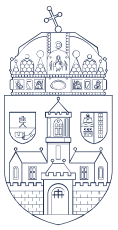


ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Alba Regia Műszaki Kar
Természettudományi és Szoftvertechnológiai Intézet

SZAKDOLGOZAT

Házi napelem rendszer termelésének vizualizálása

OE-AMK

2023

Hallgató neve: Budai Bence

Hallgató törzskönyvi száma: T001006/FI12904/A-M

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Óbudai Egyetem
Alba Regia Műszaki Kar
Természettudományi és Szoftvertechnológiai Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve: Budai Bence
Szakdolgozat száma: SZD2402130933072593[redacted]
Törzskönyvi száma: T001006/FI12904/A-M **Neptun kódja:** [redacted]

Szak: mérnökinformatikus
Specializáció: Felhő szolgáltatási technológiák és IT biztonság specializáció

A dolgozat címe: Házi napelem rendszer termelésének vizualizálása
A dolgozat címe angolul: Visualisation of the Production of a Home Solar System
A feladat részletezése: A hallgató feladata hogy egy alkalmazást fejlesszen, amely segítségével közérthető módon, grafikusan szemlélteti az adott napelem rendszer termelésének adatait, továbbá a tárolt adatok alapján állítson elő a felhasználó számára releváns statisztikai értékeket. A rendszer hibaesemény bekövetkezésekor legyen képes azt érzékelni és riasztást generálni. A hallgató mutassa be a rendszer működőképességét.

Intézményi konzulens neve: Dr. Beszédes Bertalan

A kiadott téma elévülési határideje:2026. 07. 10.
Beadási határidő: 2024. 05. 15.

A szakdolgozat: Nem titkos.

ÓBUDAI EGYETEM Alba Regia Műszaki Kar Természettudományi és Szoftvertechnológiai Intézet Székesfehérvár 1.

Kiadva: Székesfehérvár, 2024. 04. 03. P.H. .....
Intézetigazgató

A dolgozatot beadásra alkalmasnak találom: 2024. 05. 15.

..... belső konzulens

..... külső konzulens

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Alba Regia Műszaki Kar
Geoinformatikai/Mérnöki/
Természettudományi és Szoftvertechnológiai Intézet

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a dolgozat saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült dolgozatban található eredményeket az Óbudai Egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja, a titkosításra vonatkozó esetleges megkötések mellett.

Kelt: Székesfehérvár, 2024 május 9.

.....

hallgató aláírása

Tartalomjegyzék

1. Bevezetés 1
1.1 Bevezetés 1
1.2 A problémafelvetés 2
2. A megvalósítás eszközei 3
2.1 A hardver 3
2.1.1 Villamos mérőóra 3
2.1.2 Napelem 5
2.1.3 ESP32 és fotorezisztor 8
2.1.4 Szerver 14
2.1.5 Hálózat 16
2.2 A szoftver 22
2.2.1 Proxmox 22
2.2.2 MQTT broker 29
2.2.3 MySQL szerver 32
2.2.4 NodeRed környezet 36
3. A megvalósítás 44
3.1 Tervezés 44
3.2 Az adatok eltárolása 45
3.2.1 Abban az esetben amikor online a napelem invertere 45
3.2.2 Abban az esetben amikor nincs online állapotban a napelem invertere . 47
3.3 Az eltárolt adatok megjelenítése, vizualizálása 48
3.3.1 Pillanatnyi adatok 49
3.3.2 Pillanatnyi teljesítmények 50

3.3.3 Grafikonok 51
3.4 Kommunikáció 54
3.4.1 Hibaesemény detektálás 55
3.4.2 Felhasználói parancs végrehajtás 55
4. Továbbfejlesztési lehetőségek 56
5. Összefoglaló 57
6. Summary 58
7. Hivatkozások 59
8. Ábrajegyzék 61

1. BEVEZETÉS

1.1 Bevezetés

Napjainkban az otthoni felhasználásra szánt naperőművek létesítése egyre nagyobb léptékben növekszik, de felmerül a kérdés, hogy megéri-e? A szakdolgozatom segítségével próbálok rávilágítani a napelemes rendszer optimális kihasználásának lehetőségeire.

A napelemes rendszer működésének alapvetően 3 féle módja lehetséges. A termelt felesleges energia tárolásának szempontjából beszélhetünk úgynevezett sziget üzemű, hibrid, és visszatápláló napelem rendszerekről, de ezeket a módozatokat alább részletesebben is ismertetem. Mi is az a felesleges energia? A napelemek a nap sütésének függvényében termelik a villamos energiát viszont az ezúton termelt energia által ellátott épület az esetek többségében vagy teljes mértékben elhasználja, sőt még ezen felül további energiát is igényel, vagy csak egy bizonyos részét használja el a napelemek által termelt energiának.

Az ideális egyensúlyi helyzet miszerint az ellátott épület és a termelt energia különbsége pontosan nulla legyen nagyon nehezen fenntartható állapot így mindenféleképpen szükség van egy olyan energia tárolási módszerre, ami szükség szerint eltárolja a felesleges energiát és amennyiben egy olyan relációs helyzet következik be miszerint nagyobb az energiaszükséglet mint a napelem által előállított energiamennyiség azesetben az előzőekben taglalt nap folyamán eltárolt felesleges energiából kiegészítve a rendszer biztosítani tudja az igénynek megfelelő mennyiséget. Visszatérve a napelemes rendszer működésének módozataihoz, miszerint a rendszer háromféle módon működhet.

A sziget üzemű napelemes házi erőművet nagyvonalakban úgy kell elképzelni, mint egy napelemes kertben leszúrható fényforrást. Nincs szükség külső betáplálásra teljes mértékben a napfény energiájára hagyatkozik ezt pedig úgy érik el, hogy a rendszer a nap folyamán termelt felesleges energiát előre gondosan megtervezett akkumulátor egységek töltésére használja majd a nap sütésének csökkenése arányában graduálisan egyre nagyobb mértékben erre az eltárolt mennyiségre hagyatkozva megszakítás nélkül üzemel tovább és biztosítja az áramellátást.

A visszatáplálós rendszer energia tárolását ezzel szemben csak átvitt értelemben nevezhetjük tárolásnak mivel valójában nem történik energiatároló egységek töltése. Itt a felesleges energia a villamos energiát mérő egységen keresztülhaladva az áramszolgáltató rendelkezésére bocsátóik, aki eljuttatja ezt az energiát az olyan háztartások, illetve vállalatok részére, akiknek szükségük van rá. Ezt a kifele menő energiamennyiséget folyamatosan méri az áramszolgáltató. A visszatáplálós rendszer esetében nincsen helyileg eltárolt energia így az áramszolgáltató által biztosított energiából vételez abban az esetben amikor a nap energiája nem áll rendelkezésre. Az áramszolgáltató és a napelemrendszer tulajdonosa továbbiakban ügyfél között létrejött szerződés pedig arról dönt, hogy a visszatáplált energiamennyiségért cserébe milyen kedvezményben részesíti az ügyfelet. Jelen esetben a nappal termelt visszatáplált felesleges energiából kivonása kerül az áramszolgáltató hálózatából vételezett energia mennyiség és az ezen energiák különbsége adja azt az energiamennyiséget, amit a szolgáltató az ügyfél felé számláz. Ezt nevezzük a köztudatban is jelen levő úgynevezett szaladó elszámolásnak.

A hibrid üzemű napelemrendszer pedig a már említett két rendszer ötvözete. Rendelkezik helyi energiatároló berendezésekkel, de amint ezek telítődtek energiával a rendszer kvázi visszatápláló üzemmódúként üzemel tovább. A napenergia kiesése esetében pedig elsődlegesen a helyi energiatárolókból vételezi az energiát.

1.2 A problémafelvetés

A dolgozatomban a visszatápláló módozatú házi naperőmű esetét vettem górcső alá ugyanis ez a legelterjedtebb rendszertípus az alacsonyabb költség és magasabb hatékonyság miatt hisz nincsen hőveszteség az akkumulátor egység töltése közben, valamint az akkumulátorokban tárolt energiát a töltéshez és a hálózati feszültségre emeléshez is át kell alakítani, amelyek mind-mind veszteséges folyamatok. A napelemrendszer engedélyeztetése és jóváhagyása után felcsatlakoztatásra került az áramszolgáltató hálózatára és elindult a visszatáplálás. Az első hét üzemelés után kezdett körvonalazódni a probléma miszerint a ház energia fogyasztása és a termelt villamos energia nagymértékben eltérnek ezzel arra következtetve, hogy a napelem rendszer által termelt villamos energia nagy része felesleges ebből pedig az következik hogy visszatáplálásra kerül, úgymond kárba

vész az áramszolgáltató által biztosított igen alacsony felvásárlási tarifák miatt, így született az ötlet miszerint valamilyen módon optimalizálni kellene a háztartás fogyasztását a naperőműves termelés függvényében, egy olyan funkció segítségével amely közérthető módon szemlélteti az energiaáramlást, segíti a felhasználóját az optimalizálásban.

2. A MEGVALÓSÍTÁS ESZKÖZEI

A szakdolgozatom ezen részében részletesebben taglalom a projektem megvalósítását lehetővé tevő alkotóelemeket melyek az idő előrehaladtával összecsiszolódnak, eggyé olvadva pedig egy elképzelést, ötletet életre keltve egy hasznos funkcióval egészíti ki a hétköznapokat. Az informatika világában alapvetően két fő kategóriába lehet sorolni az eszközöket: Hardver és Szoftver. A hardver a fizikai eszköz mely a valós világunkban létezik, rendelkezik tömeggel, térfogattal és egyéb fizikai tulajdonságokkal melyekkel leírható és jellemezhető ilyen például a processzor, a memória, a merevlemez vagy egy mikrokontroller. A szoftver pedig a számítógép azon része, amely a számítógép nyelvén íródott algoritmusok összessége melyeket, utasítások, elágazások meghatározott sorozata alkot. Ezt a részét a számítógépnek nem tudjuk tapintani, nincs tömegük és egyéb fizikai jellemzőjük ilyen például egy kódszerkesztő vagy egy webböngésző is. Ezen eszközök és technológiák összhangban egy egységként működve segítik mindennapjainkat.

2.1 A hardver

Ebben a részben részletesebben taglalom a dolgozatom során felhasznált hardvereket, ismertetem jellemzőiket.

2.1.1 Villamos mérőóra

A mérések szempontjából kulcsfontosságú elem, eszköz a villamos mérőóra, melyet hétköznapi nevén villanyóraként szoktunk emlegetni. Ez az eszköz méri a rajta keresztülfolyó áram mennyiségét melynek mértékegysége az amper, a feszültséget, melynek mértékegysége a volt és ezen változók segítségével kalkulációkon át számolja a teljesítményt, fogyasztást, és egyéb hasznos fizikai jellemzőket az áram szempontjából. Esetemben ez

a villamos mérőóra 3 fázist tud egyszerre mérni mely fázisonként 24 amper áramerősséget képes átereszteni. Teljesítmény szempontjából ez azt jelenti, hogy P (teljesítmény) = U (feszültség) * I (áramerősség), behelyettesítve 240 V * 24 A = 5760 W (Watt) teljesítményt képes átereszteni fázisonként a hálózatot védő kismegszakító leoldása, megszakítása nélkül. Ha ez a mennyiséget megszorozzuk hárommal megkapjuk azt a maximum teljesítmény mennyiséget, amit egy adott pillanatban vételezhetünk az áramhálózatból tehát 3 * 5760 W = 17 280 W ez az a maximális teljesítmény mennyiség, amely egy adott pillanatban áramolhat a villanyórán keresztül mind fogyasztás és visszatáplálás szemszögéből.

1. ábra Villamos mérőóra

Ez a mérőóra az áramszolgáltató által biztosított Sanxing SX631 típusú villamos mérőóra mely rendelkezik egy P1 porttal, csatlakozási lehetőséggel melyen keresztül az ügyfél

hasznos adatokat nyerhet ki a mérőórából. Ez a kimenet egy soros adatátviteli technológiát alkalmazó kommunikációs port melynek sebessége 115200 Baud. Ez azt jelenti hogy egy máspdperc alatt 115200 bitet, számítógépes világ kontextusában értett legkisebb méretű adat, képes továbbítani a vevő félnek. Ez a kommunikációs interfészt egy RJ-12-es azonosítóval rendelkező csatlakozón érheti el a felhasználó. Az adatok dekódolása után elérhetővé válnak olyan adatok mint például a pillantnyi fázisonkénti feszültség, áramerősség, összes exportált valamint importált energia mennyisége kWh-ban és még sok más. [1]

Ennek a kommunikációs portnak az olvasásához szükséges egy eszköz mely a fogadott adatokat dekódolás után továbbítja a wifi hálózatot használva a további feldolgozást végző eszközök számára. Ez az eszköz pedig egy Homewizard nevezetű holland cég által gyártott P1 Meter eszköz mely az előzetes applikációból végezhető konfiguráció után automatikusan csatlakozik WiFi hálózathoz és egy API-t biztosít mely egy interfész a programok számára. Ezen API-n keresztül elérhetőek a villamos mérőóra által P1 porton továbbított előzőleg taglalt változók melyet a Lokális hálózatra csatlakoztatott bármilyen eszköz képes olvasni.

2.1.2 Napelem

A napelem mint energiatermelő eszköz sok változtatáson és kísérleten keresztül ment át mire elnyerte azt a formáját és tulajdonságait mely lehetővé tette a háztartások egyre növekvő villamos energia igényének megbízható kiszolgálását. Alapvetően a visszatápláló üzemű napelemes rendszert két fő egység alkotja ezek a többnyire a tetőszerkezetre erősített napelemek, valamint a közelében a falon elhelyezett inverter egység alkotja. A napelemek fő alkotóelemei a szilikon és a az úgymond szennyezésre használt nemesfémek és ásványok. A gyártás során felhasznált elemek és technológiák a felhasználási terület függvényében kerülnek kiválasztásra. Ilyen felhasználási terület lehet például a föld körüli pályán keringő különféle feladatokat ellátó műholdakon elhelyezett napelemek, vagy éppenséggel a föld felszínén elhelyezett napelemek is, ezek a területek több szempontból különböznek egymástól, ilyen szempont lehet például a napenergia. A Nap, mint égitest

többféle hullámhosszú sugárzást bocsát ki magából, ennek a kibocsátott energiának vannak tartományai, amelyek a szabad szemmel is láthatóak, ez a fény, azonban ez csak egy töredéke a nap által sugározott energiáknak. A nem látható hullámhosszon helyezkednek el a látható spektrumot közrefogva az Ultra Ibolya (Ultra Violet röviden UV) valamint az Infra Vörös (Infra Red röviden IR), utóbbit érzékeljük hőként, például télen egy kandalló mellett kuporogva. A napenergia, mint komplex energia ezen különböző hullámhosszú energiák egyvelege, nagyban eltér az űrben és a Föld felszínén, ennek magyarázata pedig a légkörben található ózon réteg, páratartalom és egyéb gázok, szennyeződések, amelyek a sugárzás bizonyos tartományaiban szűrőként működnek. Ezen tényezők pedig befolyásolják azt hogy a napelemet milyen sugarak érik el, milyen sugárzás elnyelésére, átalakítására kell optimalizálni a gyártás és tervezés során. Napjainkban a két fő típusú napelemet használnak a háztartások ellátására ezek pedig az úgynevezett polikristályos és a monokristályos. A monokristályos napelem arról kapta a nevét, hogy a napelemet adó szilikont egyetlen egységes kristály alkotja ezáltal nagyobb hatékonyságot és terhelhetőséget biztosít a magasabb előállítási költség rovására. Ezzel szemben a polikristályos napelem vagy másnéven multikristályos napelem alacsonyabb hatékonysággal rendelkezik mivel az előállítása egyszerűbb ezáltal az árat is alacsonyabban tartva. A két típus külsőleg onnan ismerhető fel, hogy a monokristályos napelemet több gyakran nyolcszög alakú napelem szeletecske alakít egy nagyobb panelen elhelyezve, a polikristályos napelem pedig egy egységes felületű panel. Ezek a panelek a tetőszerkezethez erősítve párhuzamos vagy soros kötésben táplálják az invertert a nap sugárzása által generált egyenárammal.

2. ábra Napelemek

Ezzel a közvetlenül a napelemek által termelt árammal több probléma is felmerül. Egyrészről az elektromos hálózatban Magyarországon 50 Hz frekvenciájú és 240V feszültségű váltakozó áram köznyelven váltóáram van jelen, ez pedig azt jelenti, hogy a fázis vezetéken ötvenszer vált az áram irányt egy másodperc leforgása alatt szemben a napelem esetével, ahol az áram iránya konstans, nem változik. Más szempontból megközelítve a napelemekből származó feszültség ingadozhat a napszak és időjárási viszonyok függvényében, ezokból kifolyólag szükség van egy olyan eszközre, amely a napelemekből érkező egyenáramú, ingadozó feszültségű villamos energiát átalakítja a háztartás ellátására is alkalmas villamos energiává melynek feszültsége stabilizált, áramirányának változása pedig adaptívan az áramszolgáltató hálózatában található frekvenciához igazított.

Az inverter mint eszköz szintén végez méréseket és adatokat szolgáltat a kimeneti, bemeneti oldalon érkező, avagy távozó villamos energiát jellemezve, valamint közvetít olyan adatokat, mint például az inverter egység hőmérséklete.

3. ábra Az inverter

Szakdolgozatom esetében az inverter és a napelemek együttesen ideális körülmények között 12kW villamos energia előállítására képesek.

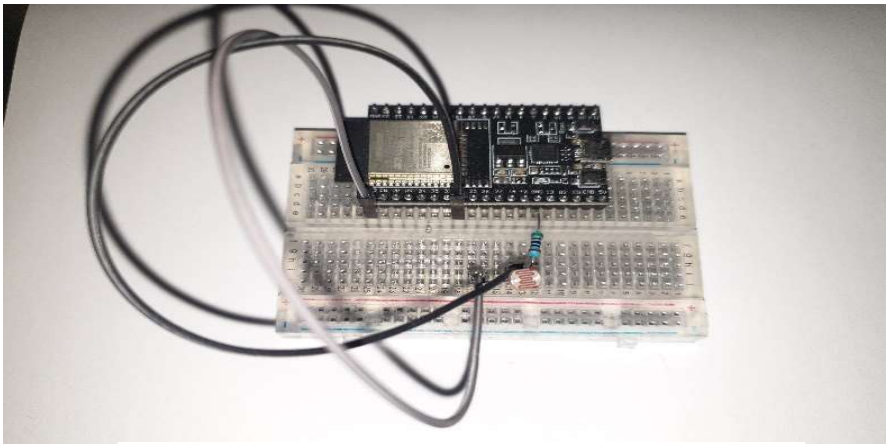
```
{
    "flg": 1,
    "tim": "20240419195136",
    "tmp": 402,
    "fac": 4997,
    "pac": 0,
    "sac": 0,
    "qac": 0,
    "eto": 148907,
    "etd": 521,
    "hto": 5569,
    "pf": 0,
    "wan": 0,
    "err": 0,
    "vac": [
        2375,
        2409,
        2397
    ],
    "iac": [
        9,
        9,
        9
    ],
    "vpv": [
        1497,
        1495
    ],
    "ipv": [
        0,
        0
    ],
    "str": []
}
```


4. ábra Inverter API

Az inverter is hasonlóan a villamos mérőórában elhelyezett eszközhöz, rendelkezik egy API interfésszel, amelyen keresztül olyan adatokat tudunk kinyerni, mint a napelemek felől érkező feszültségek, áramerősségek vpv, ipv, valamint a villamos hálózatra csatolt feszültségek és áramerősségek vac és iac néven. Fontos megjegyezni, hogy az ábrán látható adatokat el kell osztanunk tízzel a valós adat eléréséhez.

2.1.3 ESP32 és fotorezisztor

Ezt az eszközt használtam a szakdolgozatomban a fényerősség méréséhez. Azt az esetet vizsgáltam ezzel az eszközzel, hogy amennyiben fényt érzékel az eszköz elküld a fény



5. ábra ESP32 és fotorezisztor

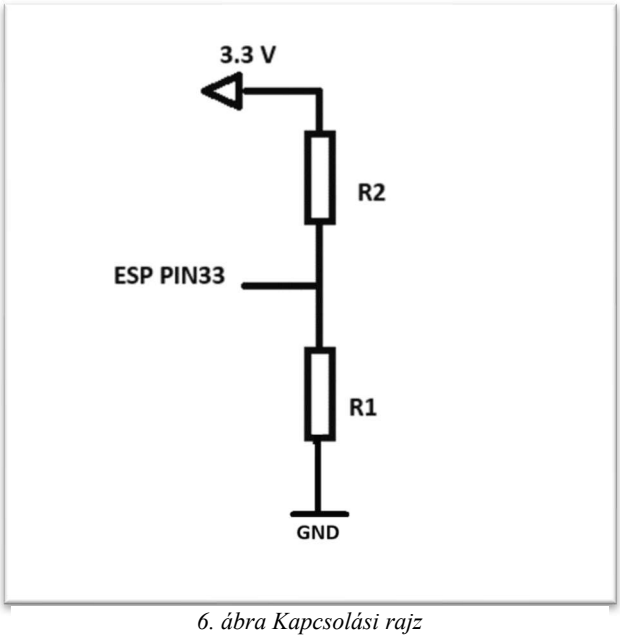
erősségével egyenesen arányos értéket a beérkező adatokat feldolgozó szerver számára így amennyiben elegendő fénymennyiség áll fent, de a napelemek nem termelnek energiát azesetben egy riasztást küld a felhasználó számára, hogy valami okból kifolyólag a napelem invertere nem termel villamos energiát, úgynevezett offline állapotba került.

Az ESP32 eszköz egy komplexebb mikrokontroller melyet az Espressif Systems vállalat gyárt. általánosságban egy mikrokontroller alatt egy olyan eszközt értünk amely rendelkezik egy processzorral amely a különféle számításokat végzi és ezek függvényében változtatja a kimeneteket, rendelkezik háttértárral melyen az előre megírt algoritmusok tárolódnak amelyek alkotják a programot amelyet a processzor olvas lépésről lépésre a végrehajtás során és rendelkezik operatív tárral melyben eltárolja a végrehajtás során keletkezett részeredményeket, elágazások kiértékeléséhez szükséges változókat végül pedig a ki és bemeneti lábakkal melyeken keresztül összekapcsolhatjuk eszközünket a külvilággal. Ezeken felül ez a chip kiegészül olyan integrált extra funkciókkal, mint például Wi-Fi konnektivitás, valamint a Bluetooth képességek, ezek pedig a kommunikáció szempontjából nagyban leegyszerűsítik a fejlesztők munkáját, ezzel számtalan programozási órát spórolva. Az ESP32 elődje egy ESP8266 névre hallgató SoC volt, mely rövidítés arra utal, hogy a rendszer melynek részei a processzor, operatív tár, kommunikációs portok, modulok egyetlen chipre vannak integrálva, ezzel helyet spórolva, átviteli sebességeket gyorsítva, hibalehetőségeket csökkentve. Az ESP32 esetében a processzor egy Tensilica

nevezetű cég által fejleszettt Xtensa LX6 típusú processzor mely 32 bites címtérrel rendelkezik és az utasításokat 160 vagy 240 MHz órajel frekvenciával hajtja végre. Operatív tár szempontjából a mikrokontroller 520 kilobájt memóriával rendelkezik, amelyet opcionálisan négy, nyolc, vagy 16 megabájt flash memóriával egészítenek ki háttértár feladatának betöltésére. Wi-Fi vezeték nélküli kommunikációt tekintve képes a 801.11 hálózati szabvány b, g és n generációit alkalmazni. A modul melyen a chip található 38 forrponttal rendelkezik mely forrpontok különböző célokat látnak el ilyen célok lehetnek például a 3,3 Voltos tápellátás, különféle ki- és bemenetek, vezetékes kommunikáció kivezetései, a chip működését engedélyező bemenet. [2] A mikrokontrolleren megtalálhatóak olyan bemenetek melyek egy ADC konverter bemeneti pontjára vannak kötve. Ez az ADC konverter lényegében egy feszültségmérő eszköz mely bizonyos időközönként végez egy mérést, ennek a mérésnek a gyakoriságát a mintavételezési frekvencia, a mérésnek a pontosságát pedig a felbontás írja le. Erre az eszközre, konverterre, azért van szükség mert a számítógépek így a mikrokontrollerek is bináris számrendszert alkalmaznak működésuk során melynek következménye, hogy két állapotú lehet egy bemenet: van feszültség tehát igen, nincs feszültség tehát nem. Ezzel az igen, nem adattal közvetlenül nem lehetséges a feszültségi szintek leírása a bemeneten mivel a feszültség bármekkora értéket felvehet a mikrokontroller pedig közvetlenül csak azt a tényt tudja érzékelni, hogy van-e feszültség, avagy nincs. Tételezzük fel, hogy a mikrokontrollerünk minimum 0 és maximum 5 Volt intervallumú feszültségi szintek értelmezésére képes, a feszültég pedig lineárisan növekszik. 0 Volt feszültségnél a bemenet 0 ahogy elkezd emelkedni a feszültség 2 Voltnál a bemenet még mindig 0 mindaddig amíg át nem lép egy küszöbértéket amelyen fellül a bemenet szaturálódik, kinyit, és a bemenet 1-re vált. Példánkban ez a feszültség legyen 2.5 Volt amely alatt a bemenet értéke 0 felette pedig 1. Ezt az előbb definiált problémát oly módon hivatott megoldani az analóg digitális konverter, röviden ADC hogy az adott pillanatban mért feszültséghez egy a felbontás alapján korlátok közé szorított arányos értéket rendel. Erre azért van szükség a szakdolgozatom során mert a fotorezisztorral alkotott feszültségosztóból távozó feszültséget veszi alapul a mikrokontroller és az ADC által hozzárendelt értéket továbbítja a szerver részére.

A fotorezisztor egy olyan speciális ellenállás mely a fény fotonjait elnyelve az elnyelt mennyiséggel arányosan változik az ellenállása, ezt a tulajdonságot pedig különböző fémek speciális kémiai reakciók által keletkezett sóival érik el.

A mikrokontroller bemenetét elvi szinten kondenzátornak tekinthetjük így, ha egy feszültségi forrást a csatlakozik a bemenetre egy ellenálláson keresztül az a bemenet bizonyos idő elteltével telítődik és ez egy olyan problémához vezet, hogy ha a bemeneten 2,65 V feszültség mérhető de az idővel csökken 2,55 Volt-ra az ADC előfordulhat hogy 2,6 Volt-ot érzékel ezáltal rontva a mérés pontosságát. Ezt a problémát viszonylag egyszerűen lehet orvosolni egy földpotenciálra, valamint a bemenetre kötött relatív nagyobb ellenállással, mely segítségével úgymond lehúzzuk a feszültséget, megszűntetve ezt a kapacitív energiafelesleget, pontosítva a mérést.

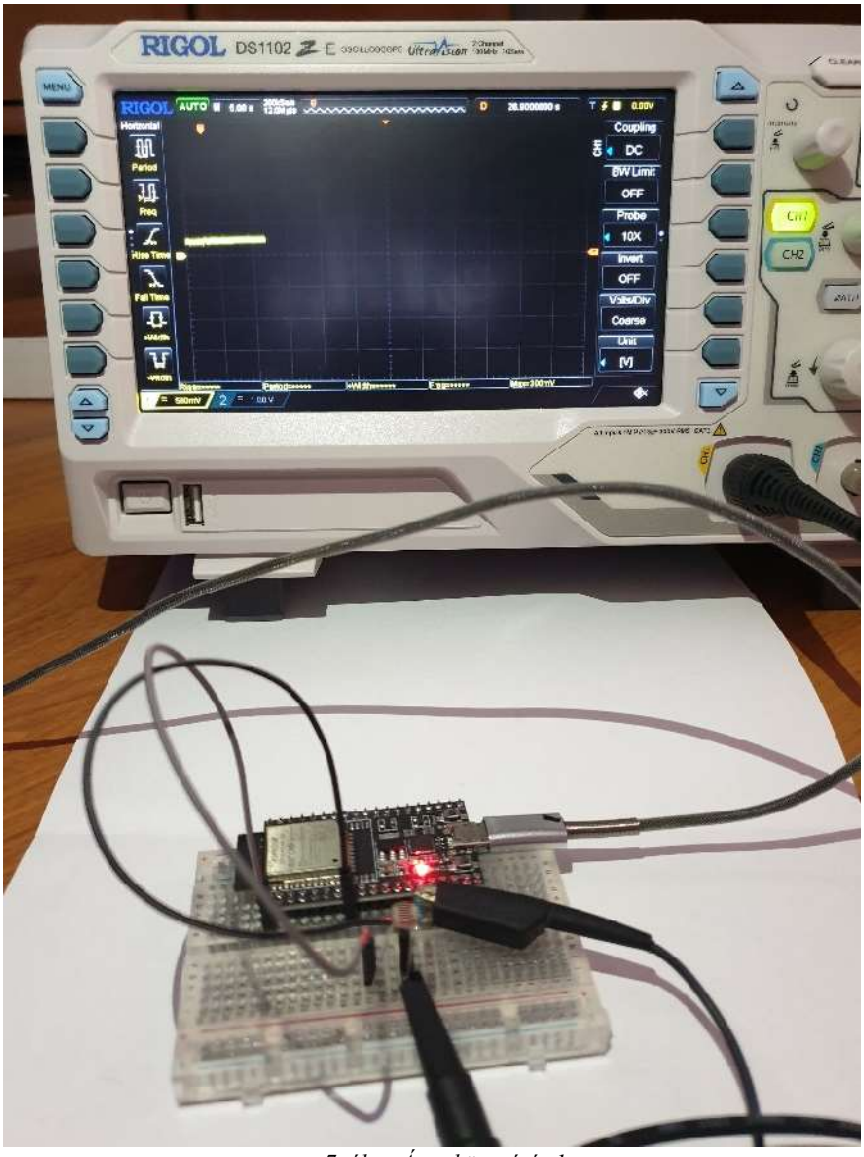


6. ábra Kapcsolási rajz

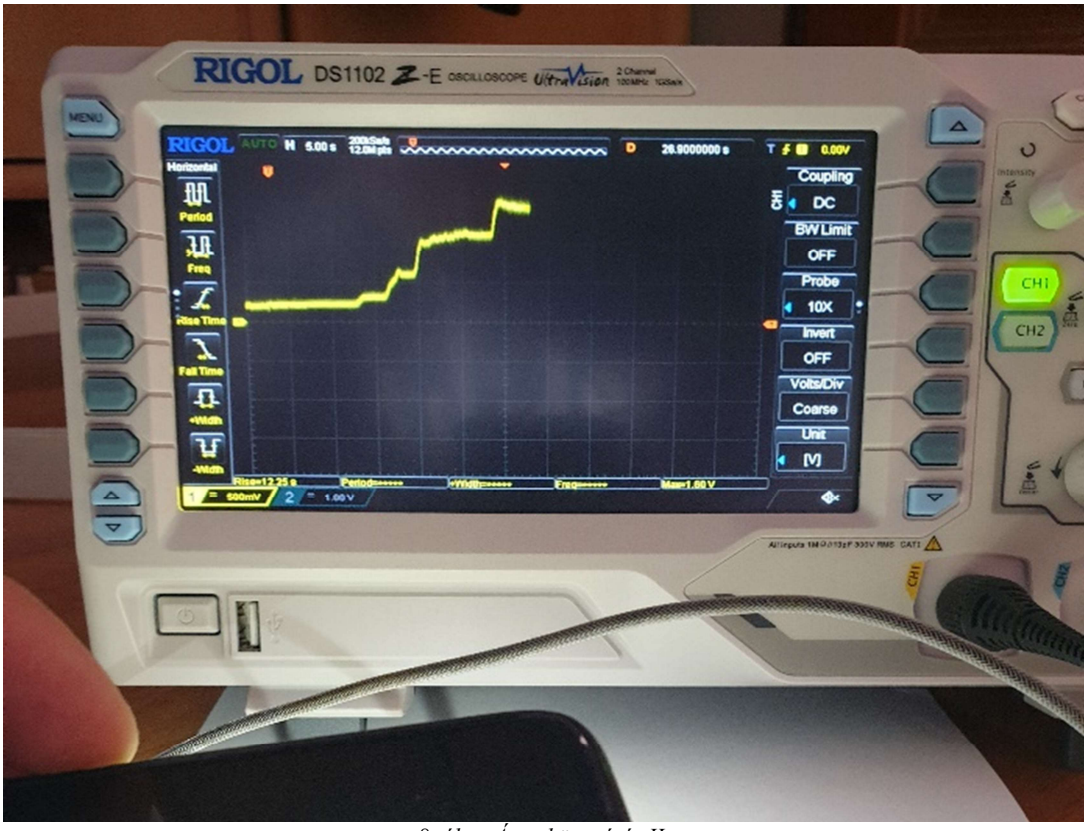
A kapcsolási rajzon láthatóak a komponensek melyen R1 a fentebb definiált lehúzó ellenállás melynek ellenállása magasabb, mint R2. R2 a rajzon a fotorezisztor helyét jelképezi melynek egyik kivezetése az ESP bemenetére, valamint R1 ellenállásra csatlakozik másik kivezetése pedig a tápfeszültségre van kötve. Ez az áramkör gyakorlatban passzív feszültségosztó áramkörként is definiálható. A gyakorlatban pedig ez az áramkör a következtetőképen működik, ahogy egy fényforrás egyre közelebb kerül az eszközhöz egyre

több fényt nyel el, a fotorezisztor ellenállása csökken így a mikrokontroller bemenetén a feszültség ezzel arányosan nő mely jelenséget a következő ábrák szemléltetnek.

Az ábrán található oszcilloszkóp kijelzőjéről megállapíthatjuk az ESP32 bemenetén detektálható feszültséget. Leolvasás után arra következtethetünk, hogy a jelenlegi körülmények között 250 mV feszültség mérhető a bemeneten.



7. ábra Áramkör mérés 1.



8. ábra Áramkör mérés II.

A fényforrás fokozatos közelítésével az ábrán látható módon emelkedik a feszültség, a képernyőről pedig leolvasható, hogy a feszültség egészen 1.5 Volt-ig emelkedett, ezekhez az értékekhez pedig a bemeneten található analóg digitális konverter 12 bites felbontáson az alábbi ábrán látható értékeket rendelte.

9. ábra ADC konverzió eredménye

2.1.4 Szerver

A szerver egy olyan eszköz a hálózaton, amelyen olyan feladatokat lát el szolgáltatásokat nyújt, amelyekre több felhasználónak is igénye lehet akár helyi hálózaton akár az interneten keresztül a világ bármely pontján. A szerverek olyan eszközök melyek elveikben hasonlóak a mindennapi számítógépekhez azonban mégis megtalálható néhány fontos különbség. A szerverek többnyire egy arra kialakított helységben helyezkednek el mely helység általában por szűrőkkel ellátott, valamint klimatizált. Ezekre az intézkedésekre abból az okból kifolyólag van szükség mert ezeket a számítógépeket arra tervezték, hogy folyamatosan üzemeljenek megállás nélkül így csökkentve a túlmelegedés, valamint a por lerakódásának kockázatát és az ezen okokból következő leállásokat. A folyamatos, leállítás nélküli üzemelés mellett fontos szempont a redundancia. Ez a fogalom szószerinti fordításban azt jelenti, hogy felesleges, viszont az informatika világában inkább a tartalékra vonatkozik. Abban az esetben, ha egy szerverre azt a jelzőt alkalmazzák, hogy redundáns az a gyakorlatban azt jelenti, ha a szerver kiesik egy váratlan hiba miatt, abban a pillanatban a helyét egy azzal megegyező feladatot ellátó vagy akár teljes mását képző szerver átveszi és helyettesíti ameddig az eredeti állapot vissza nem áll. Ez a redundancia a szerver gépen belül is jelen van például tápegység vagy hálózati kártya formájában. A redundáns tápegység segítségével lebonyolítható az üzemeltetése a szervernek oly módon, hogy az egyik tápegység az áramszolgáltató hálózatából vételezi a villamos energiát míg a másik tápegység egy szünetmentes tápellátást biztosító eszköz aljzatába csatlakoztatva tartalékként funkcionál az áram szolgáltatás szünetelése esetén. Az ily módon üzemeltetett szervereknél akár szünetmentes tápellátást biztosító egység cseréje is lehetséges leállás nélkül a közvetlen betáplálásnak köszönhetően. A redundáns hálózati kártyákat olyan esetekben lehet alkalmazni, mint például egy föld alatti kábel elvágása miatti hálózat kiesés esetén vagy egy hálózati eszköz meghibásodása esetén egy más földrajzi lokációval, útvonallal rendelkező hálózat veszi át a fő hálózat szerepét. Napjainkban abban az esetben, ha egynél több hálózati kártya áll rendelkezésre, azesetben dönthet úgy a rendszergazda, hogy link aggregációt alkalmazva a két hálózati kártyát együttesen bocsátja a feladatok ellátására így a sebességük összeadódnak, viszont amennyiben az egyik hálózat

kiesik abban az esetben az összeköttetés nem szűnik meg csak az átviteli sebesség csökken. A redundáns fogalom adattárolás formájában is megnyilvánulhat, ez azt jelenti, hogy az adatok nem kizárólag egy merevlemezen tárolódnak. Az ilyen redundáns adattárolási topológiáját, hierarchiáját írja le a RAID. A RAID mint fogalom az 1970-es években került be a köztudatba jelentése pedig nem más, mint redundáns tömbje a független lemezeknek. A RAID-et három fő módszer, technika alkot ezek pedig a csíkozás mellyel a tároló kapacitás növelhető, a tükrözés mellyel az olvasási sebesség növelhető, és a paritás mellyel pedig a hibatűrést lehet növelni. Ezen módszerek alkalmazásával, variációival, több RAID úgynevezett szintet is kialakítottak az olyan szempontok figyelembevételével, mint például a célkitűzés, maximálisan megengedhető hibás merevlemezek száma és a rendelkezésre álló merevlemezek száma. Két alap RAID szint a RAID 0 és RAID 1, az első esetben csíkozás történik igy a merevlemezek kapacitása összeadódik annak a kárára, hogy amennyiben az egyik merevlemez meghibásodik az egész kötet olvashatatlanná válik így minden adat elvész. A második esetben tükrözés történik így kapacitás meghatározója a merevlemezek egyenkénti kapacitása viszont az ilyen módon konfigurált tömb az adatokat úgy tárolja, hogy egyik merevlemez teljes mása a másik merevlemeznek, ennek következtében, ha az egyik merevlemez kiesik abban az esetben a hibás merevlemez eltávolításával egy új merevlemez helyezhető be majd ez az újonnan behelyezett merevlemez a még működő merevlemez segítségével adatokkal lesz feltöltve így helyreáll a normál működés. A RAID adattárolást megvalósíthatja mind hardveres eszköz és szoftver is, előbbi az operációs rendszer számára transzparens így csak egy virtuális lemezt lát az operációs rendszer és írás vagy olvasás esetén a hardveres RAID vezérlő irányítja a forgalmat, kezeli a paritást a merevlemezek között elosztva míg a szoftveres RAID esetén az operációs rendszer látja el ezeket a feladatokat, mivel ez a feladat műveletigényes és nem egy dedikált hardveres eszközön történik a szerver erőforrásait használja. [3]

A munkám során egy DELL OptiPlex 990 típusú számítógépet használtam szerver gyanánt így hardveres vezérlő hiányában és korlátozott számú merevlemez csatlakoztatási lehetőség következtében szoftveres megvalósítású RAID 1 szintet használtam a redun-

dáns adattárolás megvalósításához. A RAID 1 tömböt 2 darab 500 GB kapacitású merevlemez alkotja. A számítógép fő egysége a processzor egy Intel által tervezett és gyártott i5 termékcsaládba tartozó második generációs eszköz mely 4 maggal rendelkezik és maximális órajelfrekvenciája pedig 3.10 GHz. Operatív tár szempontjából 12 GB DDR3 szabványú memóriával láttam el a rendszert melyet összerakáskor többször is teszteltem a számítógép korából fakadó aggályaim miatt. A RAID tömbben konfigurált merevlemezeken kívül megtalálható még egy harmadik merevlemez is melyen az operációs rendszer fogal helyet. A másik két számítógép szintén szerver funkciót látnak el viszont más feladatok ellátására lettek konfigurálva.

10. ábra A szerver funkciót ellátó számítógépek

2.1.5 Hálózat

A számítógépes hálózat fontos alkotóeleme az informatikának hisz ezen keresztül zajlik a kommunikáció a számítógépek és egyéb eszközök kommunikációja. Legelterjedtebb formája az úgynevezett rezes közeg, mely azt jelenti, hogy az adatok elektromos impulzusok sorozataként kerülnek továbbításra a hálózat eszközei között. Kisebb kiterjedésű

hálózatok esetében ez a réz alapú kábel lehet az UTP kábel mely mozaikszó az árnyékolatlan csavart érpár szavak angol megfelelőinek kezdőbetűiből tevődik össze. Ezen átviteli közeg körülbelül 100m hatótávolsággal rendelkezik és gyakran 1 Gbps adatmennyiség átvitelére képes. Fizikai tekintetben ez a kábel általában szürke külső köpennyel rendelkezik melynek az eltávolítása után 8 darab réz lehet elkülöníteni egymástól mely vezetékek kettesével álnak párban és ezek a párok a szabványban egy meghatározott hosszon meghatározott számú csavarással kerülnek párosításra az interferencia kiküszöbölésének érdekében. A színek és párok pedig a következőképpen alakulnak, a narancssárgát csavarják össze a narancssárga fehérrel a gyártás során, a zöldet a zöld fehérrel, a kéket a kék fehérrel és végül a barnát a barna fehérrel. Ezen szálak futnak a kábelben mely kábelnek a végződéseként tekinthető a szabványosított csatlakozó melynek az azonosítója az RJ-45. Ezt a csatlakozót krimpeléssel egy erre kialakított célszerszám által okozott préseléssel erősítik a szakemberek a kábelre így megbízható csatlakozást kialakítva. Ebben a csatlakozóban egy előre meghatározott szabvány szerinti sorrendben kell elhelyezni krimpelés előtt ez a sorrend pedig gyakran a narancssárga-fehér, narancssárga, zöld-fehér, kék, kék-fehér, zöld, barna-fehér és barna. Ennek az átviteli közegenek hátránya az árnyékolás hiányában az interferenciára való hajlam, amely zajként nyilvánul meg az átvitel során, valamint a viszonylag kis hatótáv. Réz alapú közeghez tartozik az internetszolgáltatói szinten használt koaxiális kábel mely viszont már védett az interferencia ellen és nagyobb körülbelül 500 méteres hatótávval rendelkezik viszont hátrányai közé kell sorolni azt a tényt hogy közvetlenül sajnos nem lehet csatlakoztatni a vég eszközökhöz mint például a szerverek és asztali számítógépek mindenképpen szükségessé válik egy modem nevezetű eszköz mely az Ethernet kereteket a koaxiális hálózaton keresztül továbbítani illetve fogadni tudja. Ez a kábel fizikai tekintetben egy masszívabb merevebb kábel melyet egy fehér köpeny borít. A külső borítást eltávolítva egy árnyékoló fém háló látványa tárul fel mely gyakran kiegészül egy alumínium fólia borítással is. Ezt a réteget eltávolítva egy vastagabb szigetelő réteg következik melynek közepén található egy tömör réz vezeték melyen keresztül zajlik a kommunikáció. Az utóbb ismertetett koaxiális átviteli közeg helyét veszi át a rohamos ütemben terjedő optikai adatátviteli közeg melynek szállító eszköze az üvegszál. Ez az átviteli közeg alkalmas akár 40-100 km áthidalására is, ellenáll

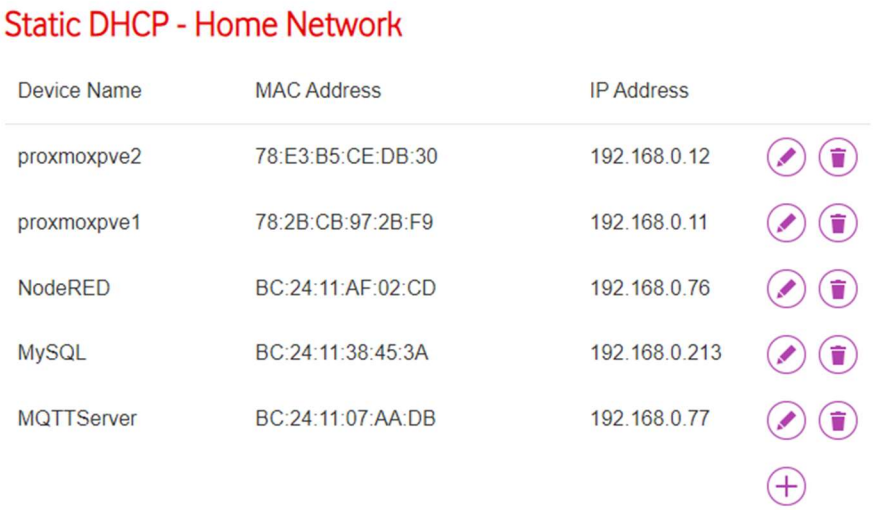
az elektromos interferenciának, átvitel sebességének szempontjából pedig alkalmas a 10 Gbps sebességű átvitelre is. Hátránya a sérülékenysége melyet a kábel gyártása során különböző anyagokkal próbálnak csökkenteni, mint például a kevlár szálak. Átviteli közegek közé tartozik még a rádiófrekvenciás átvitel mely a vezeték nélküli kapcsolatokat hivatott megvalósítani. Ilyen vezetéknélküli átviteli közeg a Wi-Fi mely a 802.11 hálózati szabvány szerint működik. Legelterjedtebb változata napainkban a 802.11n és 802.11ac mely szabványok 2.4 és 5 Ghz es frekvencia tartományban üzemelnek. Hátrányuk az olykor-olykor instabil kapcsolat, kis hatótáv, alacsony sebesség viszont mivel vezeték nélkül történik a kommunikáció az így csatlakoztatott eszközök mobilisek, az felhasználói eszköz akkumulátoros üzeme esetén nem szükséges a vezetékes kapcsolat létesítése sem hálózati eszközköz sem áramforráshoz.

Ezen átviteli közegek hiányában nem jöhetett volna létre az internet, nem lehetne elérni egy szervert a világ bármely pontjáról, így a napelemes termelési adatokat és az azok felhasználásával generált grafikonokat is csak pendrive segítségével lehetne kinyerni a szerverből. Esetemben a felsorolt adatátviteli közegek mindegyike megtalálható. Az internetszolgáltató az utcában megtalálható alállomásig optikai, üvegszálas úton biztosítja a kapcsolatot majd innen koaxiális kábelt alkalmazva létesít összeköttetést a háztartásokkal, szerződött partnerekkel. Ezen összeköttetés létesítéséhez az internetszolgáltató biztosít egy hálózati ezközt mely régebben csak egy modem volt, de napjainkban már egy úgynevezett SOHO router is egyben. SOHO, mint rövidítés, mozaikszó az angol Small Offcie Home Office szavakból áll össze. Ez az eszköz általában magában foglal egy routert mely a forgalomirányításért felel a hálózaton, egy switch-et mely elosztóként funkcionál a vezetékes hálózaton, valamint egy hozzáférési pontot, röviden AP-t mely a vezetéknélküli hozzáférést teszi lehetővé a hálózathoz. Ezen modemmel kiegészített SOHO router után találhatóak meg a vezeték nélküli hálózatot használó eszközök, az UTP kábelek, az ezt az összeköttetést alkalmazó egyéb hálózati, valamint felhasználói eszközök. Az összeköttetés megvalósult viszont a kommunikáció szempontjából vannak még hiányosságok, ez pedig a kommunikációs protokoll, mely definiálja az adatok áramlásának szabályait. A hálózati kommunikációt több réteg szerint lehet osztályozni, ezen rétegeket

pedig az OSI modell írja le. Ez a modell hét különböző réteget definiál. Az első réteg a fizikai réteg mely a fentebb definiált réz, optikai, vagy mikrohullámú közegeket gyűjti egy csoportba, ezen rétegen keresztül áramolnak egy meghatározott szekvencia szerint a bitek. A következő szint, azaz a második az adat kapcsolati réteg szintje, mely szinten a szekvenciális bitek sorozatai nagyobb egységeket például Ethernet kereteket alkotnak. Ezekben az Ethernet keretekben különféle szegmensek találhatóak melyek közül két szegmens a hálózatra csatlakozó eszköz címét tartalmazza. Általában ezek az eszközök két féle címmel rendelkeznek, ezek a MAC cím és az IP cím, ezek alapján lehet azonosítani az eszközöket a hálózati kommunikáció során. A MAC cím egy 16 bites számrendszerben generált cím mely hatszor két karakterből áll. Ezt a címet a gyártás során égetik az eszközbe ebből következik, hogy ez a cím a legtöbb esetben nem változtatható meg. Az IP cím pedig a MAC cím alapján a DHCP szerver által kerül az eszközhöz rendelésre. Az IP címeknek 2 változata van jelen napjainkban ezek pedig az IPv4 és az IPv6 címek. Az IPv4 verzió 10-es számrendszerben leírható címeket foglal magában melyek tagolás tekintetében négyszer hármas egységeket képeznek. Ezeket a MAC cím IP cím hozzárendeléseket az ARP tábla tárolja. Az Ethernet keretben található címeket tartalmazó szegmensek egyike a feladó, küldő eszköz címe a másik pedig a cél eszköz címét specifikálja. A keret szegmenseiben megtalálható még egy előtag, egy adat szegmens és a CRC kód mely segítségével az esetleges átvitelben keletkezett hibák detektálhatóak. A harmadik rétegben, azaz a hálózati rétegben történik az előzőleg bemutatott Ethernet keretek áramlásának irányítása, a forgalom szabályozása mely feladatokat az esetek többségében a router nevezetű eszköz lát el. [4] [5]

A munkám során a router által alapértelmezetten beállított 192.168.0.0 /24-es hálózatot használtam. A hálózati címet követő /24-kifejezés a hálózati maszk mely azt határozza meg hogy a hálózatban hány kiosztható cím áll rendelkezésre. A /24 maszk hasonló módon az IP címhez a következőképpen írható fel: 255.255.255.0. Ezen maszk határozza meg azt a tényt, hogy az IP cím első három bináris számrendszerben vett oktettje nem változik így a 192.168.0 minden kiosztott cím esetében ugyan az marad, valamint azt is meghatározza, hogy összesen 255 darab kiosztható cím áll rendelkezésre. Mivel a DHCP

szerver egy MAC címhez bármilyen hálózatban kioszthatóIP címet rendelhet így a szervert minden IP cím hozzárendelés után más IP címen lehetne elérni, ez pedig problémát jelent a munkamenet során mivel vannak olyan folyamatok melyek a szervert az előzetes konfiguráció során elmentett IP cím alapján érik el így egy változás után a szerver elérhetetlenné válna. Erre kínál megoldást a DHCP szerver azon funkciója mellyel a felhasználó előre definiálhatja, fixálhatja azt, hogy a szerver egy bizonyos MAC címhez milyen IP címet rendeljen. Munkám során az internetszolgáltató által biztosított routerben megvalósított DHCP szervert a következő ábrán látható módon konfiguráltam.

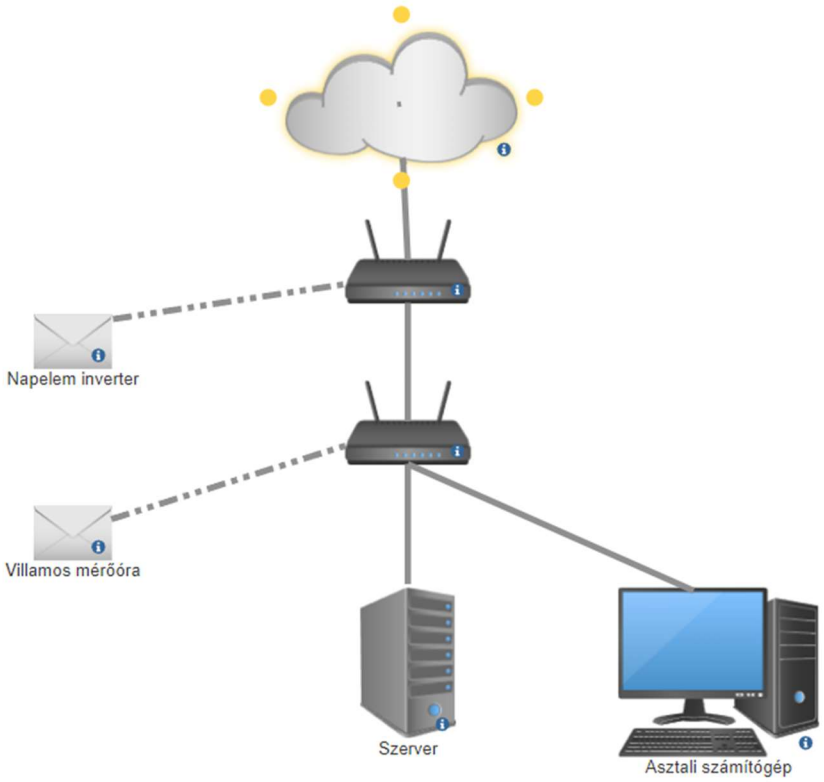


11. ábra DHCP: cím allokálás

Az ábráról leolvasható például a MySQL adatbázist kezelő szerver MAC címe, valamint az is hogy ez a szerver a hálózaton belül a 192.168.0.213-as címen érhető el. Amennyiben az a cél, hogy egy eszköznek az IP címe ne változzon megoldható oly módon is hogy az eszköz hálózatra csatlakoztatásakor manuálisan hozzárendelünk egy olyan IP címet amely még nincsen használatban és ezt az eszköz számára statikus módon konfiguráljuk, így a hálózatra csatlakoztatáskor már rendelkezik az eszköz IP címmel és nem fog kérést küldeni a DHCP szerver felé. Az ezzel az előzőleg leírt módszerrel konfigurált eszközök a villamos mérőórában elhelyezett adatokat továbbító eszköz, valamint a napelemek által ellátott inverter. Ezen eszközök annak a kockázatnak vannak kitéve, hogy nem ismerik a már hálózaton kiosztott IP címeket így már ha az előzetesen konfigurált IP cím már egy

másik eszköz számára ki lett osztva és használatban van a hálózaton, majd ezen fellül pedig csatlakoztatásra kerül az eszköz, az azt eredményezi, hogy IP cím ütközés, duplikáció keletkezik a hálózatban ezáltal hiba keletkezik a forgalomirányításban.

A topológa a hálózaton található eszközök egymáshoz viszonyított elhelyezkedését írja le, valamint a kapcsolatokat szemlélteti az eszközök között. Munkám során a következő ábrán látható topológia áll fent melyen a folyamatos vonal a vezetékes kapcsolatot szemlélteti a szaggatott vonal pedig a vezeték nélküli kapcsolatot.



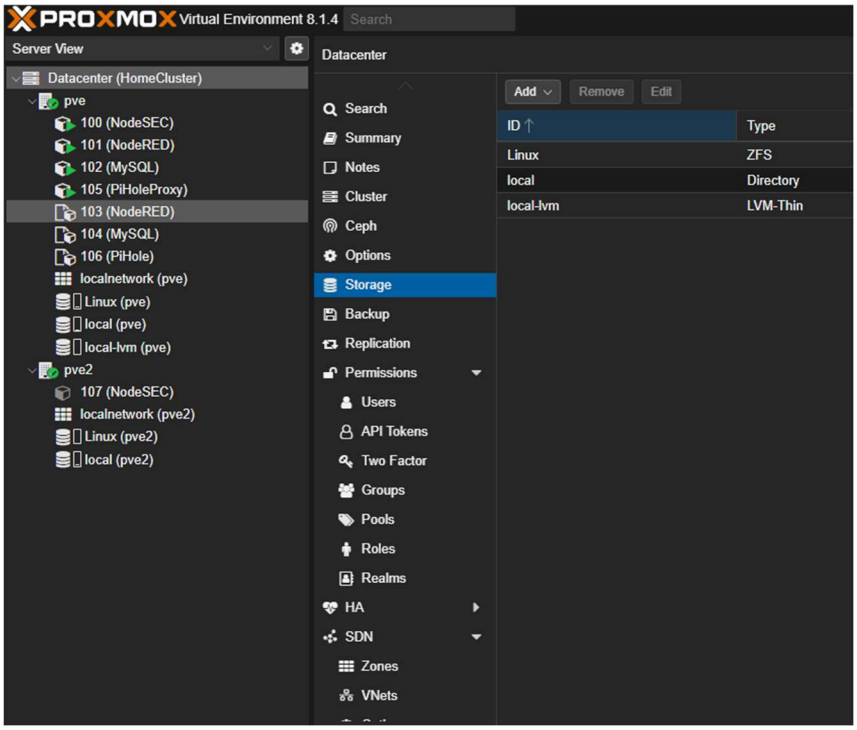
12. ábra Hálózati topológia

2.2 A szoftver

A dolgozatom ezen részében ismertetem a munkám során használt szoftvereket, azok szerepét a megvalósítás során.

2.2.1 Proxmox

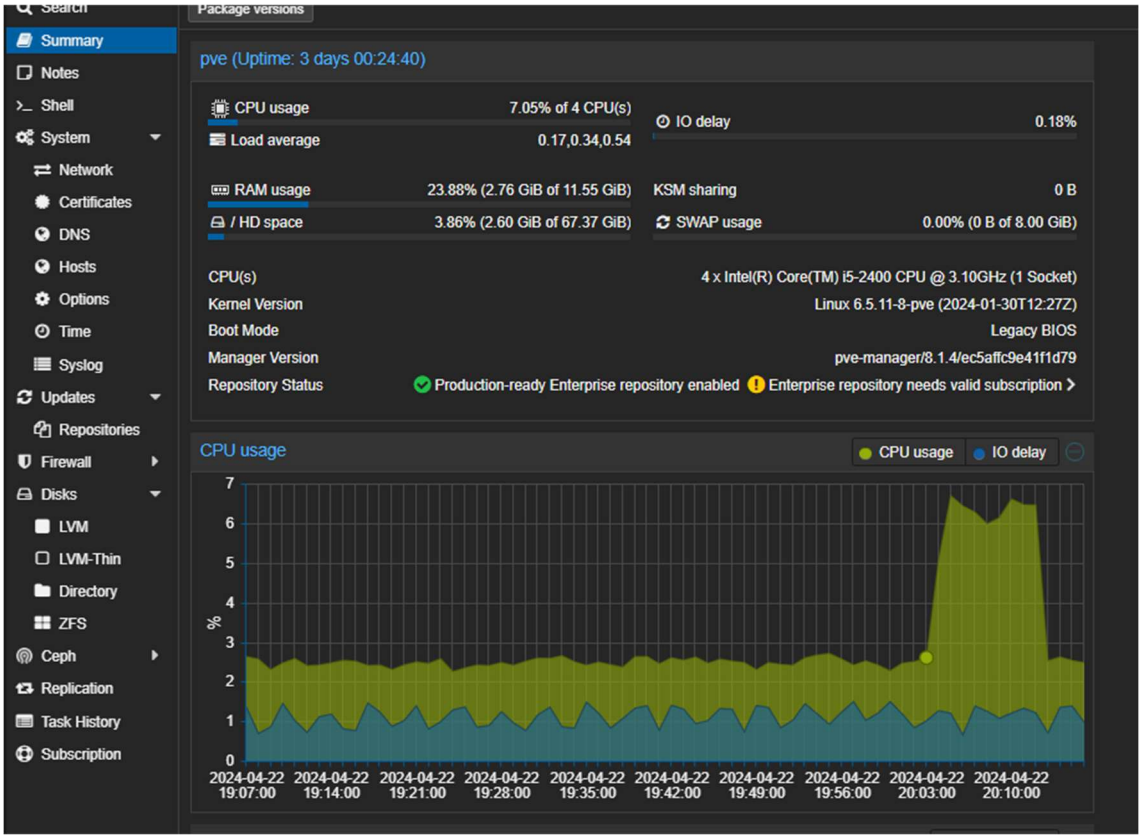
A proxmox egy nyílt forráskódú linux alapú operációs rendszer, hypervisor melynek fő célja a virtuális számítógépek, valamint containerek felügyelete, valamint futási környezetének megteremtése. A virtuális számítógépeknek ez a rendszer KVM technológiát alkalmazó környezetet biztosít mely rövidítés az angol Kernel-alapú Virtuális Munkaállomás kifejezésekből tevődik össze. Ezen virtualizációs technológia kernel szinten allokál virtuális hardvereket egy számítógéphez, ezekért a virtuális hardverekért pedig a QEMU felel. A rendszer feltelepítése után egy web böngészőből elérhető grafikus interfészen keresztül konfigurálható a szerver.



13. ábra Proxmox kezdőképernyő

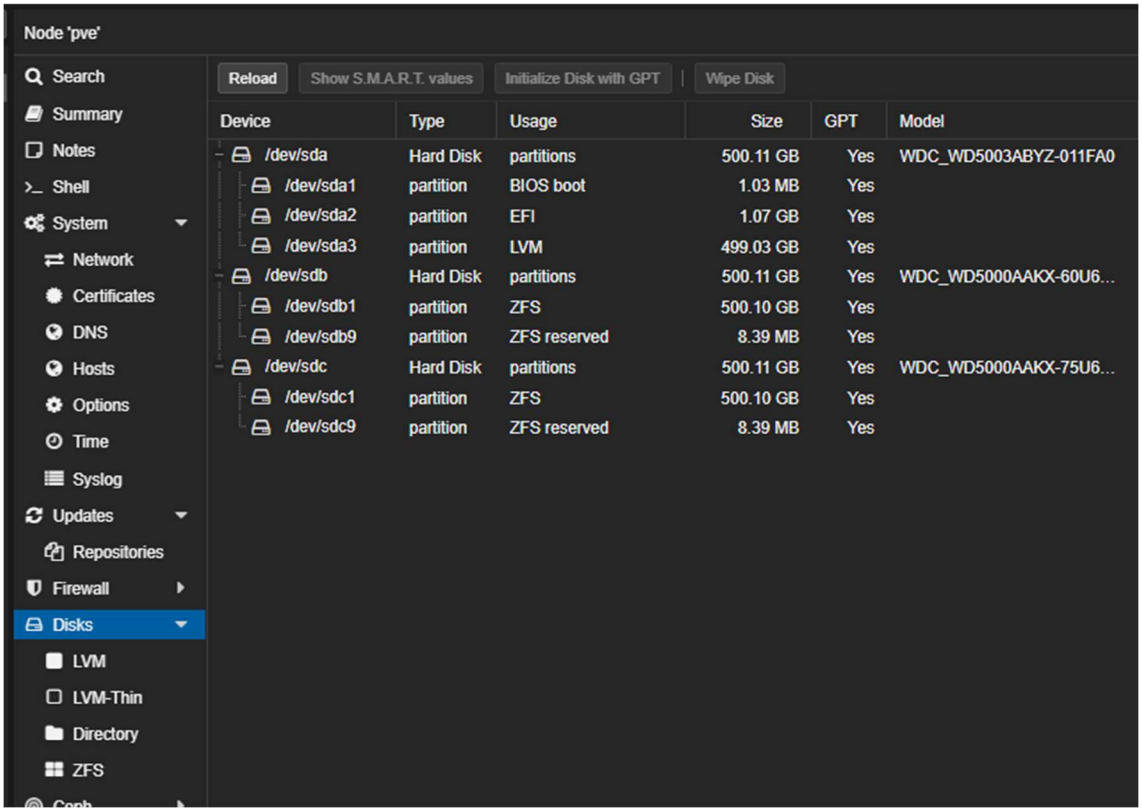
A felületen a telepítés során megadott felhasználónévvel és jelszóval való bejelentkezés után a fentebb beszúrt ábrán látható látvány mutatkozik meg. Látható, hogy a felületen

való navigálás a bal oldalon legördülő menürendszerre hasonlító módon lehetséges. A szerver nézet kiválasztása után a rendszer oly módon jeleníti meg a virtuális gépeket és containereket, valamint a szerveren létrehozott tárolókat, hogy azok elhelyezkedési helye a szempont, tehát ha a rendszerünk több szervert tartalmaz melyek össze vannak fűzve egy úgynevezett fürtbe, esetemben ennek a fürtnek az elnevezése a HomeCluster, abban az esetben a tartalmazó szerver neve alatt jelennek meg a szóban forgó objektumok. Az általam használt fürtben két szerver foglal helyet melyek neve a pve és pve2. A szerver nevére kattintva láthatjuk az adott node adatait, beállításait.



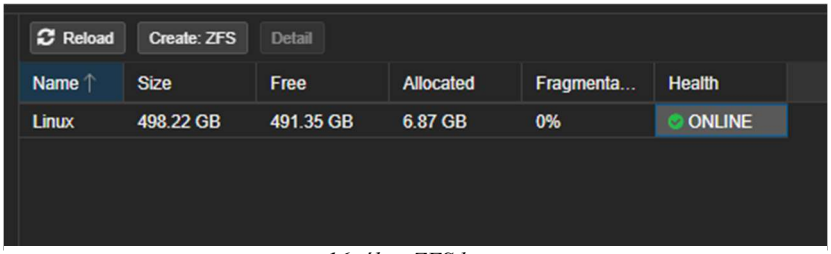
14. ábra Szerver menüje

Az összegzés felületen olyan statisztikai adatok láthatóak mint például a processzor kihasználtsága, az operatív tár telítettsége az ezek alapján vezetett grafikonok és a processzor pontos típusa.



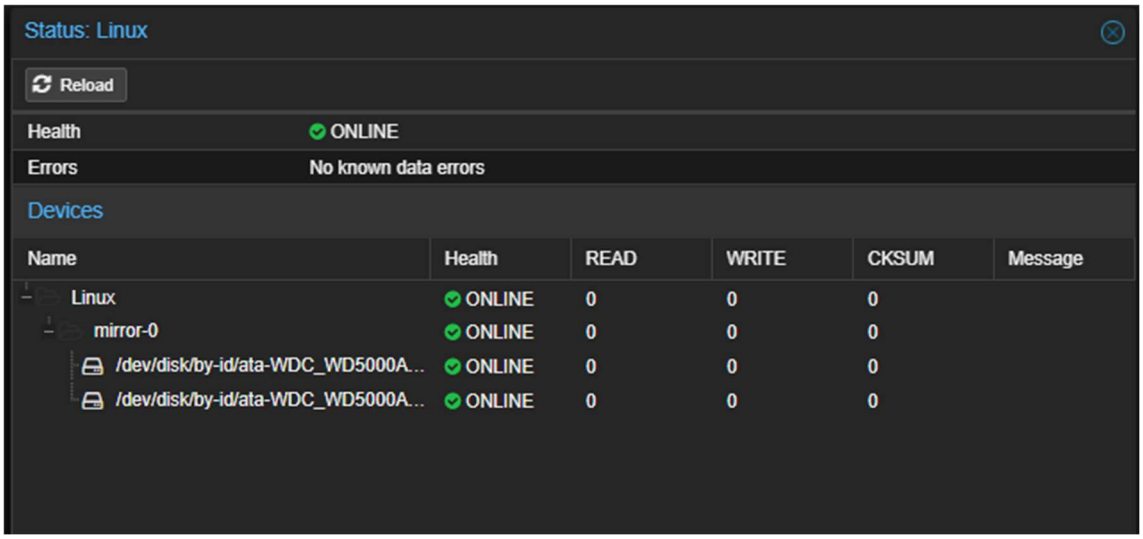
15. ábra Szerver merevlemezei

A lemezek menüpont alatt megtalálhatóak a szerverben elhelyezett merevlemezek, valamint az azokon található partíciók. Esetemben a hardvernél már említett három darab merevlemez látható az első merevlemezen foglal helyet az operációs rendszer, valamint a virtuális gépekhez és containerekhez használt lemezképfájlok. A második és harmadik merevlemez a RAID tömbben konfigurált lemezek. A hardver résznél említésre került miszerint a RAID megvalósítása szoftveresen történik, erről a megvalósításról a ZFS gondoskodik. Ez a ZFS lényegében egy nyílt forráskódú fájlrendszer melynek fejlesztése során három fő cél volt. Az adatok integritásának megőrzése tehát ha sérül az adat akkor azt képes legyen észlelni és kijavítani, az adattárolás kötetekben való megvalósítása, és minél nagyobb sebesség elérése melyet az operatív tár közbeiktatásával tudtak a fejlesztők új szintre emelni.



16. ábra ZFS kötet

Ahogyan az ábra is szemlélteti a ZFS kötetet Linux néven hoztam létre mely kötet 500 GB tárolókapacitással rendelkezik, melyből 6.88 GB már allokálva lett. A kötet nevére duplán kattintva a következő ablak mutatkozik.

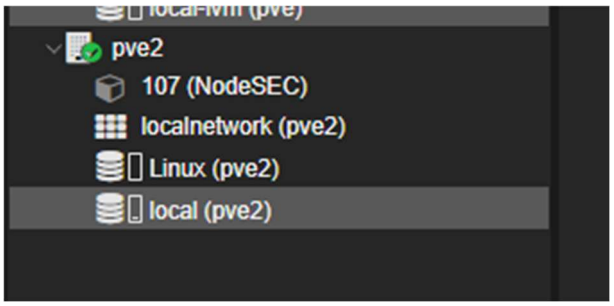


17. ábra ZFS kötet merevlemezei

Itt pedig az látható, hogy a Linux kötetet két merevlemez alkotja mely lemezek egymás tükörképei és jelen pillanatban a merevlemezek kifogástalan állapotban vannak.

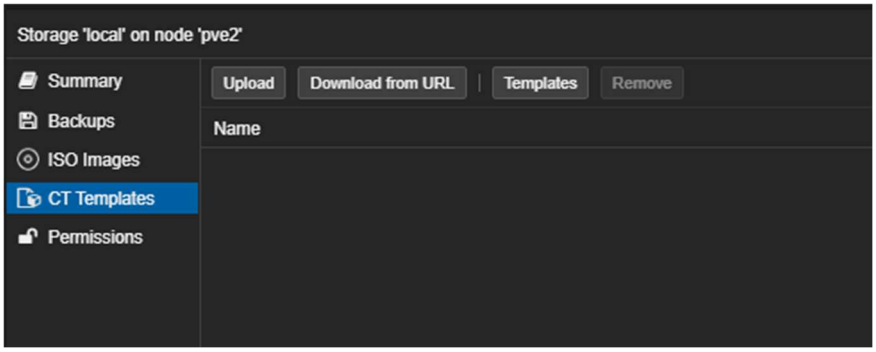
A virtuális gépek és containerek létrehozása a következőképpen zajlik. Először el kell dönteni, hogy melyik szerveren kívánunk létrehozni a példának okáért egy containert. Én a pve2 nevű szervert választottam. El kell navigálni a szerveren abba a adat tárolást megvalósító kötetbe amelyben a lemezképek tárolódnak. A szerveren ez a local kötet.

[6]



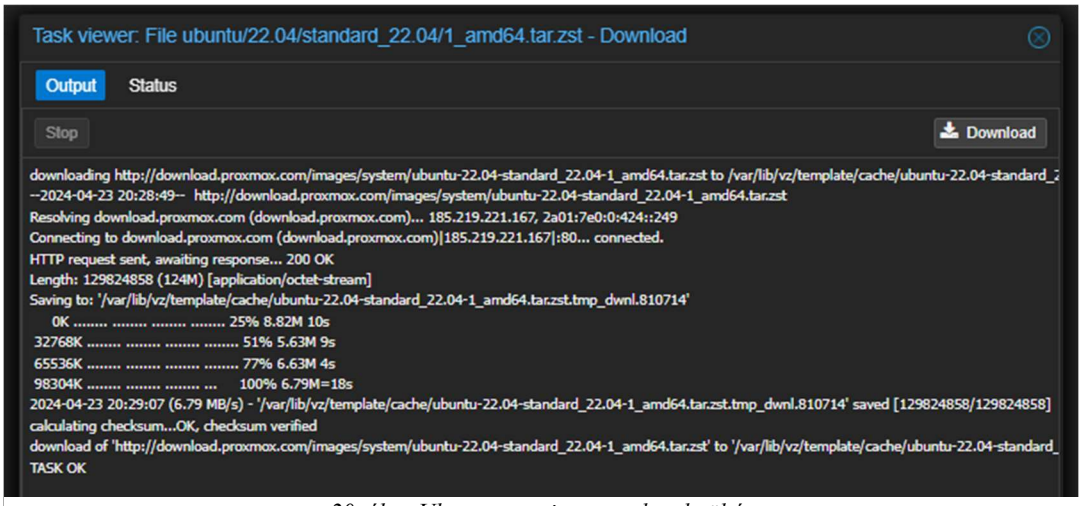
19. ábra Szerver local kötet

A megnyitás után a menüben navigálva a CT Templates lehetőséget választva a következő menü jelenik meg.



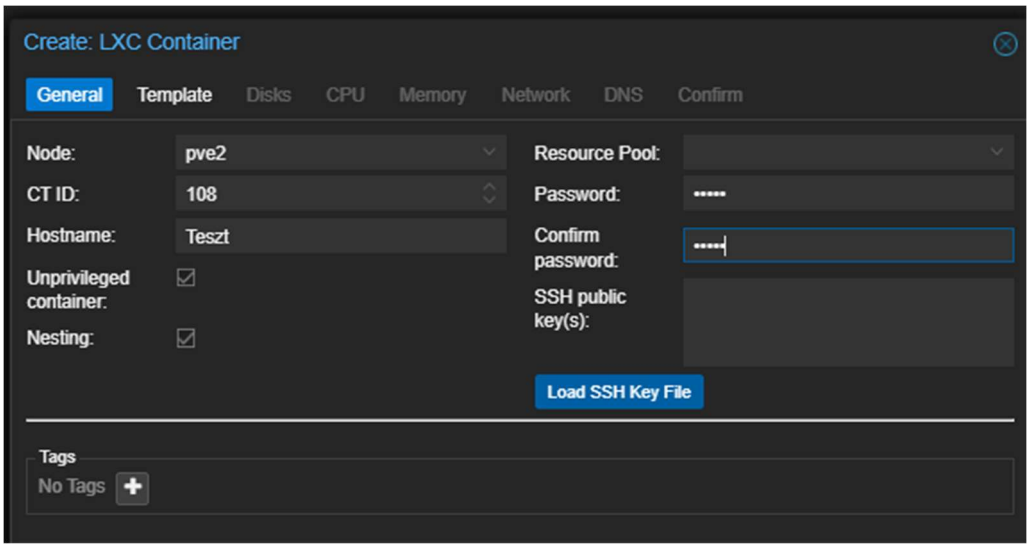
18. ábra CT Templates almenü

Ezen a felületen a Templates gombra való kattintás után megannyi főként linux alapú operációs rendszer közül válogathat a felhasználó, én egy ubuntu rendszert választottam a példához melynek letöltése közben a következő ablak tájékoztat a folyamat állapotáról.



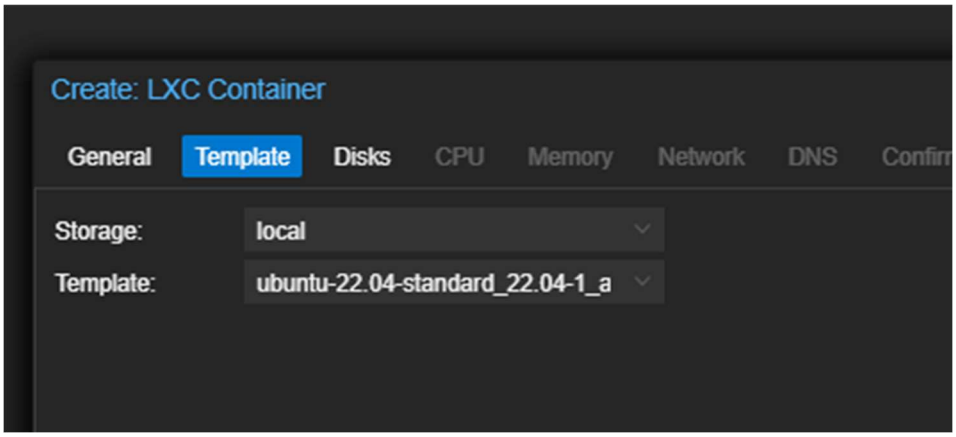
20. ábra Ubuntu container template letöltés

Jelen állapotban készen áll a szerver az új container létrehozására. Rendelkezésre áll a képfájl melyből a container készül, a szerveren elegendő szabad tárhely és erőforrás áll rendelkezésre, így a létrehozás folyamatát a szerver nevén való jobb kattintással lehet elindítani majd a container létrehozás lehetőséget választva az alábbi ábrán látható ablakban elkezdődik a konfigurációs folyamat.



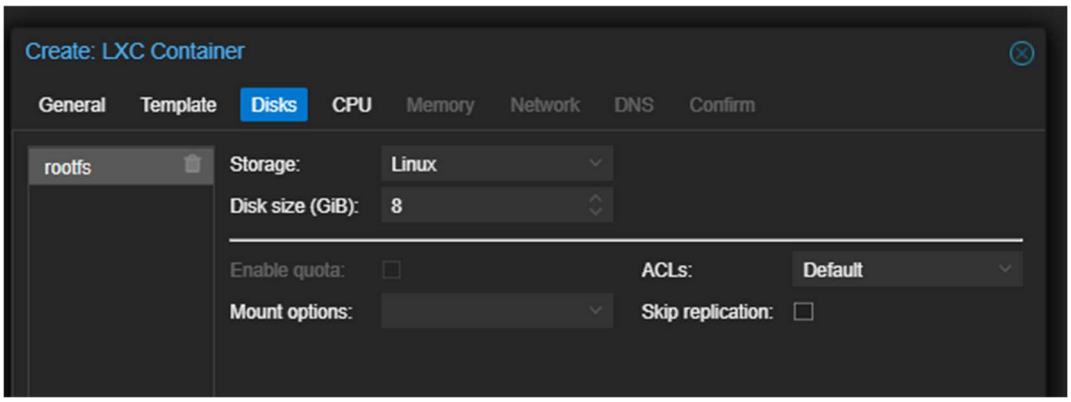
21. ábra Container konfigurációs folyamat I.

A hosztnévként Teszt-et adtam meg valamint a bejelentkezéshez használt jelszó ugyancsak Teszt. A következő képernyőn a képfájlt kell kiválasztanunk.



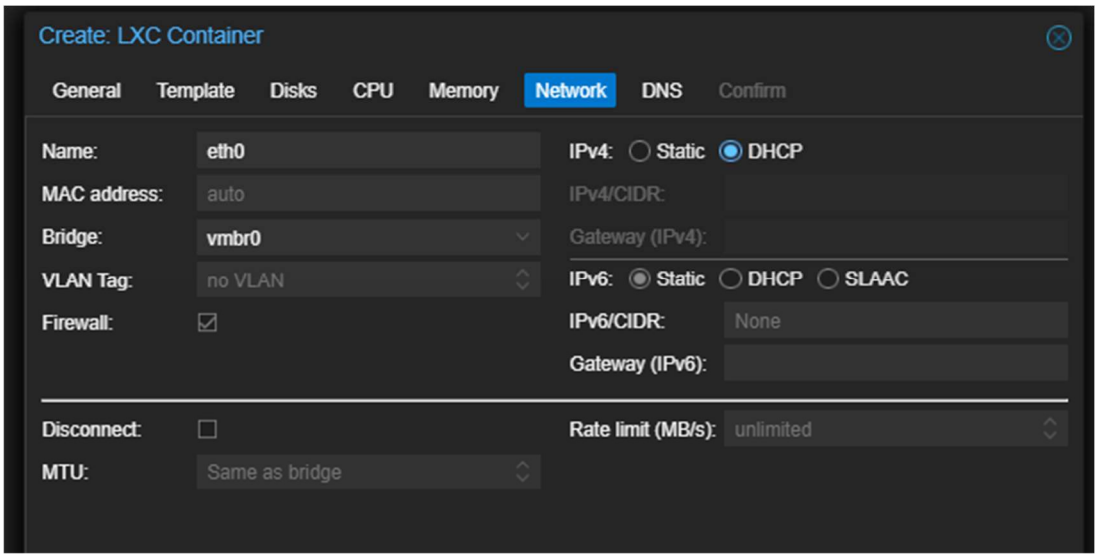
22. ábra Container konfigurációs folyamat II.

A képfájl kiválasztása után van lehetőség a háttértárat megvalósító kötet kiválasztására.



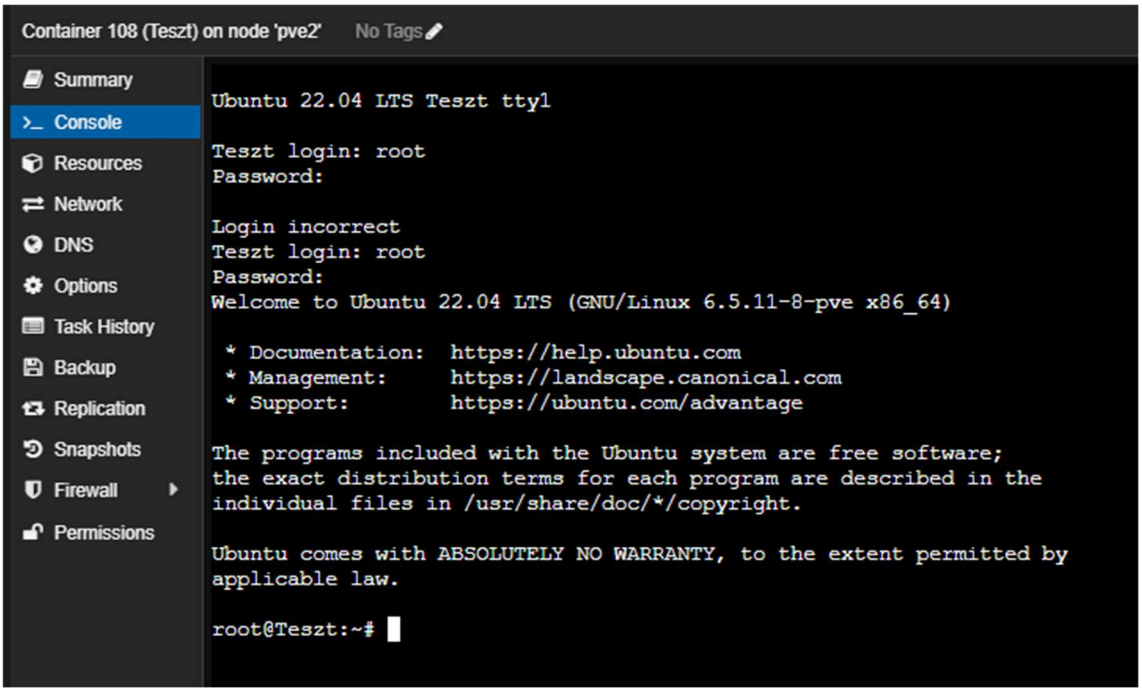
24. ábra Container konfigurációs folyamat III.

Az ábrán látható módon a Linux elnevezésű ZFS fájlrendszerű kötetből 8 GB méretű tárolókapacitást allokáltam a container számára. A processzor és memória menükben megadtam, hogy a rendszer egy processzor magot biztosítson a processzor számára, valamint 512 MB operatív tárat bocsátson a container rendelkezésére. A hálózat fülön két féle módszerrel lehet konfigurálni az IP címet a hálózati interfész kiválasztása után. Ez pedig a statikus és a dinamikus, a dinamikus IP cím kérés esetében a virtuális számítógép a DHCP szerver segítségével allokál IP címet.



23. ábra Container konfigurációs folyamat IV.

A létrehozás után a console menüt választva el is lehet indítani a containert.



25. ábra Container futás közben

A container egy olyan virtuális számítógép mely az operációs rendszer szintű virtualizáció segítségével valósul meg ezzel lehetővé téve azt, hogy több izolált linux alapú futtatási környezet jöjjön létre egy linux kernel használatával. A kernel az operációs rendszer azon része mely az alkalmazások, programok és a hardver eszközök között helyezkedik el és ezen egységek közötti kommunikációt bonyolítja le.

2.2.2 MQTT broker

Az MQTT mint szolgáltatás 1999-ben látott napvilágot, fő célja az IoT eszközök közötti kommunikáció lebonyolítása. Az IoT eszközök olyan internetre vagy hálózatra csatlakoztatott eszközök, amelyek egy vagy több szenzorral vannak ellátva és ezen szenzor által méréseket végeznek majd ezen mért adatokat egy másik eszköz számára továbbítják a hálózaton további feldolgozás céljából. Alapvetően ezek az IoT eszközök kis számítási teljesítménnyel rendelkeznek, tároló kapacitásuk is alacsony így minden lehetőséget meg kell ragadni annak érdekében, hogy a kód, amely az eszközön fut minél jobban optimalizált legyen. A kommunikáció ezekkel az eszközökkel többféle módon is megvalósítható,

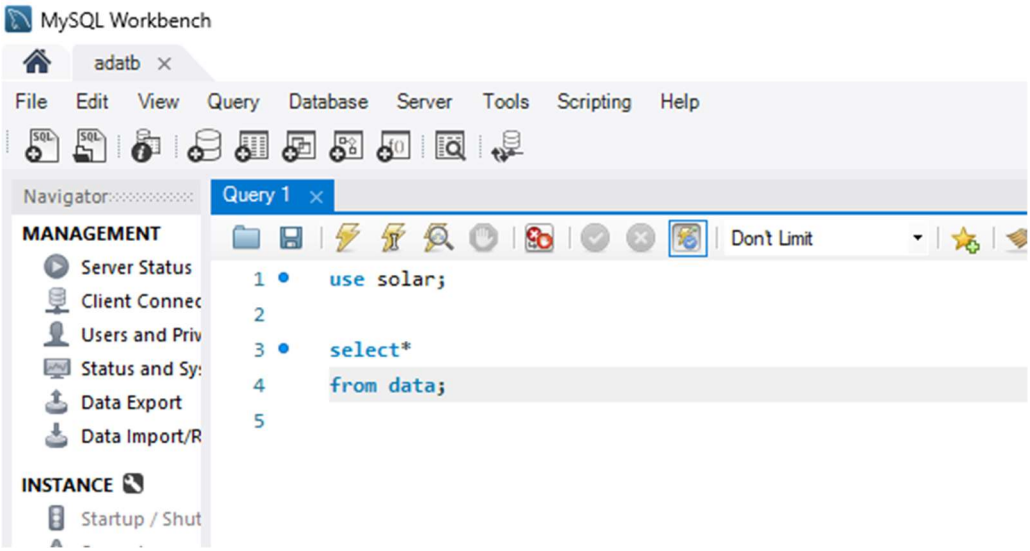
ezen módozatok egyike az MQTT. Ez egy olyan átviteli protokoll mely oly módon működik 3 eszköz összhangban üzeneteket küld egymásnak. Ezen eszközök az MQTT kliens, amely maga az IoT eszköz, az MQTT broker mely a szerver szerepét tölti be a folyamat során és maga a kliens, felhasználó vagy applikáció, amely az adatokat fogadja. A broker alapvetően csak egy köztes eszköz mely az üzeneteket befogadja az IoT eszköztől és elosztja a kliensek között. Ez a rendszer esemény vezérelten üzenetek küldése segítségével kommunikál. Ezt az üzenetet úgy kell elképzelni, mint egy változót. Van egy címe, amely a topic és van egy törzs része az üzenetnek, amely az adatot tartalmazza. Ezt a címet mind az IoT eszköz és a kliens is ismeri. Az MQTT kliens a topic segítségével publikál egy adatot tartalmazó üzenetet, amely beérkezik a szerverre, brókerre. Itt az üzenet, adat elosztásra kerül azok a brókerre csatlakozott kliensek között, amely kliensek feliratkoztak a szóban forgó üzenet, adat topic-jára. Az MQTT fejlesztése során figyelembe a hatékonyságot a kis számításigényt és az ingadozó stabilitású WiFi kapcsolatok okozta problémák kiküszöbölésére is fordítottak időt az opcionális titkosítási lehetőséggel. Munkám során az ESP32 mikrokontrollerem forráskódjának írása során implementáltam ezt a kommunikációs lehetőséget az egyszerűsége és megbízhatósága miatt. A mikrokontrolleren futó forráskód látható az ábrán mely forráskód utolsó sora felel az üzenet küldésért

```
83  void setup() {
84    pinMode(BUILTIN_LED, OUTPUT);
85    Serial.begin(115200);
86    setup_wifi();
87    client.setServer(mqtt_server, 1999);
88    client.setCallback(callback);
89
90    pinMode(33,INPUT);
91    analogSetAttenuation(ADC_6db);
92  }
93
94  void loop() {
95
96    if (!client.connected()) {
97      reconnect();
98    }
99    client.loop();
100
101   unsigned long now = millis();
102   if (now - lastMsg > 10000) {
103     lastMsg = now;
104     value = analogRead(33);
105     snprintf (msg, MSG_BUFFER_SIZE, "%ld", value);
106     client.publish("esp32", msg);
107   }
108 }
```

26. ábra ESP32 forráskód, MQTT broker

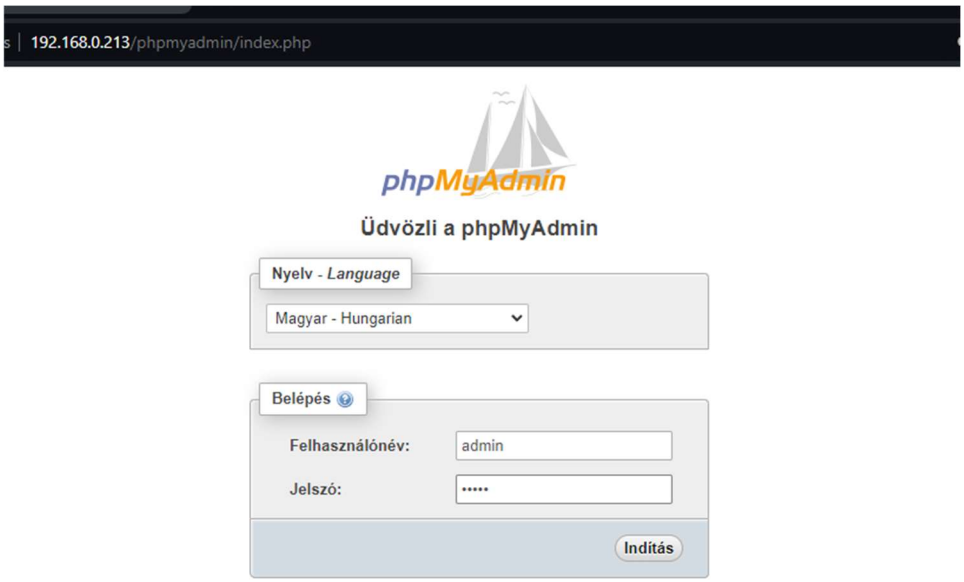
2.2.3 MySQL szerver

Az adatok tárolása kulcsfontosságú pontja a tervezésnek hisz ez határozza meg az adatok elérésének lehetőségeit, azok reprezentálásának, szűrésének módszereit. A munkám során az adatok folyamatosan érkeznek majd eltárolásra kerülnek és az eltárolt adatok olvasása után szűréseken, kisebb modifikációkon, műveleteken keresztül kerül megjelenítésre a felhasználó számára. Az adatok szűrésének minél egyszerűbb és hatékonyabb megvalósítása volt a szempont az adattárolást megvalósító technológia kiválasztása során így a választás a MySQL szerverre esett. Ez ezzel a technológiával az adatokat adatbázisokba lehet szervezni, majd azokat menedzselni. Az adatbázisok oly módon szervezik struktúrákba adatokat, hogy a különböző táblákban mezők segítségével jelentést rendelnek az adathoz majd ezen mezők elemei alkotják a rekordokat. Munkám során első lépésként létrehoztam egy linux alapú containert melyen feltelepítettem a szükséges csomagokat a MySQL szerver futtatásához. Ez után kezdetét vette a konfigurálás melyet a kezdetekben a MySQL Workbench nevezetű program segítségével parancssoros módon hajtottam végre majd a hatékonyabb munka és a nagyobb átláthatóság miatt feltelepítettem a phpMyAdmin felületet melyen keresztül már könnyedén vizuális módon tudtam konfigurálni az adatbázist.



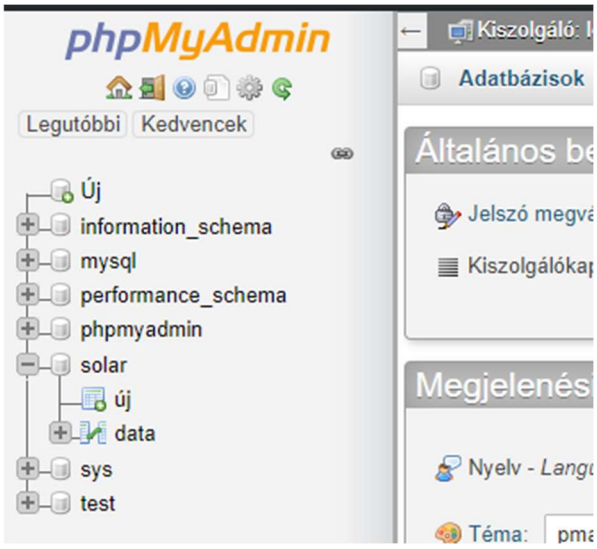
27. ábra MySQL Workbench

A Workbench felületét a fentebb található ábra szemlélteti, mely felületen beírt két darab parancsból azt a következtetést lehet levonni, hogy a konfiguráció során létrehozott adatbázist solar-ként neveztem el, azon belül pedig egy táblát helyeztem el mely táblát pedig data néven lehet hivatkozni. Ez a phpMyAdmin felületen a következőképpen néz ki.

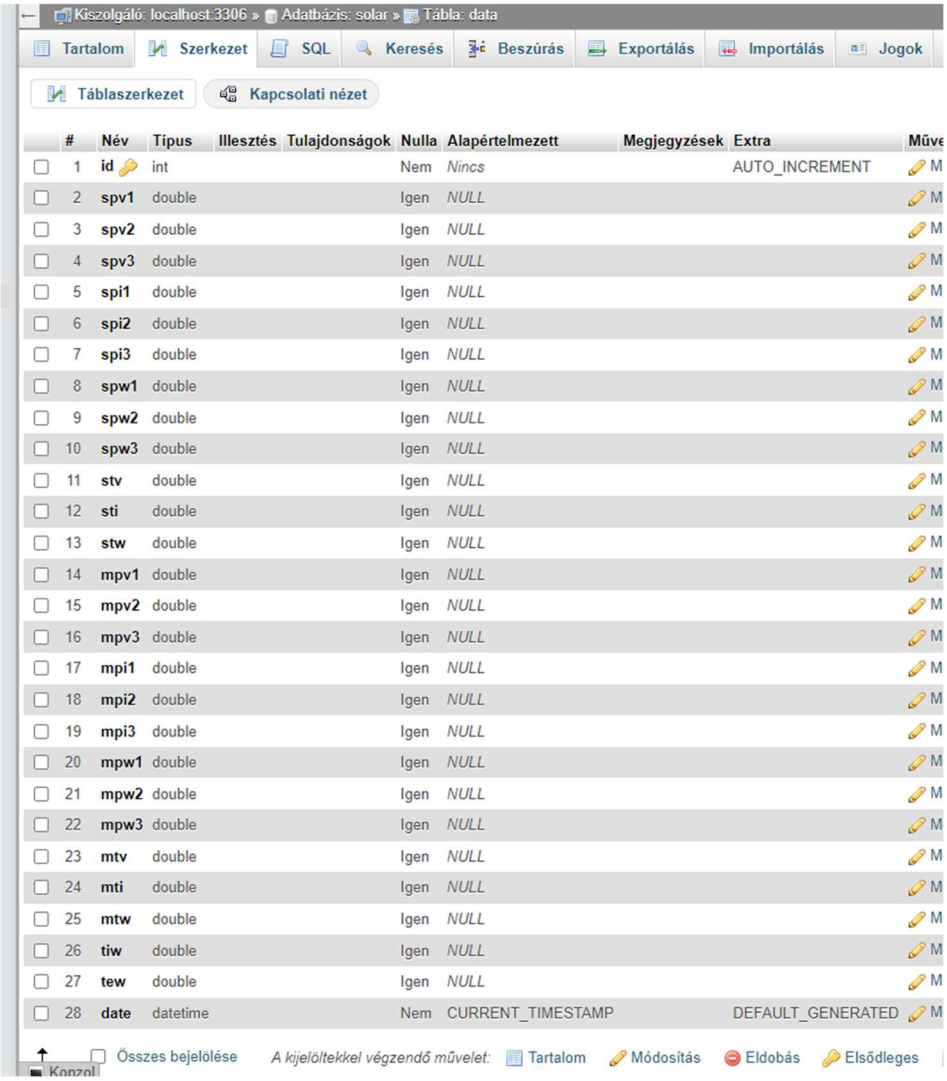


28. ábra phpMyAdmin belépés

A felületen való belépés után pedig láthatóvá válnak az adatbázisok a szerveren.



29. ábra phpMyAdmin adatbázisok



30. ábra phpMyAdmin solar tábla mezők elnevezése

Az adatokat tartalmazó tábla megnyitása után a szerkezet menüben lehetőség van a mező nevek megtekintésére, szerkesztésére, új mezők beszúrására.

A mezők elnevezései a következőképpen alakítottam ki.

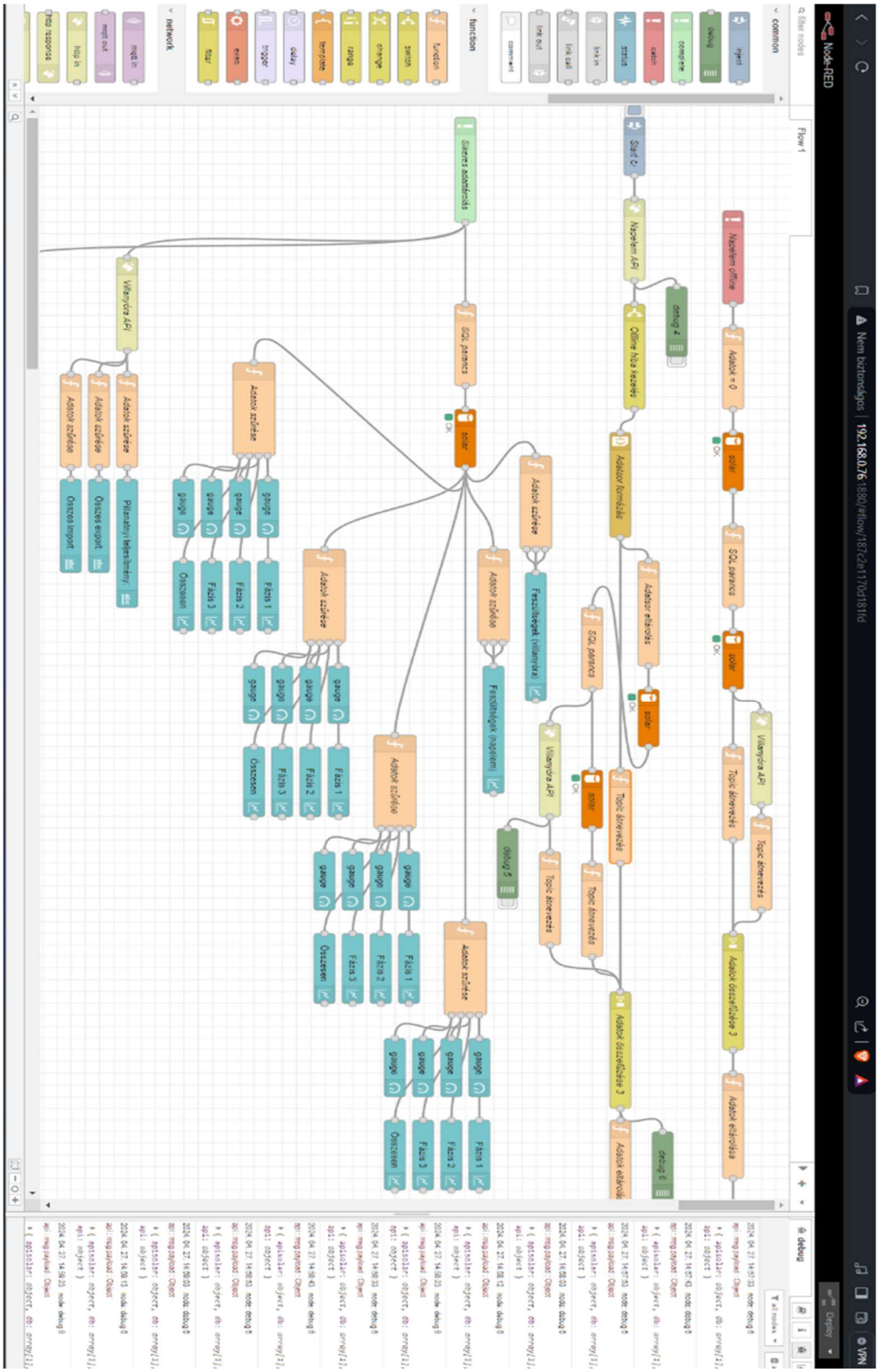
- id – minden sor egyedi azonosítója mely folyamatosan inkrementálódik
- sp… - az így kezdődő mezők a napelem invertere által mért adatok
    - …v, …i, …w ezen mezők tartalmazzák sorrendben a feszültségeket fázisonként, az áramerősségeket, és a teljesítményeket szintén fázisokra bontva
- mp… - az így kezdődő mezők a villamos mérőóra által mért adatok
    - …v, …i, …w ezen mezők tartalmazzák sorrendben a feszültségeket fázisonként, az áramerősségeket, és a teljesítményeket szintén fázisokra bontva
- st… - az így kezdődő mezők a napelem invertere által mért adatok
    - …v, …i, …w ezen mezők összegezve tartalmazzák a következő sorrendben a feszültségeket, az áramerősségeket, és a teljesítményeket.
- mt… - az így kezdődő mezők a villamos mérőóra által mért adatok
    - …v, …i, …w ezen mezők összegezve tartalmazzák a következő sorrendben a feszültségeket, az áramerősségeket, és a teljesítményeket.
- tew, tiw – mezők tartalmazzák a villamos mérőóra által mért összes exportált, valamint importált energiamennyiséget
- date - az adott rekordhoz tartozó dátum és időbélyeg mezője.

mpw1	mpw2	mpw3	mtv	mti	mtw	tiw	tew	date
1288.6088458880442	1526.4353835521767	1535.9557705597788	723.5	18.04077…	-4351	4369.869	11377.43	2024-03-17 13:59:22
1371.7693370165746	1613.438535911602	1618.7921270718232	724	19.07596…	-4604	4369.869	11377.443	2024-03-17 13:59:32
1340.8393587617472	1584.2666666666669	1586.8939745715868	723.6	18.70480…	-4512	4369.869	11377.455	2024-03-17 13:59:42
1326.0315375051803	1810.8333333333333	1572.1351291614862	723.900000…	19.51346…	-4709	4369.869	11377.468	2024-03-17 13:59:52
1298.620458943876	1778.6045894387612	1783.7749516173621	723.4	20.15869…	-4861	4369.869	11377.482	2024-03-17 14:00:02
1610.1747266435984	1848.5678615916954	1856.2574117647057	722.5	22.06865…	-5315	4369.869	11377.496	2024-03-17 14:00:12
1571.7266841886844	1818.6182044542813	1824.6551113570342	722.900000…	21.63840…	-5215	4369.869	11377.51	2024-03-17 14:00:22
1530.0583874982713	1775.4173419997235	1783.5242705020053	723.1	21.10980…	-5089	4369.869	11377.525	2024-03-17 14:00:32
1622.490582711958	1864.8381662524164	1875.6712510356253	724.2	22.21375…	-5363	4369.869	11377.539	2024-03-17 14:00:42
1521.7827781617138	1758.9229302004146	1774.2942916378713	723.5	20.95908…	-5055	4369.869	11377.553	2024-03-17 14:00:52
1528.4985481194692	1768.9985481194692	1781.502903761062	723.2	21.06650…	-5079	4369.869	11377.567	2024-03-17 14:01:02
1296.9425373134327	1782.7205638474293	1792.3368988391376	723.6	20.19154…	-4872	4369.869	11377.581	2024-03-17 14:01:12
1475.2020425062103	1717.6262765663814	1726.1716809274083	724.6	20.36351…	-4919	4369.869	11377.595	2024-03-17 14:01:22
1555.4318056322472	1804.796024295969	1810.7721700717834	724.4	21.41054…	-5171	4369.869	11377.609	2024-03-17 14:01:32
1497.6802181115406	1742.5166344561017	1754.803147432358	724.400000…	20.68208…	-4995	4369.869	11377.622	2024-03-17 14:01:42
1297.1765452538632	1778.6385209713026	1791.1849337748345	724.8	20.14072…	-4867	4369.869	11377.636	2024-03-17 14:01:52
1274.5582192727775	1763.5283561454444	1767.913424581778	723.3	19.92534…	-4806	4369.869	11377.65	2024-03-17 14:02:02

31. ábra Betekintés a MySQL adatbázisba

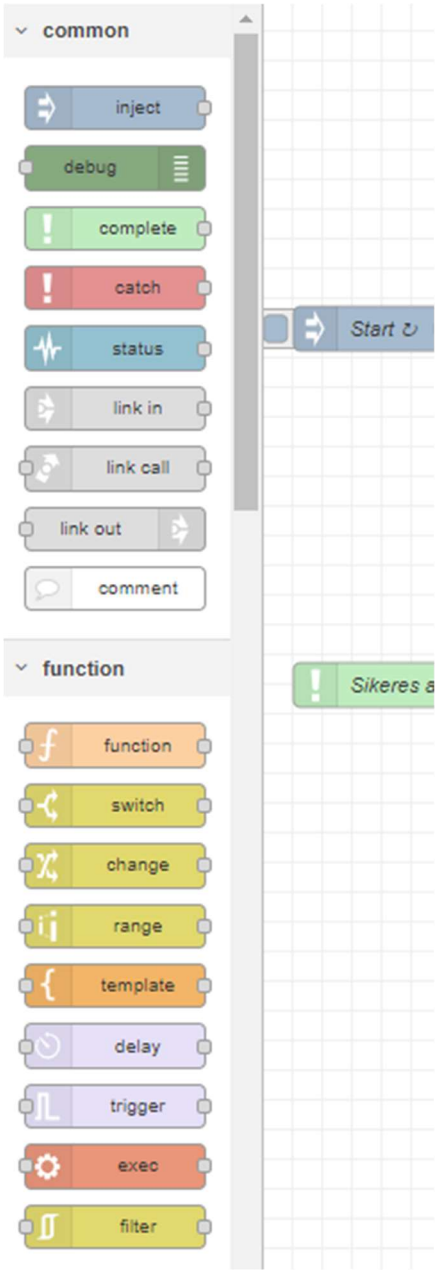
2.2.4 NodeRed környezet

A NodeRed programozási környezet szintén egy linux alapú containerben foglal helyet a szerveren. A telepítési folyamat hasonlóan az egyéb linux alapú programokhoz és kiegészítőkhöz parancssoros konzol segítségével történik. A NodeRed alapvetően egy félig meddig grafikus programozási környezetként titulálható rendszer melyet a benne foglalt csomópontok és az azokat összekötő hálózatok alkotnak. Ezen hálózatokban úgynevezett üzenetek áramolnak mely üzenetek különböző adatokat hordoznak magukkal. Ilyen adat például a fejlécként tekinthető topic, amely az üzenet azonosításában segít, vagy a payload, amely a változókat, és adat tömböket hordozza. A felület 3 fő részből tevődik össze. Bal oldalon található az eszköztár, középen található a szerkesztő felület, jobb oldalon pedig a konfigurációért és a hibakeresésért felelő mező található.



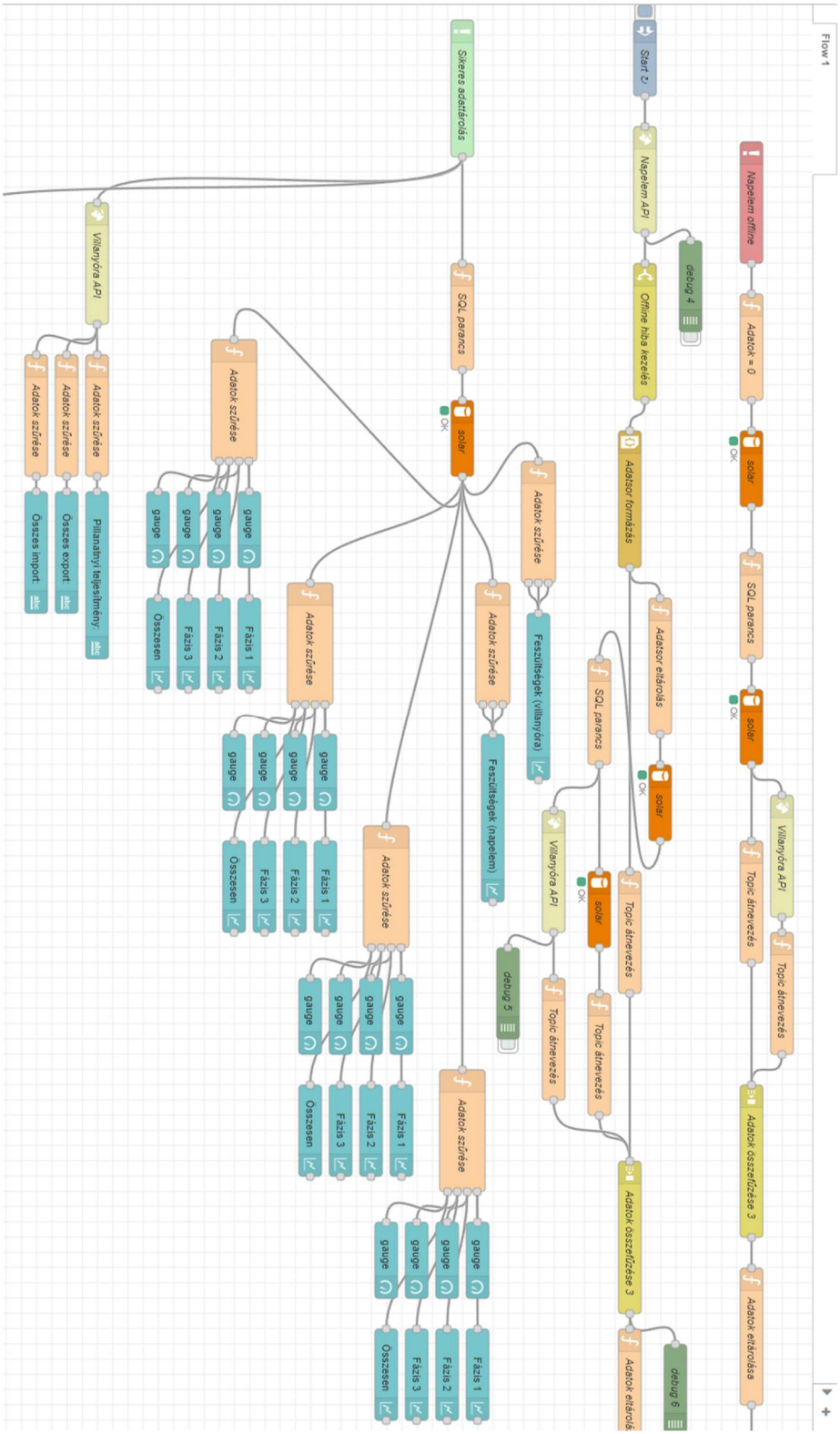
32. ábra NodeRed fejlesztői felület

A bal oldali menüben találhatóak a csomópontok későbbiekben node-ok, amelyek mindegyike egy adott feladat ellátását hivatott megvalósítani. Ilyenek node-ok lehetnek az események, függvények, elágazások.



33. ábra NodeRed eszköztár

A középső felületre kerülnek a node-ok, amelyek drag-and-drop, húzd és vidd módszer segítségével helyezhetőek a szerkesztő felületen, elhelyezés után pedig ezen a felületen van lehetőség a node-ok közötti kapcsolatok létrehozására.

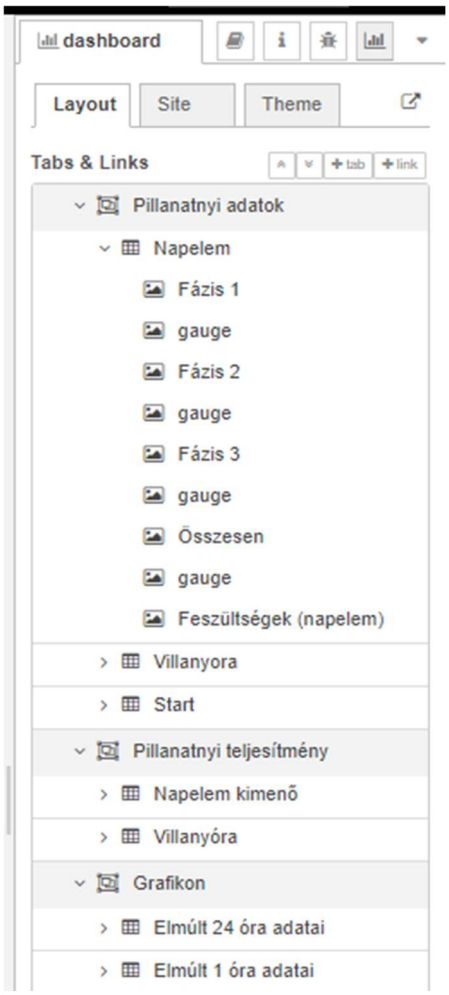


34. ábra NodeRed szerkesztő felület

A harmadik felület pedig a leginkább a hibakeresés és a felhasználói felület konfigurációjának szempontjából releváns ugyanis az üzenetfolyamba a debug node-ok segítségével lehet úgymond belehallgatni mely üzenetek megjelenítése ezen a harmadik szegmensen történik a következő ábrán látható módon.

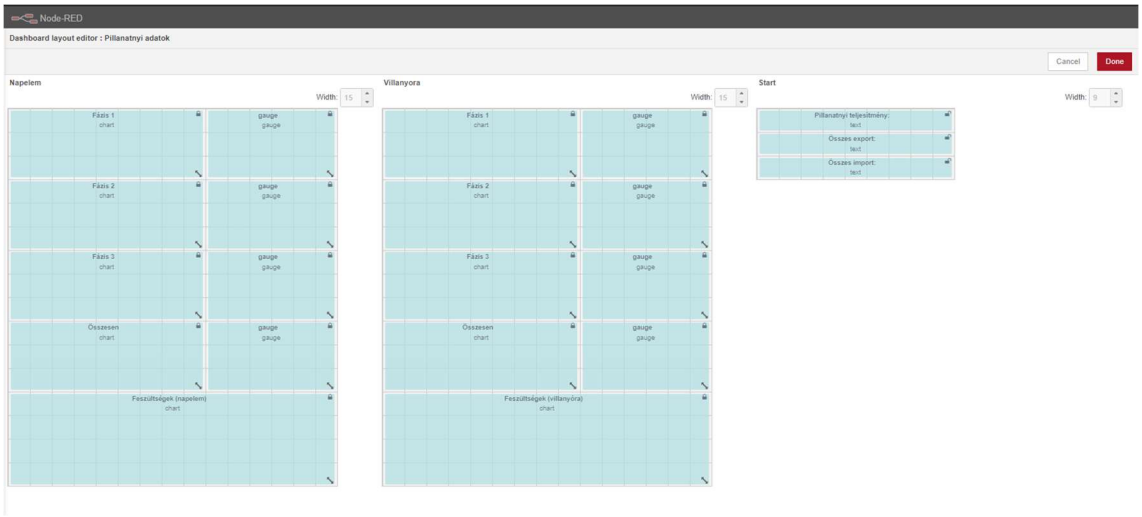
35. ábra NodeRed debug felület

Az ábráról megállapítható, hogy az elkapott üzenet a hatos számú debug node-on keresztül érkezett és egy komplex adatstruktúrát üzenetet tartalmaz. A következő alfülön pedig a böngészőből elérhető felület konfigurációs oldala található mely a következő ábrán látható módon néz ki.



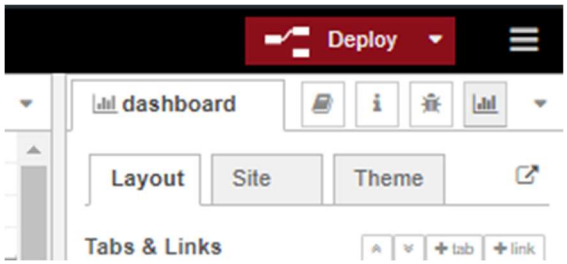
36. ábra NodeRed felhasználói felület konfigurációs menü

Az ábrán látható, hogy 3 oldalból, nagyobb menüből áll a felhasználói felület mely oldalakon belül csoportok helyezkednek el, amely csoportok tartalmazzák a megjelenített grafikus elemeket. A csoport nevére kattintva, jelen esetben pillanatnyi adatok, megjelenik egy layout szerkesztési opció mely segítségével lehetőség nyílik a grafikus felhasználói felület elrendezésének szerkesztésére.



37. ábra NodeRed oldal elrendezés szerkesztése

A fentebb felsorolt felületeken végrehajtott módosítások a jobb oldal felső sarkában elhelyezett deploy elnevezésű gomb megnyomásával léptethetőek érvénybe.



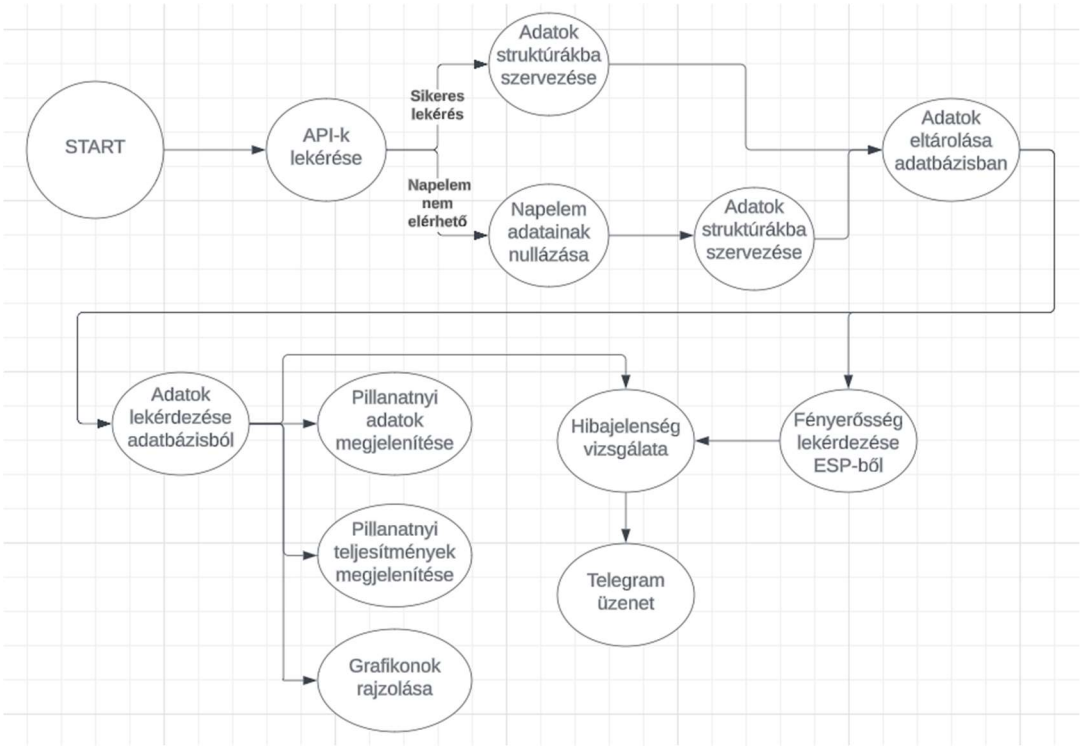
38. ábra NodeRed deploy gomb

A mellette lévő lenyíló menüben pedig a szerkesztői környezetre vonatkozó beállítások találhatóak, valamint ebben a menüben nyílik lehetőség kiegészítőként telepíthető modulok telepítésére mellyel az alap rendszerben nem implementált funkciókat megvalósító node-ok érhetőek el.

3. A MEGVALÓSÍTÁS

3.1 Tervezés

Tervezés során a feladatok prioritásának felállítása során az adatok eltárolását tartottam a legfontosabbnak így, ha a program futása során valami probléma merülne fel a beolvasott adatok eltárolásával, az esetben a felhasználói felületen is láthatóvá válik a hibajelenség. Szempontként tekintettem arra, hogy a kijelzés során minél kevesebb művelet váljék szükségessé ennek okán pedig oly módon valósítottam meg az adatok eltárolását, hogy az adatbázis már olyan mezőket is tartalmaz melynek értékei az eltárolás folyamata keretében kerülnek kiszámításra. A program futása a következő ábrán látható folyamatábra szerint valósul meg.



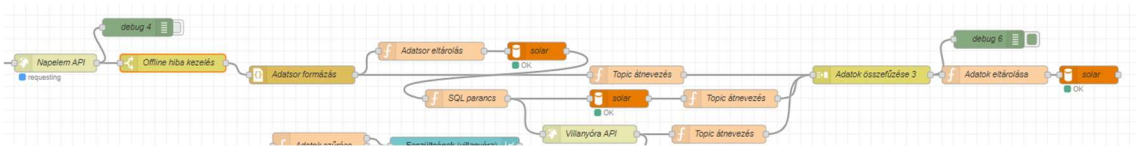
39. ábra Program futását leíró folyamatábra

3.2 Az adatok eltárolása

A program futása egy üres üzenet elküldésével indul. Ezen üzenet keletkezése, igazodva a villamos mérőóra által definiált időintervallumhoz, tíz másodpercenként kerül kiküldésre, ezzel biztosítva szinkronizációt és indokolatlanná téve az adatok beérkezését, változását detektáló kód fejlesztését, mivel a villamos mérőórában elhelyezett eszköz által szolgáltatott API oly módon működik, hogy a villamos mérőóra által a P1 felhasználói interfészen keresztül tíz másodpercenként érkező adatsort eltárolja és amint az API lekérdezésre kerül a legutóbb eltárolt adatsort fogja szolgáltatni, így minden lekérdezés esetén elérhetőek az adatok és a tíz másodperces várakozással töltött időintervallum pedig biztosítja azt hogy minden lekérdezésnél a legfrissebb adatok álljanak rendelkezésre. A probléma viszont abban az esetben merül fel amikor az inverter által szolgáltatott API-kerül lekérdezésre, ugyanis olyan időjárási viszonyok mellett, avagy abban a napszakban, amelyben nem áll rendelkezésre elegendő mennyiségű napenergia az inverter egy olyan energiatakarékos állapotba kerül, amely következtében a Wi-Fi modult lekapcsolja így az API nem áll rendelkezésre, a program futása során hiba keletkezik. Ezen okból kifolyólag az adattárolás két féle módon történhet.

3.2.1 Abban az esetben amikor online a napelem invertere

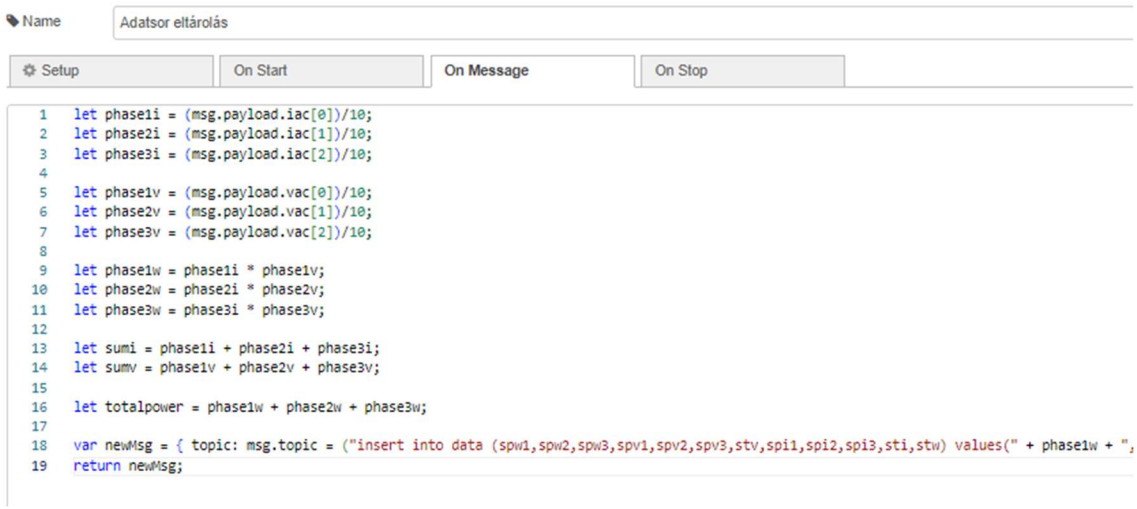
Abban az esetben amikor a napelem inverterének API-ja lekérdezhető állapotban van az alábbi utat járja be a program futása során.



40. ábra Inverter online adatfolyam

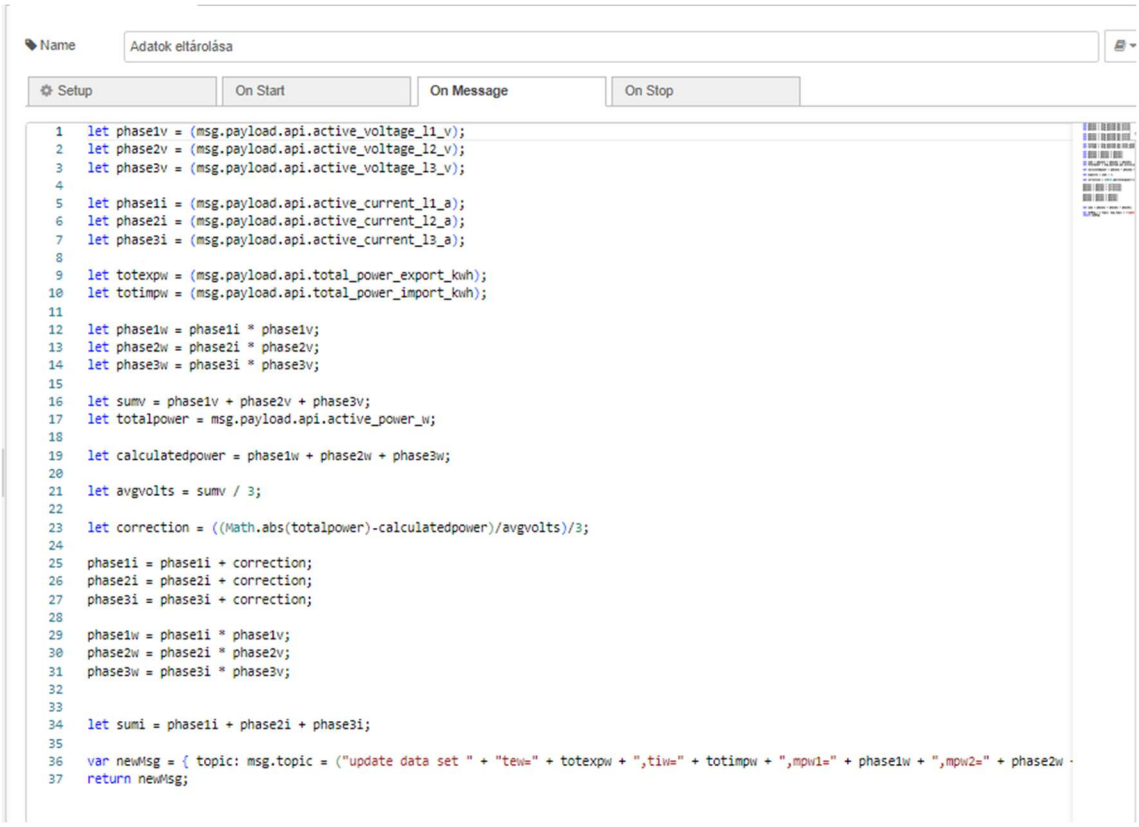
Elsőként az inverterből érkező adatok kerülnek lekérdezésre egy http request elnevezésű node segítségével mely lekérdezés eredményeképpen egy nyers adatstruktúrát ad át a következő node-nak. A következő node vizsgálja azt, hogy az inverter API elérhető állapotban van-e. Amennyiben sikeres volt a lekérdezés azesetben a következő node az adatsort automatikusan JSON formátumból Objektum formátumúvá alakítja át, így az adatsor elemei hivatkozhatóvá válnak. A következtő node a már hivatkozható adatokat előkészíti az

eltárolásra, valamint az API-n keresztül rendelkezésre nem álló adatokat, mint például a fázisokra lebontott teljesítmények, a rendelkezésre álló adatok alapján kiszámolja, majd egy olyan üzenetet küld tovább a következő narancssárga adatbázist képviselő node számára melynek a topic mezőjében kerül elhelyezésre a végrehajtandó adatbázis manipulációt végző adatfeltöltő parancs.



41. ábra Adat tárolás előkészítése, napelem

Ezáltal elhelyezésre kerültek a napelem adatai az adatbázisban viszont a villamos mérőórából származó adatok helyén NULL érték szerepel mivel ezek nem lettek feltöltve. Szükségessé válik egy adatbázis lekérdezés mely eredménye a legutoljára beszúrt rekord azonosítója. Ezzel a folyamattal párhuzamosan a villamos mérőórából is lekérdezésre kerülnek az adatok majd a félig feltöltött adatbázis, a legutolsó rekord azonosítója és a villamos mérőóra adatai birtokában a szálak összeforrnak, az adatok összefűzésre kerülnek majd ismét egy adatbázis manipulációs parancs kiadása következik mely lokalizálja a legutolsó adatsort beszúrja a rekordban a hiányzó adatokat. Ebben a node-ban kapott helyet a villamos mérő egység adatainak formázását, valamint a hiányzó adatok előállítását megvalósító kód is.

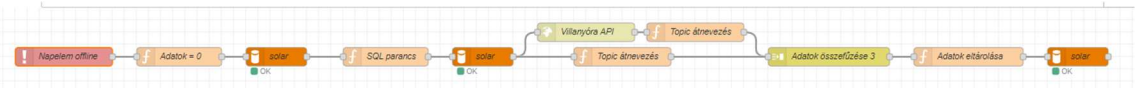


42. ábra Adat tárolás előkészítése, villanyóra

A fejlesztés során azt a megállapítást tettem, hogy a villamos mérő egység API által szolgáltatott fázisokra lebontott teljesítménye összegezve nem egyezik a tényleges energiaáramlás adataival melynek oka az, hogy az interfészen keresztül olvasható fázisokra lebontott áramerősség csak egész pontosságú, így egy korrekciót kellett alkalmaznom a szintén az API-n keresztül elérhető összegzett tényleges teljesítmény alapján mely a P1 felhasználói interfészen aktív teljesítmény néven található meg.

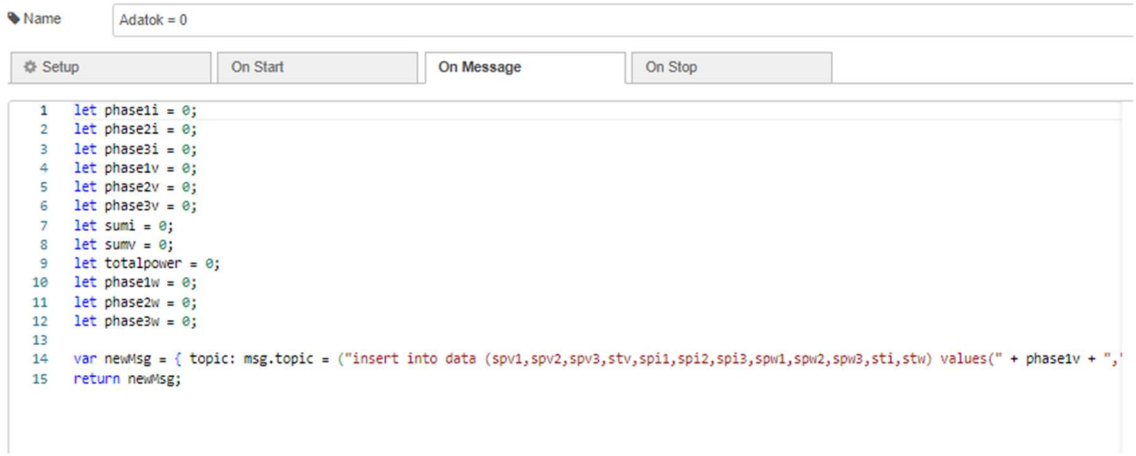
3.2.2 Abban az esetben amikor nincs online állapotban a napelem invertere

Amennyiben a napelem inverterének lekérdezése nem lehetséges, egy hibaesemény fog bekövetkezni mely hibaesemény bekövetkezésekor egy arra a hibára definiált node fog érvénybe lépni és a program futása azon az ágon folytatódik tovább.



43. ábra Inverter offline adatfolyam

Mivel az inverter adatai nem állnak rendelkezésre és az adatbázisban nem optimális NULL értékeket hagyni így fel kell tölteni az adott rekordban kiesett értékek helyét, mégpedig oly módon, hogy az a későbbi folyamatok során ne okoz gondot. Ezt a következő ábrán látható módon oldottam meg.



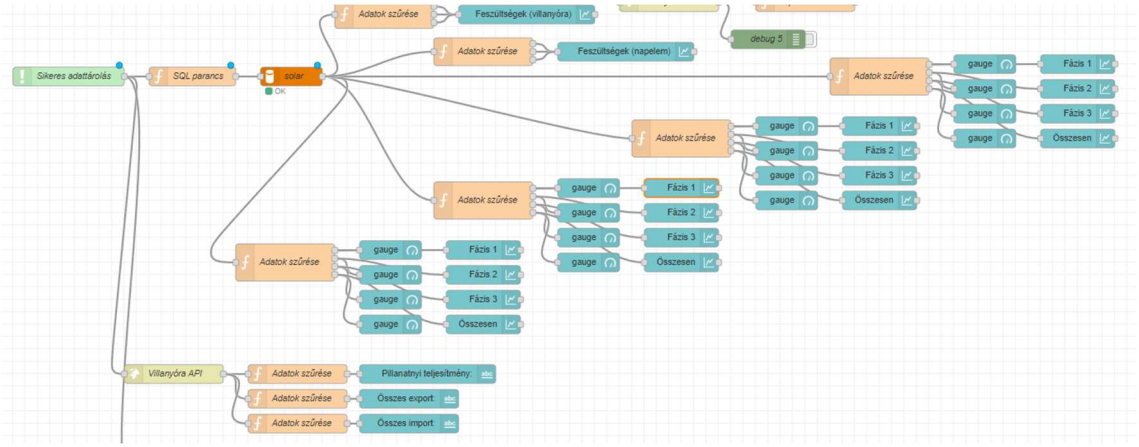
44. ábra Inverter adatainak helyettesítése

Minden olyan adatot, amely az API-n keresztül érkezne, valamint ezen adatokkal való műveletek eredményeként születne, 0 értékkel helyettesítettem majd ezeket egy adatbázist manipuláló parancsban összefoglaltam és elküldtem a parancsot az adatbázist képviselő node részére. A folyamat további része hasonlóan az előző esethez kerül végrehajtásra.

3.3 Az eltárolt adatok megjelenítése, vizualizálása

A következő fejezetekben mutatom be, hogy az eltárolt adatok adatbázisban alkotott nehezen átlátható mátrixából hogyan is lesz a felhasználó számára is egyszerűen értelmezhető grafikus felület. A NodeRed felület bemutatása során fel-fel bukkantak részletek a grafikus felület elemeiről. A webes grafikus felhasználói interfész megnyitása után az oldal bal felső sarkából legördülő menüből három lehetőség közül választhat a felhasználó, ezek pedig a pillanatnyi adatok, a pillanatnyi teljesítmények és a grafikonok. Ezen

opciók közül választhat a felhasználó melyre kattintás után megjelenik a kiválasztott oldal, melyen a megjelenítésért felelős elemek csoportokra bontva jelennek meg. A megjelenítésért felelős elemek adatai az adatbázisban való sikeres eltárolás eseménye hatására frissülnek.



45. ábra Adat vizualizálás folyamata

3.3.1 Pillanatnyi adatok

46. ábra NodeRed pillanatnyi adatok

A pillanatnyi adatok megjelenítése esetén az ábrán látható módon jelennek meg az elemek. Két fő csoport alkotja az oldalt, a bal oldali csoport felel a napelem inverterének

adatai megjelenítéséért, a jobb oldali csoport pedig a villamos mérőóra által szolgáltatott adatokat jeleníti meg. A következő inverter által mért adatok láthatóak az oldalon ebben a sorrendben:

- A fázisokhoz tartozó áramerősségek, jelen esetben 3 fázis
- Az összesített áramerősségek
- A fázisokhoz tartozó feszültségek

A következő csoportban található az előző mintájára szervezett az villamos mérőóra által mért adatokat megjelenítő grafikonok. A jobb felső sarokban kiegészítésként megjelenítésre került a villamos mérőóra által mért pillanatnyi áramirány és teljesítmény mennyiség, az összesen felszerelés óta exportált és importált energiamennyiség.

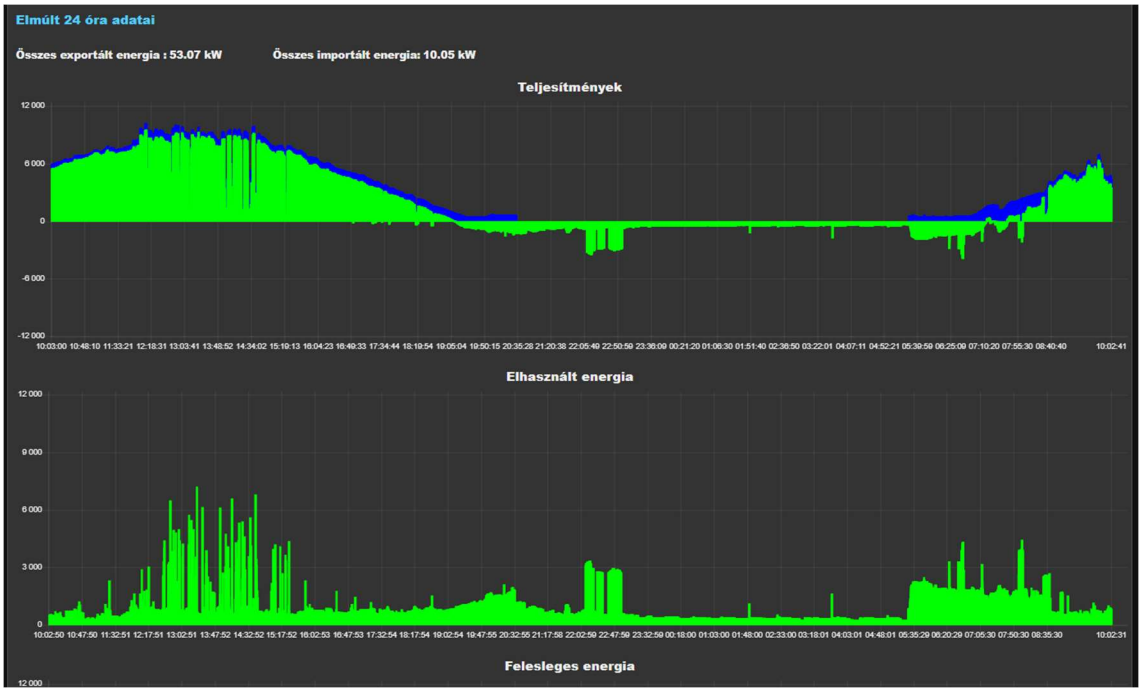
3.3.2 Pillanatnyi teljesítmények

47. ábra NodeRed pillanatnyi teljesítmények

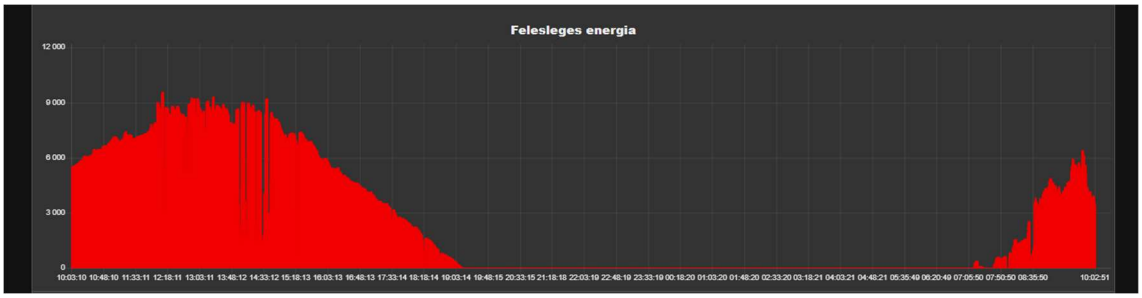
A pillanatnyi teljesítmény oldalon láthatóvá válnak az egyes fázisokon átfolyó teljesítmények. A jobb oldalon találhatóak a napelem inverterén át az elektromos hálózatba távozó teljesítmények fázisok szerint, valamint az utolsó sorban összesítve. A jobb oldali csoportban az előzőleg leírt módon kerülnek megjelenítésre a villamos mérőóra által mért adatok. A grafikonokról leolvasható, hogy a napelem átlagosan fázisonként körülbelül kétezer watt teljesítményt táplál be az elektromos hálózatba melyből az első fázison hét-

száz watt kerül felhasználásra a háztartás által a második fázison kétszáz watt kerül felhasználásra a harmadik fázison pedig teljes mértékben felhasználásra kerül a termelt energiamennyiség. Az utolsó sorban található grafikonokról leolvashatjuk, hogy a rendszer energiatermelésének a körülbelül ötven százaléka kerül felhasználásra a háztartás által, a felesleges energia pedig visszatáplálásra kerül az áramszolgáltató hálózatába.

3.3.3 Grafikonok



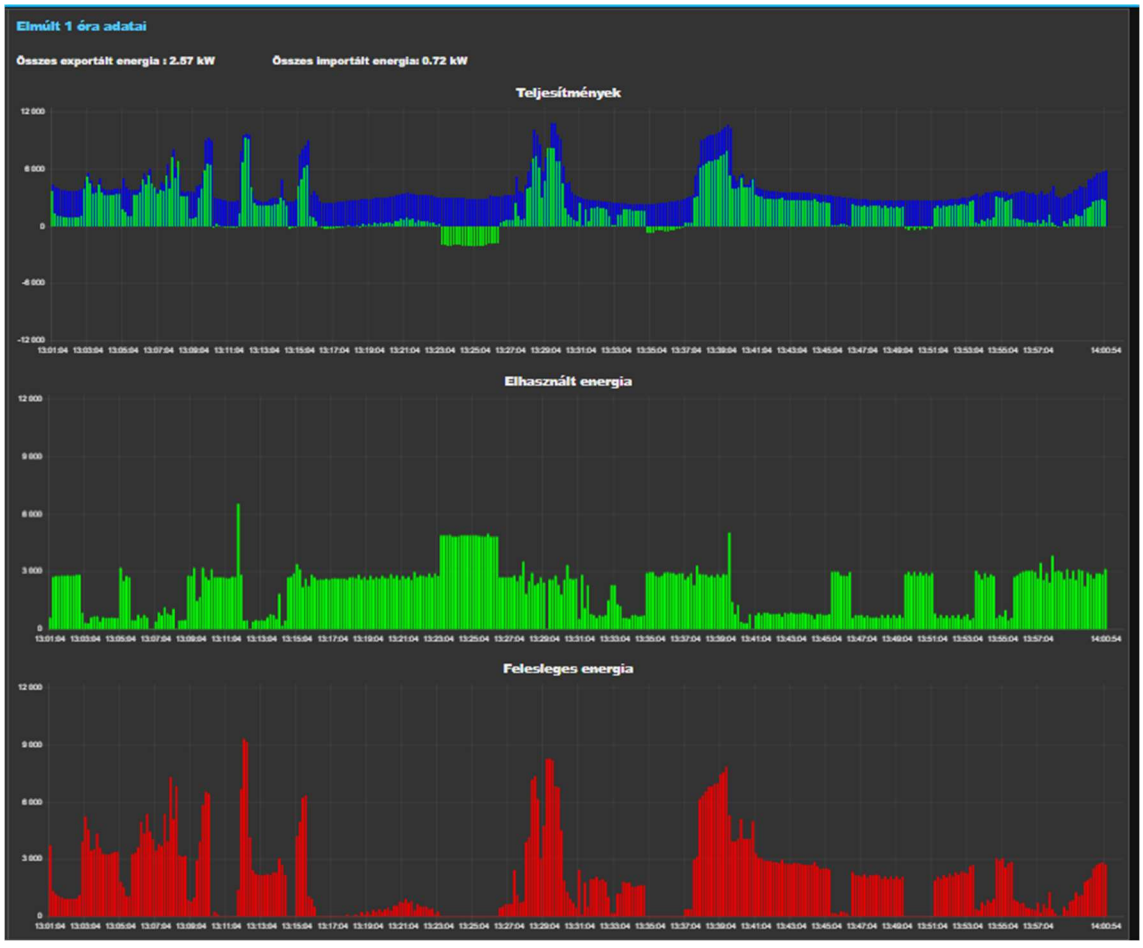
48. ábra NodeRed grafikon 24 óra I.



49. ábra NodeRed grafikon 24 óra II.

Ezen a felületen a pillanatnyi teljesítmények kerültek megjelenítésre az idő függvényében ábrázolva. A fentebb látható grafikonon az elmúlt 24 óra adatai láthatóak az alább beszúrt

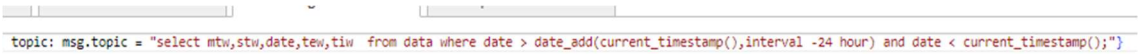
ábrán pedig az elmúlt 1 óra adatai láthatóak. Immáron jól látható napszak szerint is a termelés és fogyasztás arány, nem szükséges a pillanatnyi teljesítményeket megjelenítő oldal folyamatos szemmel tartása. Az első grafikonon zöld színnel látható a villamos mérőóán áramló teljesítmény, a kékkel pedig látható a napelem által termelt teljesítmény. A kettő különbözete adja a középső ábrát mely azt szemlélteti, hogy a termelt energiának mekkora része került felhasználására a háztartásban, a harmadik piros ábra pedig szemlélteti, hogy hogyan alakult az áramszolgáltató hálózatába való visszatáplálás a vizsgált időintervallum alatt. Kiegészítésként megjelenítésre került a grafikonok felett a vizsgált időszak alatt exportált és importált energiamennyiség.



50. ábra NodeRed grafikon 1 óra

Ez a megjelenítési mód ellenben az előző, pillanatnyi esetekkel, nem szimpla adatbázis lekérdezést használ mely lekérdezés az utolsó rekordot adja eredményül az adatbázisban,

hanem egy komplexebb lekérdezést alkalmaz mely figyelembe veszi a jelenlegi időt és a lekérdezésben meghatározott intervallumot és az ezen előre definiált időintervallumon belül keletkezett rekordokat adja eredményként.



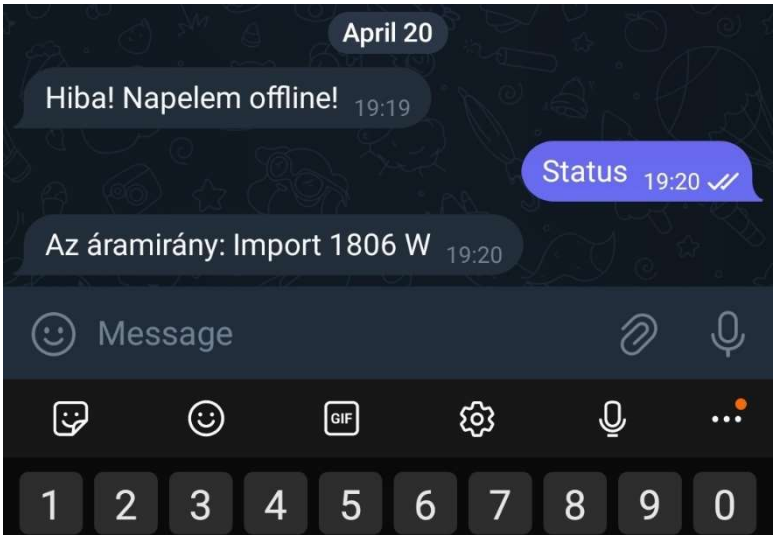
51. ábra Adatbázis lekérdezés

A nagy mennyiségű adatsorok, rekordok miatt a megjelenítéséhez egy külön függvény vált szükségessé, amely a rekordokat a grafikonok számára is értelmezhető formátumúra alakítja.

52. ábra Rendszerező függvény

3.4 Kommunikáció

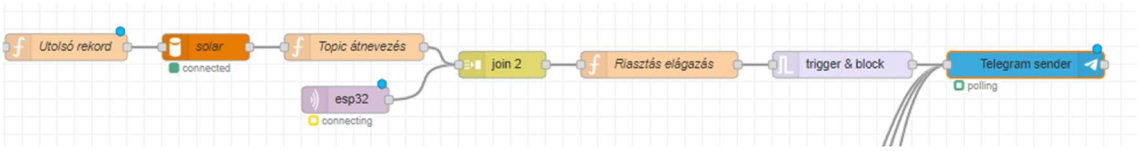
A kommunikációt a NodeRed telegram kiegészítő modul telepítése után kezdhettem el konfigurálni. Elsőként telepítenem kellett a telegram alkalmazást a telefonomra, majd regisztráció után a BotFather elnevezésű szolgáltatást vettem igénybe, amely segítségével a szükséges adatok megadása után létrehoztam egy telegram botot, valamint a bothoz tartozó chat felületet. A bot alapvetően úgy viselkedik, mint ha egy ember ülne a chat felület túloldalán pedig az igazság az, hogy a programom küldi és fogadja az üzeneteket. Az újonnan létrehozott botom azonosítóit a NodeRed felületen megadva már működésre is lehetett bírni. Az ábrán látható módon megvalósítottam egy riasztást mely értesíti a felhasználót, valamint implementáltam egy felhasználó által kiadható parancsot.



53. ábra Telegram felület

3.4.1 Hibaesemény detektálás

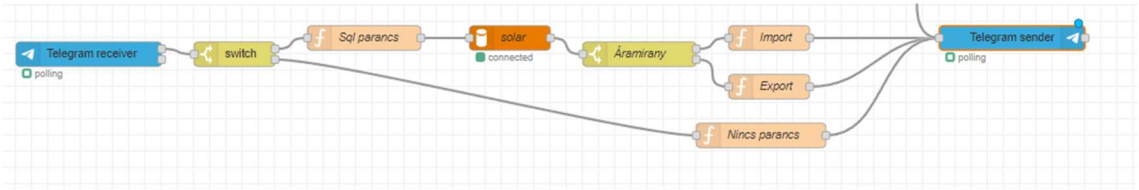
A riasztást kiváltó hibaesemény pontosabban arra vonatkozik, hogy amennyiben az ESP32 által mért fénymennyiség magasabb egy bizonyos határértéknél és a napelem invertere offline állapotba kerül az esetben egy üzenetet küld a felhasználó számára, hogy valamilyen okból kifolyólag a környezeti körülmények ideálisak a termelés számára viszont nem történik termelés.



54. ábra Riasztás folyamata

Az ábrán az látható, hogy a futás során elsőként a program lekérdezi az adatbázisban szereplő utolsó rekordot, valamint ezzel párhuzamosan lekérdezi az ESP32 által mért fénymennyiséget az MQTT brokertől ezen adatokat egyesíti és amennyiben a feltételek teljesülnek tehát hiba áll fenn azesetben üzenetet küld a program a felhasználó részére.

3.4.2 Felhasználói parancs végrehajtás



55. ábra Telegram parancs

Először a telegram bot vevő node-jának szerkesztői felületen való elhelyezése után tudtam kísérletezni a fogadott üzenet formátumával, struktúrájával. Ez után megírtam azt, hogy amennyiben a felhasználó elküldi a status parancsot azesetben lekérdezi az adatbázisból a jelenlegi áramirányt és az áramló teljesítményt. Amennyiben elírás történik, avagy nem a status parancs érkezik azesetben a program jelzi, hogy nincs ilyen parancs.

4. TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK

A program jelen állapotában még több kiaknázatlan fejlesztési lehetőséget kínál megvalósításra. Pontosabb fényerősség mérő szenzorok elhelyezésével a napelemek által borított területen pontosabb információkat kaphatunk a pillanatnyi fényviszonyokról így statisztikákat lehet előállítani a napelemek által termelt energiáról mely segítségével olyan állapotokat is lehet detektálni, mint például a nagy mértékben szennyeződött. tisztítást igénylő napelemek. A felhasználói felület jelenleg webes felületen keresztül érhető el így a mobiltelefonokon való megtekintés kissé nehézkessé válik, indokolttá válik a mobiltelefonos applikáció fejlesztése. Riasztások tekintetében további riasztásokat lehet implementálni, mint például a túlfeszültség vagy a magas áram által kiváltott riasztás.

5. ÖSSZEFOGLALÓ

A dolgozatom írása során bemutattam a napelemek legnépszerűbb telepítési konfigurációit, amelyeket otthoni környezetben meg lehet találni, definiáltam a rendszer részét képző inverter feladatát. A napelemek által termelt energiát valamiféle úton fel kell használni, hogy megtérüljön a bele fektetett tőke így bemutattam ezen az úton termelt energiának az útját, hogy hogyan keletkezik az úgymond felesleges energia, valamint, hogy ez a felesleges energia hová is tűnik az elektromos hálózatban. A dolgozatom célja ennek a felesleges energiatöbbletnek a minimalizálása annak érdekében hogy a napelemes rendszer minél hatékonyabban üzemeljen mégpedig oly módon hogy vizuálisan grafikonok segítségével szemlélteti a felhasználó számára hogy a vizsgált pillanatban avagy napszakban milyen arányban keletkezik a napelemek által termelt energia és ennek a termelt energiának mekkora része kerül visszatáplálásra, más szóval kifejezve optimalizálja az általa ellátott háztartás napelemes energiával történő üzemelését. Munkám során bemutattam a felhasznált eszközöket mind szoftver, mind hardver szempontjából így például program futását, magját képviselő NodeRed fejlesztői környezetet, a mögötte elhelyezkedő MySQL adatbázist és ESP32-es típusú mikrokontrollert, valamint az ezek kommunikációját biztosító hálózatot. Kitérőt tettem a hálózaton történő kommunikáció alapköveire, mint például az átviteli közegek és a különböző címek, amelyek hiányában a dolgozatom létre sem jöhetett volna.

6. SUMMARY

During the writing of my thesis, I presented the most popular installation configurations of solar plants that can be found in a home environment, I defined the role of the inverter that is part of the system. The energy produced by solar panels has to be used in some way to recoup the capital invested in it, so I showed the way the energy is produced, and way taken by this produced energy. I showed how the so-called excess energy is generated, and where this excess energy goes in the power grid. The aim of my thesis is to minimize this excess energy in order to make the photovoltaic system operate as efficiently as possible by visually illustrating with the help of graphs for the user what proportion of the energy produced by the solar panels and how much of this generated energy is fed back into the power grid, in other words, optimizing the household consumption supplied by solar energy. During my work, I presented the tools used both from the point of view of software and hardware. Tools such as the NodeRed development environment representing the running and core of the program, the MySQL database and ESP32 type microcontroller behind it, as well as the network providing their communication. I detoured the cornerstones of communication on the network, such as transmission media and various addresses, without which my thesis would not have been possible.

7. HIVATKOZÁSOK

[1] E-on, „P1 port felhasználói interfész,” 10 Február 2023. [Online]. Available: https://www.eon.hu/content/dam/eon/eon-hungary/documents/Lakossagi/aram/muszaki-ugyek/p1_port%20felhaszn_interfesz_taj_%2020230210.pdf. [Hozzáférés dátuma: 14 április 2024].

[2] Espressif, „ESP32 mikrokontroller,” [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf. [Hozzáférés dátuma: 19 április 2024].

[3] Wikipedia, „RAID adattárolási módszer,” [Online]. Available: https://en.wikipedia.org/wiki/RAID. [Hozzáférés dátuma: 20 április 2024].

[4] Wikipedia, „OSI modell,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/OSI_model. [Hozzáférés dátuma: 22 április 2024].

[5] Wikipedia, „Ethernet keret,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Ethernet_frame. [Hozzáférés dátuma: 22 április 2024].

[6] FreeBSD, „ZFS fájrendszer,” FreeBSD, [Online]. Available: https://docs.freebsd.org/en/books/handbook/zfs/#:~:text=ZFS%20is%20an%20advanced%20file,a%20checksum%20of%20the%20data.. [Hozzáférés dátuma: 22 április 2024].

[7] „Ganesh V. Karbhari, Dr. Pragya Nema. Adaptive Solar Energy Management System based on Internet of Things. International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; Volume 8 Issue III Mar 2020,” [Online]. Available:

https://www.ijraset.com/fileserve.php?FID=27017. [Hozzáférés dátuma: 1 április 2024].

[8] „Isak Borg, August Dixelius, David Östlund. Interactive Visualization of Solar Energy Data. Sjalvstandigt arbete i informationsteknologi," [Online]. Available: https://uu.diva-portal.org/smash/get/diva2:1325802/FULLTEXT01.pdf. [Hozzáférés dátuma: 11 április 2024].

[9] „Duarte OG, Rosero JA, Pegalajar MdC. Data Preparation and Visualization of Electricity Consumption for Load Profiling. Energies. 2022; 15(20):7557," [Online]. Available: https://doi.org/10.3390/en15207557. [Hozzáférés dátuma: 8 április 2024].

[10] „Györök György, Tihomir Trifonov, Alexander E. Baklanov, Bertalan Beszédes, Svetlana V. Grigoryeva, Aizhan Zhaparova. A Special Robust Solution for Battery Based Power Supply. In: Orosz, Gábor Tamás (szerk.) 11th International Symposium on Applied Informat," [Online].

[11] „Attila Sáfár, Bertalan Beszédes. Educational Aspects of a Modular Power Management System. In: Orosz, Gábor Tamás (szerk.) AIS 2019 : 14th International Symposium on Applied Informatics and Related Areas organized in the frame of Hungarian Science Festiva," [Online].

[12] „Mosavi Amirhosein, Bertalan Beszedes, Imre Felde, Nadai Laszlo, Gorji Nima E. Electrical characterization of CIGS thin-film solar cells by two- and four-wire probe technique. MODERN PHYSICS LETTERS B 2020 p. 2050102 , 16 p. (2020)," [Online].

8. ÁBRAJEGYZÉK

1. ábra Villamos mérőóra 4
2. ábra Napelemek 6
3. ábra Az inverter 7
4. ábra Inverter API 8
5. ábra ESP32 és fotorezisztor 9
6. ábra Kapcsolási rajz 11
7. ábra Áramkör mérés 1 12
8. ábra Áramkör mérés II. 13
9. ábra ADC konverzió eredménye 13
10. ábra A szerver funkciót ellátó számítógépek 16
11. ábra DHCP: cím allokálás 20
12. ábra Hálózati topológia 21
13. ábra Proxmox kezdőképernyő 22
14. ábra Szerver menüje 23
15. ábra Szerver merevlemezei 24
16. ábra ZFS kötet 25
17. ábra ZFS kötet merevlemezei 25
18. ábra CT Templates almenü 26
19. ábra Szerver local kötet 26
20. ábra Ubuntu container template letöltés 26
21. ábra Container konfigurációs folyamat I. 27
22. ábra Container konfigurációs folyamat II. 27
23. ábra Container konfigurációs folyamat IV. 28
24. ábra Container konfigurációs folyamat III. 28
25. ábra Container futás közben 29
26. ábra ESP32 forráskód, MQTT broker 31
27. ábra MySQL Workbench 32
28. ábra phpMyAdmin belépés 33

29. ábra phpMyAdmin adatbázisok 33
30. ábra phpMyAdmin solar tábla mezők elnevezése 34
31. ábra Betekintés a MySQL adatbázisba 35
32. ábra NodeRed fejlesztői felület 37
33. ábra NodeRed eszköztár 39
34. ábra NodeRed szerkesztő felület 40
35. ábra NodeRed debug felület 41
36. ábra NodeRed felhasználói felület konfigurációs menü 42
37. ábra NodeRed oldal elrendezés szerkesztése 43
38. ábra NodeRed deploy gomb 43
39. ábra Program futását leíró folyamatábra 44
40. ábra Inverter online adatfolyam 45
41. ábra Adat tárolás előkészítése, napelem 46
42. ábra Adat tárolás előkészítése, villanyóra 47
43. ábra Inverter offline adatfolyam 48
44. ábra Inverter adatainak helyettesítése 48
45. ábra Adat vizualizálás folyamata 49
46. ábra NodeRed pillanatnyi adatok 49
47. ábra NodeRed pillanatnyi teljesítmények 50
48. ábra NodeRed grafikon 24 óra I. 51
49. ábra NodeRed grafikon 24 óra II. 51
50. ábra NodeRed grafikon 1 óra 52
51. ábra Adatbázis lekérdezés 53
52. ábra Rendszerező függvény 53
53. ábra Telegram felület 54
54. ábra Riasztás folyamata 55
55. ábra Telegram parancs 55