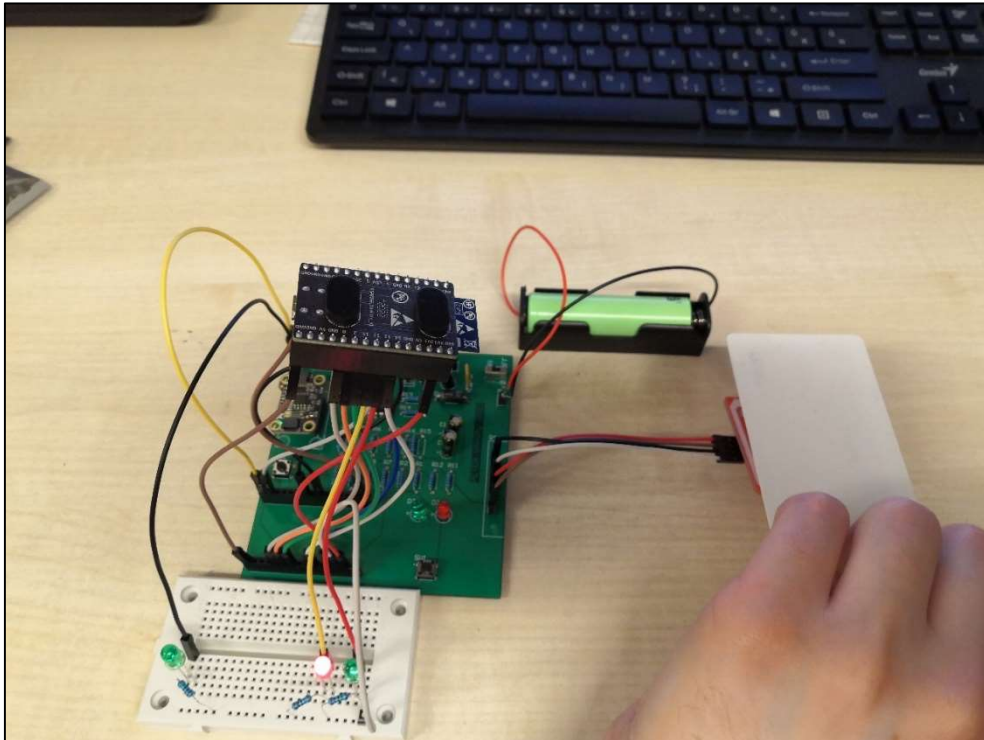
- Belépési adat törlése: <R RFID_UID{20}>. Az RFID_UID a felhasználó tag-jének uid-ja 10 bájt, hexadecimális formátumban. A {}-ben lévő számérték az adott mező szélességét fejezi ki karakterekben (bájtokban).
- Idő beállítása: <T IDŐ{10}>. Az IDŐ a pontos idő, ami alapján a moduloknak a beléptetést engedélyezéséről döntenük kell. A {}-ben lévő számérték az adott mező szélességét fejezi ki karakterekben (bájtokban).
- Használati feljegyzések listázása: <L>.
- Belépési adatok listázása: <Q>.
- Használati feljegyzések listájának törlése: <C>.

III. HASZNÁLAT BEMUTATÁSA, TESZTELÉS

A rendszer használatához az első lépés egy felhasználóhoz tartozó tag hozzáadása.

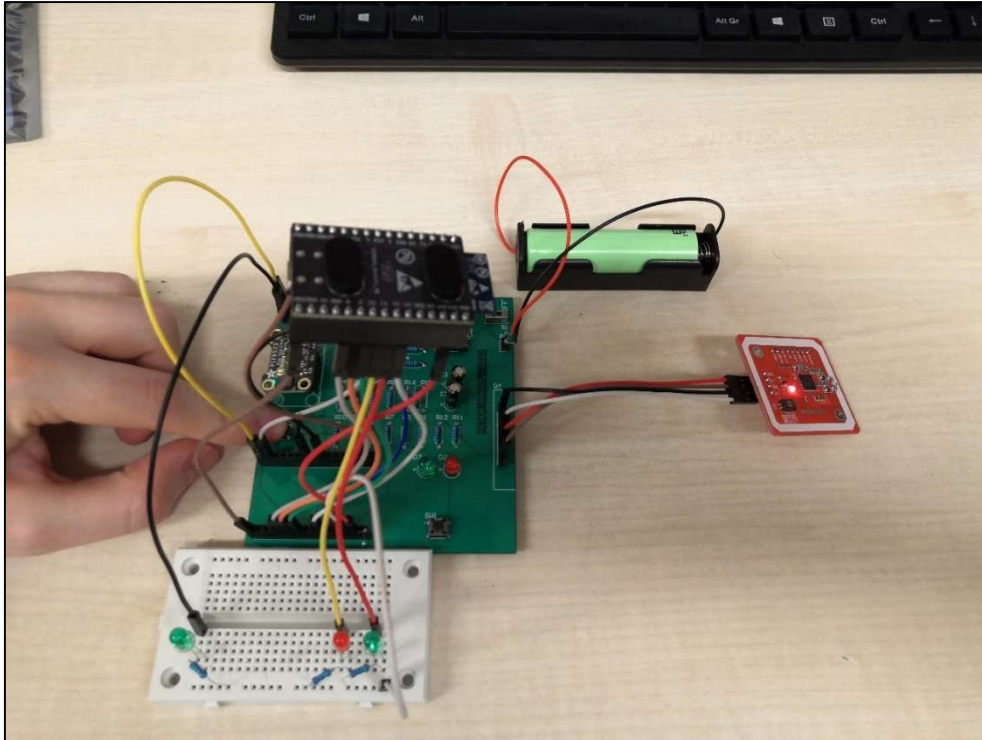
Ennek lépései:

1. Távoli modulon található ébresztő gomb (SW1) megnyomása, majd a tag olvasóhoz érintése. Ekkor a modul nem ismeri fel a tag-et, így az olvasás után a piros LED néhány másodpercig világít, és a relé nem változtatja állapotát.



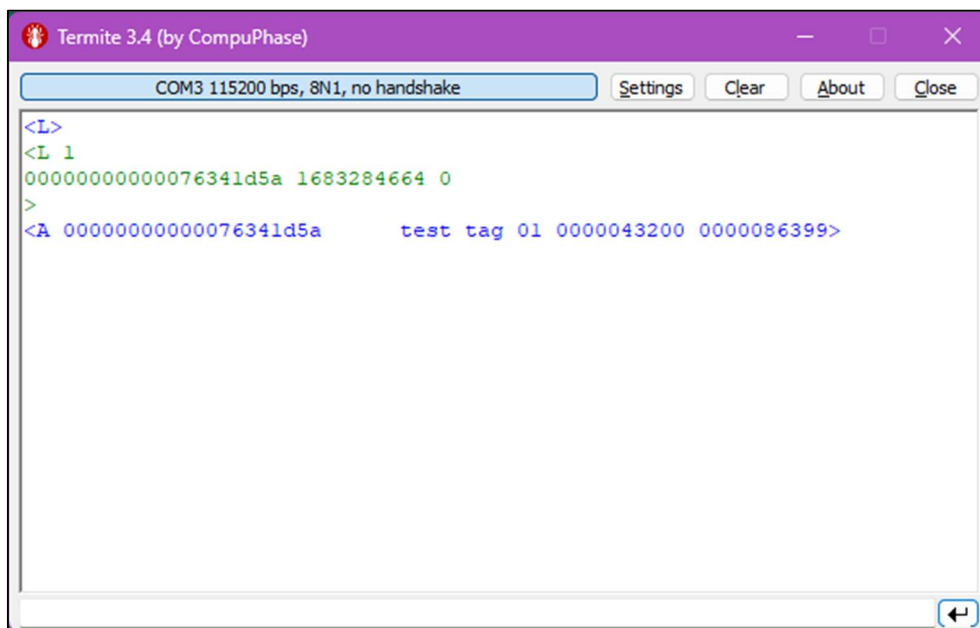
Ábra 20: Használat 1.

2. Az 1. lépés során feljegyzés készült a távoli modulban. Ahhoz, hogy ezt a központi modul azonnal megkapja, újra kell indítani a távoli modult és meg kell várni, amíg a modulok közötti kommunikáció lezajlik. Ezt a piros LED kikapcsolása jelzi.



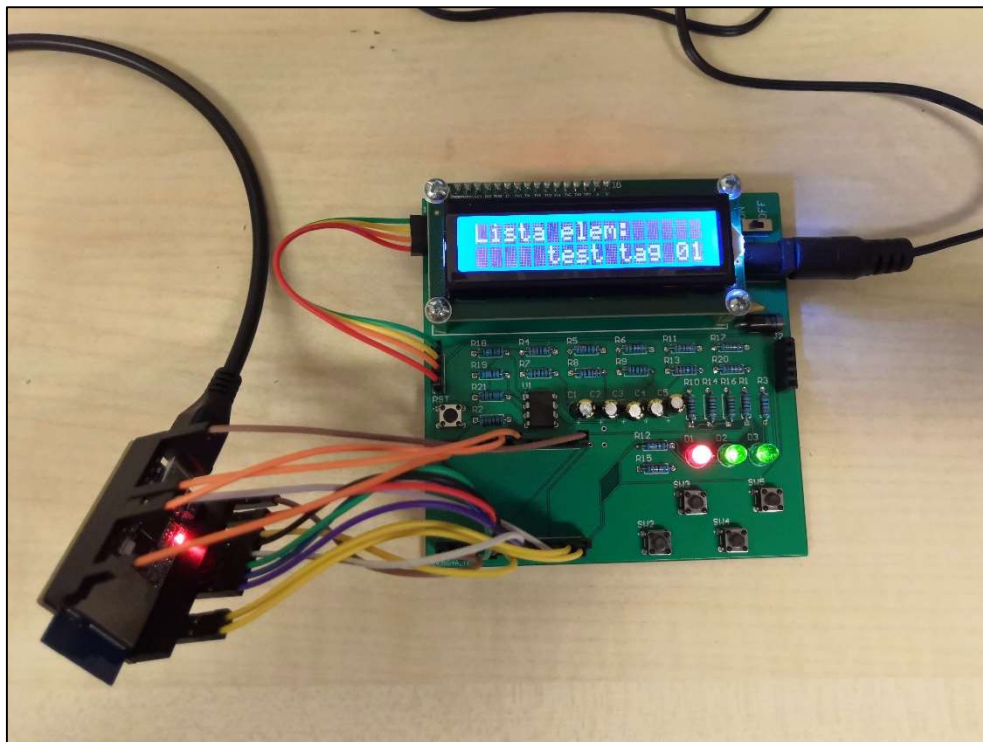
Ábra 21: Használat 2.

3. Újraindítás után, sikeres kommunikáció esetén a központi modulon meg kell jelenjen a feljegyzés. Ennek ellenőrzésére az <L> utasítással ki lehet listázni a feljegyzéseket. Az <L> utasítás arra is jó, hogy kiderítsük a tag UID-jét. A UID ismeretében a felhasználót hozzá lehet adni a belépési adatbázishoz. (A képen a kezdési idő 12:00, zárási idő 23:59.)



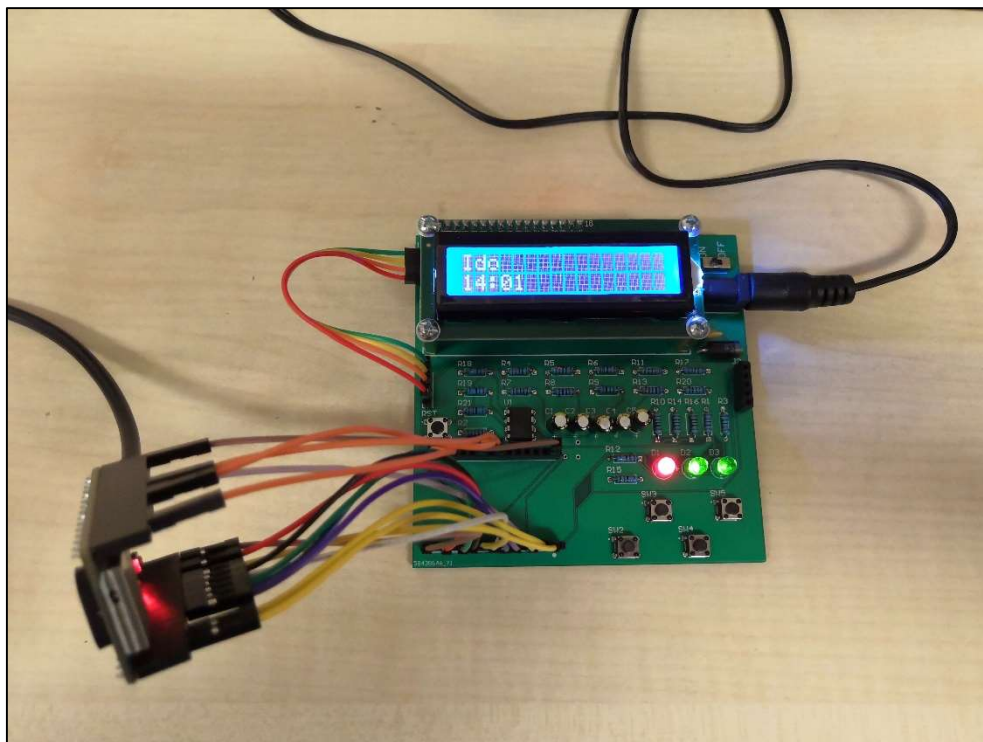
Ábra 22: Használat 3.

4. A központi modul helyi felhasználói felületén is ellenőrizni lehet, hogy a felhasználót sikerült-e az adatbázishoz adni.



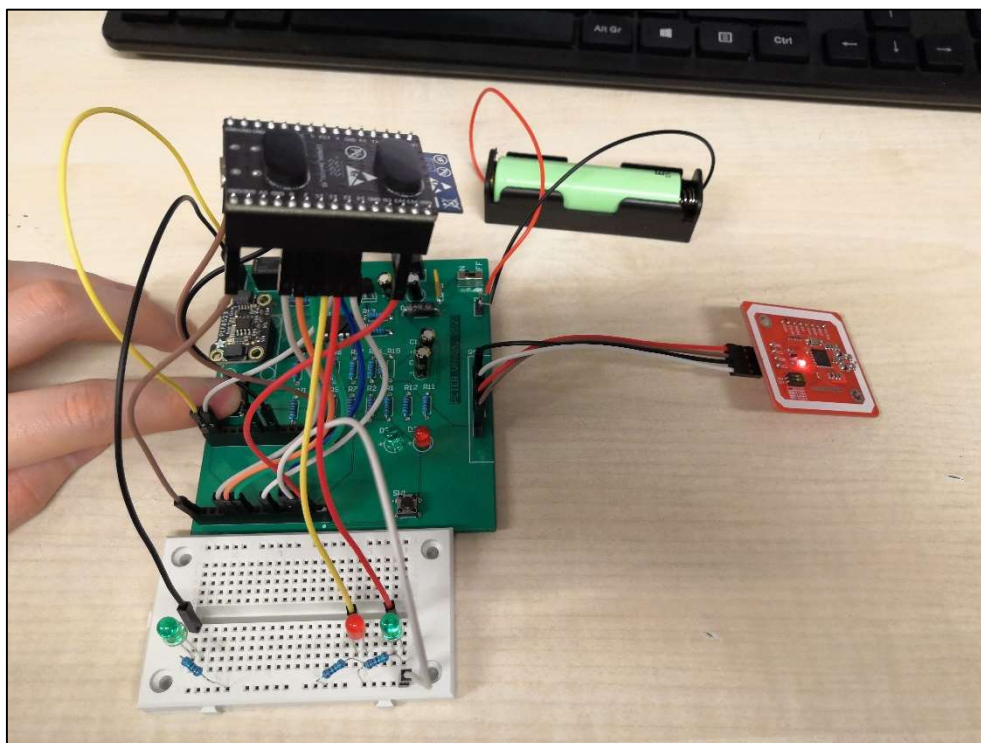
Ábra 23: Használat 4.

5. Az idő megfelelő beállítását is ellenőrizni lehet. Ha nincs megfelelően beállítva, akkor módosítani kell.



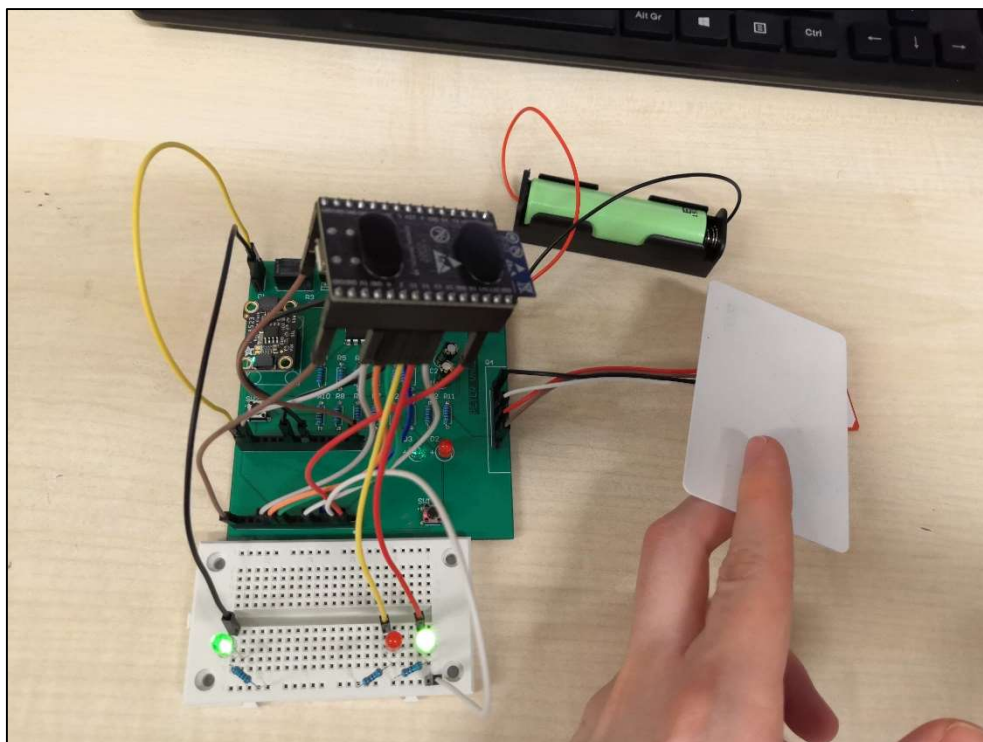
Ábra 24: Használat 5.

6. A távoli modult újra kell indítani az adatok azonnali frissítésének érdekében.



Ábra 25: Használat 6.

7. Ha az idő megfelelő, akkor az új taggal már engedélyezett a belépés. Ezt a távoli modul A zöld LED és a relé néhány másodpercig tartó bekapcsolásával jelzi. (A lenti képen a tesztpanel bal szélén lévő zöld LED világítással szemlélteti a relé bekapcsolását. A nyomtatott áramkörbe épített LED-ek helyett külső LED-eket használtam, mert a képen látható modulon fordított polaritással sikerült beforrasztani azokat.)



Ábra 26: Használat 7.

Az idő beállítása a központi modul helyi kezelő felületén a következő módon történhet:

- A központi modul gombjai segítségével el kell navigálni az „Idő” menüpontba.
- Az „Idő” menüpontban ENTER-t nyomva elérhetővé válik az idő szerkesztői felülete.
- A jelenleg szerkesztés alatt álló mező '*' karakterekkel van körülvéve.
- A LEFT és RIGHT gombokkal lehet az értéket csökkenteni vagy növelni.
- A baloldali mezőben az óra van, a jobboldaliban a perc.
- A kijelölt mezőt az ENTER lenyomásával lehet változtatni, visszalépni és menteni a BACK gombbal lehet.

IV. TAPASZTALATOK

A beléptető rendszer tervezése a legnagyobb és legkomplexebb feladat volt amin eddig dolgoztam.

A feladat megoldását a feladat követelményeinek és megoldási tervének szöveges és képi megfogalmazásával kezdtem. Ez a rész nem jelentett különösebb nehézséget.

Következő lépésként a hardvert kezdtem el tervezni, ezen belül a komponenseket választottam ki. Ez egy hosszadalmas folyamat volt, mivel nem volt semmilyen tapasztalatom elektronikai komponensek beszerzésében, illetve az alkalmazandó komponensek tekintetében csak minimális megkötést tartalmazott a feladatkiírás, ezért a „bőség zavarába” kerültem.

Miután mindkét modulhoz kiválasztottam a főbb alkatrészeket, elkezdtem tervezni a kapcsolási rajzot. A tervező szoftver használatában nem volt sok tapasztalatom, ezért ezzel lassan haladtam. A komponensek felvétele kihívást jelentett, a végső megoldásban hibák is maradtak, amelyeket csak a nyomtatott áramkörök legyártása után fedeztem fel, ezért a végső megoldásban is fellelhetőek. A használni kívánt komponensek listája ekkor került véglegesítésre.

A kapcsolási rajz elkészülte után a nyomtatott áramkört terveztem meg. Ez folyékonyabban ment már, mint a kapcsolási rajz tervezés. Ebben a részben nem találtam később hibát.

Nyomtatott áramkört életemben először rendeltem. A JLCPCB-vel gyártattam le, ez egy Kínai nyomtatott áramkör gyártó cég. A legyártott lemez az elképzeléseimnek megfelelően alakult.

Az áramköri komponenseket saját kezűleg forrasztottam fel a nyomtatott áramköri lemezre. Forrasztással kapcsolatban is nagyon kevés tapasztalatom volt, azonban végül két LED kivételével minden komponenst sikerül működőképesen beforrasztanom. A forrasztás során azt tapasztaltam, hogy a forrasztási felületeket túl kicsire hagytam, ezért az ón olykor nehezen tapadt meg rajta. További tapasztalatom, hogy a tűskesorokhoz túl kicsi furatátmérőt adtam meg. Ezt a problémát meg tudtam egyszerűen oldani azzal, hogy hüvelysortokat raktam a helyükre.

A beágyazott program készítést a távoli modul kódjával kezdem. A C++ és az objektum orientált programozás szokatlan dolog volt számomra, ezért ez a projekt többnyire „C stílusban” készült. A legtöbb időt az EEPROM kezelés és az adatok tárolásának megtervezése vette igénybe. Az EEPROM-nál törekedtem arra, hogy minél kevesebbszer írjam felül az EEPROM tartalmát, ezzel lassítva az elhasználódást.

A távoli modul esetén a munkát jelentősen hátráltatta az, hogy az RFID modul (MFRC522 alapú, ismeretlen gyártótól), amit eredetileg használni terveztem, nem működött. Sokáig próbálgattam a megjavításával, de végül a cseréje mellett döntöttem. Az új RFID olvasó egy PN532 alapú, ismeretlen gyártótól származó modul. Az áramköri kapcsolást nem kellett módosítani a csere miatt, mert az új modul az eredetileg is bekötött UART-ot használja.

A központi modul készítésekor már jobban törekedtem az objektumorientált programozásra. Itt különös figyelmet fordítottam az adatszerkezetek hatékony felépítésére, mivel több programrésznek is hozzá kell férnie a tárolt belépési adatokhoz. A helyi felhasználói felület támogatja a legalapvetőbb funkciókat. Az idő beállítási lehetőség szükséges ahhoz, hogy a központi modul teljes funkcionalitással működjön táp kimaradás után is, számítógépes kapcsolat nélkül. A jelenlegi verzió csak az óra és perc beállítását támogatja, ezért pontos dátum és idő beállításához a számítógépes felületet kell használni.

Az eredeti tervek szerint egy számítógépes felhasználói program is tartozott volna ehhez a projekthez, ez azonban az idő hiánya miatt nem készült el. Ez az oka például annak, hogy az időt Unix formátumban kell megadni a parancsokban.

Összességében a projekt megvalósítását sikeresnek ítélem. A rendszer teljesíti a specifikációját. A modulok megfelelően működnek önmagukban és összekapcsolva is. A tesztek során képes voltam hozzáadni új RFID tag-eket a rendszerhez, és az ezen tag-ekhez tartozó adatokat módosítani. A tag-eket a beállított módon detektálta és engedte be a beléptető rendszer.