



Széchenyi István Egyetem
Gépészmérnöki, Informatikai és Villamosmérnöki Kar

TMDK Dolgozat

**Rádiófrekvenciás spektrumanalizátor távoli elérése,
adatok tárolása, vizualizációja, eszköz vezérlés**

Mészáros András

(III. éves Mérnökinformaticus B.Sc. hallgató)

Konzulens: Németh Péter Ernő (tanársegéd)

Széchenyi István Egyetem
2022. október
Győr

Tartalomjegyzék

Rádiófrekvenciás spektrumanalizátor távoli elérése, adatok tárolása, vizualizációja, eszköz vezérlés	1
Tartalomjegyzék.....	3
1 Absztrakt / kivonat	5
2 Bevezetés	6
3 Mérési körülmények	8
3.1 Spektrumanalizátorok Magyarországon	8
3.2 A műszer.....	10
4 Kapcsolatteremtés	11
4.1 Csatlakozás hálózaton keresztül	11
4.1.1 Mi a VPN?	12
4.2 A kommunikáció	12
4.3 Vezérlés	13
5 A szoftver.....	14
5.1 Szoftverkörnyezet.....	14
5.1.1 Socket.....	14
5.1.2 PyQt	15
5.1.3 NumPy	16
5.1.4 Anaconda	16
5.1.5 PyQtGraph	17
5.2 Grafikus felület.....	17
5.2.1 Elérhető funckiók.....	17
5.2.2 Kontrasztos mód	18
5.3 Adattárolás.....	18
5.4 Emailküldő rendszer	19

5.5	Rutin mérési feladat az eszköz segítségével	21
5.5.1	DVB-T teszt	21
5.5.2	FM teszt	22
5.5.3	Vészüzemmód.....	22
5.5.4	Stresszteszt.....	23
6	A jövő.....	27
6.1	Mobilalkalmazás.....	27
6.2	Távoli Ki-Be kapcsolás	28
6.3	Emailküldő rendszer	29
6.4	Adatbázis	30
6.5	Lekérdezések	31
6.6	Kompatibilitás	31
6.7	Üzemeltetés, támogatás	32
7	Kapott eredmények, tapasztalatok	33
8	Ábrajegyzék	34
9	Irodalomjegyzék	35

1 Absztrakt / kivonat

A TMDK dolgozatom témája a spektrumanalizátor mérőműszer távoli elérése, adatok kinyerése, tárolása, vizualizációja, illetve a műszer vezérlésének megvalósítása.

A spektrumanalizátor a rádiófrekvenciás spektrum egy széles frekvencia tartományába eső jelek spektrumát méri, majd ennek függvényében jeleníti meg a jel teljesítményt egy diagramm formájában a saját kijelzőjén.

Vannak olyan speciális mérési feladatok, ahol a körülmények nem engedik meg, hogy ember is tartózkodjon a mérési helyszín közelében, vagy csak nem szeretnénk elutazni az akár több száz kilométerre lévő mérési helyszínre, ahol az eszköz található fizikailag. Az újabb eszközök, már képesek erre a funkcióra, viszont az általunk SZESAT szakkollégium által használt eszköz, erre még nem képes és ez különösen akkor jelent problémát, mikor az egyetem nyitvatartási idején kívül szeretnénk használni a berendezést.

Munkám fő célja az előbb említett problémák kiküszöbölése a spektrumanalizátor távoli elérésének megvalósításával interneten keresztül.

Dolgozatomban bemutatom, hogy miként oldottam meg egy általam készített számítógépes programmal a spektrumanalizátor távoli elérését. Ezen felül bemutatom, hogy milyen funkciókkal láttam el a programot, amelyekkel a rutin mérési feladatok folyamán keletkező igényeket, ilyenek például az adatok tárolása, vizualizációja, email értesítés küldése kiugró értékeknél, vagy akár mérési intervallumok távolról való állítása ki lehet elégíteni. Emellett számba vettem, hogy milyen további funkciókkal lehetne tovább bővíteni a programot, és ezeknek a funkciók fejlesztésének közben milyen problémákat kellene leküzdeni.

2 Bevezetés

Számtalan ok van arra, hogy miért érdemes a rádiófrekvenciás spektrumanalizátort távolról elérni, vezérelni. Legfőképpen a mérési helyszínek elhelyezkedése ad erre indokot. Számtalan helyzet van, ahol az emberi szervezetre káros fizikai behatások lehetnek, ilyen lehet például a radioaktív sugárzás, túlságosan magas vagy túlságosan alacsony környezeti hőmérséklet, vagy csak simán nehezen megközelíthető helyen van a mérési helyszín (például egy hegytetőn).

A koronavírus járvány következtében pedig előtérbe kerültek a különböző otthoni munkavégzési formák, melyek mára, különösen az IT szektorban alapvető igénynek számítanak egy friss munkavállalónál, hogy munkahetének egy bizonyos részét otthoni munkavégzéssel tölthesse. Ehhez pedig elengedhetetlen, hogy munkaeszközei távolról is elérhetőek lehessenek, legyen az a céges Notebookjával VPN-en keresztül való becsatlakozás, vagy akár maga a spektrumanalizátor elérése ezen a platformon.

Az én esetemben a SZESAT szakkollégium által használt Agilent E4440A típusú spektrumanalizátorhoz készítettem el egy számítógépes programot, mellyel meg tudom valósítani az eszköz távoli elérését, az adatok kinyerését, tárolását, vizualizációját és még egy sor napi rutin mérési feladatokat megkönnyítő extra kényelmi funkciót, amikkel egyszerűbbé tudjuk tenni a felhasználóink életét. Az elsődleges célunk az alap távoli elérés biztosítása volt. Mi esetünkben azért volt erre szükség, mert a spektrumanalizátor egy rendkívül drága és nehezen mozgatható eszköz, amit az egyetemi laborunk kölcsönbe kapott a Nemzeti Média- és Hírközlési Hatóságtól (továbbiakban: NMHH), oktatási, és kutatási célokra, hogy valós körülmények között készülhessünk arra, ami a távközlési iparágban fog minket várni. Ennek fényében az eszközt csak a labor területén használhatjuk, ellenőrzött körülmények között. Ez a labor, mivel az egyetem területén belül található, csakis a nyitvatartási időn belül tudjuk látogatni, ami szorgalmi időszakban nem jelent különösebb problémát, viszont a nyári leállások és szabadságolások alkalmával az egyetem korlátozott nyitvatartási idővel rendelkezik. Méréseket pedig ezekben az időszakokban is végeznünk kell, ezért szükségessé vált, hogy távolról tudjuk vezérelni az eszközt.

Magához a spektrumanalizátor működésének a pontos megértéséhez és szakszerű kezeléshez sok évnyi villamosmérnöki tapasztalat, tudás szükséges, amivel én nem rendelkezem, hiszen csak egy egyszerű Mérnökinformaticus hallgató vagyok, az én képzési

formámnak, semmilyen szinten nem része a villamosságtan. Az én feladatom csupán az eszközhöz távoli eléréséhez szükséges program elkészítése volt. Dolgozatomban is a szoftverre és az informatikai megoldásaimra fektettem a hangsúlyt, nem pedig magára az eszközre és a mérésekre, bár alap szinten említést teszek róluk is.

3 Mérési körülmények

3.1 Spektrumanalizátorok Magyarországon

Leggyakoribb eset, hogy miért szükséges számunka a spektrumanalizátor távoli elérése az a távolság. Magyarországon az NMHH a leggyakoribb felhasználója ezeknek az eszközöknek. Mivel nekik tevékenységi körükből adódóan az ország minden pontján tudniuk kell betartatniuk az elektronikus hírközlésről szóló törvényt, ezért az egész országot lefedő mérési hálózatra van szükségük, hogy tudják ellenőrizni valós időben, történt-e jogosulatlan sávfelhasználás. Erre a célra speciális automata mérőállomásokat telepítettek, melyet az erre a célra létrehozott egységesített rendszerbe integráltak. Ezen belül egy magyar fejlesztésű *Országos Spektrum és Interferencia Monitoring* (SIMON) névre keresztelt speciális szoftverrel valósítják meg a távoli elérés funkcióját. Ez egy jelenleg 55 mérőállomásból álló hálózat, mely az ország minden pontján megtalálható. Egy ilyen telephelyben megtalálható egy vagy akár több rádiófrekvenciás spektrumanalizátor, melyeket, a budapesti központból távolról el tudják érni interneten keresztül.



1. ábra Kihelyezett NMHH mérőállomás [26]

Ezek az állomások általában 30 MHz- 6 GHz közötti frekvenciatartományban képesek vizsgálni a jel spektrumát. Ebben a sávban helyezkednek el a különböző műsorszóró adók, telekommunikációs sávok. A mérőállomások segítségével pedig képesek vagyunk ezeket a

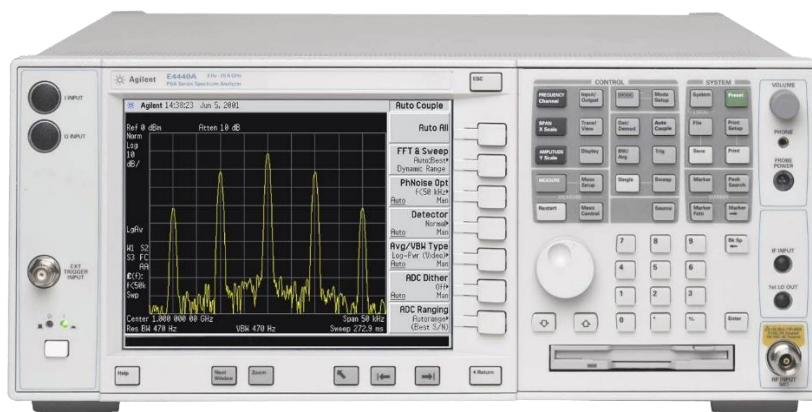
jelforrásokat azonosítani, továbbá az irányukat is meghatározni. Erre azért van szükség, hogy meg tudjuk védeni a fogyasztókat, illetve a jogkövető magatartást tanúsító piaci szereplőket azáltal, hogy monitoringozzuk az általuk előre bejegyzett frekvenciákat, hogy ne lehessen illetéktelen felhasználás ezen a területen, ami zavart okozhatna. Amennyiben egy ilyen zavart észlel az állomás, származzon az bármilyen forrásból, mint például egy szándékos zavaró forrás, akkor jelez a Budapesti központnak, ahonnan az NMHH munkatársai elindulnak és a mérőállomástól kapott paraméterek (jel teljesítmény, irány) alapján meglátogatják a zavar forrását. Ilyenkor pedig visznek magukkal egy mobil spektrumanalizátort, hogy a helyszínen pontosan meg tudják állapítani, hogy honnan is jön pontosan a jel, majd ez alapján elvégzik a szükséges eljárásokat, hogy megszüntessék a zavarforrást. A mérőállomásokon és a Budapesti központon kívül vannak még az NMHH-nak több telephelye, viszont azon részlegük, amelyik a korábban vázolt tevékenységet végzi csak Budapesten található, ezért is fontos szempont számukra, hogy távolról vezérelhetőek legyen a mérőállomások, hogy ne kelljen súlyos időbeli és anyagi ráfordítások árán kiszállniuk Budapestről az akár az ország teljesen más szegletében található mérőállomáshoz. Amennyiben bármilyen szoftveres módosítást akarnak végrehajtani, akkor ezt is távolról meg lehet tenni. (Például a vizsgált frekvenciatartomány végpontjainak állítása.)



2. ábra Országos Spektrum és Interferencia Monitoring (SIMON) mérőállomás rendszer [22]

3.2 A műszer

Az általunk használt eszköz egy Agilent E4440A típusú rádiófrekvenciás spektrumanalizátor. Ezen belül is a nagytudású, sok beállítási lehetőséggel rendelkező spektrumanalizátorok közé tartozik. Ez 3 Hz-től 26.5 GHz- es frekvenciatartományban képes mérni a jelek spektrumát és a hozzájuk tartozó amplitúdót. A frekvenciatartományt a kijelzője X tengelyén, az amplitúdót pedig az Y tengelyén jeleníti meg az eszközt.



3. ábra Agilent E4440A típusú rádiófrekvenciális spektrumanalizátor [1]

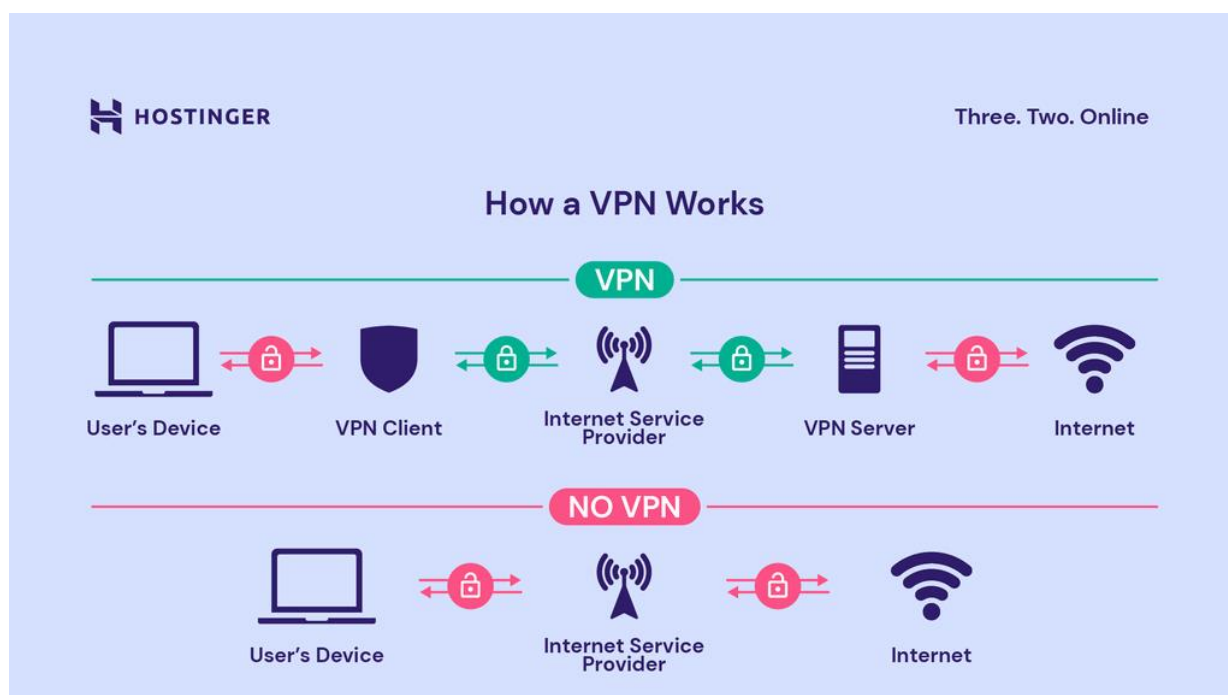
4 Kapcsolatteremtés

4.1 Csatlakozás hálózaton keresztül

A szoftver fejlesztésének megkezdése előtt az elsődleges feladatom az eszközzel való kommunikáció felépítése volt. Mivel ezelőtt semmilyen módon nem kommunikáltunk külsős eszköztől még helyi hálózaton az eszközzel nehéz dolgom volt. Szükségem volt az eszköznek az IP címére, illetve a Port számára. A gyártói leírásból [2] kiderült, hogy az eszköz IP címe állítható a spektrumanalizátor saját felületén, ezt be is állítottam egy általam tetsző címre, viszont a portszám nem volt adott. Első körben direkt módon kötöttem rá a számítógépem az eszköz hálózati csatlakozójára internetkábelén keresztül és megpróbáltam teszt adatcsomagokat küldeni, viszont itt nem jártam sikerrel. Ezek után rákötöttem a labor által használt routerre a spektrumanalizátort internetkábelén keresztül, majd rácsatlakoztam a router egy másik fizikai portjára a számítógépen keresztül. A Wireshark [3] nevezetű szoftverrel felmértem az összes belső hálózaton áthaladó csomagot. Ekkor egy óriási listát kaptam tele Port számokkal. Ezek után parancssorból teszt adatcsomagokat kezdtem kiküldeni a címekre egyesével, az általam az eszközön korábban beállított IP címre, és mellé a korábban listázott Port számokat kezdtem próbálgatni, majd hosszú idő után megtaláltam a keresett kombinációt. Ezen a szinten képesek voltunk már helyi hálózaton belül kommunikálni az eszközzel, viszont ez számunkra nem volt elégséges, hisz az alap problémát, miszerint nem lehetett az eszközzel külső hálózatról interneten keresztül kommunikálni még nem oldottuk meg. Ehhez pedig első körben Port Forwardingot kellett beállítanom a labor routerjén, viszont ez még önmagában nem számított volna biztonságos megoldásnak ezért össze csatoltam a már meglévő VPN szolgáltatásunkkal.

4.1.1 Mi a VPN?

A virtuális magánhálózatok (VPN) [4] segítségével képesek vagyunk titkosítani a személyes adatainkat, és elrejteni a valódi IP-címeinket egy nyilvános hálózaton. Ez az egyik leggyakoribb megoldás az internetes adatvédelem és biztonság védelmére. Az adathalászat nevezetű számítógépes támadásokat is nagy mértékbe ki lehet a segítségével küszöbölni. A hagyományos távoli asztal szolgáltatásokhoz képest az a nagy előnye, hogy itt nem csak egy adott eszközre lehet rácsatlakozni távolról, hanem az egész hálózatra. Több típusa is létezik, az általunk használt távoli elérésű Cisco AnyConnect [5] VPN szolgáltatás a leggyakoribb verzió. Alapvetően arra a célra lett kifejlesztve, hogy a cégek a saját belső hálózatukat ki tudják úgy terjeszteni, hogy távolról be tudjanak rá csatlakozni az alkalmazottaik és az összes hálózati funkciót el tudják érni biztonságos módon, csakúgy, mint a fizikálisan is bent lévő társaik, akik



4. ábra Távoli elérés megvalósítása VPN-el és nélküle [25]

direkt módon érik a belső hálózatot.

4.2 A kommunikáció

Mint az korábban említettem az eszközzel TCP/IP [6] protokollon keresztül kommunikálunk. Ez egy olyan nyílt forráskódú iparági modell, amelyet a hálózaton keresztüli kommunikációval kapcsolatos problémák megoldására fejlesztettek ki. Emellett ingyenes és bárki által használható. További nagy előnye, hogy minden egyes hálózati eszközhöz saját egyedi IP címet rendel, mellyel így később könnyebb tervezni például a mi programunk esetén is, hogy privát és fix a spektrumanalizátor IP címe. Ennek ellenére belefejlesztettem egy menüt

a programba, hogy át lehessen ezt az IP címet állítani kézzel egy grafikus felületen anélkül, hogy módosítani kellene a programkódot, mert bármi megtörténhet, például az, hogy átírja valaki magában az eszközben a címet.

4.3 Vezérlés

A protokollon keresztül szöveges üzenetként küldjük tovább a parancsokat, ezzel imitálva a gombnyomásokat az eszközön. Ezek a parancsok az SCPI (Command for Programmable Instruments) szabványt követik. Ezt a protokollt még a HP fejlesztette ki a különböző mérőműszerek kommunikációjának, vezérlésének szabványosításához, jelenleg pedig egy nemcégspecifikus konzorcium kezeli ezt. Az SCPI parancsok ASCII szöveges sztringek, amelyek a fizikai rétegen (pl. IEEE-488.1) keresztül jutnak el a műszerhez. A parancsok egy vagy több kulcsszavak sorozatai, amelyek közül sokan paramétert is felvesznek. Ilyen parancsok például a `FREQ:STAR`, (kezdőfrekvencia állítás), vagy akár a `TRAC2:MODE VIEW` (trace mód állítása). Visszafelé is ilyen és ehhez hasonló szöveges parancsokat kapunk majd ezek dekódolásával ki tudjuk olvasni a nekünk szükséges paramétereket (például adott Frekvenciához tartozó jelszint). Majd ezeknek tárolásával, feldolgozásával és végül megjelenítésével értékes grafikonokat tudunk előállítani.

5 A szoftver

5.1 Szoftverkörnyezet

Miután képesek lettünk kommunikálni az eszközzel a következő lépés a program elkészítése volt. Mindenekelőtt azonban el kellett döntenem, hogy milyen platformra szeretnénk ezt fejleszteni. Adódott jogosan az igény, hogy egyaránt működjön mobil, illetve hagyományos asztali eszközökről is. A programozási nyelv kiválasztásánál pedig az volt az elsődleges szempont, hogy a lehető legtöbb dobozos megoldást tudjam alkalmazni, hogy ne kelljen minden egyes funkciót saját magamnak elkészítenem nulláról, lehetőleg az alap mechanikákat csak konfigurálnom kelljen. Viszont a mai világban nem elég csak elkészíteni egy szoftvert, mert az önmagában semmire sem elég, azt üzemeltetni is kell és az sem opció, hogy ezt a feladatot csak én tudjam ellátni, mert bármi történhet velem és ebben az esetben használhatatlan lesz a szoftver. Ezért figyelembe kellett vennem, hogy kik a felhasználói célcsoportom. A SZESAT szakkollégiumban többnyire Távközlés szakirányos villamosmérnök hallgatók vannak, akik bár nem programozók ők is elég behatóan tanulnak Python programozást. Nem véletlenül, hiszen a Python egy egyszerűen megérthető magas szintű programozási nyelv és szinte bármi megvalósítható benne. Kifejezetten ajánlott mikrokontroller programozáshoz, számtalan külső könyvtár érhető el hozzá, melyek megkönnyítik a programozást. Ezért is választottam ez a programozási nyelvet és ökoszisztémát és mivel az egész program a későbbi külsős üzemeltetésre lett fejlesztve rengeteg kommentet is csináltam a programkódon belülre, hogy annak a fejlesztőnek akinek esetleg a jövőben bele kell módosítania a programba könnyebben megértse annak működését és logikáját.

5.1.1 Socket

Több ilyen külső könyvtárat is felhasználtam, melyek közül a *Socket*et emelném ki.

```
def connectToDevice(self):
    ip = self.ipAddressEdit.text()
    port = int(self.portEdit.text())
    self.device = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    self.device.connect((ip, port))
    self.device.send("*IDN?\n\r".encode())
    self.idn=self.device.recv(1024).decode()
```

5. ábra Csatlakozás az eszközhöz Socket segítségével

A Socket külső könyvtár segítségével mindössze ennyi sornyi programkóddal képesek vagyunk megvalósítani a hálózati kommunikációt a számítógépünk és a spektrumanalizátor között. Óriási előnye ennek a csomagnak, hogy univerzális módon képes működni szinte bármilyen, hálózati kommunikációra képes eszközzel, legyen az a túloldalt egy Spektrumanalizátor, nyomtató, vagy akár egy Oszcilloszkóp. Nekünk csak annyi a dolgunk, hogy megadjuk számára az IP címet, amire csatlakoznia kell, illetve a hozzá tartozó Port számot, majd a megfelelő szállítási protokoll (ez esetben a SOCK_STREAM paraméter a TCP/IP-nek felel meg) kiválasztásával, már képesek is vagyunk kommunikálni az eszközzel, az összes eszközspecifikus háttérfolyamatot pedig a Socket elintézi helyettünk. Nekünk innentől kezdve, csak a send (küldés) és recv (fogadás) parancsokat kell használnunk.

5.1.2 PyQt

Másik fontos eszköz a PyQt [7] volt. Ezt a felhasználói felületek elkészítéséhez használtam, bár ennél jóval többre is képes lenne ez a könyvtár. Többek között vannak benne dobozos megoldások, melyekkel meg tudjuk valósítani akár a többszörös feladatkezelést, adatbázisműveleteket, grafikus kártyák kezeléséhez szükséges API-kat (pl.: OpenGL), hálózatkezelést, vagy akár vektorgrafikus képeket is tudunk vele alkalmazni. Az én programom esetén viszont főként a grafikus megoldásoknál alkalmaztam, hiszen felhasználói igény volt, hogy mindenféle kijelző méreten és felbontáson képes legyen a program megfelelően megjeleníteni, emellett reszponzívnak is kellett lennie. Emellett természetesen platformfüggetlen is, így nem kell külön szoftveres implementációt csinálnunk a grafikus felületek esetén egy Windows alkalmazás, vagy egy Linuxos rendszer esetén. [8]

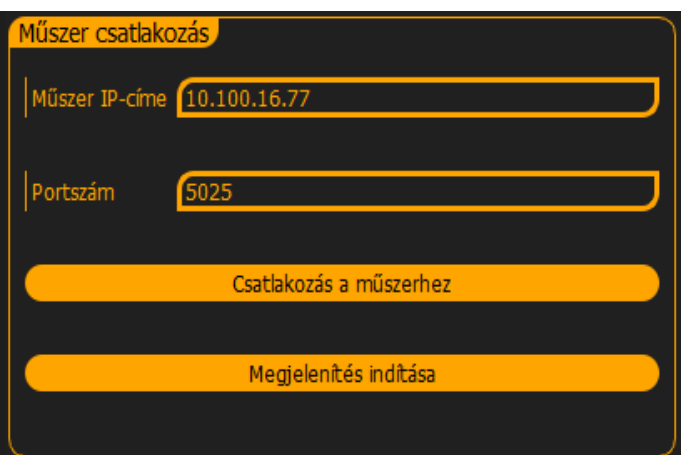
```
groupbox = QGroupBox(" Műszer csatlakozás ")
self.ipAddressLabel = QLabel(" Műszer IP-címe", self)
self.ipAddressEdit = QLineEdit('10.100.16.77', self)
self.portLabel = QLabel(" Portszaám")
self.portEdit = QLineEdit('5025', self)

self.connectButton = QPushButton("Csatlakozás a műszerhez", self)
self.plotButton = QPushButton("Megjelenítés indítása", self)

grid = QGridLayout()
grid.addWidget(self.ipAddressLabel, 0, 0)
grid.addWidget(self.ipAddressEdit, 0, 1)
grid.addWidget(self.portLabel, 1, 0)
grid.addWidget(self.portEdit, 1, 1)
grid.addWidget(self.connectButton, 2, 0, 1, 2)
grid.addWidget(self.plotButton, 3, 0, 1, 2)

groupbox.setLayout(grid)
return groupbox
```

7. ábra Műszer csatlakozás GIU megvalósítása
PyQt-val



8. ábra Műszer csatlakozás GUI

5.1.3 NumPy

A következő külsős könyvtár a *NumPy* [9] volt. Erre azért volt szükség, mert a Python a többi népszerű objektumorientált programozási nyelvvel ellentétben nem rendelkezik beépített tömbkezelő algoritmusokkal, csak listákkal. Számomra viszont elengedhetetlenek voltak a kétdimenziós tömbök, hiszen ebben tároltam futásidőben a beérkező adatokat. Le lehetett volna őket képezni az alapból beépített lista adattárolási struktúrákkal, de nem lett volna se célszerű, se optimális, mivel a beérkező adatokkal kétdimenziós függvényeket terveztünk később tárolni adatbázisban. Később innen pedig feldolgozni, majd megjeleníteni őket és ezt a hagyományos listákkal annyi kiegészítő extra művelet árán lehetett volna elérni, hogy a valós idejű megjelenítést el kellett volna engednünk, mert annyira erőforrásigényesek lettek volna a listákkal.

5.1.4 Anaconda

A következő kiterjesztés az *Anaconda* [10] nevezetű csomagkezelő rendszer volt. Ez elengedhetetlen a modern szoftverfejlesztésben, olyan, mint az *NPM* [11] a Webfejlesztésnél. Nélküle nem lennénk képesek a korábban említett kiegészítőket sem használni, ennek a kiterjesztésnek a dolga összemenedzselni a különböző verziójú és kiterjesztésű kiegészítőket, hogy egészen futtathatóak legyen Python környezetben zökkenőmentesen. Külön kiemelném az *Anaconda Navigator* [12] nevezetű funkcióját, hiszen itt grafikus módon képesek vagyunk kezelni a különböző bővítményeinket és akár egyetlen kattintással tudjuk őket frissíteni, aktiválni, törölni, vagy újakat beszerezni. Továbbá egyetlen kattintással képes az általunk előre beállított futtatási környezetet elindítani, szimulációval, fejlesztőkörnyezettel, vagy akár adatbáziskezelő szoftverrel együtt is, mindezt automatizált módon. Nekünk fejlesztőknek pedig innentől csak egyetlen dologgal kell foglalkoznunk, a programozással, az összes többi körítést az Anaconda megoldja helyettünk. A legnagyobb funkciója pedig az, hogy az összes beállításunkat felhőben tárolja, ezzel pedig megvalósulhat a platformokon átívelő valós idejű fejlesztés is.

```
PS C:\Users\Andris>
PS C:\Users\Andris>
PS C:\Users\Andris>
PS C:\Users\Andris>
PS C:\Users\Andris> Conda install numpy
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.

Retrieving notices: ...working... done
```

9. ábra Csomag telepítés Anaconda CLI parancsokkal

✓ mkl_random	○ Intel (r) mkl-powered package for sampling from common probability distributions in
✓ numba	○ Numpy aware dynamic python compiler using llvm
✓ numexpr	○ Fast numerical expression evaluator for numpy.
✓ numpy	○ Array processing for numbers, strings, records, and objects.
✓ numpy-base	○ Array processing for numbers, strings, records, and objects.
✓ numpydoc	○ Numpy's sphinx extensions
✓ pytables	○ Brings together python, hdf5 and numpy to easily handle large amounts of data.

8. ábra Csomag telepítés Anaconda Navigator grafikus felületen

5.1.5 PyQtGraph

És el is érkeztünk a legfontosabb kiegészítőhöz, ami nélkül nem készülhetett volna ez a dolgozat, ami nem más, mint a *PyQtGraph*. [13] Ez szintén egy GUI tervezést elősegítő bővítmény, csakúgy, mint a korábban említett *PyQt*, viszont ebben az esetben nem felhasználói felületeket építettem vele, hanem a beépített grafikus könyvtárainak segítségével valósítottam meg az összes grafikonos megjelenítést. Az eszköz segítségével lehetőségünk lenne, gyertya, kör, vagy akár sávdigrammok megjelenítésére, de a mi esetünkben elegendő volt a beépített vonaldiagrammot alkalmazni, mert alapból a spektrumanalizátor is a saját kijelzőjén ilyen fajta módon jeleníti meg az adatokat.

5.2 Grafikus felület

5.2.1 Elérhető funkciók

A szoftver kezelőfelületén elérhető minden egyes funkció, ami a műszer rutin felhasználásához szükség.

Többek között lehetőségünk van állítani a műszeren:

- Hálózati beállításait
 - IP címét
 - Portszámát
- Frekvencia beállításait
 - Kezdő érték
 - Vég érték
 - Center
 - Span
 - Mértékegységeket
- Trace beállításait
 - Trace 1
 - Trace 2
 - Trace 3
- Amplitúdó beállításait
 - Scale

○ Csillapítás



10. ábra Hagyományos megjelenítési mód

5.2.2 Kontrasztos mód

A szoftver több megjelenítési témával is rendelkezik, emellett szabadon is lehet állítani, hogy épp milyen megjelenítési színeket használjon a program grafikus felületén megjelenítésre, hogy mindenfajta fényviszony között megfelelően olvasható legyen. Ez a funkció az eredeti eszközön csak korlátozott keretek között lehetséges. Egyik ezek közül a kontrasztos mód, ami eredetileg extrának készült, majd a felhasználói igények alapján végül ez a fekete-narancssárga lett az alapértelmezett UI szín. Ennek a módnak az is a jelentősége, hogy minden fajta látási körülmény mellett, függetlenül a külső beható fényekre a szoftver kitűnően olvasható. Ezen felül éjszakai felhasználás esetén is kíméli a szemet, továbbá energiatakarékosabb is. [14] De természetesen a hagyományos világos Fehér-Kék megjelenítési mód továbbra is elérhető az almenükön keresztül.

5.3 Adattárolás

A műszer másodpercenként kiolvasott adatait nem lehet direkt módon a grafikon kirajzolását végző függvénynek adagolni, mert akkor vibráció lépne fel a megjelenítés folyamán. Ezért egy közbenső réteg beiktatásával kellett ezt a problémát orvosolnom. Normális esetben ezt egy adatbáziskezelő szoftver lenne, viszont ehhez infrastrukturális beruházások lennének szükségek (különálló server, szoftverlicenszek), ezért úgy oldottam meg az adattárolást, hogy egy CSV fájlban tárolom soronként a beérkező nyers stringeket feldolgozatlanul egy időbélyeggel ellátva. Majd a megjelenítés folyamán innen olvasom őket vissza. Időbélyegre

azért is van szükség, hogy visszakereshetőek legyenek utólag az adatok, és az emailküldő rendszerrel ezt a paramétert is kiküldjük a felhasználónak.

5.4 Emailküldő rendszer

Rádiófrekvenciás méréseket olyan cézzal is készíthetünk, hogy figyeljük, hogy bizonyos határértékeken belül történik-e a jelek erősségének változása. Viszont nem várható el, hogy folyamatosan üljön egy ember a szoftver mellett és figyelje a grafikon változását. Ezért valamilyen formában jeleznünk kell a felhasználónak, hogy rendellenes esemény történt/zajlik éppen. A felugró ablak nem számít teljes értékű megoldásnak, mert az életszerűtlen, hogy bekapcsolt számítógép mellett valakinek folyamatosan a szoftverünk legyen az aktív ablakon. Ezért értesítést kell küldenünk számára. Erre a legalkalmasabb és legköltséghatékonyabb megoldás az email küldése jelenti.

Határérték beállítások

Start	603	MHz
Stop	704	MHz
Határérték	20	dBm

Email cím: SZESAT@SZE.HU Hozzáadás

NJSZK@SZE.HU Törlés

GVIK@SZE.HU Törlés

TDK@SZE.HU Törlés

11. ábra Emailküldő rendszer grafikai felülete

Az alábbi felületen be tudjuk állítani, hogy az általunk vizsgált Frekvencia tartományban milyen határértéknél küldjön email-es értesítést a rendszert. Ezen felül be tudjuk állítani, hogy milyen címekre küldjük ki az üzenetet. Maximum 4 címre tudunk küldeni. Ennek felhasználói

felület tervezési korlátjai vannak, nem pedig technikaiak) A kikerkeztett emailben pedig láthatjuk az összes paramétert, ami a kiugró értékhez tartozó összes paramétert tartalmazza időbélyeggel társítva, csakúgy, mint a CSV log fájlban. Jelen pillanatban nem rendelkezek saját SMTP szerverrel, mert ez költséggel járna, ezért a munkahelyi szervert kellett igénybe vennem erre a célra.

SPA határérték túllépés

Meszaros, Andras (G/P5-5) <andras2.meszaros@audi.hu>
Címzett: "mandras2001@gmail.com" <mandras2001@gmail.com>

Határérték túllépés történt ekkor: 2022-11-17 11:26:13.174168

Érték: -139.506 dBm

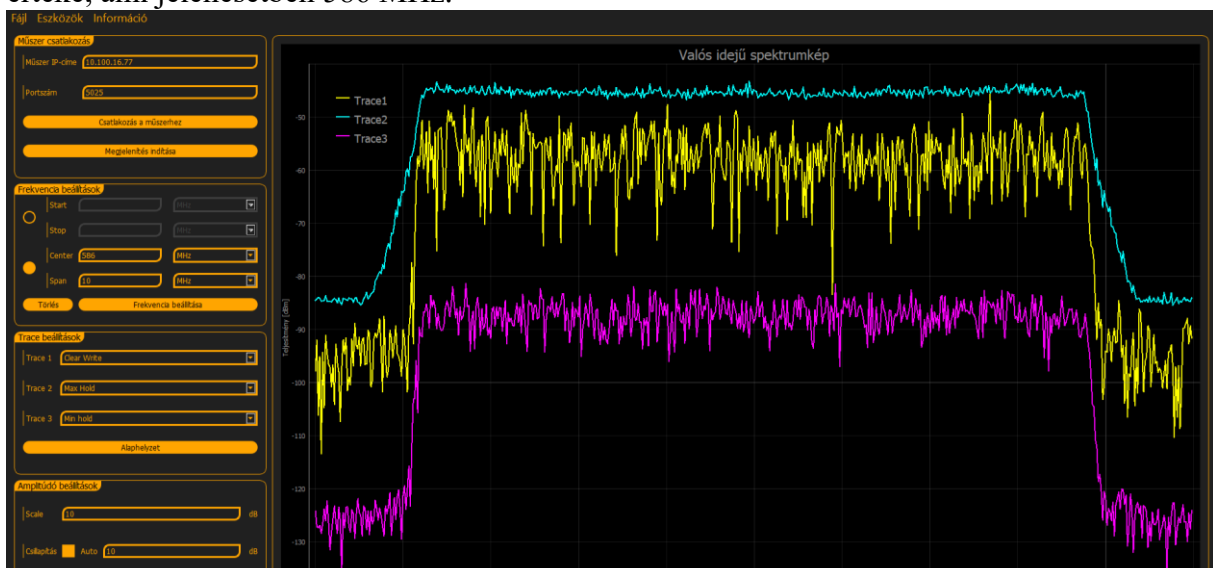
INTERNAL

12. ábra A rendszer által kiküldött email

5.5 Rutin mérési feladat az eszköz segítségével

5.5.1 DVB-T teszt

Tesztként kipróbáltuk VPN-en keresztül bejelentkezve a kollégiumból az eszközt és ráállítottuk a szoftveren keresztül a Győr/Szabadhegy — TV-URH gerincadó állomás. Ez egy 8MHz széles DVB-T2 földfelszíni műsorszóró adó. A programunkon kijelzett értékeket elmentettük, a laborban ülő kolléga pedig rögzítette a spektrumanalizátor kijelzőjén megjelenő értékeket, majd összehasonlítottuk őket és tökéletes egyezést mutattak. A fogadáshoz szükséges beállítási paramétereket pedig az *FMDX-ről* [15] nyertük, ilyen például a Center Frequency értéke, ami jelen esetben 586 MHz.



13. ábra DVB-T teszt

5.5.2 FM teszt

Nem csak DVB-T tesztek tudunk elvégezni a szoftverrel, hanem bármilyen olyan mérést meg tudunk rajta keresztül valósítani, mint amit az alap spektrumanalizátor kezelő felületén. Ilyen lehet például egy FM rádió teszt, jelen esetben a Petőfi rádió győri elérését teszteltük. [16]



14. ábra FM teszt

5.5.3 Vészüzemmód

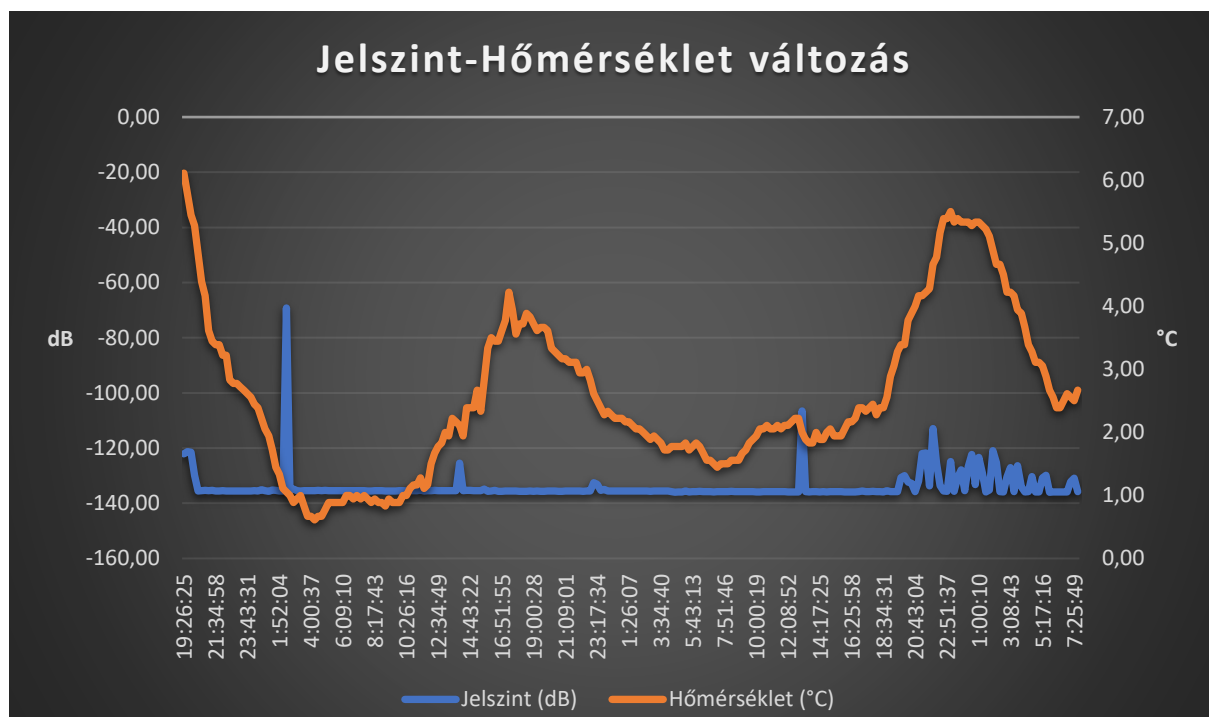
A szoftver rendelkezik egy vészüzemmóddal is. Ez azt jelenti, hogy azután, hogy ráálltunk egy frekvenciára és onnan nem kapunk vissza adatokat automatikusan átállítja a beállításokat a rendszer, hogy a solti adóállomást [17] tudjuk fogni és amennyiben nem kapunk onnan sem vissza jelet, akkor egy felugró hibaüzenetet közvetít felénk az alkalmazás, hogy a műszer körül van a hiba, nem pedig a szoftverrel. Ezt afféle referencia állomásként alkalmazzuk, mert ez az adóállomás 4:25 és 22:31 között fogható [18]

5.5.4 Stresszteszt

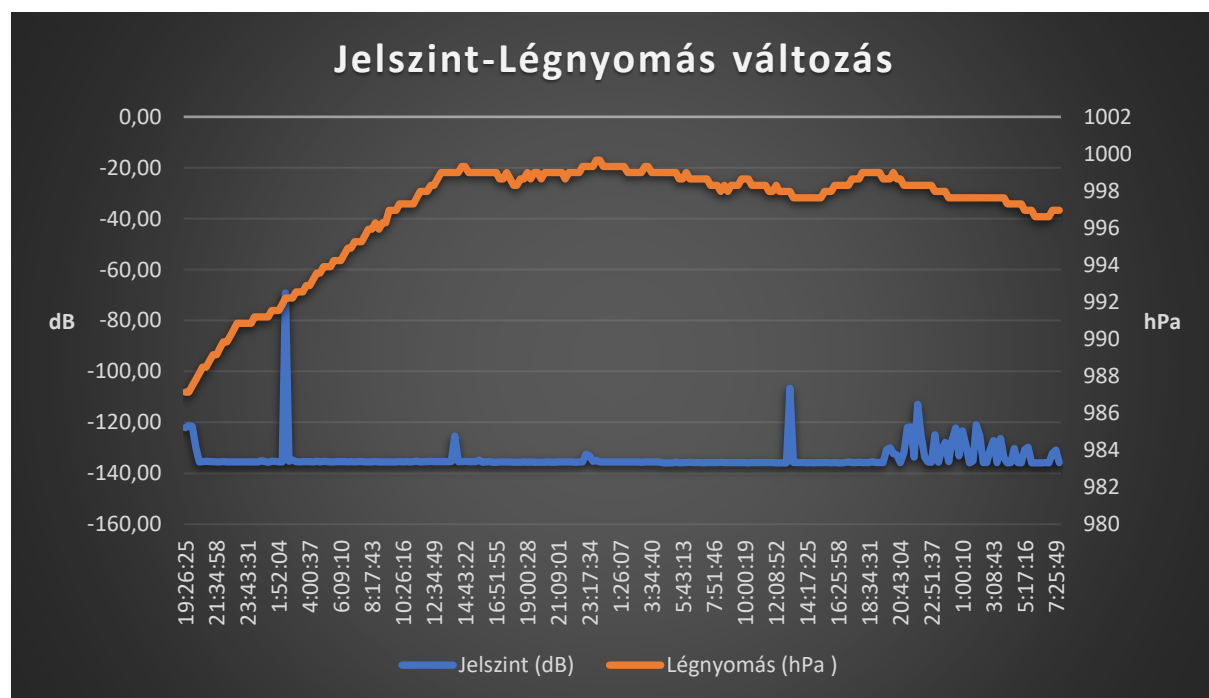
A következő tesztet egy 60 órás folyamatos mérés volt, melyet 2022.10.01 és 2022.10.04 között hajtottam végre. Az eszközt arra tervezték, hogy folyamatosan használat mellett akár éveken át is képes legyen megállás nélkül üzemelni, viszont a szoftveremet még nem teszteltem ilyen körülmények között. Ezért egy úgynevezett Hörmann forrásra álltam rá, ami nem más, mint egy amatőr rádiós jeladó. Ezen felül a mérési körülmények változására is kíváncsi voltam, mint például a hőmérsékletre, páratartalomra, szélerősségre, vagy akár a légnyomásra. Ebben nagy segítségemre volt, hogy az egyetemünk tetején helyezkedik el egy időjárás mérőállomás csupán pár méterre az antennámtól, amivel végeztem a tesztet, így igencsak pontos képet kaphattam arról, hogy milyen tényezők befolyásolják a mérést. Ennek az időjárásállomásnak a mérési adatai pedig mindenki számára nyilvánosan elérhetőek és nemcsak az élő adatokat láthatjuk, hanem 1 évre visszamenőleg lekérdezhetjük a múltbéli értékeket 5 perces lebontásban [19]. Első fontos kihívásunk a program stabilitásával kapcsolatos hiba volt, hiszen nem szabályos időközönként a program kifagyott. Miután többször sikerült reprodukálni a hibát, és lett elegendő mérési adatom, arra jutottam, hogy a hiba a spektrumanalizátor adatküldésében található. Időnként egy hibás adatcsomagot küld tovább a műszer, melyet nem volt képes kezelni a szoftverem. Ezt a hibát úgy tudtam kezelni, hogy automatikus újraindulás funkcióval láttam el a programomat erre az esetre. Ebben az esetben a program leáll, majd újraindítás után ugyanabba a CSV fájlba folytatja az írást. A folyamat körülbelül 4 másodpercet vesz igénybe, ekkora veszteséggel kell számolnunk a spektrumanalizátor hibás adatküldése esetén. A 60 órás mérés folyamán összesen 7x fordult ez elő, melyet a szoftver lekezelt, így körülbelül 30 másodpercnyi adatunk veszett kárba összesen 60 óra alatt, ami elfogadhatónak számít.

Külön kihívásnak számított a különböző forrásból származó adatok összeszinkronizálása. A spektrumanalizátorhoz készített szoftverünk másodpercenként négyszer kérdezi le az eszköz aktuális jelszint értékét, ami 60 óra alatt több mint 864000 sornyi adatot jelentett, míg az időjárás állomás ugyanennyi idő alatt csak 720 sornyi adatot küldött. Emellett figyelembe kellett venni a Microsoft Excel [20] grafikonkészítő szoftverének is a határait, ami maximum 255 sornyi adatból képes grafikont készíteni, így végül ehhez képest igazítottam a léptéket, ami 15 perces bontás lett. Ezt úgy értem el, hogy készítettem egy alapszintű C# konzolos alkalmazást, ami beolvasta a spektrumanalizátor szoftverünk által készített több mint 864000 soros CSV fájlt, majd 388-as (864000/255) szeletekre bontotta ezt

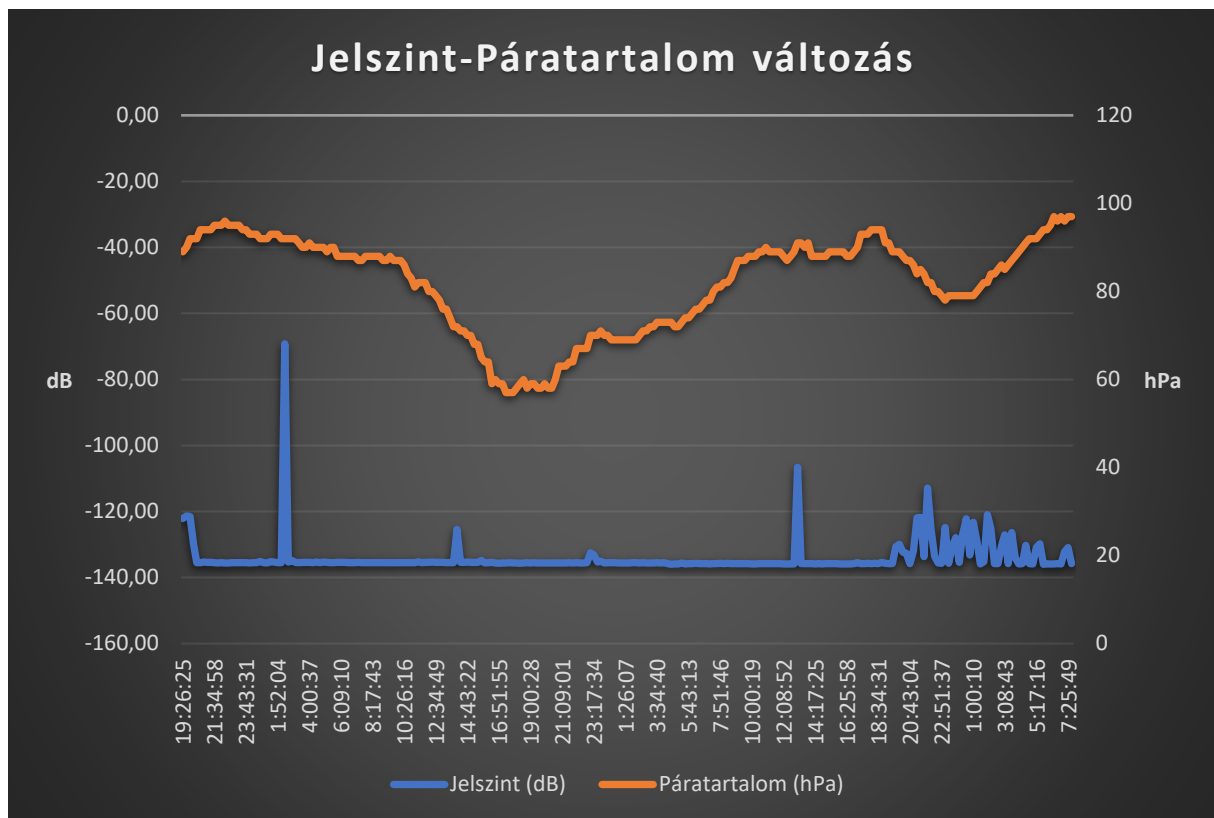
és az egyes szeletekben található értékeket átlagolta, így kaptunk végül 255 darab értéket. Ugyanezt megcsináltam az időjárás adatokkal is, majd a kapott értékeket a Microsoft Excel segítségével egy grafikonra illeszttem. A mérési értékeket pedig az alábbi grafikonokon szemléltetném:



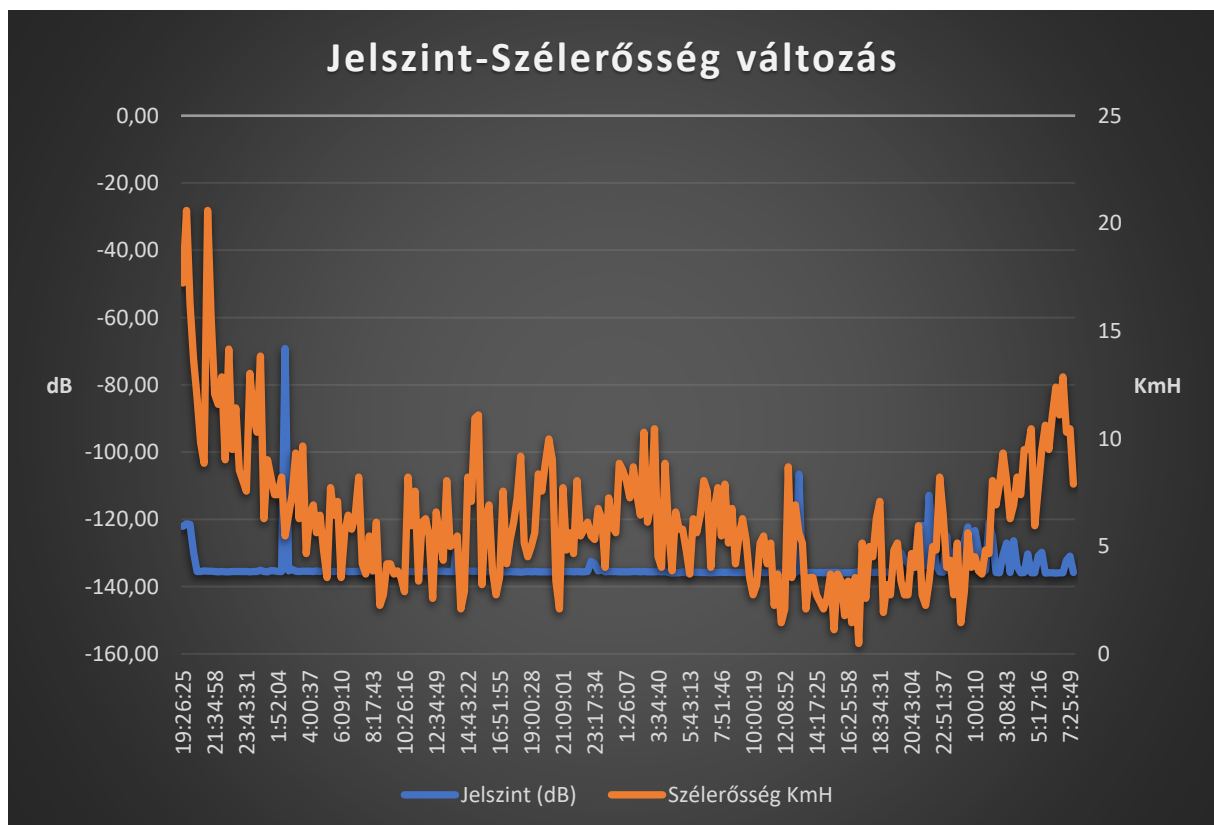
16. ábra Jelszint-Hőmérséklet változás diagram



15. ábra Jelszint-Légnyomás változás diagram



18. ábra Jelszint-Páratartalom változás diagram



17. ábra Jelszint-Szélerősség változás diagram

5.5.4.1 Stresszteszt összegzés

A fenti grafikonok alapján nem lehet egyértelműen kijelenteni, hogy kizárólag egy bizonyos környezeti feltétel változása felelős az átviteli közeg minőségének változásáért, viszont az jól látható, hogy ezek a különböző környezeti tényezők (hőmérséklet, páratartalom, légnyomás, szélereősség) együttesen már képesek változtatni az átviteli közegen. Ami számunka viszont fontosabb az magának a jelszintnek a változása. A grafikon első felén felfedezhetünk egy kiugrást, ez mérési hibának tudható be, hiszen csak 1 másodpercig volt jelen és ennyi idő alatt nem képes az átviteli közeg összetétele jelentősen változni. A környezeti adatok is ezt az állítást igazolják, hiszen nem érzékelünk semmilyen komolyabb eltérést ebben az időintervallumban egyik paraméter kapcsán sem. Ezen felül több rövid intervallumú kitérést is észleltünk (14:43-nál., illetve 12:08-nál), és ezek is az előző oknál fogva mérési hibának tudhatóak be. Jelentősebb viszont a grafikon végén 20:43-nál kezdődő hosszabb intervallumú kitérés, itt viszont már nem mérési hibáról, hanem egyértelmű tartósabb átviteli közeg minőségváltozásról beszélhetünk, amit igazolnak is a környezeti paraméterek.

6 A jövő

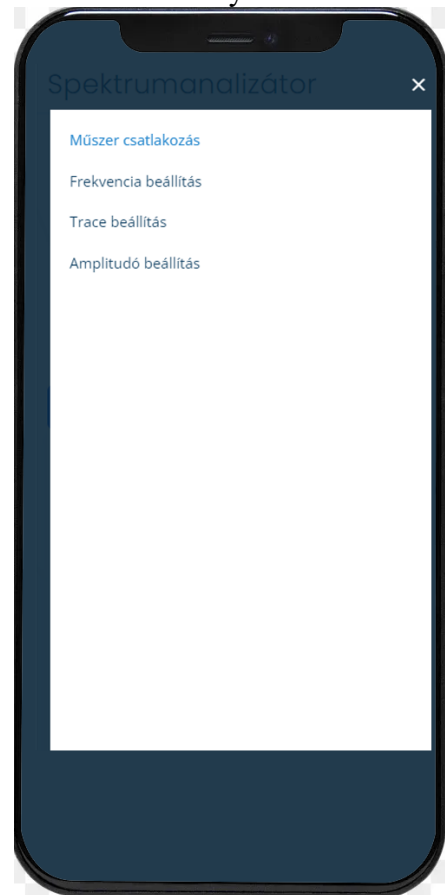
6.1 Mobilalkalmazás

A jövőben több más kényelmi funkciót is meg szeretnék valósítani. Ilyen lenne például a mobilalkalmazás elkészítése. Erre azért lenne szükség, mert sok esetben egy terepi mérésnél, mikor egyik kezünk jellemzően foglalt, mert vagy egy szerszám van benne, vagy csak simán állítunk vele valamit, a másik kezünkbe kellene fognunk a notebookot, amin fut a szoftverünk, hogy élőképet kaphassunk a mérésünkről. Ez felettébb instabil és veszélyes művelet, ezért célszerű lenne az alkalmazásunkat portolni mobil eszközökre is. Ennek a nehézségét az adja, hogy mivel itt még nagyobb a szórás az eszközök és a platformok között, ezért nem lenne lehetőség egyetlen Python fájlal elvégezni ezt a funkciót, hanem érdemes lenne a mai kor irányelvei szerint webalkalmazásban gondolkodnunk. Viszont ehhez már szükségünk lenne úgynevezett Front-End illetve Back-End oldalra is. Ezek mindegyike egy külön tudományág, de a Back-End oldal tartogat főként technikai kihívásokat számunkra. Nem a fejlesztés miatt, hanem úgynevezett infrastrukturális igények miatt. Ilyenkor már szerveres megoldásokra is szükség lenne, hogy távolról tudjuk elvégezni magának az oldalnak a ki renderelését, hogy ne a végfelhasználó eszközén kelljen ezt megtenni. Emellett kellene nekünk külön adatbáziskezelési megoldások is, szintén külön hardverekkel és megoldásokkal. A fejlesztési többletidőn túl (ilyenkor már nem játszik, hogy egyetlen Python fájlal oldjunk meg mindent) jelentős anyagi költségekkel járna a szerver oldal kiépítése és üzemeltetése, hiszen ez nem egy egyszeri költséget jelent, hanem egy folyamatosan jelentkezőt. Viszont onnantól kezdve, hogy az adatokat már felhőben tárolnánk megvalósulhatna a mérések platformok közötti átvitele.

Például egy általunk terepen végzett mérést az irodában azonnal láthatnák a kollégák számítógépen keresztül. El is készítettem ennek a webalkalmazásnak a látványtervét is.



19. ábra Webalkalmazás
kezdőoldala mobilon



20. ábra Webalkalmazás menüje
mobilon

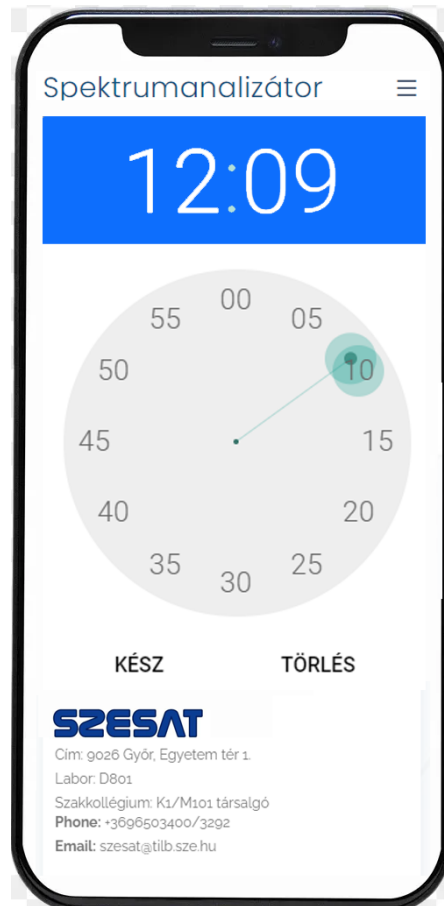
6.2 Távoli Ki-Be kapcsolás

Másik fontos funkció a távoli kikapcsolás lehetősége lenne. Erre azért lenne szükség, mert a szabványos SCPI parancsok között ilyen nem létezik, viszont más műszereknél, ahol lehetséges a távvezérlés alap funkciónak számít ez. Visszafelé is érdemes lenne megcsinálni az eszközállapot kapcsolást, viszont a spektrumanalizátor gyári dokumentációja alapján nincs lehetőségünk Wake-On-LAN funkcióra (ez a protokoll felelne azért, hogy az eszköz kikapcsolt állapotban a LAN portjára beérkező speciális adatcsomag hatására bekapcsolja az eszközt.). Ezért mindenképpen valami hardveres alapú megoldásra lenne szükségünk ezen funkció megvalósításához, mert jelen állapotában egy hétvégi mérés úgy néz ki, hogy az eszközt bekapcsolva hagyjuk egész időtartamra, majd a legközelebb, mikor az egyetem nyitvatartása megengedi számunkra kikapcsoljuk az eszközt. Ez viszont nem feltétlenül egészséges a műszer szempontjából és negatívan tudja befolyásolni az eszköz élettartamát a folyamatos üzemeltetés. Emellett nem is gazdaságos, hogy a konkrét mérési időtartamokon kívül is járattunk kell az

eszközt, mert nincs lehetőségünk KI-BE kapcsolni az távolról azokra az időtartamokra, mikor ténylegesen szükségünk lenne rá.



22. ábra. Tervezett Webalkalmazás aktív eszközigő beállító menüje mobil nézetén



21. ábra Tervezett Webalkalmazás aktív eszközigő beállító menüjének dátum választó gombra előugró menüje mobil nézetén

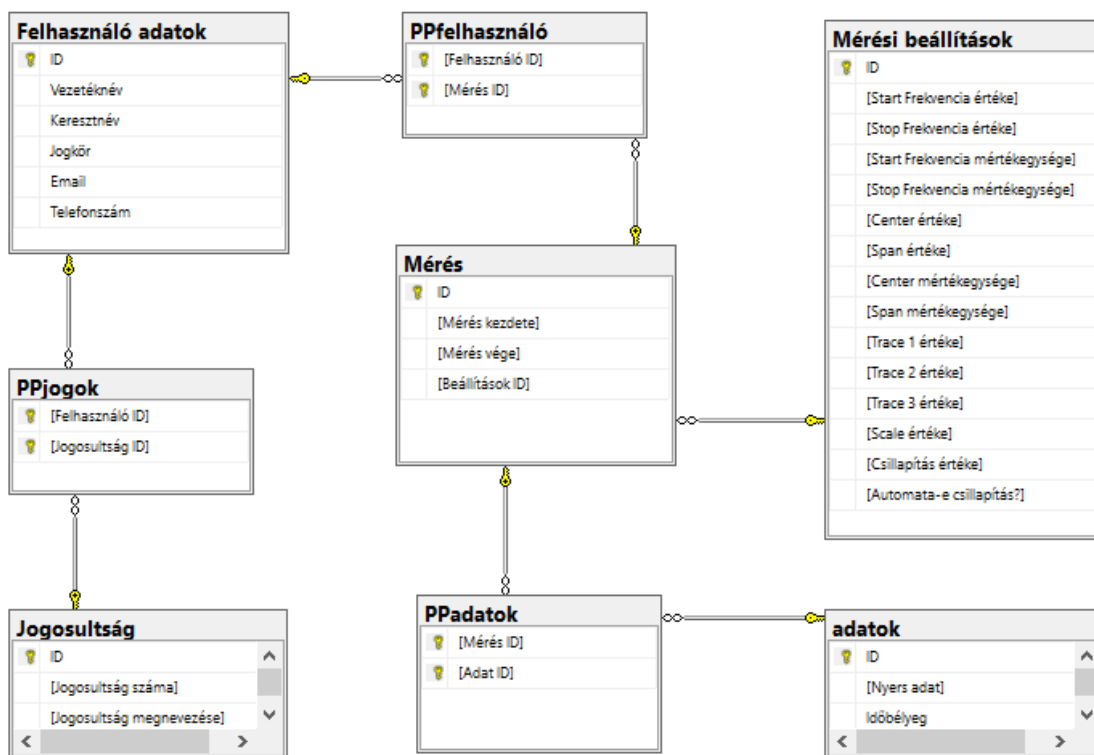
6.3 Emailküldő rendszer

Az emailküldő rendszer folyamán az alapokat már leraktam, képes vagyok kiugró értékeknél egyszerű szöveges üzeneteket küldeni. Viszont a jövőben mindenképpen szeretném a beállítások lehetőségeit bővíteni. Továbbá be szeretnék vezetni egy felhasználó autentikáció alapuló rendszert, ahol lennének különböző jogosultsági szintek, és ezekhez a szerepkörökhöz szeretnék automatikus email riportokat társítani. Pl.: Vezető jogosultságú felhasználók minden hétnek a végén kapjanak egy összesítő riportot. Itt egy PDF-et terveznék küldeni, amihez természetesen kellene egy PDF generáló modul is, de miután meghatároznánk a lehetséges eseteket, hogy mikről szeretnénk ilyen értesítést küldeni, utána már lehetne automatizálni a működést. Ezen felül szeretnék saját SMTP szervert is igénybe venni, hogy függetleníteni tudjam magam a munkáltatóm rendszerétől, ami után már egyedi küldő fél mezőt is tudnék alkalmazni. (például: hiba@spa.sze.hu) Továbbá lehetőségem nyílna egyedi tárgy mező

alkalmazására is, ahova akár már a konkrét jelszint értéket is be tudnám írni a programmal, ami azért lenne jelentős, mert amennyiben utólag nézi valaki vissza a posta fiókját, és nagy számú értesítő emailt kap, akkor már azelőtt tudja őket szortírozni, mielőtt megnyitná őket, ezzel időt és energiát spórolnánk meg a felhasználóknak.

6.4 Adatbázis

Mint említettem a Webalkamazásnál már nem lenne elegendő az adatokat CSV fájl-ban tárolni, hanem már rendes adatbáziskezelő szoftver lenne szükséges. El is készítettem az adatbázis részét MS SQL-ben. [21]



23. ábra Az adatbázis felépítése

Későbbiekben pedig a back-endet .NET-ben fogom írni, és azon belül is Entity Frameworköt fogok használni, amihez amúgy is jól passzol az MS SQL-ben elkészített adatbázisom.

6.5 Lekérdezések

Onnantól kezdve, hogy rendelkezésünkre áll majd az adatbázis, nem csak feltölteni tudjuk oda az adatokat, hanem lekérdezni is. Létrehoztam a Webalkalmazáson belül egy felületet,



24. ábra Mérés eredmények megjelenítése Webalkalmazás segítségével Mobilon

ahol a Be-Ki kapcsoláshoz menühöz hasonlóan meg tudjuk adni a vizsgálandó időintervallumot, majd egy csúszka segítségével tudunk a lekérdezett sávon belül lépkedni. A csúszka mozgatásának hatására pedig változik valós időben a grafikon képe, annak megfelelően, hogy az éppen vizsgált időpontban, ahol a csúszka választója éppen áll milyen értékek vannak az adatbázisban eltárolva.

6.6 Kompatibilitás

A jövőben nem csak kizárólag az Agilent E4440A típusú eszközzel szeretném, hogy a program kommunikáljon, hanem lehetőség szerint az összes SCPI szabványt használó rádiófrekvenciális spektrumanalizátorral, amik a piacon fellelhetők függetlenül márkától és évjáratától.

6.7 Üzemeltetés, támogatás

Mivel fontos azzal az esettel számolnunk, hogy nem biztos, hogy a jövőben én fogom továbbfejleszteni ezt az alkalmazást, ezért olyan szemlélet módban kellett azt végig fejlesztenem, hogy minél egyszerűbb legyen megérteni a programkódot. Így van ez az alkalmazás üzemeltetésével is, ezért létrehoztam egy 40 diás felhasználói kézikönyvet prezentáció formájában, ahol lépésről lépésre bemutatom az összes funkcióját a programnak



25. ábra Felhasználói kézikönyv

közérthetően. Ehhez különböző jelölésekkel, főként nyilakkal és ikonokkal mutatom meg, hogy adott funkció eléréséhez mit és milyen sorrendben kell kattintani/ gépelni. A cél az volt, hogy ne csak a villamosmérnök hallgatók tudják kezelni ezt a szoftvert, hanem olyanok is, akiknek nincsen villamossági előképzettsége. Sőt a tesztek alapján, olyan hallgató is tudta kezelni a programot készség szinten a prezentáció után, aki amúgy nem is műszaki vonalon tanul, és nem tudta ezt megelőzően, hogy mi is az a spektrumanalizátor.

7 Kapott eredmények, tapasztalatok

Az előző fejezetekben részletezett funkciók elkészítésével úgy gondolom, hogy sikerült elérnem az eredetileg kitűzött céljaimat, amiket a bevezetésben is részleteztem. Ezt bizonyítja, hogy az 5.5-ös fejezetben lévő tesztek sikerrel lefutottak, majd a kapott mérési eredmények a szoftveren megegyeztek az eszközön szereplő értékekkel. Ezen felül az eredeti célkitűzést, miszerint csak vezérelni lehessen az eszközt sikerült is túlteljesítenem. Ez a felhasználói visszajelzések tükrözik legjobban, akik túlnyomó részben azt nyilatkozták, hogy a szoftver a felhasználói kézikönyvem elolvasása nélkül is, nemhogy egyértelműen használható, hanem még mikor a laborban tartózkodnak is szívesebben vezérlik a szoftveren keresztül az eszközt, mint magán a mérőműszer kezelőfelületén.

Összességében úgy gondolom, hogy az elkészített munkám hiánypótló és az ezen a területen dolgozó emberek dolgát megkönnyíti az adatok kereshetőségének egyszerűsítésével, a vizualizáció gyárinál magasabb szintre emelésével, és az extra kényelmi funkciók sikeres bevezetésével. A szoftver sikerét pedig az bizonyítja legjobban, hogy egyre több és több új felhasználói igény születik további funkciókra, mert látják a felhasználók a benne rejlő további lehetőségeket és a program jövőjét.

8 Ábrajegyzék

1. ábra Kihelyezett NMHH mérőállomás [26].....	8
2. ábra Országos Spektrum és Interferencia Monitoring (SIMON) mérőállomás rendszer [22]	9
3. ábra Agilent E4440A típusú rádiófrekvenciális spektrumanalizátor [1]	10
4. ábra Távoli elérés megvalósítása VPN-el és nélküle [25]	12
5. ábra Csatlakozás az eszközhöz Socket segítségével	14
6. ábra Műszercsatlakozás GUI	15
7. ábra Műszer csatlakozás GIU megvalósítása PyQt-val	15
8. ábra Csomag telepítés Anaconda Navigator grafikus felületen	16
9. ábra Csomag telepítés Anaconda CLI parancsokkal	16
10. ábra Hagyományos megjelenítési mód	18
11. ábra Emailküldő rendszer grafikai felülete	19
12. ábra A rendszer által kiküldött email	20
13. ábra DVB-T teszt	21
14. ábra FM teszt	22
15. ábra Jelszint-Légnyomás változás diagram	24
16. ábra Jelszint-Hőmérséklet változás diagram.....	24
17. ábra Jelszint-Szélerősség változás diagram	25
18. ábra Jelszint-Páratartalom változás diagram.....	25
19. ábra Webalkalmazás kezdőoldala mobilon	28
20. ábra Webalkalmazás menüje mobilon	28
21. ábra Tervezett Webalkalmazás aktív eszközidő beállító menüjének dátum választó gombra előugró menüje mobil nézetén	29
22. ábra. Tervezett Webalkalmazás aktív eszközidő beállító menüje mobil nézetén.....	29
23. ábra Az adatbázis felépítése.....	30
24. ábra Mérés eredmények megjelenítése Webalkalmazás segítségével Mobilon	31
25. ábra Felhasználói kézikönyv.....	32

9 Irodalomjegyzék

- [1] Avalontest, „Avalontest,” 2022. [Online]. Available: <https://avalontest.com/keysightagilent-e4440a-spectrum-analyzer>. [Hozzáférés dátuma: 4 November 2022].
- [2] K. Technologies, „Keysight,” 2017. [Online]. Available: <https://www.keysight.com/us/en/assets/7018-02149/technical-overviews/5990-3990.pdf>. [Hozzáférés dátuma: 4 november 2022].
- [3] „Wireshark,” 2022. [Online]. Available: <https://www.wireshark.org/>. [Hozzáférés dátuma: 4. November 2022].
- [4] Wikipédia, „Virtual private network,” 4. November 2022. [Online]. Available: https://en.wikipedia.org/wiki/Virtual_private_network. [Hozzáférés dátuma: 4. November 2022].
- [5] C. S. Inc., „Cisco AnyConnect,” 2014. [Online]. Available: <https://www.cisco.com/c/en/us/support/security/anyconnect-secure-mobility-client-v4-x/model.html#DataSheetsandProductInformation>. [Hozzáférés dátuma: 10 10 2022].
- [6] Wikipédia, „TCP/IP,” 3. Január 2022. [Online]. Available: <https://hu.wikipedia.org/wiki/TCP/IP>. [Hozzáférés dátuma: 4. November 2022].
- [7] R. Computing, „What is PyQt?,” 2022. [Online]. Available: <https://riverbankcomputing.com/software/pyqt/>. [Hozzáférés dátuma: 4. November 2022].
- [8] M. FitzPatrick, „PyQt5 Widgets,” 16. Augusztus 2022. [Online]. Available: <https://www.pythonguis.com/tutorials/pyqt-basic-widgets/>. [Hozzáférés dátuma: 4. November 2022].

-
- [9] „Array programming with NumPy,” 2020. [Online]. Available: <https://doi.org/10.1038/s41586-020-2649-2>. [Hozzáférés dátuma: 4 11 2022].
- [10] A. S. Distribution, „Anaconda Documentation,” Anaconda Inc., 2020. [Online]. Available: <https://docs.anaconda.com/>. [Hozzáférés dátuma: 4. November 2022].
- [11] L. Byte, „Mi az NPM? (Csomópont csomagkezelő a programozásban,” 16. Március 2022. [Online]. Available: <https://vidabytes.com/hu/que-es-npm/>. [Hozzáférés dátuma: 4. November 2022].
- [12] Anaconda, „Anaconda Navigator,” 2022. [Online]. Available: <https://docs.anaconda.com/navigator/index.html>. [Hozzáférés dátuma: 4. November 2022].
- [13] PyQtGraph, „PyQtGraph,” 2021. [Online]. Available: <https://www.pyqtgraph.org/>. [Hozzáférés dátuma: 4. November 2022].
- [14] N. Zoltán, „Tényleg jó a sötét mód?,” 6. Július 2019. [Online]. Available: <https://pcworld.hu/pcwpro/sotet-mod-dark-mode-tippek-265485.html>. [Hozzáférés dátuma: 4. November 2022].
- [15] FMDX, „Győr/Szabadhegy — TV-URH gerincadó állomás,” 2022. [Online]. Available: <https://www.fmdx.hu/transmitters-hng-szabadhegy.htm>. [Hozzáférés dátuma: 4. November 2022].
- [16] MyOnlineRadio, „Petőfi Rádió frekvencia - Petőfi Rádió frekvenciák,” 2022. [Online]. Available: <https://myonlineradio.hu/mr2-petofi-radio/frekvencia>. [Hozzáférés dátuma: 4. November 2022].
- [17] FMDX, „Solt — középhullámú (KH) adóállomás,” 2022. [Online]. Available: <https://www.fmdx.hu/transmitters-hng-solt-kh.htm>. [Hozzáférés dátuma: 4. November 2022].
- [18] V. Vámos, J. Csősz és É. Balla, „2 MW-os rekonstrukció Solton,” Június 2019. [Online]. Available: <http://www.radiovilag.hu/images/184-185o19.pdf>. [Hozzáférés dátuma: 4. November 2022].

-
- [19 Wunderground, „Győr SZESAT - IGYR41,” 2022. [Online]. Available:
] <https://www.wunderground.com/dashboard/pws/IGYR41>. [Hozzáférés dátuma: 21
11 2022].
- [20 M. Corporation, „Microsoft Excel,” Microsoft Corporation, 2022. [Online].
] Available: <https://office.microsoft.com/excel>. [Hozzáférés dátuma: 11 10 2022].
- [21 Microsoft, „SQL Server technical documentation,” 2022. [Online]. Available:
] <https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16>.
[Hozzáférés dátuma: 4. November 2022].
- [22 NMHH, „NMHH,” 12 május 2020. [Online]. Available:
] https://nmhh.hu/cikk/212302/Az_NMHH_orzi_a_hetkoznapok_nyugalmat.
[Hozzáférés dátuma: 4 11 2022].
- [23 S. László, MŰSZERISMERTETŐ SEGÉDLET a Laboritórium 2. c. tárgy
] méréseihez, Első szerk., Budapest, Pest: BME VIK , 2008.
- [24 Cameradepo, „IP hálózatok elmélete, egyszerűen,” 2018. [Online]. Available:
] <https://web.archive.org/web/20181010031835/http://cameradepo.hu/termektamogatas/ip-halozatok-elmelete-egyszeruen#ddns>. [Hozzáférés dátuma: 4 November 2022].
- [25 Hostinger, „What Is a VPN: How Does It Work and Should You Use It?,” 31.
] Augusztus 2022. [Online]. Available: <https://www.hostinger.com/tutorials/what-is-vpn>. [Hozzáférés dátuma: 4. November 2022].
- [26 NMHH, „Spektrum Monitoring,” [Online]. Available: <http://emirpub-prod.nmhh.hu/pubrendszer-web/spektrummonitoring/spektrumMonitoringLocationSelect.jhtml>. [Hozzáférés dátuma: 4. November 2022].
- [27 N. Jennings, „Socket Programming in Python,” 21. Február 2022. [Online].
] Available: <https://realpython.com/python-sockets/>. [Hozzáférés dátuma: 4. November 2022].

-
- [28 G. & D. J. F. L. Van Rossum, Python reference manual, Centrum voor
] Wiskunde en Informatica Amsterdam., 1995.
- [29 Wikipédia, „NumPy,” 18. November 2018. [Online]. Available:
] <https://hu.wikipedia.org/wiki/NumPy>. [Hozzáférés dátuma: 4. November 2022].
- [30 JavatPoint, „Python sys module,” 2022. [Online]. Available:
] <https://www.javatpoint.com/python-sys-module>. [Hozzáférés dátuma: 4. November 2022].
- [31 Education-wiki, „Mi a TCP / IP?,” 2022. [Online]. Available:
] <https://hu.education-wiki.com/3737338-what-is-tcpip>. [Hozzáférés dátuma: 4. November 2022].
- [32 Wikipédia, „Standard Commands for Programmable Instruments,” 30,
] Október 2022. [Online]. Available:
[https://en.wikipedia.org/wiki/Standard_Commands_for_Programmable_Instrument](https://en.wikipedia.org/wiki/Standard_Commands_for_Programmable_Instruments)
s. [Hozzáférés dátuma: 4. November 2022].
- [33 Wikipédia, „Standard parancsok a programozható eszközökhöz,” 2022.
] [Online]. Available:
https://hu.frwiki.wiki/wiki/Standard_Commands_for_Programmable_Instruments.
[Hozzáférés dátuma: 4. November 2022].