

GNSS alapú navigációs rendszer RTK bázisállomása

2024

Szerző

Eredits Dániel

Témavezetők

Krecht Rudolf, Boros Norbert

Tartalomjegyzék

1. Bevezetés.....	4
1.1 Feladat definiálása.....	4
1.2 Felállított követelményrendszer	4
1.2.1 Funkcionális követelmények.....	5
1.2.2 Nem funkcionális követelmények.....	6
2. Használt technológiák	7
2.1 Global Navigation Satellite System (GNSS)	7
2.2 Real-Time Kinematic (RTK).....	8
2.3 Mobil RTK bázisállomás hardveres felépítése.....	10
2.3.1 GNSS vevő.....	10
2.3.2 Egykártyás számítógép.....	11
2.3.3 Akkumulátor.....	12
2.3.4 Akkumulátor állapot mérése.....	13
2.3.4.1 Hardveres felépítés	13
2.3.4.2 Szoftveres felépítés	14
2.4 Mobil RTK bázisállomás szoftveres felépítése	15
2.4.1 NTRIP szerver indítása.....	15
2.4.2 Virtual Private Network (VPN)-re való csatlakozás	15
2.4.3 LED visszajelzések	16
2.4.4 Shell script használata	16
3. Mérési eredmények	17
3.1 Mérés célja	17
3.2 Mérés menete	17
3.3 Mérési eredmények	17
3.3.1 Általános GNSS vevő.....	17

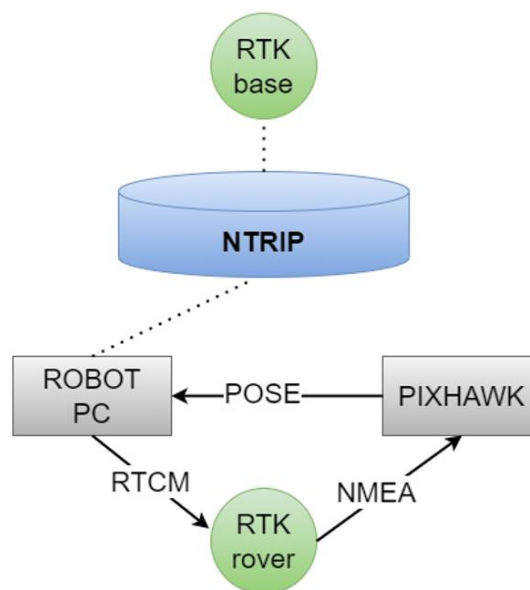
3.3.2 RTK Float és RTK Fixed.....	18
3.3.3 Antenna és antenna nélküli survey-in.....	20
3.3.4 Akkumulátorral végzett mérések.....	20
3.4 Levont következtetések	22
4. Összefoglalás.....	23
5. Továbbfejlesztési célok	24
Ábrajegyzék	25
Irodalomjegyzék.....	26

1. Bevezetés

1.1 Feladat definiálása

A dolgozatom témájaként fejlesztett Real-Time Kinematic (RTK) bázisállomás egy olyan navigációs rendszer támogatásához készült, amely segíti a zalaegerszegi ZalaZone Járműipari Tesztpálya kiszolgálását forgalomtechnikai terelőbóják automatizált kihelyezésével. A navigációs rendszer kettő Global Navigation Satellite System (GNSS) vevőből áll, melyet egy RTK rendszer támogat a centiméter pontos helyzetmeghatározás elérése érdekében. A navigációs adatokat egy Cube Orange autopilot olvassa ki és továbbítja a robotplatform Robot Operating System 2 (ROS2) rendszerének, az aktuális orientációval együtt.

A dolgozatom a fentebb leírt navigációs rendszert támogató saját fejlesztésű RTK bázisállomás fejlesztését mutatja be, illetve a vele elvégzett tesztek eredményét és ezeknek az eredményeknek az elemzését.



1. ábra A navigációs rendszer blokkvázlata

1.2 Felállított követelményrendszer

A saját fejlesztésű RTK bázisállomás problémamentes fejlesztése érdekében felállítottam egy funkcionális és egy nem funkcionális követelményekből álló listát. Ezeket a lépéseket követve, minden egyes kritériumot sorban kielégítve alakítottam ki a navigációs rendszer számára megfelelő bázisállomást.

A követelményrendszer kiterjed minden, egy RTK bázisállomás számára fontos funkció megvalósításához szükséges probléma megoldására.

Véleményem és tapasztalatom szerint a tervezési folyamat során egy jól átgondolt és megfelelően kialakított követelményrendszer táblázat nagy mértékben tudja csökkenteni a fejlesztési időt és meg tud előzni számos, a fejlesztés során felmerülő problémát.

1. Táblázat Az RTK bázisállomás felállított követelményrendszere

Követelmény	Tervezett megoldás
Saját szoftver futtatása	Egykártyás számítógép
Saját NTRIP szerver	Pygnssutils python könyvtár használata
Mobilitás	Akkumulátor felszerelése
Elérhetőség	Virtual Private Network (VPN)-re való csatlakozás
Külső visszajelzések	LED visszajelzők használata
Gyorsabb, pontosabb rendelkezésre állás	Külső antenna csatlakoztatása
Hatótáv, sebesség növelése	5G antenna felszerelése
Akkumulátor állapotának olvasása	Feszültség és árammérő szenzor használata

1.2.1 Funkcionális követelmények

Egy mobil RTK bázisállomás létrehozásánál a leglényegesebb szempont a megfelelő mobilitás, illetve a kellő nagyságú üzemidő biztosítása, a robotplatform zökkenőmentes működése érdekében egy saját, a projekt igényeit kiszolgáló szoftver fejlesztésével, illetve hardver rendszer kialakításával.

A mobilitás elérése érdekében szükséges a bázisállomást felszerelni egy külső, tölthető akkumulátorral, hogy az elindításhoz ne kelljen semmilyen külső áramforrás, szó szerint bárhol elérhető legyen. E követelmény kielégítése érdekében fontos egy olyan akkumulátor használata, amely tudja biztosítani a készülék számára a megfelelő üzemidőt.

A saját fejlesztésű szoftver futtatásához egy lehetséges megoldás egy egykártyás számítógép (pl.: Raspberry PI), amely kapcsolatban áll a ZED-F9P chippel felszerelt GNSS vevővel és rendelkezik a megfelelő hardverspecifikációval, a szoftver zavartalan futtatása érdekében.

A bázisállomás minél pontosabb és gyorsabb helyzetmeghatározása érdekében, a GNSS vevőhöz csatlakoztatva lett egy Antenna, melynek hatékonyságát tesztekkel is bizonyítom dolgozatom egy későbbi fejezetében.

Végül a fejlesztés során is, illetve a mindennapi használatban rendkívül fontos a külső visszajelzés, hogy pontosan tudni lehessen, a készülék rendelkezésre állást, illetve akkumulátorállapotát. Erre a megoldás több LED visszajelző felszerelése, ami jelzi többek közt a bázis elérhetőségét, illetve az akkumulátor töltöttségi állapotát. Az ezekre írt saját fejlesztésű programokat dolgozatom egy későbbi fejezetében mutatom be.

1.2.2 Nem funkcionális követelmények

Az RTK rendszer működéséhez szükséges egy NTRIP szerver, amin keresztül a Radio Technical Commission for Maritime Services 3 (RTCM3) adatok továbbításra kerülnek. A navigációs rendszer saját NTRIP szerverrel rendelkezik, amely induláskor automatikusan elindul. Ezt a pygnssutils, nyílt forráskódú szoftver segítségével teszi meg.

Az RTCM3 adatok továbbításához az RTK bázisállomásnak és az RTK rovernek egy hálózaton kell lenniük. Erre a problémára jelent megoldást a VPN kapcsolat, mivel ennek segítségével nem szükséges egy komplett hálózat kiépítése akár egy tesztpályára, hanem a VPN hálózatra csatlakozva gördülékenyen folyhat az RTCM3 adatok továbbítása.

A LED visszajelzéshez egy külön, szintén induláskor induló szoftver került fejlesztésre, ezt részletesebben a dolgozatom egy későbbi részében fejtem ki.

Az 5G hálózatok sokkal nagyobb adatátviteli sebességet tesznek lehetővé, mint az előző generációs rendszerek, például az 4G. Az 5G jelentős előnye az alacsony késleltetés vagy latency. Ez a jellemző kulcsfontosságú a valós idejű alkalmazásokhoz, mint például az autonóm járművek vezérlése. Az alacsony latency lehetővé teszi, hogy az eszközök gyorsan és megbízhatóan kommunikáljanak egymással, ami kritikus fontosságú lehet olyan területeken, mint az ipari automatizálás.

2. Használt technológiák

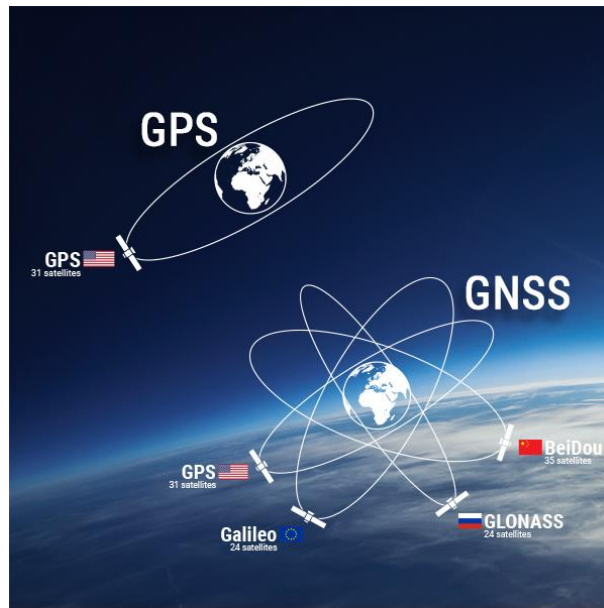
2.1 Global Navigation Satellite System (GNSS)

A Global Navigation Satellite System (GNSS) az egyik legfontosabb navigációs rendszer, amelyet a Föld körül elhelyezett műholdak és földi állomások alkotnak. Célja a globális helymeghatározás és idősinkronizáció biztosítása. Ezek a rendszerek kritikus fontosságúak számos alkalmazásban, például a közlekedésben, a katasztrófa-elhárításban, a hadviselésben és a helymeghatározáson alapuló szolgáltatásokban.

A GNSS működési elve a műholdak és a földi vevők közötti összeköttetésen alapul. A műholdak, amelyek precíz idősinkronizációt tartanak fenn, folyamatosan rádiójeleket sugároznak a földi vevőknek. Ezek a jelek tartalmazzák az időpontot, amikor a jel sugározva lett, valamint a műhold aktuális pozícióját. A földi vevők általában olyan állomások, amelyeket a GNSS-t használni kívánó felhasználók telepítenek, és ezek folyamatosan kommunikálnak a keringő műholdakkal, hogy pontos helyzetmeghatározást érjenek el.

A GNSS több különböző navigációs rendszer műholdjait használja, mint például a Global Positioning System (GPS), a GLONASS, a BeiDou és a Galileo. Ezek a rendszerek összehangoltan működnek annak érdekében, hogy még pontosabb helyzetmeghatározást nyújtsanak a felhasználóknak. A GNSS által nyújtott pontosság általában 1-2 méteres tartományban van, ami lehetővé teszi a széles körű alkalmazásokban történő hatékony és megbízható helymeghatározást.

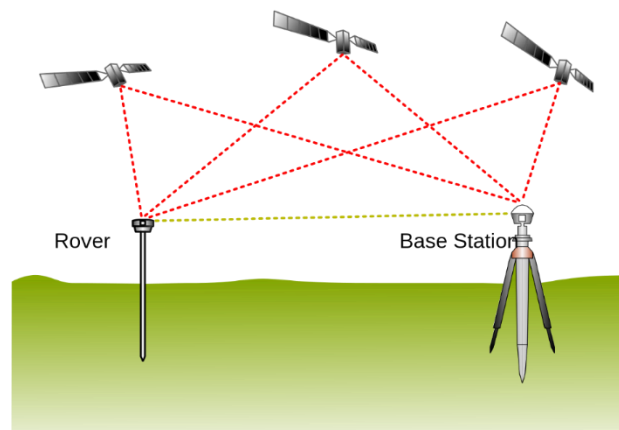
A GNSS rendszerek és vevők fontos szerepet játszanak a modern technológia számos területén, és folyamatos fejlesztéseknek köszönhetően egyre nagyobb pontossággal és megbízhatósággal képesek működni, ami hozzájárul az innovációhoz és a társadalmi előrehaladáshoz. [1] [2]



2. ábra GNSS navigációs rendszer

2.2 Real-Time Kinematic (RTK)

Az Real-Time Kinematic (RTK) egy olyan fejlett technika, amelyet a GNSS rendszerek általános hibáinak kiküszöbölésére fejlesztettek ki. Ezek a hibák lehetnek például a műholdak jeleinek időbeli torzulásai, vagy a troposzférában jelentkező pontatlanságok. Az RTK rendszer lényege, hogy a GNSS vevő (rover) mellett egy bázisállomás (base station) is használatban van a helyzetmeghatározás közben. A bázisállomás a Radio Technical Commission for Maritime Services version 3 (RTCM3) korrekciós adatokat továbbítja az Networked Transport of RTCM via Internet Protocol (NTRIP) szerveren keresztül a rover felé. Ezáltal a rover nem csak a műholdak jeleit használja fel, hanem a bázisállomástól érkező korrekciós adatokat is figyelembe veszi a helyzetének meghatározásakor. [3]



3. ábra RTK rendszer működési elve

Az RTK módszerrel a rover egy úgynevezett kétvonalas mérés technikát alkalmaz, amely segítségével a bázisállomástól érkező korrekciós adatokat használva centiméteres pontossággal határozhatja meg a helyzetét. Az RTK rendszer kiépítése és működtetése során a rover pontosságának státusza két állapotot vehet fel: RTK Float és RTK Fixed.

Az RTK Float állapotban az RTK algoritmus valamilyen okból nem tudja pontosan meghatározni a vevő pontosságát, ezért a GNSS vevő valamivel rosszabb pontosságot ér el, mint a centiméteres helyzetmeghatározás esetén. Ezek az okok lehetnek például természeti körülmények, épületek árnyékolása vagy a hálózattal történő jelvesztés. Ezen problémákra gyakran megoldást nyújt egy külső antenna használata.

Az RTK Fixed állapotban viszont az előzőleg említett problémák nem merülnek fel, vagy legalábbis a hatásuk minimális, elhanyagolható. Ebben az esetben az RTK rendszer képes centiméteres pontossággal meghatározni a rover pozícióját mind statikus, mind mozgó állapotban. Ehhez elengedhetetlen az RTK bázisállomás és az RTK rover megfelelő konfigurációja, valamint az NTRIP szerver megfelelő működése. [3]

Az általam fejlesztett navigációs rendszer egy speciális antenna segítségével javítja mind a rover, mind a bázisállomás megbízhatóságát és hatékonyságát. Ennek köszönhetően az RTK rendszer általános pontossága és megbízhatósága tovább növekszik, ami különösen fontos a precíziós helymeghatározási alkalmazásokban. A fejezetben kifejtett állításaimat és módszereimet részletesebben bemutatom és alátámasztom a mérési eredmények fejezetben, amelyek az általam végzett tesztek alapján kerülnek vizualizálásra és értelmezésre.



4. ábra DA910 multi-band GNSS antenna

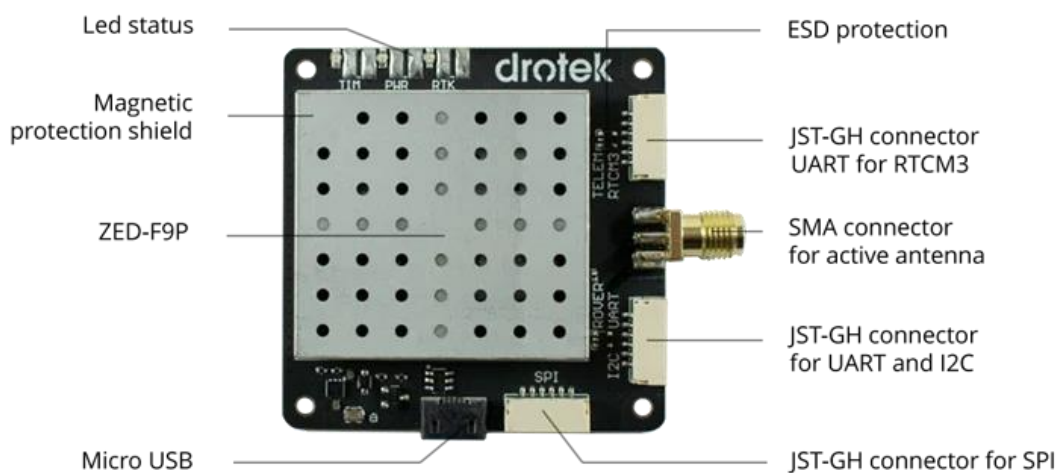
A fejezetben kijelentett állításaimat a mérési eredmények fejezetben fejtem ki részletesebben és támasztom, vizualizálom általam elvégzett mérések alapján.

2.3 Mobil RTK bázisállomás hardveres felépítése

2.3.1 GNSS vevő

Az RTK bázisállomáshoz elengedhetetlen egy dedikált GNSS vevő, ezért egy, erre a célra fejlesztett ZED F9P chippel rendelkező SIRIUS RTK GNSS ROVER lett a tokból kiemelve, és a megfelelő módon konfigurálva, hogy a hardver képes legyen ellátni a bázisállomás szerepét az RTK rendszerben. Az említett eszköz az 5. ábrán látható.

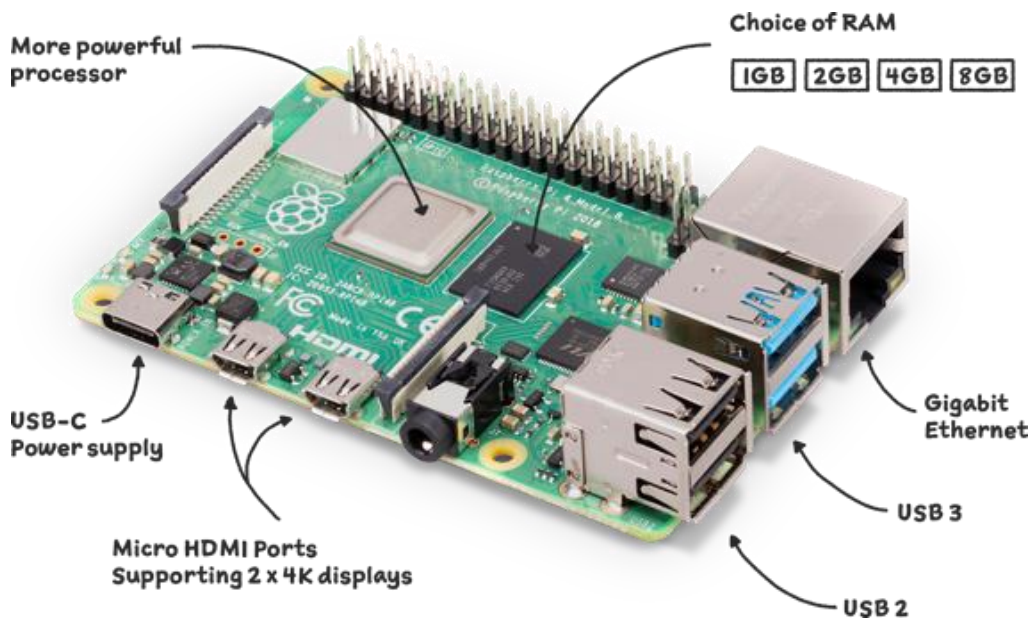
Azért esett erre a típusú GNSS vevőre a választás, hiszen képes a GPS, GLONASS, BeiDou, Galileo navigációs rendszerem műholdjait együtt kezelni, illetve egy RM3100-as magnetométerrel lett felszerelve. Továbbá nem utolsó szempont volt a hardver kiválasztásakor, hogy mind Windows, mind Linux alapú operációs rendszerekhez már létezik NTRIP szerver indítására



5. ábra Sirius RTK GNSS Rover, tokból kiemelve

2.3.2 Egykártyás számítógép

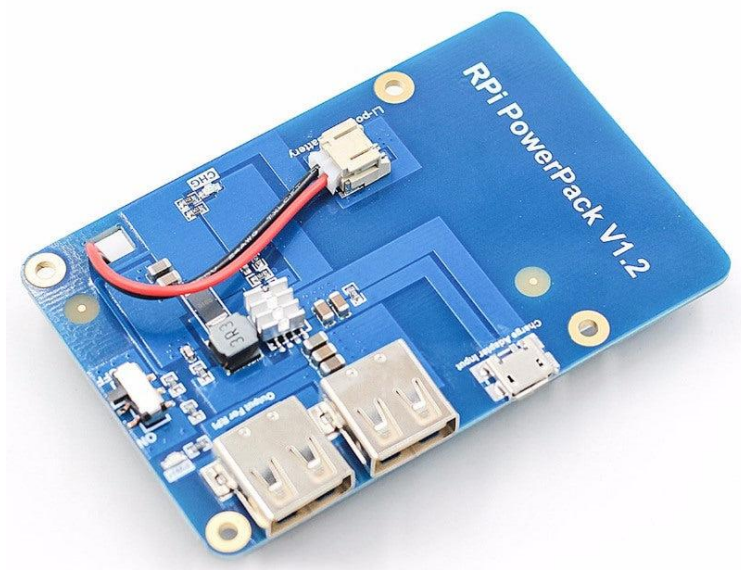
Az előző pontban bemutatott GNSS vevő egy, a 6. ábrán látható Raspberry Pi 4 Model B egykártyás számítógéphez van csatlakoztatva, melyre egy ubuntu 20.04 operációs rendszer került telepítésre. Azért esett a választás erre a mikrokontrollerre, mivel a negyedik generációs Raspberry Pi a legfejlettebb modellek egyike, és kiváló teljesítményt kínál a méretéhez képest, amit egy RTK bázisállomás megbízhatóság szempontjából megkövetel. Az eszköz erős négy magos processzorral rendelkezik, akár 8 GB RAM-mal is elérhető, és számos csatlakozási lehetőséget kínál, például USB bemenetet, melyen keresztül a GNSS vevő csatlakozik a számítógéphez, illetve micro HDMI csatlakozót amely, nagyban megkönnyítette a szoftverfejlesztés és tesztelés folyamatát. [5]



6. ábra Raspberry Pi 4 Model B

2.3.3 Akkumulátor

Az RTK bázisállomás mobilitása szempontjából kritikus pont egy megfelelő teljesítményű, újratölthető akkumulátor hozzákötése a hardverhez, amely a Raspberry Pi 4 egykártyás számítógépet látja el árammal, hogy az zökkenőmentesen tudja biztosítani a rajta futó NTRIP szerver elérhetőségét az RTK roverek számára.



7. ábra RPi PowerPack V1.2

Az akkumulátor, mely a bázisállomást ellátja árammal, egy, kifejezetten a Raspberry Pi egykártyás számítógépekhez készült akkumulátor, a Raspberry Pi Lithium Battery Power Pack V1.2, mely a 7. ábrán látható, hardverspecifikációja pedig az 2. táblázatban került részletezésre. Az üzemidőt a készülék gyártója megközelítőleg 9 órára ígérte, ezt méréssel fogom bemutatni dolgozatom egy következő fejezetében.

2. Táblázat RPi PowerPack V1.2 hardverspecifikáció

Specifikáció megnevezése	Specifikáció értéke
Akkumulátor kapacitása	3800 mAh
Maximális kisütési áram:	1,8 A
Terhelés nélküli kimeneti feszültség	5,1V ± 0,1V
Szabványos töltési áram/feszültség	1,0A/5,0V
Teljesen feltöltött Li-ion akkumulátor kikapcsolási feszültsége	4,18V - 4,2V

2.3.4 Akkumulátor állapot mérése

2.3.4.1 Hardveres felépítés

A bázisállomás akkumulátor állapotának nyomon követését egy INA 219 feszültség és árammérő szenzor hajtja végre.

Az INA219 egy nagy pontosságú, kétkimenetes digitális áram- és feszültségmérő eszköz. Az eszköz funkciója, hogy pontosan mérje az áramot és a feszültséget egyes részekben vagy eszközökben. Általában nagyobb áramok mérésekor alkalmazzák, például akkumulátorok töltésének vagy kisütésének ellenőrzésekor, illetve napelemes rendszerekben.

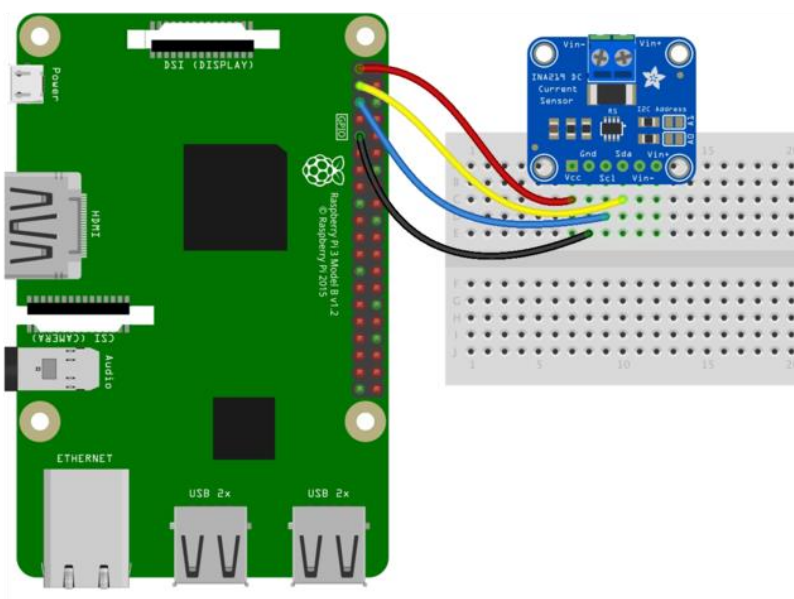
Egyik kiemelkedő jellemzője a nagy pontosság, ami lehetővé teszi a precíz méréseket. Ezenkívül alacsony energiafogyasztása is fontos előny, különösen olyan alkalmazásokban, ahol az energiahatékonyság kiemelt szerepet játszik, mint például a bázisállomás fogyasztásának optimalizálása.

3. Táblázat. Raspberry Pi és INA219 összekötési táblázat

Raspberry Pi 3 Model B pin	INA219 pin
3V3 Power	VCC
GPIO 2 (SDA)	SDA
GPIO (SCL)	SCL
Ground	GND

Az INA219 készüléket csatlakoztatni a Raspberry Pi-hoz, a mikrokontroller GPIO portjain keresztül, áramot, földelést biztosítva lehet, melyet a 8. ábra és a 3. táblázat alapján végeztem el a fejlesztés során.

Ezt követően szükséges csatlakoztatni a mérőeszközhöz a mérni kívánt készüléket, ami jelen esetben az RPi PowerPack V1.2 akkumulátor. Csatlakoztatás után gond nélkül olvasásra kerülnek a várt adatok, melyet a dolgozatom következő fejezetében részletezek.



8. ábra Raspberry PI 3 B és INA 219 kötési rajz

2.3.4.2 Szoftveres felépítés

Az akkumulátor állapotmérő szoftvere egy olyan implementáció, amely a már korábban bemutatott INA219 áram- és feszültségmérő chipet alkalmazza a Python programozási nyelv használatával.. A kód ezt az eszközt használva folyamatosan figyeli és rögzíti az áram- és feszültségadatait.

Először is, a szükséges modulokat importálja, majd inicializálja az INA219-et az I2C buszon keresztül. Az I2C busz egy elterjedt kommunikációs protokoll, amely lehetővé teszi a digitális eszközök közötti adatcserét. Az INA219 konfigurálása után a kód létrehoz egy naplófájlt, amelybe az áram- és feszültség adatokat fogja jegyezni használatkor

Azért fontos a naplófájl létrehozása minden használatkor, mivel ha valamilyen hiba történik, a naplófájlokból vissza lehet követni a hiba okát és hamarabb megoldást találni. Illetve ezen naplófájlok segítségével további kimutatások végezhetők.

A fő ciklusban a kód állandóan lekérdezi az INA219 szenzor busz- és shunt-feszültségét, az áramot és a teljesítményt. Ezekből kiszámítja a teljes feszültséget, az áramfogyasztást és a töltőáramot. Az adatokat egy naplófájlba írja, valamint kiírja a konzolra is, így a felhasználó is követheti a folyamatot fejlesztés, tesztelés közben.

2.4 Mobil RTK bázisállomás szoftveres felépítése

2.4.1 NTRIP szerver indítása

Az NTRIP szerver indítása érdekében szükséges egy olyan szoftver amely képes az ZED F9P chippel rendelkező GNSS vevőkön elindítani egy saját, dedikált NTRIP szervert. Kutatásom alapján jelenleg a projekthez leginkább illő és megfelelő nyílt forráskódú szoftver a pygnssutils python könyvtár..

A pygnssutils GNSS CLI segédprogramok és python osztályok sorozata, amely az alábbi könyvtárak köré épül:

- pyubx2 – UBX elemző és generáló könyvtár
- pynmeagps – NMEA elemző és generáló könyvtár
- pyrtcm – RTCM elemző könyvtár
- pyspartn – SPARTN elemző könyvtár

A rengeteg egyéb tulajdonsága mellett azért pygnssutils-ra esett a választásom, mivel lehetővé teszi egy NTRIP szerver CLI alapú elindítását, illetve CLI alapú kapcsolódást a már elindított NTRIP szerverre. A kiadott parancsban kapcsolódni kell a megfelelő soros porthoz, amelyen keresztül a GNSS vevővel tudunk kapcsolódni, továbbá meg kell adnunk a készülékünk IP címét is és autentikáció után könnyedén lehet csatlakozni az elindított NTRIP szerverre.

Az autentikációhoz a felhasználónevet és jelszót a PYGPSCLIENT_USER és a PYGPSCLIENT_PASSWORD környezeti változók segítségével lehetséges megváltoztatni az alapértelmezettől.

2.4.2 Virtual Private Network (VPN)-re való csatlakozás

Az RTK rover és az RTK bázisállomás közötti RTCM3 adatok továbbításához NTRIP szerveren keresztül elengedhetetlen, hogy a két GNSS vevő egy hálózaton legyen. Egy tesztpályára egy megfelelő hatótávú, gyorsaságú és megbízható hálózat kiépítése rendkívül költséges. Erre a megoldás a két vevőt egy közös VPN hálózatra csatlakoztatva elkerülhető egy komplett hálózat kiépítése, csupán két hálózati adapter integrálása szükséges az RTK rover és az RTK bázisállomás egykártyás számítógépéhez. Ezzel a megoldással az RTCM3 adatok veszteségi lehetősége is megoldódik.

A funkció implementáláshoz az OpenVPN szoftver Command Line Interface (CLI)-s használata tűnt a legmegfelelőbbnek, mely segítségével mind a bázisállomás, mind a rover elindítás után automatikusan csatlakozik a megfelelő VPN kulcs használatával a közös hálózatra. [6]

2.4.3 LED visszajelzések

Az RTK bázisállomás fejlesztési és tesztelése fázisánál, valamint a mindennapi használatánál egyaránt fontos a felhasználó számára a külső visszajelzés, melyre legjobb megoldásnak egy LED visszajelző felszerelését találtam.

Ugyan magán a GNSS vevőn találhatóak LED visszajelzők, viszont úgy gondoltam, hogy az ott megjelenített információkon kívül érdemes még egyéb folyamatokról is visszajelzést adni a felhasználó számára, mint például, hogy a survey-in metódus végbement-e és kész küldeni a bázisállomás a megfelelő RTCM3 korrekciós adatokat.

2.4.4 Shell script használata

Ahhoz, hogy az NTRIP szerver automatikusan elinduljon legjobb megoldásnak egy shell scriptet találtam készíteni. A script első lépése a VPN hálózathoz való csatlakozás, hiszen az itt kapott IP cím szükséges az NTRIP szerver elindításához. Ezt követően a serial port eléréséhez szükséges admin jogot biztosítani a pygnnsutils könyvtárkollekciónak, hogy zökkenőmentesen indulhasson az NTRIP szerver. Végző lépésben pedig, az NTRIP kliens kapcsolódik rá a saját NTRIP szerverére, hogy a LED-hez szükséges RTCM 1005 adat olvasása lehetséges legyen.

A végén a scriptet a *contrab* parancssori eszköz segítségével időzített műveletté lett alakítva, így az eszköz indulásakor automatikusan végbemennek a kiadott utasítások.

3. Mérési eredmények

3.1 Mérés célja

A projekt során elvégzett mérések fő célja annak az igazolása, hogy az általam fejlesztett mobil bázisállomás hatékonysága utoléri a kereskedelmi vállalatok által forgalmazott bázisállomások által nyújtott teljesítményt.

Ezen felül összehasonlítom egy általános GNSS vevő pontosságát egy RTK-val támogatott navigációs rendszerével, amely alapkövetelménye a projektnek melynek keretén belül a bázisállomás fejlesztésre került.

3.2 Mérés menete

A mérést a mobil bázisállomással végeztem, illetve az 1. ábrán látható, a mobil robotplatformon használt RTK rover konfigurációjából készítettem egy másolatot, így a mérések abban a környezetben, azokkal az eszközökkel lettek elvégezve, amelyek a projekt keretein belül hosszú távon is használatban lesznek.

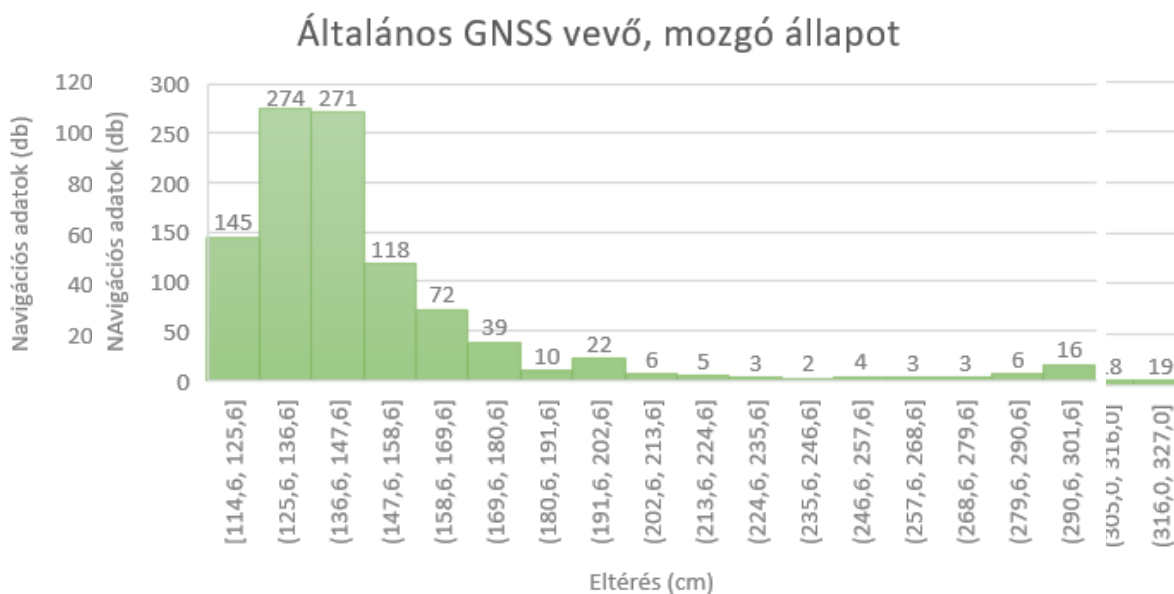
A GNSS vevők pontosságát a készülékek gyártója a Drotek cég által fejlesztett u-center szoftver segítségével mértem meg és az itt kapott adatok vizualizáltam grafikonok segítségével.

3.3 Mérési eredmények

3.3.1 Általános GNSS vevő

Első sorban fontosnak éreztem egy általános GNSS vevő pontosságának bemutatását mind statikus, mind mozgó állapotban, hiszen ezen pontosság volt számomra a kiinduló pont, az itt kapott eredményt javítja tovább az RTK rendszer.

A 6. ábrán látható mérési eredmény egy általános GNSS vevő pontossága statikus állapotban, nyílt ég alatt, melynek a pontossága alig megy 1 méter alá és átlagosan 2 méter és 1 méter között ingadozik, ami beváltotta a GNSS vevővel szembeni várakozást.



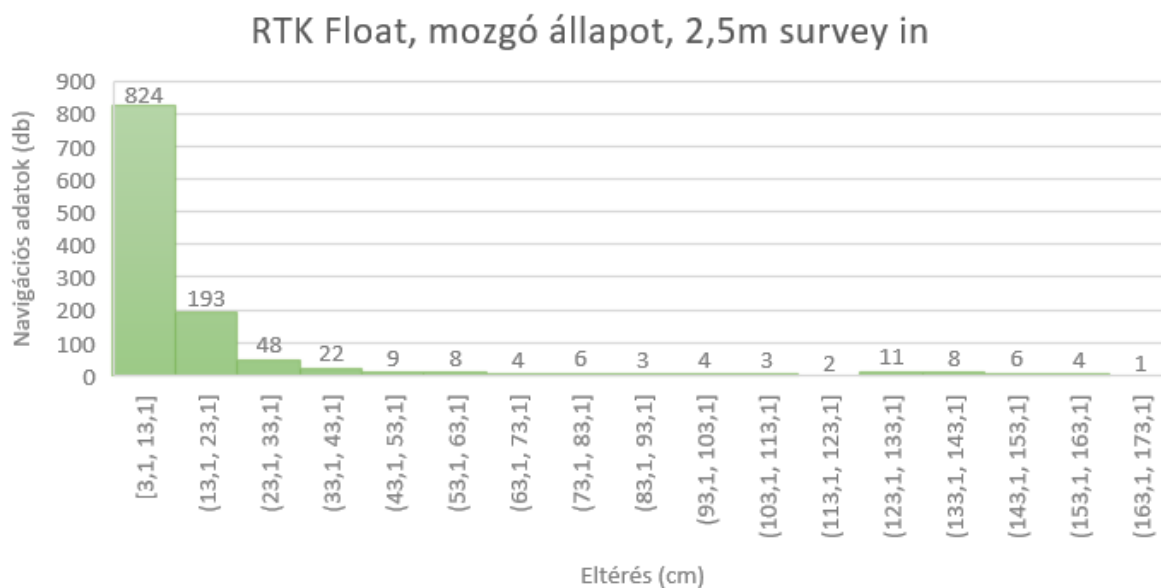
10. ábra Általános GNSS vevő mozgó állapotban

A mozgó állapotú mérés kiemelten fontos, mivel egy mobil robotplatform egyik pontjából a másikba való mozgás közben folyamatosan biztosítani kell a lehető legpontosabb helyzetét a robotnak. A 7. ábrán látható mérés eredménye egy általános GNSS vevő pontossága mozgó állapotban. Ahogy az ábrán is látható egy általános GNSS vevő mozgó állapotban 110 cm alá nem javul, és ez a pontosság nem elég a projekt követelményeinek kielégítéséhez.

3.3.2 RTK Float és RTK Fixed

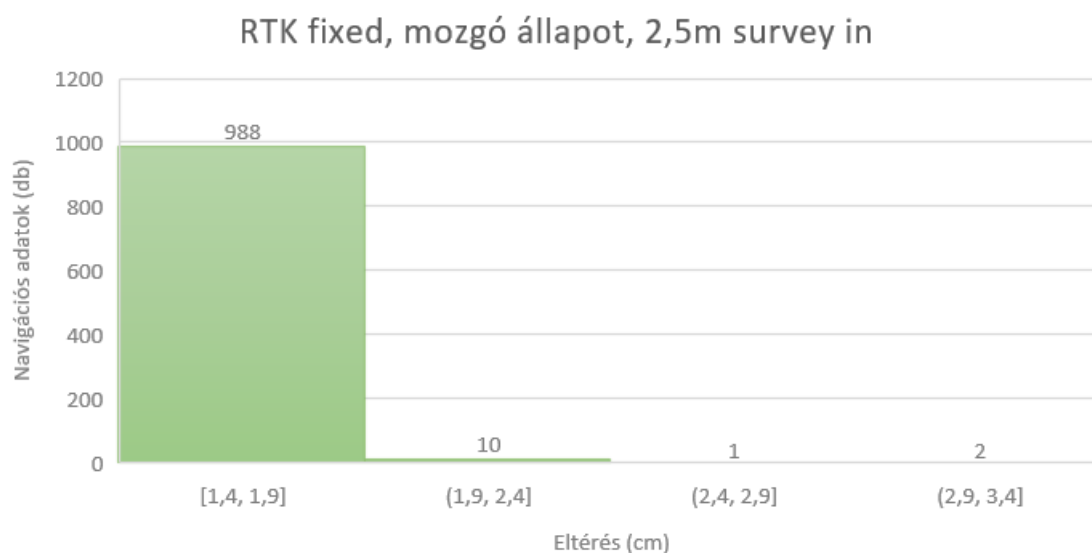
Ebben a fejezetben az RTK algoritmusának két állapotát, az RTK Float-ot és az RTK Fixed-et fogom összehasonlítani.

A 8. ábrán látható az RTK Float állapotban történt mérés eredménye, melynek során a bázisállomás 2,5 méteres survey-in közelítéssel üzemelt. Ahogy az ábrán is látható a nem megfelelő körülmények között mért RTK rover pontossága 3,1 cm és 13,1 cm között volt az esetek túlnyomó részén, ami bizonyítja az RTK korrigálásának a hatékonyságát, és a bázisállomás megfelelő működését. Már ez az eredmény is hatalmas előrelépés egy általános GNSS vevő pontosságához képest.



11. ábra RTK Float állapot, 2,5m survey-in esetén, mozgó állapotban

A 9. ábrán az RTK algoritmusának RTK Fixed állapotú mérési eredménye látható, melynek során a bázisállomás szintén 2,5 méteres survey-in közelítéssel működött. Ahogy az ábrán látható az RTK rover pontossága az esetek túlnyomó többségben 1,4 cm és 1,9 cm között van, amely elérte a projekt elején felállított követelményt és nagyban túlmutat egy általános GNSS vevő pontosságán. Ez az eredmény bizonyítja a projekt sikerességét, mind a navigációs rendszer kiépítése terén, mind a mobil RTK bázisállomás fejlesztése terén.



12. ábra RTK Float állapot, 2,5m survey-in esetén, mozgó állapotban

3.3.3 Antenna és antenna nélküli survey-in

A survey-in hosszának megméréséhez szintén a u-center szoftvert használtam, amely lehetővé teszi az eljárás nyomon követését. Az antennával és az antenna nélküli survey-in hosszát 3-3-szor mértem meg egyaránt és ezeknek az eredménye látható mutatom be a 2. táblázatban másodpercben megadva.

4. Táblázat Antenna és antenna nélküli survey-in mérési eredmények

	Antennával	Antenna nélkül
1. mérés	326 másodperc	732 másodperc
2. mérés	312 másodperc	756 másodperc
3. mérés	322 másodperc	695 másodperc
Átlag	320 másodperc	727,67 másodperc

A 2. táblázat eredményeként egyértelműen látható, hogy az antennával elvégzett survey-in metódus kevesebb mint fele annyi idő alatt eredményez rendelkezésre állást, mintha antenna nélkül indítanánk el. Továbbá kiemelném, hogy a survey-in procedura végbemenetelének a szórása is jelentősen kisebb antennával való használat esetén mint anélkül.

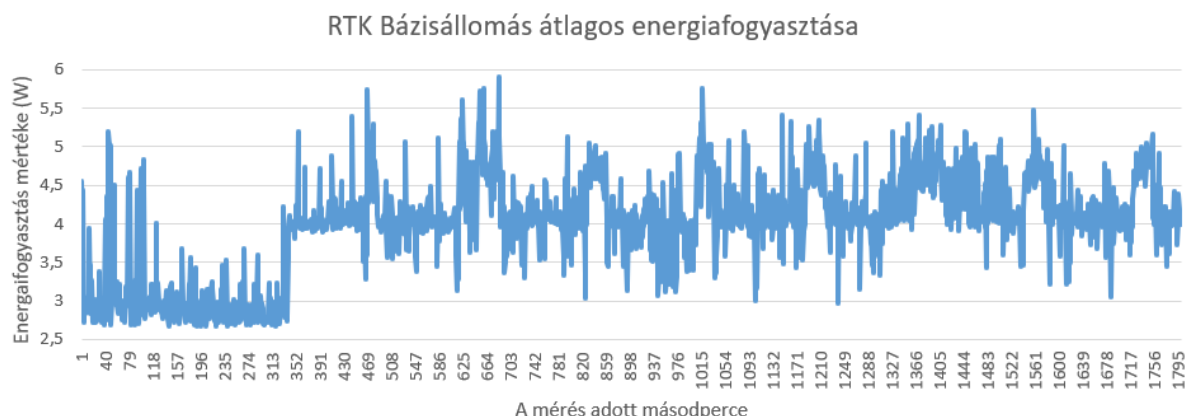
3.3.4 Akkumulátorral végzett mérések

Az RTK bázisállomás akkumulátor állapotának a mérését három különböző esetre bontottam: készenléti állapot, NTRIP szerver futtatása, illetve NTRIP szerver futtatása és a LED visszajelző használata.

A méréseket a használt Raspberry Pi 4 Model B mikrokontroller és a vele felszerelt RPi PowerPack V1.2 akkumulátorral végeztem, saját készítésű kód felhasználásával. A kapott értékeket az INA 219 műszer segítségével mértem.

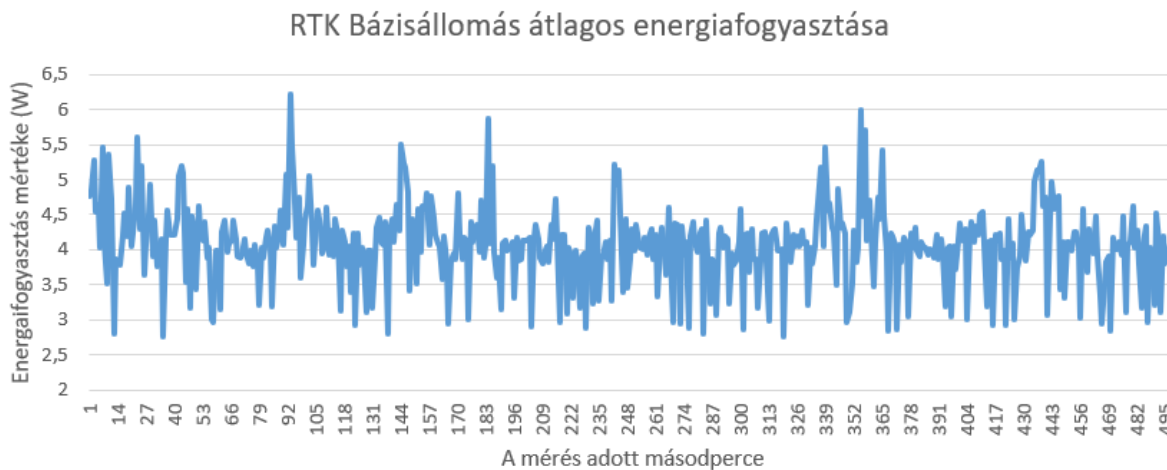
A 13. ábrán látható mérési eredmény a bázisállomás áramfogyasztását mutatja készenléti módban, illetve az áramfogyasztást mérő python nyelvű program futott a fél órás teszt alatt, ami másodpercenként mért egy értéket. Az átlagos energiafogyasztás 3,99 W lett a mérés ideje alatt. Ezt a mérést annak az érdekében végeztem el, hogy az általam összeállított hardveres mérőkörnyezet, illetve az általam írt mérőprogram megfelelően működik-e. A szakirodalom a

Raspberry Pi 4-es modell átlagos fogyasztását készenléti állapotban 3,8 W és 4 W közé teszi, tehát az alábbi méréssel igazoltam a mérés pontosságát, illetve valóságát.

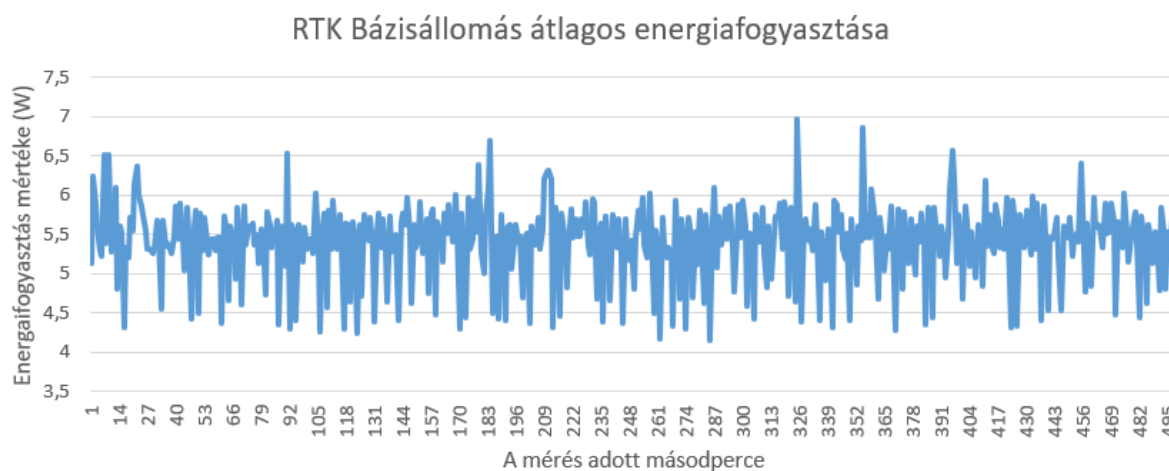


13. ábra RTK bázisállomás áramfogyasztása készenléti módban

A 14. ábrán látható mérés a bázisállomás egy kritikus használati módját hivatott mérni. Az NTRIP szerver futtatása állapot a survey-in befejeztéig, tart, utána lép életbe a 15. ábrán látható áramfogyasztás. Az NTRIP szerver futtatása közben a bázisállomás átlagos energiafogyasztása 4,08 W volt a mérés ideje alatt, amely nem kiugróan magas a készenléti állapothoz képest.



14. ábra RTK bázisállomás áramfogyasztása NTRIP szerver futtatása közben



15. ábra RTK bázisállomás áramfogyasztása NTRIP szerver futtatása, LED visszajelző használata közben

A 15. ábrán látható bázisállomás ezt az állapotot használja működése közben túlnyomó részt, mivel a survey-in befejezte után a bázisállomás jelzi, hogy készen áll a kliensek fogadására. A mérés során használt LED visszajelző viszonylag sok többlet áramfogyasztást eredményez, mivel így az átlagfogyasztás 5,4 W lett a mérés ideje alatt.

3.4 Levont következtetések

A mérési eredményekből egyértelműen levonható az a következtetés, hogy egy általános GNSS vevő és egy RTK-val támogatott navigációs rendszer pontossága alig összehasonlítható. A több mint 1 méteres pontosságról sikerült javítani az RTK rover helyzetének meghatározását 1,4 cm-re, amely egyaránt bizonyítja az RTK korrekciók hatékonyságát és a saját fejlesztésű mobil bázisállomás hatékonyságát. Illetve az elért pontosságot nem rontja az NTRIP szerver egy VPN szerveren való futtatása sem, amely nagyban növeli a navigációs rendszer mobilitását illetve modularitását.

Illetve a mérési eredmények fejezetemben továbbá bizonyításra került egy külső antenna használatának fontossága, hiszen a rendelkezésre állási időt több mint a felére csökkentette, mintha antenna nélkül lépne használatba a bázisállomás.

Továbbá bizonyítottam, hogy a rendelkezésre állás gyorsaságát, illetve megbízhatóságát nagyban javítja egy külső antenna használata az RTK bázisállomást támogatva.

Végző soron pedig bázisállomás üzemideje a felszerelt akkumulátorral garantáltan 6 óra 30 perc, ami kielégíti a robotplatform által megkövetelt üzemidőt.

4. Összefoglalás

Összességében a kijelölt követelmények kielégítésre kerültek, hiszen sikerült egy olyan mobil RTK bázisállomást fejleszteni, ami nem csak az aktuális navigációs rendszerbe, de más RTK-val támogatott navigációs rendszerekbe is könnyen integrálható.

A bázisállomás saját, automatikusan induló NTRIP szerverrel rendelkezik, mely egy VPN szerveren fut, így a hálózati kialakítás sem jelent akadályt.

Sikerült elérni a centiméter pontos helyzetmeghatározást a saját fejlesztésű bázissal ugyan úgy, mint a nagyobb, kereskedelmi cégek által gyártott RTK bázisállomás, viszont egy sokkal gazdaságosabb kivitelezésben.

5. Továbbfejlesztési célok

A projekt következő állomásaként az első számú cél különböző RTCM3 üzenetekkel való tesztek elvégzése, esetleges további pontosság elérése érdekében. Ezek a tesztek alapján újra kalibrálni a bázisállomást, hogy az RTK-val támogatott navigációs rendszer teljesítménye növekedjen.

Egyéb tesztek végzése a bázisállomás survey-in beállítását módosítva. Hasznosnak bizonyulna elvégezni a mérést, hogy kiderüljön a bázisállomás hosszabb rendelkezésre állási idő mellett jobb teljesítménnyel tud-e üzemelni.

Végül a legfőbb továbbfejlesztési cél, a követelményrendszerben meghatározott 5G modem felszerelése a bázisállomásra, az ott meghatározott indokok miatt.

Ábrajegyzék

1. ábra A navigációs rendszer blokkvázlata	4
2. ábra GNSS navigációs rendszer	8
3. ábra RTK rendszer működési elve	8
4. ábra DA910 multi-band GNSS antenna	9
5. ábra Sirius RTK GNSS Rover, tokból kiemelve	10
6. ábra Raspberry Pi 4 Model B	11
7. ábra RPi PowerPack V1.2	12
8. ábra Raspberry PI 3 B és INA 219 kötési rajz.....	14
9. ábra általános GNSS vevő statikus állapotban	18
10. ábra Általános GNSS vevő mozgó állapotban	18
11. ábra RTK Float állapot, 2,5m survey-in esetén, mozgó állapotban.....	19
12. ábra RTK Float állapot, 2,5m survey-in esetén, mozgó állapotban	19
13. ábra RTK bázisállomás áramfogyasztása készenléti módban	21
14. ábra RTK bázisállomás áramfogyasztása NTRIP szerver futtatása közben	21
15. ábra RTK bázisállomás áramfogyasztása NTRIP szerver futtatása, LED visszajelző használata közben.....	22

Irodalomjegyzék

- [1] E. D. Kaplan és C. J. Hegarty, Understanding GPS/GNSS, 2017.
- [2] B. Hofmann-Wellenhof, H. Lichtenegger és E. Wasle, GNSS – Global Navigation Satellite Systems: GPS, GLONASS, Galileo, and more, 2007.
- [3] C. Mekik és M. Arslanoglu, „Investigation on Accuracies of Real Time Kinematic GPS for GIS Applications,” pp. 22-35, 2009.
- [4] G. Weber, „Streaming Real-Time IGS Data and Products Using NTRIP,” *Federal Agency for Cartography and Geodesy*, 2006.
- [5] T. Maragathan, P. Balasubramanie és M. Vivekanadhan, „IoT Based Home Automation System using Raspberry Pi 4,” 2021.
- [6] Z. Zhang, Y.-Q. Zhang és B. Li, „An Overview of Virtual Private Network (VPN): IP VPN and Optical VPN,” *Photonic Network Communications*, %1. kötet7, pp. 213-225, 2004.