



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Rusvai Miklós

WI-FI ALAPÚ BELTÉRI LOKALIZÁCIÓ

TDK dolgozat

KONZULENS

Dr. Kovácsházy Tamás

BUDAPEST, 2023

Összefoglaló

Napjainkban egyre nagyobb az érdeklődés a széles körben használható és elfogadható árú beltéri pozíció lokalizációs technológiák iránt, amelyek a GNSS-hez hasonló, de beltéri alkalmazásokat tennének lehetővé. Erre egyre több különböző megoldás jelenik meg, melyek más-más technológiát alkalmaznak, más fizikai elvre vezetnek vissza a pozíció meghatározását. Ilyen például az ultraszélessávú kommunikáció (UWB), amely talán az egyik legjobb pontossággal rendelkezik a széles körben elérhető megoldások között. A hátránya a magas ára és a kis elterjedtsége. A Wi-Fi esetén korán kidolgozásra kerültek a nem túl megbízható térerőn alapuló megoldások, amelyek hibáinak kiküszöbölésére vezették be az UWB lokalizációs megoldáshoz hasonló fizikai elveken működő Fine Timing Measurement (FTM) szabványt (IEEE 802.11-2016-ban). Az FTM támogatása sem jellemző a Wi-Fi eszközökben sajnos, kevés ezt támogató hardver és szoftver érhető el. Ilyen eszközök az ESP32-S2 és ESP32-S3 mikrokontrollerek, melyek támogatják a Wi-Fi FTM-et a megfelelő kiegészítő szoftverekkel, viszont ezek képességeiről kevés tudományos igényességű vizsgálat érhető el. Szintén nem egyértelmű, hogy hogyan lehet ezeket rendszerbe integrálni. Ennek megfelelően dolgozatomban az ESP32-S2 és ESP32-S3 mikrokontrollerekkel elérhető pontosság vizsgálatát mutatom be tudományos igényességgel és egy ezekből építkező rendszer megalkotására teszek kísérletet.

A dolgozat a lehetséges alkalmazások és az azok által támasztott követelmények elemzése után először bemutatja a rádiós kommunikáción alapuló beltéri pozíció lokalizációs megoldások fizikai és műszaki lehetőségeit. Ezek után ismerteti a szélesebb körben alkalmazott technológiákat, és elhelyezi a Wi-Fi FTM-t ezek között. Bemutatja a Wi-Fi FTM működését, felsorolja az azt támogató elérhető eszközöket, és áttekinti a használt ESP32-S2 és ESP32-S3 hardver és szoftver környezetét. A dolgozat központi része az alkalmazott mérési eljárások, az azokat megvalósító szoftverek, az eredmények és azok értékelésének a részletezése, és az azt használó rendszer ismertetése. Ebben a részben kerülnek ismertetésre az új tudományos eredményeim is. A dolgozatot a munka során levonható következtetések és továbbfejlesztési lehetőségek ismertetése zárja le.

Abstract

Currently, there is growing interest in widely applicable and affordable indoor position localization technologies that would enable indoor applications similar to GNSS. Various solutions are emerging for this purpose, employing different technologies and physical principles to determine position. One such example is ultra-wideband communication (UWB), which perhaps offers one of the best accuracies among widely available solutions. However, it comes with a high price tag and limited adoption. In the case of Wi-Fi, early solutions based on unreliable signal strength were developed, and to address their shortcomings, the Fine Timing Measurement (FTM) standard, operating on principles similar to UWB localization, was introduced (in IEEE 802.11-2016). Unfortunately, support for FTM is not common in Wi-Fi devices, and there is limited hardware and software available for it. Devices like ESP32-S2 and ESP32-S3 microcontrollers support Wi-Fi FTM with the appropriate supplementary software, but there is a lack of rigorous scientific investigation into their capabilities. It is also not clear how to integrate these into a system. Consequently, in this thesis, I present a scientifically rigorous examination of the accuracy achievable with ESP32-S2 and ESP32-S3 microcontrollers and attempt to create a system based on these.

After analyzing potential applications and their requirements, the thesis first introduces the physical and technical possibilities of indoor position localization solutions based on radio communication. It then discusses more widely adopted technologies and situates Wi-Fi FTM among them. The thesis explains the operation of Wi-Fi FTM, lists available supporting devices, and provides an overview of the hardware and software environment used, specifically the ESP32-S2 and ESP32-S3. The central part of the thesis details the applied measurement procedures, the software used to implement them, the results, their evaluation, and the description of the system utilizing them. This section also presents my new scientific findings. The thesis concludes with a discussion of the conclusions drawn during the work and potential avenues for further development.

Tartalomjegyzék

Összefoglaló	2
Abstract	3
Tartalomjegyzék.....	4
1 Bevezetés	6
1.1 Téma értelmezése	6
1.2 A téma fontossága	6
1.3 Elméleti háttér.....	7
1.3.1 Lehetséges algoritmusok óraszinkronizációhoz	7
1.3.2 Wi-Fi [16]	8
1.3.3 Fine Timing Measurement (FTM) [20]	9
1.4 Rádiós alapú beltéri lokalizációs megoldások.....	10
1.4.1 Mérési elvek	10
1.4.2 Fizikai megvalósítás/piacon megjelenő technológiák	14
2 Alkalmazott eszközök	19
2.1 ESP32-S2.....	19
2.1.1 Leírása [7].....	19
2.1.2 Blokk vázlat	20
2.2 ESP32-S3.....	21
2.2.1 Leírása [8].....	21
2.3 A mikrokontrollerekhez tartozó könyvtár család	22
3 Mérési feladat leírása	23
3.1 Mérés tervezés	23
3.2 Méréshez használt szoftverek	24
3.2.1 Első kód: FTM_Initiator_test arduino kód:	24
3.2.2 Második kód: wifi_ftm_test2 arduino kód:	25
3.2.3 CoolTerm szoftver	27

3.3	Mérések és eredményeik az ismertett kódok felhasználásával.....	28
3.3.1	Mérések kis távolságokra	28
3.3.2	Mérések nagy távolságokra, nyílt téren.....	29
3.4	Mérések az egyetem folyosóján	30
3.5	Két node-os mérések eredményeinek kiértékelése.....	34
3.5.1	Mérés az egyik végponttól 5 méterre a másik végponttól 33 méterre.....	34
3.5.2	Mérés mindkét végponttól 19 méterre.....	34
3.5.3	Mérés az egyik végponttól 12 méterre a másik végponttól 26 méterre....	35
3.6	Mérés 2 dimenzióban.....	37
3.7	Mérés 3 dimenzióban.....	38
3.8	Összefoglalás és jövőbeli tervek.....	38
4	Hivatkozások.....	39

1. Bevezetés

1.1 Téma értelmezése

A téma a Time of Flight időmérésből származó és ebből számított távolság meghatározás. Ennek elvégzéséhez elengedhetetlen az óraszinkronizáció és késleltetés mérés elméleti hátterének megismerése, továbbá ezeknek az ismereteknek a felhasználása távolság mérés megtervezéséhez és a megtervezett távolság mérés alkalmazhatóságának vizsgálata mérésekkel. A pozíció lokalizációhoz a 3 dimenziós térgeometria és trigonometria ismerete, hogy a mért adatokból ki tudjuk számítani a mérő node pozícióját. A mérések megtervezésénél fontos szempont, hogy azokkal minél többféleképpen és minél széleskörűbben meg tudjuk vizsgálni a mérőműszerek és a mérési elv pontosságát és tulajdonságait. További fontos lépés a mérések során elmentett adatok későbbi részletes feldolgozása és kiértékelése.

1.2 A téma fontossága

Az okostelefonok és a GPS által nyújtott lokalizációs szolgáltatások elterjedésével egyértelművé vált a kültéri helymeghatározás fontossága. De a civil szféra növekvő igénye a biztonságra (gyerekek és idősek figyelése, veszélyes területekre való belépés kontrollja stb.), az ipar területén a hatékonyság növelése egyre nagyobb igényt teremt a beltéri lokalizációra is. A kültéri helymeghatározási módszerek nem nyújtanak kellő pontosságot beltéri helymeghatározáshoz, beltéri pozíció lokalizációra pedig még nem létezik egy széleskörben elterjedt, jól alkalmazható kiforrott technológia, amit a minden napokban tudnánk alkalmazni.

A Beltéri Pozícionálási Helymeghatározási és Navigációs technológia erősen kapcsolódik a piaci kereslethez. Az 1. táblázat több tipikus beltéri pozíció lokalizációs alkalmazás pontossági követelményeit és költségeit mutatja be [2].

Általánosságban elmondható, hogy a nagyobb pontosságot igénylő alkalmazásoknál a létesítmények és berendezések költségei ennek megfelelően magasabbak. Sok forgatókönyvben (például a tömegpiacon) a minimális felszerelési költség és a felhasználni kívánt berendezések költsége megkerülhetetlen tényezők [2].

A táblázat mutatta magas költségek miatt fontos, hogy találjunk egy elfogadható pontosságú megfizethető beltéri lokalizációs technológiát [2].

Alkalmazás	Pontossági követelmény	Költség
Ipar és építőipar	Centiméter szinttől milliméter szintig.	10 000 dolláros szint.
Autonóm járművek	Deciméter szinttől centiméter szintig.	1000 USD szinttől 10 000 USD szintig.
Beltéri térképezés	Deciméter szinttől centiméter szintig.	1 000 dolláros szint.
Segélyhívó	Vízszintesen deciméter, magasságban padlósinten.	1 000 dolláros szint.
Gyalogos alkalmazások	Vízszintesen méteres, magasságban padlósinten.	Használja a meglévő fogyasztói eszközöket. A infrastruktúra kiépítésének költsége az 1 000 USD-s szint per 100 m ² .
Mobil vészhelyzet	80% 50 m-en belül.	Használja a meglévő mobilrendszereket.

1. táblázat [2]

1.3 Elméleti háttér

1.3.1 Lehetséges algoritmusok óraszinkronizációhoz

A projekt elindításakor a kezdetleges cél több különböző node egy központi node-hoz történő szinkronizálása volt. Ebből a szinkronizációból tértem át az időmérésen alapuló távolság meghatározásra és pozíció lokalizációra. Az óraszinkronizáció megvalósításához első körben utána olvastam több különböző óraszinkronizációs protokollnak és ezeket összehasonlítottam. A következőkben röviden bemutatom az általam megismert óraszinkronizációs protokollokat.

1.3.1.1 Network Time Protocol (NTP) [14]

Az NTP protocol egy internetes protocol, amely óra szinkronizálásra szolgál adatcsomag cserélő, változó késleltetésű rendszerekben.

Az NTP szándéka, hogy az összes résztvevő számítógépet szinkronizálja néhány milliszekundum alatt. Az NTP használja az intersection algorithm-t, ami egy módosított verziója Marzullo algoritmusának, hogy kiválassza a pontos idő szervereket és úgy tervezték, hogy mérsékelje a változó késleltetésű rendszer hatását.

1.3.1.2 Simple Network Time Protocol (SNTP) [14]

Amikor az NTP leváltotta a régi Time Protocol-t, néhány alkalmazás túl bonyolultnak találta a teljes protokollt. Ennek a résnek a betöltésére hozták létre a Simple Network Time Protocol-t (SNTP). Ez egy egyszerűsített változata az NTP-nek, amely teljesen kompatibilis vele.

Egyszerűbb algoritmus, de alacsonyabb idő pontosságot nyújt, ezért nem tanácsos időt szinkronizálni SNTP forrásról.

1.3.1.3 Precision Time Protocol [18]

A Precision Time Protocol egy olyan protokoll, amelyet számítógépes hálózatokban óra szinkronizációra használnak. Helyi (lokális) hálózaton ezzel elérhető a szubmikronos pontosság is, amely alkalmassá teszi mérésre és irányításra a rendszert. A PTP-t pénzügyi tranzakciók szinkronizálására, mobil átjátszó tornyok átvitelére, tenger alatti akusztikus tömbökhöz alkalmazzák és hálózatokhoz melyek igénylik a precíz időzítést, de nincs hozzáférése a műholdas navigációs jelekhez.

1.3.2 Wi-Fi [16]

A Wi-Fi az IEEE által kifejlesztett, vezeték nélküli mikrohullámú kommunikációt (WLAN) megvalósító, széleskörűen elterjedt szabvány (IEEE 802.11) népszerű neve. A Wi-Fi hálózat rádióhullámokat használ a számítógépek közötti adatátvitelhez. Wi-Fi hálózat szinonimájaként a WLAN hálózat (Wireless Local Area Network) kifejezést is használják, amelynek jelentése vezeték nélküli helyi hálózat. A Wi-Fi hullámok elektromos impulzusokként vagy hullámokként terjednek a levegőben. A kibocsátott elektromos jel a megtett távolsággal arányosan folyamatosan gyengül, majd elhal. A projekt méréseinek elkészítése során 2,4 GHz frekvenciájú Wi-Fi jelet használtam, melynek a hullámhossza 12,5 cm.

A fizikai környezet különféle anyagokból épül fel, amelyek eltérő módon befolyásolják a Wi-Fi hálózat működését és a jelterjedést.

1. Visszaverődés: a Wi-Fi jel, mint rádióhullám, az egyes felületekről visszaverődhet. A visszaverődés esetén a jelek gyengülhetnek, és interferenciát okozhatnak a vevő eszközön.

2. Refrakció: a refrakció a jel, irányváltozása amikor egy közegbe lép, amelyben eltérő sebességgel halad tovább. Pl. az üveg vagy a víz képes megtörni az elektromos jeleket.

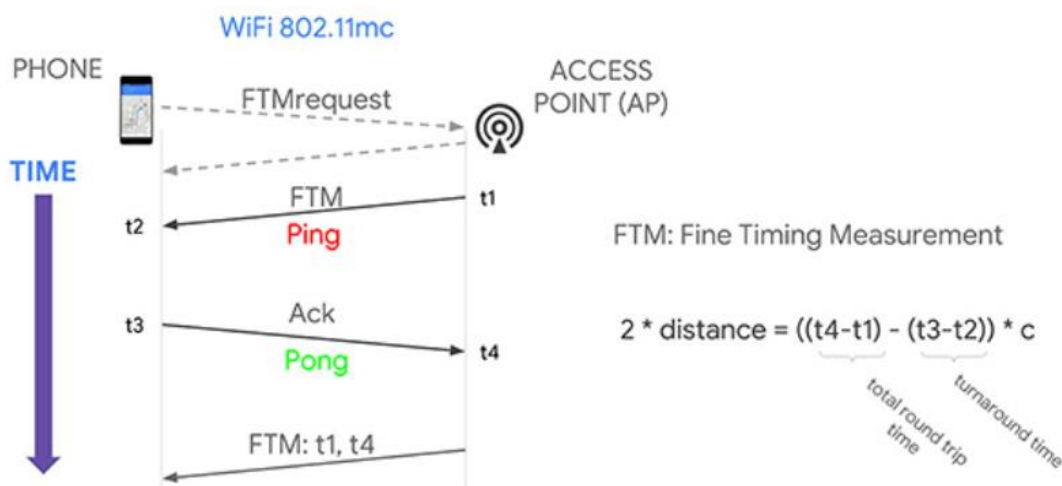
3. Diffrakció: a diffrakció (elhajlás) az a jelenség, amikor a rádióhullámok áthatolhatatlan akadályokkal találkoznak és azt körülölelik. Ebben az esetben a rádióhullám iránya és intenzitása egyaránt megváltozik. A diffrakció (rádiójel)árnyék-zónát alakíthat ki az adó és a vevő között.

4. Szóródás: ez a jelenség hasonló a refrakcióhoz, és ugyanannyira kiszámíthatatlan. A levegőben lévő por, a páratartalom, az anyagok egyenetlenségei és összetétele az elektromos jel szétszóródásához vezethetnek minden irányba. Ez jelentős hatással lehet a jel integritására és erősségére.

5. Elnyelődés: ez az egyik leggyakoribb reakció, amelyet a rádiójelek útjuk során elszenvednek. Általában egy anyag konvertálja a jel energiáját hővé.

1.3.3 Fine Timing Measurement (FTM) [20]

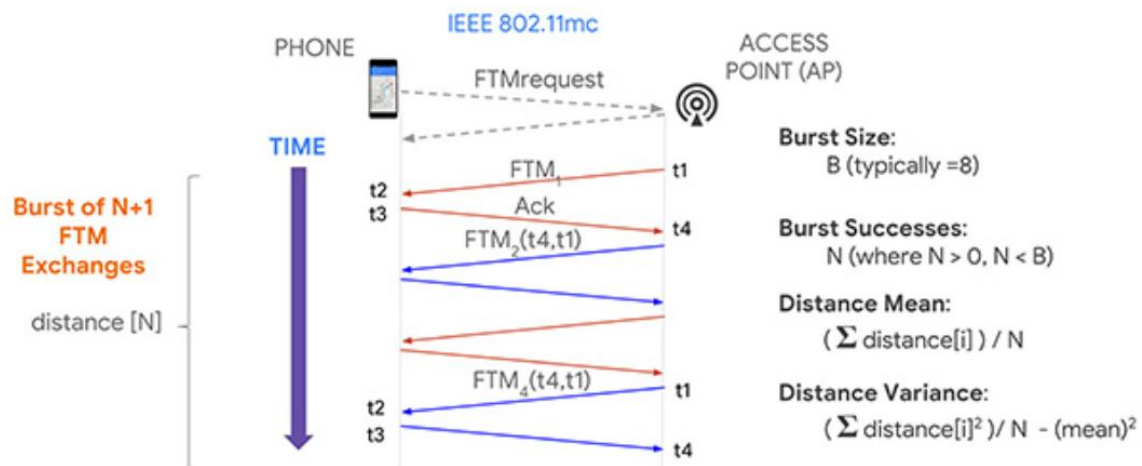
A Wi-Fi Fine Timing Measurement (FTM) funkciót az IEEE 802.11mc szabvány határozza meg, és az IEEE 802.11-2016 tartalmazza. A mérési eljárás lehetővé teszi, hogy két állomás meghatározza egymás közötti távolságot a kicserélt üzenetek terjedési idejének mérésével. Az idő, ami alatt a Wi-Fi jel elér a telefontól egy Wi-Fi csatlakozási ponthoz a levegőben arányos a kettő közötti távolsággal (körülbelül 3,3 nanoszekundum méterenként). Mióta a telefonok rendelkeznek saját belső órával és nem egy közös AP-val szinkronizálják óráikat, azóta az egyutas késleltetés időmérése nem alapulhat az üzenetváltás két végének időbélyegeinek különbségén. Szerencsére az óra ofsztetje ellentétesen befolyásolja az időbélyegek különbségét, amikor a jel ellentétes irányba halad. Végezetül meghatározható a terjedési idő (RTT) anélkül, hogy ismernénk az óra ofsztetjét. Ehhez egyszerű összeadásokat és kivonásokat kell végezni négyszer: $RTT=(t4-t1+t2-t3)$.



1. ábra FTM kommunikáció sémája és távolság számítás [20]

Fontos, hogy a második FTM “Ping” a csatlakozási ponttól tartalmazza az elküldési időpontot t1 és a megérkezés időpontját, t4, a csatlakozási ponthoz.

A terjedési idő mérése nem tökéletesen pontos, a mérésnek különböző hibája lehet, RF interferencia ugyanúgy hibát okozhat, mint a környezetben lévő tárgyak helyzete és mozgása. A megismételt mérések javíthatják a pontosságot:



2. ábra FTM kommunikáció üzenet váltása [20]

Feljegyezzük az utolsó üzenet váltást, amit nem követett még további olyan jel, mely tartalmazta a t4 és t1 időpontokat, így az utolsó üzenet váltás terjedési idejét nem tudjuk meghatározni. N darab üzenet váltással N-1 terjedési időt tudunk meghatározni.

Ismerve a terjedési idők 3, 4 vagy több AP esetén, akkor az ismert pozíciókból meghatározható az okostelefon pozíciója. Ha sem az okostelefon, sem a csatlakozási pontok pozíciója nem ismert kezdetben, akkor egyidejű lokalizációt és leképezést is lehet végezni.

Az IEEE 802.11mc Wi-Fi szabvány nyújt metódusokat a mérésekre.

1.4 Rádiós alapú beltéri lokalizációs megoldások

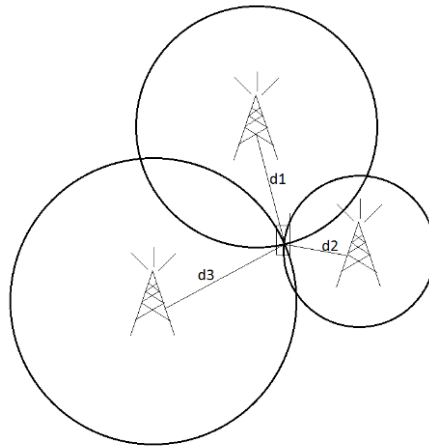
1.4.1 Mérési elvek

1.4.1.1 Time of Flight (TOF) mérésen alapuló [4]

Ha egy távolságot le szeretnénk mérni, akkor az a két végpont közötti út megtétele alatt eltelt idő, és a sebesség ismerete alapján könnyen kiszámítható. A Time of Flight (ToF) vagy más néven Time of Arrival (ToA) módszer is ezen az elméleten alapszik. Egy mobiltelefon távolságát számunkra ismert helyzetű ponttól, jelen esetben a bázisállomástól, úgy határozhatjuk meg, hogy bázisállomás kérésére a mobiltelefon jelcsomagot küld a bázisállomásnak. A jel terjedési ideje, ha a bázisállomás, és a mobiltelefon szinkronban vannak, könnyedén lemérhető, a terjedési sebesség pedig a fénysebességgel közelíthető.

Ha csak a mobiltelefon és kizárólag egy bázisállomás közötti jelterjedési időt mérjük meg, abból még nem következtethetünk a pontos pozícióra. Ahhoz, hogy egy jól behatárolható kis területre szűkíthessük az eszköz helyzetét, legalább 3, de inkább több bázisállomástól kell lemérnünk az jelterjedési időt.

A terjedési idő és a terjedési sebesség ismeretében kiszámított távolság minden egyes bázisállomás körül 3 D-ben egy gömböt határoz meg. Az összes gömböt összevetve megkapjuk azok metszéspontját, amely megmutatja a keresett mobiltelefon pozícióját.



3. ábra ToF mérés [4]

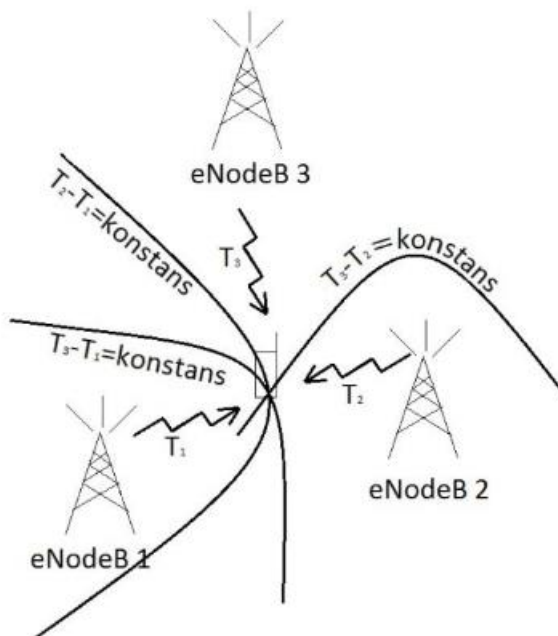
A módszernek van egy elméleti határa, amely alá nem lehet a ToA-val szorítkozni. Ez kiszámítható a jel terjedési sebességéből és a rendszer mintavételi frekvenciájából:

$$d = \frac{c}{f_s} = \frac{3 * 10^8 \frac{m}{s}}{2,4 * 10^9 Hz} = 0,125 m = 12,5 cm$$

Ez viszonylag nagy hátránya ennek az eljárásnak, viszont sok előnye közé tartozik az egyszerűség, nincs új hálózati elemre szükség az alkalmazásakor, viszonylag gyorsan eredményhez vezet, ha ezt választjuk.

1.4.1.2 Helymeghatározás az érkezési időkülönbségek mérése alapján (Time Differenc of Arrival - TDOA) [4]

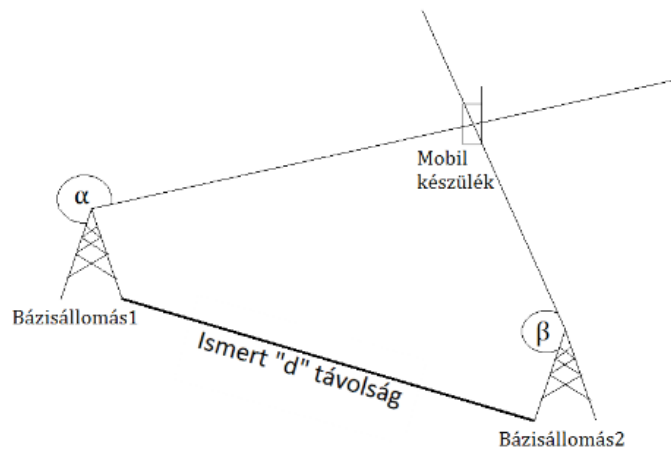
A TDOA módszernél a helymeghatározás a TOA-val ellentétben, nem az egyes bázis-állomásokból beérkező időkből számított távolságokat használjuk fel, hanem az egymáshoz viszonyított beérkezési időkülönbségekből hiperbolák segítségével számítunk helyzetet.



4. ábra TDOA módszer [4]

1.4.1.3 Érkezés szögén alapuló helymeghatározás (Angle Of Arrival -AOA) [4]

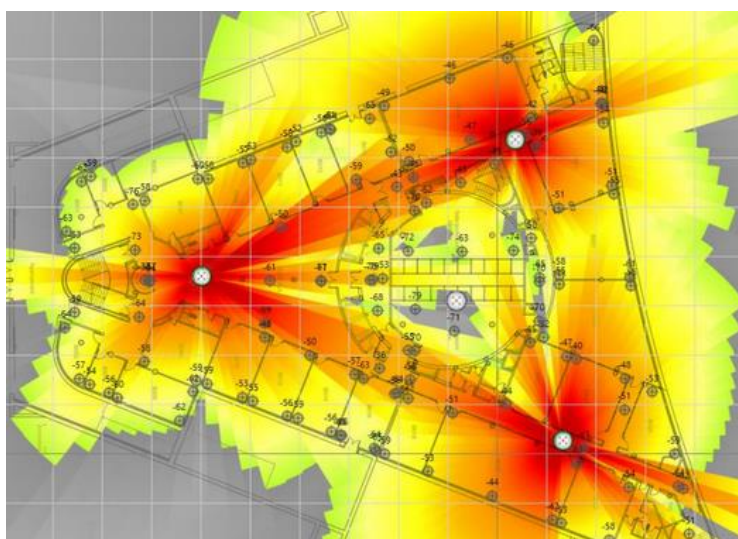
Az AOA pozicionálási eljárás alkalmazásakor, az adó által küldött jelet nem egy antenna, hanem egy antenna rendszer (legalább két antenna) veszi bázisállomásonként, és ezek az antennák mérik a beérkező jel fázisát. Az egyes antennák által mért ugyanazon beérkezett jelek között fáziskülönbség van. Ebből a fázis különbségből következtethetünk a mobiltelefon helyzeti irányára. Ha ezzel egy időben elvégezzük ugyan ezt a mérést egy másik bázisállomáson, a megkapott irányok meghatároznak egy egyenest, amelyek metszéspontjában van a meghatározni kívánt mobilkészülék.



5. ábra AOA módszer [4]

1.4.1.4 Távolságmérés a vett jel szintje alapján (RSS - Received Signal Strength) [4]

A RSS pozicionálási módszer alkalmazásakor a mobiltelefonunk által mért jelerősség a legfontosabb mérési adat, amely segítségével kiszámíthatjuk a készülék helyzetét. Ez úgy történik, hogy az adóteljesítményből, és a vett teljesítményből számítunk távolságot. Ez a távolság a bázisállomás körül egy kört határoz meg. Ezen a körön helyezkedik el valahol a mobiltelefon. Ha legalább három adótorony rendelkezésünkre áll, mindháromnál elvégezve a számításokat, ívmetszéssel könnyen megkaphatjuk a mobilkészülék helyzetét. Előnye, hogy viszonylag egyszerűen mérhető, a jelerősség mérését már régebbi generációjú hálózatokban is szabvány írja elő, de nagyon nagy hátránya, hogy figyelembe kell venni a környezetből származó hátráltató tényezőket, többek között a reflexiót, több utas terjedést, szakaszcsillapítást.



6. ábra RSSI térkép [5]

1.4.2 Fizikai megvalósítás/piacon megjelenő technológiák

1.4.2.1 Ultrawide Band [10]

Az Ultra Wideband (UWB) a Wi-Fi-hez és a Bluetooth-hoz hasonló, rádiótechnológia, amely nagyon alacsony energiaszintet használ rövid távolságú, nagy sáv szélességű kommunikációhoz a rádióspektrum jelentős részén. Az UWB hagyományos alkalmazásai közé tartozik a nem-együtműködő radar képalkotás. A legújabb alkalmazások célja érzékelő adatgyűjtés, pontos helymeghatározás és követés.

A 2000-es évek elején korlátozottan használták katonai radarokban, titkos kommunikációhoz és szív működést távolról figyelő rendszerekben. Nem igazán terjedt el, mostanában azonban újra felfedezték: fő alkalmazási területe a hely- és eszköz-meghatározás. Bár mind a Wi-Fi-t, mind a Bluetooth-t továbbfejlesztették annak érdekében, hogy segítségükkel precízebben lehessen meghatározni más berendezések helyét, az UWB alapból pontosabb náluk, kevesebb energiát fogyaszt, és várhatóan olcsóbb is lesz, ha beindul az UWB lapkák tömeggyártása [9].

Az Ultra Wideband egy olyan technológia, amely lehetővé teszi az információ átvitelét egy széles sáv szélességen (>500 MHz). Ez lehetővé teszi a jelentős mennyiségű jelenergia továbbítását anélkül, hogy zavaróan hatna a hagyományos szűksávú és hordozóhullám továbbításra ugyanazon frekvenciasávon belül. A szabályozási korlátok számos országban lehetővé teszik a rádiósáv szélesség hatékony kihasználását, és lehetővé teszik a nagy adatsebességű személyes hálózati (PAN) vezeték nélküli kapcsolatokat, hosszabb hatótávolságú alacsony adatsebességű alkalmazásokat, valamint a radar- és képalkotó rendszerek átlátszó együttműködését a meglévő kommunikációs rendszerekkel.

Az UWB technológia forradalmasította a valós idejű helymeghatározást a precíz és megbízható képességeivel. Az UWB centiméteres pontossága felülmúlja más helymeghatározási technológiákat, így ideális beltéri környezetekben, ahol a GPS jelek megbízhatatlanok lehetnek. Alacsony energiafogyasztása minimális interferenciát biztosít és lehetővé teszi a meglévő infrastruktúra együttműködését. Az UWB kiválóan teljesít nehéz környezetekben is, mivel ellenáll a többszörös visszaverődés okozta zavaroknak, így folyamatos és pontos helymeghatározást tesz lehetővé. Az UWB integrációja az IoT, AR és AI technológiával lehetővé teszi a kontextusérzékeny alkalmazások és személyre szabott élmények megnyitását. Az UWB technológia továbbfejlesztésével valós idejű helymeghatározásra gyakorolt hatása innovációt hoz,

lehetővé téve az iparágak számára, hogy magasabb szintű hatékonyságot, termelékenységet és biztonságot érjenek el.

Az UWB jelenleg még nem terjedt el széleskörben és még nagyon költséges a beszerzése ennek a technológiának.

1.4.2.2 Bluetooth Low Energy [11]

A Bluetooth Low Energy (Bluetooth LE, köznyelvben BLE) egy vezeték nélküli személyi hálózati technológia, amit a Bluetooth Special Interest Group tervezett és forgalmaz. Független a klasszikus Bluetooth-tól, és nincs kompatibilitása vele, de a Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR) és az LE képesek együttműködni. Az eredeti specifikációt a Nokia fejlesztette ki 2006-ban Wibree néven, amit 2009 decemberében integráltak a Bluetooth 4.0 verziójába Bluetooth Low Energy néven.

A Classic Bluetooth-hoz képest a Bluetooth Low Energy lényegesen csökkentett energiafogyasztást és költséget kínál, miközben hasonló kommunikációs távolságot tart fenn. A mobil operációs rendszerek, beleértve az iOS-t, Androidot, Windows Phone-ot és BlackBerry-t, valamint a macOS-t, Linuxot, Windows 8-at, Windows 10-et és Windows 11-et is natívan támogatják a Bluetooth Low Energy-t.

A Bluetooth Low Energy technológia ugyanabban a sávban működik (a 2,400–2,4835 GHz ISM sáv), mint a klasszikus Bluetooth technológia, de más csatornák halmazát használja. Míg a klasszikus Bluetooth 79 darab 1 MHz-es csatornát használ, addig a Bluetooth Low Energy 40 darab 2 MHz-es csatornát használ. Egy csatornán belül az adatok Gauss-frekvencia váltásos modulációval kerülnek továbbításra, hasonlóan a klasszikus Bluetooth Basic Rate sé-májához. Az adatsebesség 1 Mbit/s (2 Mbit/s opcióval a Bluetooth 5-ben), és a maximális továbbítási teljesítmény 10 mW (100 mW a Bluetooth 5-ben).

A Bluetooth Low Energy frekvenciaugrást használ a szűksávú interferencia problémák ellensúlyozására. A klasszikus Bluetooth is használ frekvenciaugrást, de a részletek mások; ennek eredményeként, míg mind az FCC, mind az ETSI az FHSS séma alá sorolja a Bluetooth technológiát, a Bluetooth Low Energy-t digitális modulációs technikák vagy közvetlen sorozatsebességű terjedésspektrum használó rendszerként sorolja.

A BLE támogatja a térerő alapú pozíció lokalizációt a Wi-Fi-hez hasonló módon. Továbbá támogatja az AoA pozícionálási eljárást, ahol az aktuális „bázis állomáson” több antenna

található és ezeken mérhető fázis különbséget mér. Ehhez a szabványt nem kell megváltoztatni, hanem csak speciális rádiós chipekre van szükség, amelyek több antennával rendelkeznek és képesek azokon fáziskülönbséget mérni.

1.4.2.3 Wi-Fi térerősség alapú helymeghatározás [4]

A külön kiépített infrastruktúrát nem igénylő beltéri helymeghatározási módszerek közül a leggyakrabban használt a Wi-Fi térerősség alapú módszer. Előnye, hogy Wi-Fi hálózat a legtöbb épületen belül, nem csak irodákban, de szinte minden háztartásban is ki van építve, hiszen a legtöbb informatikai eszköz számára alapvetően fontos a vezeték nélküli kommunikációs hálózat. Ezért, ha az infrastruktúra kiépítését, jeladók, jelölések elhelyezését és karbantartását el szeretnénk kerülni, a legkézenfekvőbb megoldást ezen vezeték nélküli kommunikációs hálózat felhasználása jelenti.

A Wi-Fi térerősség alapú helymeghatározás azon a megfigyelésen alapszik, hogy egy épületben elhelyezett Wi-Fi hozzáférési pontok (AP) rádiófrekvenciás jelének térerőssége egy helytől függő mintázatot alkot. Így egy tetszőleges hely koordinátái megállapíthatók, ha az adott helyen mért térerősség értékek összehasonlításra kerülnek már korábban mért értékekkel és a legjobb egyezőséget mutató korábbi érték mérésének helye közelében lesz a keresett pozíció.

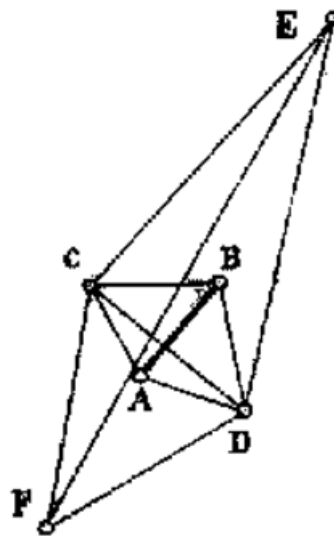
A helymeghatározás így két különálló részre bontható. Egyrészt fel kell mérni az épületen belül, a helymeghatározás szempontjából fontos helyeken, az aktuális Wi-Fi térerősség értékeket és ezeket eltárolni egy megfelelően kialakított adatbázisban. Ezt a tevékenységet még a helymeghatározás előtt el kell végezni, illetve időnként, a létrejött adatbázist frissíteni, karbantartani szükséges. A tényleges helymeghatározás ezután már megvalósítható úgy, hogy a korábbi térerősség méréseket tartalmazó adatbázisban megkeressük az aktuálisan mért térerősséggel legjobban megegyező értékeket és azok korábbi mérési pozíciója adja a keresett helyet.

A pozicionálás tehát nagymértékben a térerősség értékeken alapul, vagyis minden olyan hiba, amely a térerősség mérésekor előfordulhat, hatással van helymeghatározás pontosságára is. Ezért fontos megvizsgálni és feltárni a különböző hibaforrásokat, elemezni az egyes hibák hatását és csak ezek ismeretében valósítható meg olyan algoritmus, amely a hibák minél nagyobb részét képes szűrni, hatásukat csökkenteni.

1.4.2.4 Pozíció meghatározási módszerek: Háromszögelés [12]

Háromszögelés (lat. triangulatio), minden nagyobb területre kiterjedő mérnöki munkának első és legfontosabb mozzanata. Egyes pontoknak egymáshoz viszonyított helyzetét meghatározni a legtökéletesebb módszert a Háromszögelés adja. Azért nevezzük így, mert három pontnak egymáshoz viszonyított helyzetét azonnal ismerjük, mielőtt tudjuk, hogy mekkorák a szögei és oldalai annak a háromszögnek, amit a három pont meghatároz. És csakugyan az egész Háromszögelés nem is áll egyébből, mint ilyen háromszögek oldalhosszának és szögeinek meghatározásából. Akárhány pontot kell is felvenni, azokat, ha csak nem fekszenek egy egyenesben, mindig össze lehet kötni háromszögek bizonyos rendszerével s ha a háromszögek alkotó részeit mind ismerjük, akkor a pontoknak egymáshoz viszonyított helyzete is mind ismerős lesz. Összefüggő háromszögekben minden oldal hosszát ismerni fogjuk, amint a szögeket és egy hosszat ismerünk. Eszerint tehát az oldalak meghatározása egyetlenegy hosszának direkt megmérésére vezetődik vissza s ez az első lépés: az u. n. bázismérés.

Leggyakrabban geodéziai mérések során alkalmazzák.



7. ábra Háromszögelés [12]

A 6. ábrán látható eset számítása:

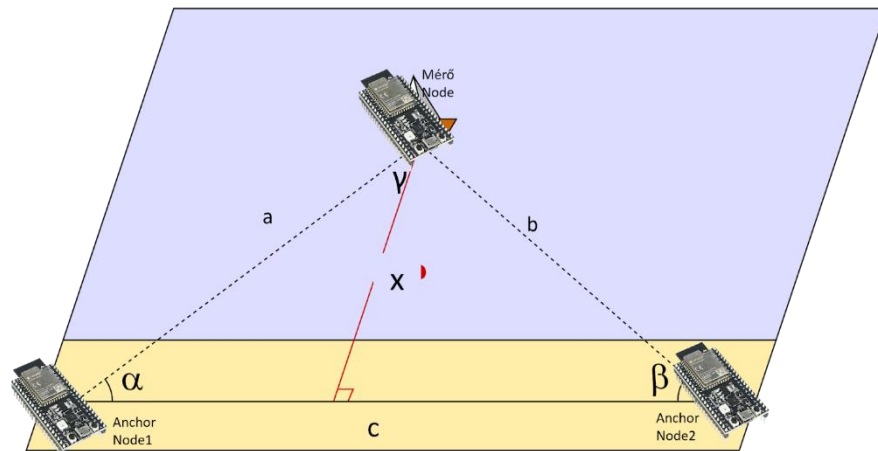
$$l = \frac{d}{\operatorname{tg} \alpha} + \frac{d}{\operatorname{tg} \beta}$$

Ebből átrendezve:

$$d = \frac{l}{\left(\frac{1}{\operatorname{tg} \alpha} + \frac{1}{\operatorname{tg} \beta}\right)}$$

Természetesen magassági mérésekre is alkalmas az itt bemutatott elveken.

Fontos felhasználása, amikor a tűzérésnek a célpontot egy megfigyelő jelöli ki [13].



8. ábra Háromszögelés a Node-okkal [13]

A GPS rendszerek elterjedése előtt az egyetlen módszer volt a földmérők, térképészek kezében. Jelentőségük a GPS terjedésével csökken, de nem szűnik meg, a geodéziában a GPS inkább az ötöd rendű ponthálózatot teszi szükségtelenné. A GPS-mérésekhez is szükség van földi alapponthálózatra, amelyek a GPS referenciapontjaiként működnek. Ezeket Magyarországon OGPSH-alappontoknak nevezik. Földi referencia nélkül a geodéziai GPS-ek sem működnek elvárható (geodéziai) pontossággal [13]. Azonban a GPS összességében máshogy oldja meg a pozíció lokalizációt.

2. Alkalmazott eszközök

A TDK mérések során az ESP32-S2 és ESP32-S3 mikrokontrollereket használtam. Ezek a mikrokontrollerek az Espressif Systems által kifejlesztett két különböző mikrovezérlő (MCU) chip, amelyek a Wi-Fi és Bluetooth kapcsolatokhoz használhatók. Ezek a chip-ek az ESP32 sorozat részét képezik, de különböző funkciókat és specifikációkat kínálnak. Léteznek ezeken alapuló modulok, amelyek rendelkeznek antennával, Flash-el stb. Továbbá léteznek ezeken a modulokon alapuló DEV kit-ek, amelyek rendelkeznek beépített programozóval és bizonyos további perifériákkal (pl. LED).

2.1 ESP32-S2

2.1.1 Leírása [7]

Az ESP32-S2 SoC sorozat egy magas integrációjú, alacsony energiafelhasználású, 2,4 GHz Wi-Fi rendszerváltozat (SoC) megoldás. state-of-the-art teljesítménye, mind az energia, mind az RF területen, ez az SoC ideális választás számos alkalmazási forgatókönyvhöz az Internet of Things (IoT), viselhető elektronikai eszközök és okos otthonok kapcsán.

Az ESP32-S2 SoC sorozat tartalmaz egy Wi-Fi alkalmazást is, amely integrál egy Wi-Fi MAC-et, Wi-Fi rádiót, basebandet, RF kapcsolót, RF balun-t, teljesítményerősítőt, alacsony zajszintű erősítőt (LNA), stb. A chip teljes mértékben megfelel az IEEE 802.11b/g/n protokollnak és teljes Wi-Fi megoldást kínál.

Ennek a chip-ek központjában egy Xtensa® 32-bites LX7 CPU található, amely akár 240 MHz-en is működhet. A chip támogatja az alkalmazásfejlesztést, anélkül, hogy szükség lenne egy gazdagép-MCU-ra.

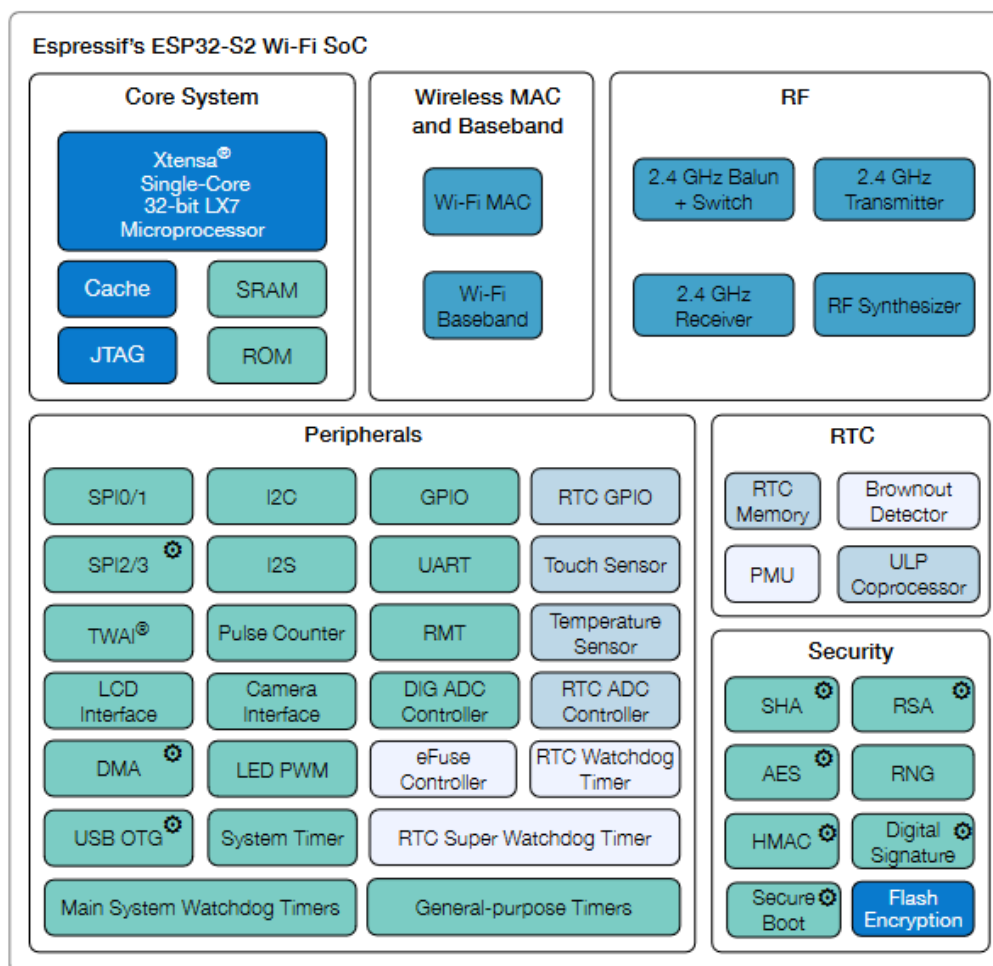
A chipen belüli memória magában foglalja a 320 KB SRAM-ot és a 128 KB ROM-ot. Támogatja több külső SPI/QSPI/OSPI flash és külső RAM chip használatát is több memóriaterület érdekében.

Az ESP32-S2 SoC sorozatot ultraalacsony energiafogyasztású alkalmazásokhoz tervezték a különböző alacsony energiafogyasztású üzemmódokkal. Az ULP mellékprocesszor ultraalacsony energiafogyasztású módban működhet. A chipben található finomított óravezérlés, dinamikus feszültség- és frekvenciaszabályozás és állítható teljesítményerősítő kimeneti teljesítménye optimális kompromisszumot teremt a kommunikációs távolság, adatsebesség és energiafogyasztás között.

A készülék gazdag perifériás interfészek széles skáláját kínálja, beleértve az SPI-t, I2S-t, UART-ot, I2C-t, LED_PWM-t, LCD interfészt, kamera interfészt, ADC-t, DAC-ot, érintésérzékelőt, hőmérsékletszenzorokat, valamint 43 GPIO-t. Emellett tartalmaz egy teljes sebességű USB On-The-Go (OTG) interfészt az USB kommunikáció lehetővé tétele érdekében.

Az ESP32-S2 SoC sorozat több dedikált hardverbiztonsági funkciót is tartalmaz. A kriptográfiai gyorsítótárazás be van építve az AES, SHA és RSA algoritmusokhoz. További hardverbiztonsági funkciókat a RNG, HMAC és digitális aláírás modulok biztosítják, valamint a Flash titkosítása és a biztonságos indítás aláírásának ellenőrzése. Ezek a funkciók lehetővé teszik, hogy a készülék megfeleljen a szigorú biztonsági követelményeknek.

2.1.2 Blokk vázlat



9. ábra ESP32-S2 funkcionális blokkdiagramja [7]

2.2 ESP32-S3

2.2.1 Leírása [8]

Az ESP32-S3 szintén egy alacsony energiafelhasználású MCU-alapú rendszerváltozat (SoC) integrált 2,4 GHz-es Wi-Fi-vel és Bluetooth® Low Energy (Bluetooth LE) technológiával. Magas teljesítményű, kétmagos mikroprocesszorból (Xtensa® 32-bites LX7), alacsony energiafogyasztású mellékprocesszorból, Wi-Fi alapbandból, Bluetooth LE alapbandból, RF modulból és számos perifériából áll. Az ESP32-S3 összeségében egy erősebb CPU-val rendelkezik, mint az S2, amely jobb teljesítményt biztosít számára bizonyos alkalmazások esetén. Továbbá ez a modul már támogatja a Bluetooth 5.0-át is.

Az ESP32-S3 kifejezetten az AIoT termékekre irányul. Ez egy továbbfejlesztett ESP32-S2, amely kétmagos, 240 MHz-ig felhúzott Xtensa 32-bites LX7 processzort használ, valamint rendelkezik AI vektor utasításokkal, valamint megbízható biztonsági titkosító motorral. A fejlesztők kihasználhatják ezeket a vektor utasításokat az ESP-DSP és ESP-NN könyvtárak segítségével az alkalmazásaik optimalizálásához. Az ESP-WHO és az ESP-Skainet SDK-k is támogatják ezt a gyorsítást. Az ESP-32-S3 sorozatú modulok képesek kezelni magas sebességű Octal SPI Flash és PSRAM-et, ami az AI alkalmazásokat sokkal simábbá teszi. Használati esetek közé tartozik:

- Offline Hangfelismerési alkalmazások
- Számítógépes látás alkalmazás
- Ébredési szó észlelési alkalmazások
- Többdimenziós Ember-gép interfész

Fontos megjegyezni, hogy az ESP32-S3 a teljesítmény és a Bluetooth verzió szempontjából erősebb, míg az ESP32-S2 egy alacsonyabb árkategóriás és energiafogyasztású.

2.3 A mikrokontrollerekhez tartozó könyvtár család

A mikrokontrollerek által használt függvények kódja nem open source, így nem látjuk, hogy pontosan mi fut le a mikrokontrolleren, amikor mi meghívunk egy függvényt. Az Espressif nem adja ki a Wi-Fi és így az FTM periféria hardver leírását, programozási felületét. Adnak egy lefordított függvény könyvtárat és az ahhoz szükséges dokumentációt és példakódot, amit be lehet építeni az alkalmazásba. A függvény könyvtárban megtalálható függvényekhez megadják a bemeneti paramétereket, hogy miket kell megadnunk a függvény lefutásához, megadják a visszatérési értékek típusát és jelentését, hogy tudjuk a későbbiekben alkalmazni és megadják a függvény rövid, egyszerű leírását, hogy mit csinál az adott függvény. A függvények leírása nem tartalmazza a függvény részletes leírását, csak a lehető legtömörebb összefoglalás, hogy az adott függvénnyel milyen műveletet lehet elvégezni. Ez nagyon kevés, de elég az alkalmazás fejlesztéshez viszont teljesen alkalmatlan az optimalizációra, hiszen nem látunk bele a részletekbe. A rendszer „működőképességét” el kell fogadnunk és csak reménykedhetünk abban, hogy a függvénykönyvtár belső implementációjába belenyúlva nem tudnánk jelentősen javítani a rendszer teljesítményén. Az FTM nem működne a megfelelő hardver nélkül, mivel a tisztán szoftveres időbélyeges pecsételést sokkal kisebb pontosságú lokalizációt tenne lehetővé (milliszekundumos nagyságrendet). Ezekből kifolyólag nem tudjuk, hogy a kód teljesen és jól kihasználja a hardver lehetőségeit. Azt tudjuk a mérésekből, hogy valószínűleg működik, mert a mért értékek hihetőek, közel vannak a működés alapján elvárthoz, de ellenőrizni nem tudjuk.

Az ilyen fajta információ visszatartás nagyban megnehezíti a munkámat, mivel nem tudom pontosan, hogy hogyan implementálták az FTM funkciót az Espressif könyvtárában. Nem látszik, hogy a kódban szerepel-e hard kódolva valamilyen konstans vagy tartalmaz-e hardver specifikus elemeket, amelyek miatt csak egy fajta hardverre alkalmazható lenne az adott függvény. További nehézség, hogy nem látszik milyen adatokat továbbítanak az FTM kommunikációban résztvevő node-ok egymásnak. Kérdés, hogy az üzenetben szerepel-e hardver specifikus adat, például az anchor node típusa. Ezen információk hiányában és a függvény „closed source” mivoltából a meglévő függvényt sem javítani, sem továbbfejleszteni nem tudom. A mérési eredmények alapján jól látható, hogy az implementációban az üzenetek hardveresen pecsételődnek, hiszen a szoftveres pecsételést sokkal nagyobb pontatlanságot eredményezne (milliszekundumos nagyságrend).

3. Mérési feladat leírása

3.1 Mérés tervezés

A TDK dolgozatom során több különböző típusú FTM mérésre alkalmas ESP32 mikrokontrollerek között végeztem FTM méréseket különböző kódok felhasználásával.

A méréseket úgy állítottam össze, hogy azokkal a mikrokontrollerek és az alkalmazott kódok és módszerek széleskörűen tesztelve legyenek. A mérések pontosságát tesztelni kell kis és nagy távolságokra, kültéren és beltéren egyaránt. A mérésekhez először csak az ESP32-S2 mikrokontroller állt rendelkezésre, így először csak ezekkel a mikrokontrollerekkel tudtam mérni. Augusztusban megjelent az Espressif-től az ESP32-S3 mikrokontroller, amely az S2-es mikrokontroller egy újabb és modernebb változata. Megjelenése után be is szereztem ebből a mikrokontrollerből és elkezdtem ugyanazokat a méréseket elvégezni, mint korábban az S2-es-sel a két különböző mikrokontrollerrel kapott eredmények összehasonlítása végett. Az eredmények összehasonlításával meggyőződhetünk, hogy az S3 valóban fejlődött-e az FTM szempontjából.

A mérések első lépése volt, amikor a szobámban a kódokat elkészítettem és feltöltöttem a mikrokontrollerekre annak érdekében, hogy tesztelni tudjam a kapcsolatot és a kommunikáció létrejöttét a két mikrokontroller között. Miután sikerült a mikrokontrollereket felkonfigurálni és a kód esetleges hibáit kijavítani, méréseket kezdtem végezni különböző távolságokra a szobán belül. Ezekkel a mérésekkel azt vizsgáltam, hogy milyen pontossággal lehet meghatározni a két node távolságát egy zárt szobán belül. A méréssel fontos tesztelendő szempont volt, hogy mennyire rontják a mérés pontosságát a falakról visszaverődő jelek és ezeket mennyire tudja kezelni a mérőmikrokontroller.

A mérések során a második lépés volt azok elvégzése szabadtéren, nagy távolságokra. Ezeknek a méréseknek a lényege az, hogy meghatározzam milyen távolságra mérnek a mikrokontrollerek a legpontosabban. A mérés lényege, hogy itt nincsenek falakról visszaverődő hullámok. Egyetlen zavaró jelként a talajról visszaverődő és tovaterjedő jel van jelen, de ez nagy távolságok esetén szinte elenyésző távolság különbséget eredményez, ha véletlenül ezt érzékelné a mérőmikrokontroller.

3.2 Méréshez használt szoftverek

Az FTM mérések implementációjára 2 különböző kóddal próbálkoztam. Mindegyik kóddal különböző távolságokra végeztem méréseket egymás után több alkalommal. A mérési eredményeket minden egyes esetben elmentettem és később feldolgoztam. A kódok közös része, hogy az egyik mikrokontrollert beállítják FTM kommunikációra alkalmas Access pointnak, a másik mikrokontrollert pedig egy FTM kommunikációra alkalmas válaszoló egységnek. Az Access pointnak nem lehetett jelszót beállítani mert a mikrokontroller nem tudta kezelni a jelszót és abban az esetben, ha a helyes jelszót másolással vittem át az egyik kódból a másik kódba, akkor is hibásnak jelezte a beírt jelszót. Azonban, ha nem állítottam be jelszót az Access pointnak akkor könnyedén tudtam hozzá csatlakoztatni a másik mikrokontrollert és tudtam méréseket végezni. A mérési eredményeket külön-külön egy-egy .txt fájlba mentettem el a CoolTerm nevű szoftver segítségével. A következő bekezdésekben ismertetem az általam alkalmazott kódokat és szoftvereket.

3.2.1 Első kód: FTM_Initiator_test arduino kód:

Ennek a kód változatnak az elkészítése során felhasználtam a Példa Library FTM_Initiator_test és FTM_Responder_test minta kódjait. Az elkészített kódomban ezeknek egy módosított kiegészített változatát implementáltam és fordítottam le a mikrokontrollerekre. Ez a változat úgy folytatja le az FTM kommunikációt, hogy meg kell adni a kódban, hogy milyen SSID-jű Access pointtal szeretnénk kommunikálni. A megkapott SSID-val rendelkező AP-ra felcsatlakozik, majd ezután elküldi a felcsatlakozott AP-nak az FTM request-et és elkezdődik a mérés. Az elkészített kódom úgy számolja a távolságot, hogy megméri a két mikrokontroller közötti késleltetési időt és ebből a terjedési sebesség ismeretében kiszámolja a két mikrokontroller közötti távolságot. A kód a kommunikáció kezdetétől a kommunikáció végéig eltelt időt úgy határozza meg, hogy az üzenet kiküldésének időpillanatát belepeccsételi egy időbélyeggel az elküldött üzenetbe, majd az üzenet visszaérkezésének időpillanatát szintén belepeccsételi az üzenetbe annak megérkezésekor. Az eltelt időt az üzenet feldolgozásakor az üzenetben szereplő két időbélyeg különbségét veszi eltelt időnek. Ezt az időt elosztja kettővel a kód és megszorozza a terjedési sebességgel, ami jelen esetben a fénysebesség. Az ilyen módon kiszámított értéket jelzi ki végül a kód a soros monitoron. A kódon belül lehet állítani az elküldött frame-ek és a burst periódusok számát, továbbá természetesen meg lehet adni annak az Access point SSID-ját, amivel az FTM kommunikációt szeretnénk folytatni. A méréseket többféle frame és burst

periódus számmal teszteltem és az eredmények alapján azt tapasztaltam, hogy a mérések akkor a legpontosabbak, ha a frame-ek számát 32-re, a burst periódusok számát pedig 4-re állítom be.

```
14:46:39.311 -> Connecting to FTM Responder
14:46:39.872 -> .
14:46:39.872 -> WiFi Connected
14:46:39.872 -> Initiating FTM session with Frame Count 16 and Burst Period 200 ms
14:46:40.152 -> 0
14:46:40.152 -> 0
14:46:40.152 -> FTM Estimate: Distance: 7.93 m, Return Time: 52 ns
14:46:40.478 -> 0
14:46:40.478 -> 0
14:46:40.478 -> FTM Estimate: Distance: 2.86 m, Return Time: 19 ns
14:46:40.757 -> 0
14:46:40.757 -> 0
14:46:40.757 -> FTM Estimate: Distance: 5.33 m, Return Time: 35 ns
14:46:41.085 -> 0
14:46:41.085 -> 0
14:46:41.085 -> FTM Estimate: Distance: 5.33 m, Return Time: 35 ns
14:46:41.364 -> 0
14:46:41.364 -> 0
14:46:41.364 -> FTM Estimate: Distance: 4.94 m, Return Time: 32 ns
14:46:41.689 -> 0
14:46:41.689 -> 0
14:46:41.689 -> FTM Estimate: Distance: 7.93 m, Return Time: 52 ns
14:46:41.969 -> 0
14:46:41.969 -> 0
```

10. ábra A sorosmonitoron megjelenő adatok az első kód futása közben

3.2.2 Második kód: wifi_ftm_test2 arduino kód:

Ez a kód változat több funkciót valósít meg egyben. Ebben a kódban az Access pontokkal való kommunikációt teljesen új irányból közelítettem meg, ebben az esetben a mérő mikrokontrollernek nem kell felcsatlakoznia az AP-kra, ahhoz, hogy FTM kommunikációt tudjanak folytatni, hanem az arra alkalmas eszközökkel anélkül is tud kommunikációt végrehajtani. A mérő mikrokontroller először szkenneli, hogy milyen Wi-Fi Access pointokat lát elérhető távolságban. Miután ezeket megtalálta, megvizsgálja, hogy melyek alkalmasak FTM kommunikációra. Az FTM kommunikációra alkalmas Wi-Fi Access pontokkal FTM kommunikációt kezdeményez és a kommunikációból a korábban leírt módon meghatározza a távolságát az Access ponttól. Miután ezt minden erre alkalmas Access ponttal elvégezte, akkor szkenneléssel újra kezdi a folyamatot. A szkennelés után kiírja a soros monitorra az összes megtalált Access point SSID-ját, a csatorna számát, amin kommunikál, az RSSI értéket, hogy mekkora a csillapítás és a MAC címét az AP-nak.

Ezt a kódot tovább fejlesztettem, hogy a vegyes rendszereket is kezelni tudja, ahol több különböző típusú mikrokontroller található. A vegyes rendszerek jelenlegi legnagyobb problémája, hogy nem tudunk belenyúlni, hogy miket küldjenek az üzenetekben egymásnak a mikrokontrollerek az FTM kommunikáció során. A különböző típusú mikrokontrollereknek különböző késleltetése és ofszetje van, amit a mérések során kompenzálni kell. Ennek kiküszöbölésére jelenleg az SSID-ba kódoltan jelzem a mérő mikrokontrollernek, hogy milyen típusú az adott AP. Az SSID vizsgálatának eredményének megfelelően kompenzálja ki a mérést a kód. A későbbiekben egy webszerveren tárolt adatbázisból lehetne ezeket az adatokat lekérdezni és nem a programban hard kódolva megoldva.

A kódot kiegészítettem az adatmentéshez szükséges részekkel. Sajnos az interneten utána kerestem és megismertem adatmentési és fájl írási eljárások az Arduino nyelvben nekem nem működtek többszöri próbálkozásra sem, mert vagy nem bírta kiírni az adott fájlba a program az eredményeket, vagy ha ő hozott létre magának egy fájlt, ahova kiírta az eredményeket, azt én nem találtam meg a számítógépen, még a rejtett fájlok között sem. Ezek következtében alternatív megoldás után néztem, mely során a CoolTerm nevű szoftvert találtam és a későbbiekben ennek a segítségével mentettem el a mérések során keletkező adatokat a későbbi feldolgozásra.

```

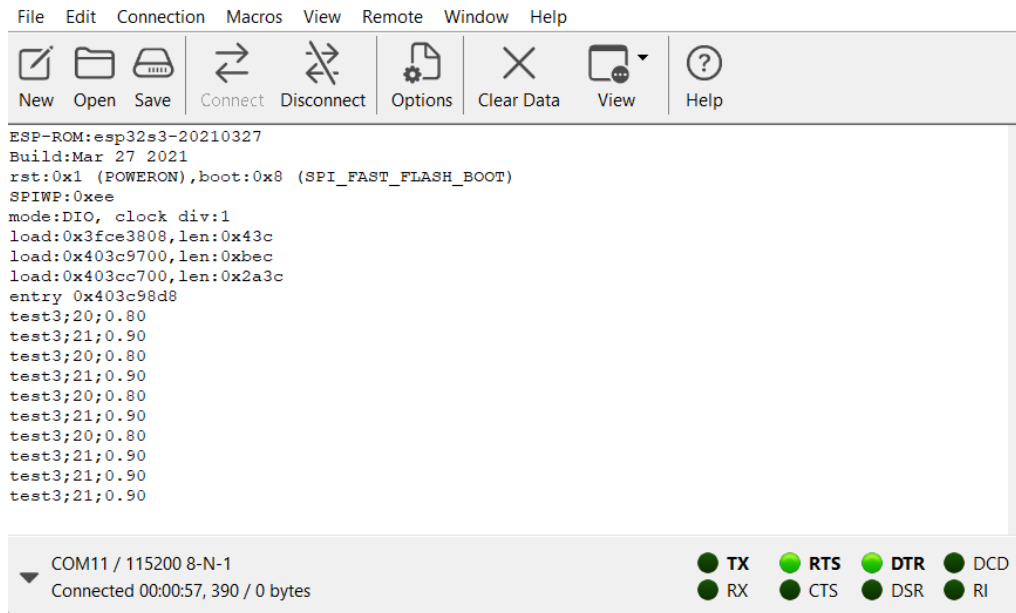
14:42:45.488 -> Scanning
14:42:48.017 ->
14:42:48.017 -> *****
14:42:48.017 -> test2 | 1 | -45 | 1 | 7C:DF:A1:1:E2:5D
14:42:48.017 -> Inside FTM req
14:42:48.017 -> Requesting FTM session with Frm Count - 32, Burst Period - 400mSec
14:42:48.017 -> 7C:DF:A1:1:E2:5D
14:42:48.017 -> ftm req sent
14:42:50.915 -> Estimated RTT - 32 nSec, Estimated Distance - 4.81 meters
14:42:50.915 -> test1 | 1 | -50 | 1 | 7C:DF:A1:1:F3:BF
14:42:50.962 -> Inside FTM req
14:42:50.962 -> Requesting FTM session with Frm Count - 32, Burst Period - 400mSec
14:42:50.962 -> 7C:DF:A1:1:F3:BF
14:42:50.962 -> ftm req sent
14:42:53.856 -> Estimated RTT - 21 nSec, Estimated Distance - 3.25 meters
14:42:53.856 -> Cudy-0708 | 8 | -51 | 0 | F4:A4:54:83:7:8
14:42:53.856 -> 1615-Juhuu | 2 | -61 | 0 | 80:1F:2:E6:36:B6
14:42:53.856 -> 1713-AN-225 | 9 | -61 | 0 | E8:DE:27:D5:B2:D2
14:42:53.856 -> 1713-LegjobbRouter | 1 | -65 | 0 | B0:BE:76:2B:89:59
14:42:53.856 -> 1716-SmukiNat | 4 | -65 | 0 | C0:A5:DD:8:91:2C
14:42:53.856 -> SCH-Test | 11 | -67 | 0 | 54:8A:BA:6F:BC:41
14:42:53.856 -> KSZK-MGMT | 11 | -67 | 0 | 54:8A:BA:6F:BC:46
14:42:53.903 -> SCH-Test2 | 11 | -67 | 0 | 54:8A:BA:6F:BC:4B
14:42:53.903 -> 1314_szobapizza | 7 | -80 | 0 | C0:C9:E3:E1:FE:76
14:42:53.903 -> 1816-ZambOS_v1.0 | 11 | -80 | 0 | 14:D6:4D:DF:CA:D6
14:42:53.903 -> Vodafone-70C0 | 11 | -86 | 0 | AC:22:5:86:4A:2B
14:42:53.903 -> *****
14:42:53.903 -> 2 FTM SSID found

```

11. ábra A soros monitoron megjelenő adatok a második kód futása során

3.2.3 CoolTerm szoftver

Ez egy olyan letölthető program mellyel a soros portokat lehet monitorozni, rácsatlakozni, adatokat küldeni és fogadni továbbá természetesen a soros portokon megjelenő adatokat elmenteni egy .txt fájlba. A program rendkívül egyszerűen alkalmazható és nagyon letisztult kezelő felülettel rendelkezik.

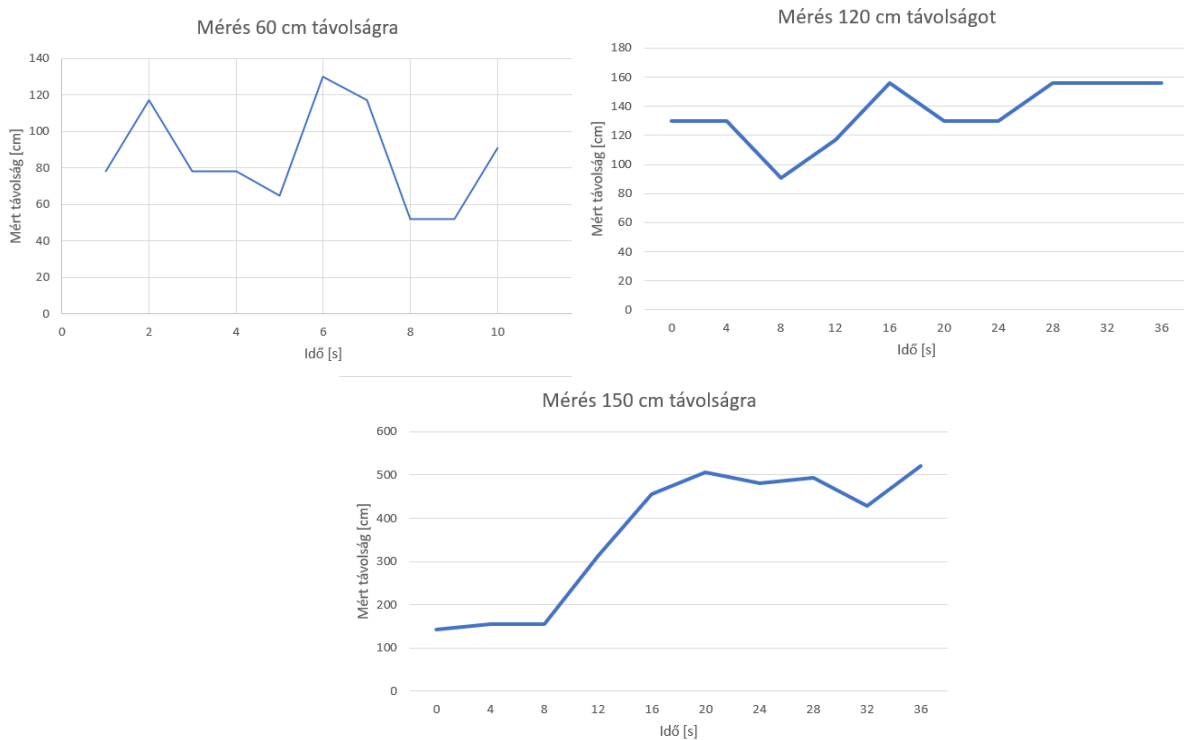


12. ábra CoolTerm szoftver működés közben

3.3 Mérések és eredményeik az ismertett kódok felhasználásával

3.3.1 Mérések kis távolságokra

Méréseket végeztem szobán belül kis távolságokra és a mérési eredményeket feldolgoztam, diagramokat készítettem belőlük a mért adatok vizualizálása végett.



13. ábra Mérési eredmények szobán belül

Az ESP32-S2 mikrokontrollerek a 30 cm-t nem tudják érdemben kimérni mert, túl közel vannak a mikrokontrollerek egymáshoz. A mikrokontrollerek elektromágneses hullámokkal kommunikálnak, amik fénysebességgel terjednek, ami $v_t=3 \cdot 10^8$ m/s. Ez a terjedési sebesség $\lambda=13$ cm körül hullámhosszt eredményez. A mikrokontrollerek időmérési pontossága nanoszekundum pontosságú, ami a távolság mérés szempontjából 30 cm körüli érték, ebből következik, hogy ki tudja-e mérni a 30 cm távolságot vagy nem. Abban az esetben, ha rosszkor fogja a jelet akkor nem tudja kimérni.

Ha a két mikrokontroller között van a távolság kimérésére használt fém mérőszalag, akkor az nagyban befolyásolja a mérés eredményét. Sokkal megnöveli a mért késleltetési időt

és az ebből számított távolságot. Amikor kivettük közülük a mérés kicsit pontosabb lett, de nem sokkal. A mikrokontroller a valós értéknél jóval nagyobb késleltetést mért és ebből jóval nagyobb távolságot számított, mint a valóságos.

60 centiméteres mérésnél, ha csak magukban voltak a mikrokontrollerek 2 és 3 métert mért, ha egy ember a közelükben állt, akkor 50 és 117 cm között mért. Ha a földön, ha az asztalon mértünk mindig nagyobb távolságot mért, mint a valóságos.

Nem stabil a kommunikációjuk, ha jól kapja el a jelet akkor egészen közeli értékeket mér, de sokszor instabil a jel és akkor nyilvánvalóan helytelen értéket mér. A mérés során voltak periódusok amikor egészen pontosan mért, volt, amikor sokkal nagyobb értéket mért és volt, amikor sokkal kisebb értéket mért.

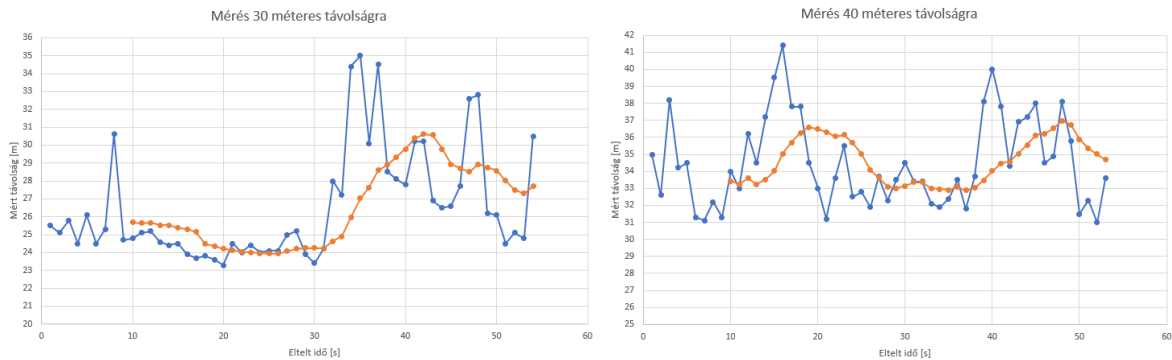
A 150 cm-es mérésnél szintén sokkal nagyobb értékeket mért a mikrokontroller.

A mérést olyan szobában végeztem, ahol elektromos padlófűtés van, nem tudom ez zavarja-e a mérést.

3.3.2 Mérések nagy távolságokra, nyílt téren

A méréseket elvégeztem szabadterben nagy, több tíz méteres távolságokra és a mérési eredményeket feldolgoztam, diagramokat készítettem belőlük a mért adatok vizualizálása végett.





14. ábra Szabadtéri mérések eredményei

Az ESP32-S2 mikrokontrollerrel történő mérések nagyobb távolságokra arányaiban sokkal pontosabbak, de van a mérésnek egy konstans eltérése a valós értéktől. Ez az eltérés adódhat az ofszet rossz meghatározásából vagy a műszer mérési pontatlanságából. A mérés 15 métertől kezdve kezd pontosabb és stabilabb lenni, ennél nagyobb távolságokra a mérés és a két mikrokontroller közötti kapcsolat stabil volt és rengeteg mérést tudtunk végezni rst vagy egyéb külső beavatkozás nélkül és nem állt le a kapcsolat.

Van egy konstans 5 méter körüli hiba, de a mérési eredmények mindig arányosan nőnek a mérendő távolsággal. A mérés előtt elvégeztem a mikrokontrollerek ofszet mérését és ez alapján kinulláztam a mikrokontrollerek mérését, de a tapasztalatok azt mutatták ez nem volt megfelelő. A mérési távolságok növelésével arányosan egyre kisebb távolságokat mért a ténylegeshez képest a mikrokontroller. Szerintem ez azért lehetséges mert a nullatávolságnál mért ofszetben még sokkal nagyobb arányban szerepel az adatfeldolgozás hibája/késleltetése, mint nagy távolságoknál. Nagyobb távolságoknál a terjedési idő sokkal számottevőbb, mint kis távolságoknál és az adatfeldolgozás ideje is sokkal kisebb a mérés teljes időtartamához képest.

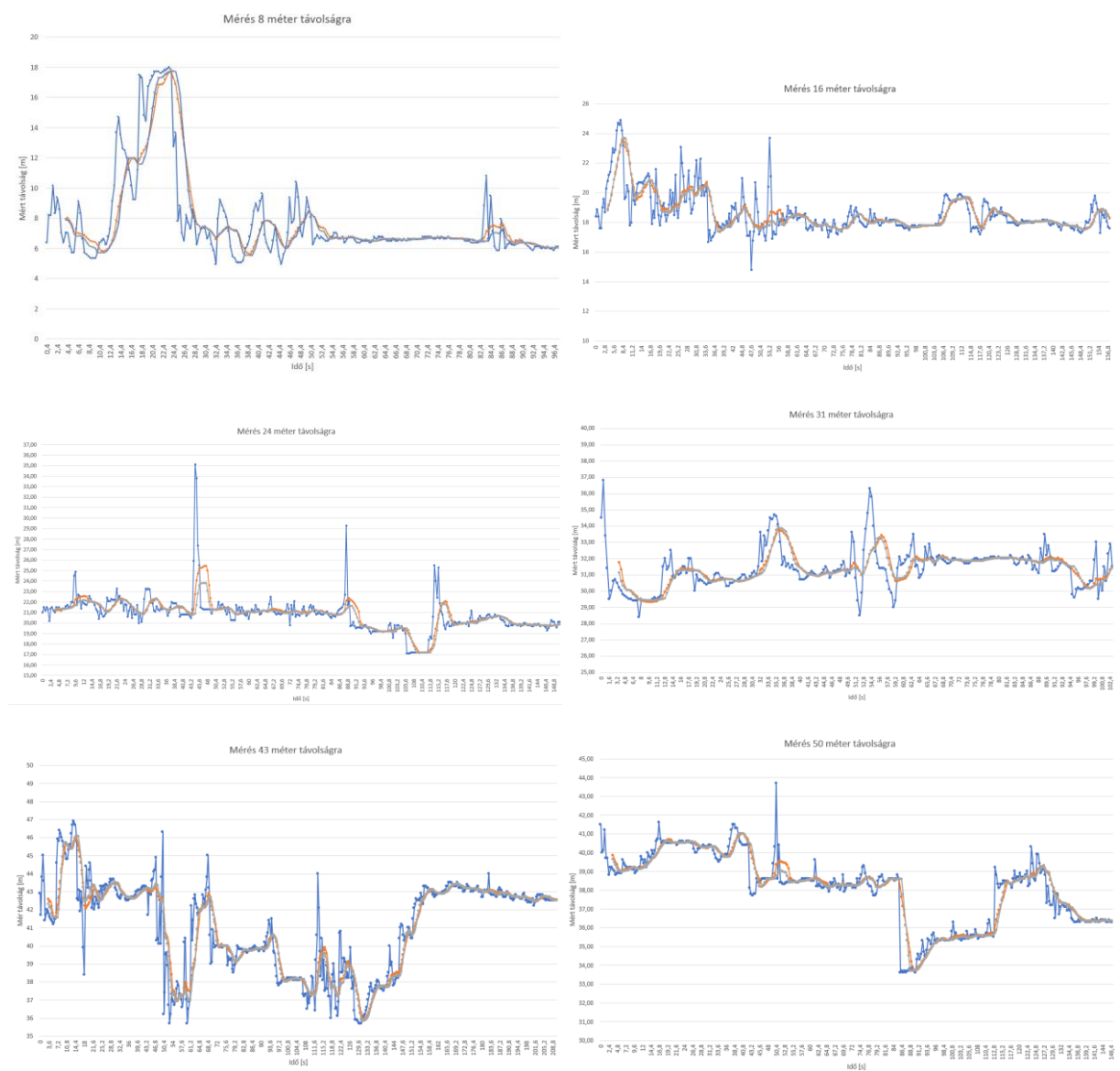
3.4 Mérések az egyetem folyosóján

A méréseket elvégeztem az egyetem épületének folyosóján is különböző távolságokra. A folyosó, mint egy nagy cső funkcionált, ami vezette a jelet, itt sokkal kevesebb visszaverődő jel volt jelen, ami zavarhatta a mérést. A mérési eredmények mentését nem sikerült megoldani kóddal ennek következtében használtam a CoolTerm64Bit nevű alkalmazást, amely lényegében soros monitorként funkcionál és minden adatot, ami a soros monitoron megjelenik elmenti az előre megadott fájlba. A méréseket úgy végeztem, hogy a mérő mikrokontroller a számítógéppel együtt fixen maradt a folyosó egyik végén, míg az Access point mikrokontrollert egy külső

akkumulátorra csatlakoztattam és elhelyeztem a folyosón különböző távolságokban. A mérési eredményeket feldolgoztam, diagramokat készítettem belőlük a mért adatok vizualizálása végett.

A méréseket az egyetem folyosóján 8, 16, 24, 31, 43 és 50 méteres távolságra végeztem el. A folyosó korlátai végett nem tudtam ezeknél nagyobb távolságokra mérni, továbbá a nyílt téren végzett mérések eredményei alapján ennél nagyobb távolságra nem is lett volna értelme mérni.

A mérési eredményeket ábrázoló diagramok:



15. ábra Mérési eredmények az egyetem folyosójáról

Pontozott vonaldiagram a mérési eredményekről (kék), a mérési eredmények tízes mozgóátlagáról (narancssárga) és a mérési eredmények tízes mozgóátlaga hibatűrő átlagolással (szürke), ahol a 2 legnagyobb és 2 legkisebb értéket kivesszük az átlagolásból.

A mérési eredményeknek kiszámoltam az olyan értékeit, amik szemléletes információkat hordoznak számunkra.

Tényleges távolság	8 m	16 m	24 m	31 m	43 m	50 m
Átlag	7,80 m	18,62 m	20,82 m	31,40 m	41,03 m	38,12 m
Eltérés	-0,20 m	2,62 m	-3,18 m	0,40 m	-1,97 m	-11,88 m
Relatív eltérés	0,0244	0,1641	0,1325	0,0129	0,0459	0,2376
Medián	6,63 m	18,10 m	20,80 m	31,40 m	42,10 m	38,50 m
Módusz	6,63 m	17,8 m	21,3 m	32 m	42,8 m	38,6 m
Szórás	2,969 m	1,377 m	1,735 m	1,243 m	2,550 m	1,889 m
Relatív szórás	0,380	0,074	0,083	0,040	0,0621	0,050

A kiszámolt értékekből látszik, hogy a 8 méterre történő mérésnek nagyon nagy szórása van, ami bizonytalanná teszi a helyzet meghatározást kis távolságon. A számolt eredményekből látszik, hogy a mérésnek körülbelül azonos a bizonytalansága plusz és mínusz irányba is mert az átlag érték nagyon közel esik a tényleges távolsághoz. Az átlag és a tényleges érték különbsége szinte a felbontás nagysága, amely attól függhet, hogy éppen hogyan kapja el az érkező hullámot a vevő. A pont diagramból pedig az látszik, hogy mínusz irányba gyakrabban téved kisebb hibával, míg plusz irányba kevesebbszer téved nagyobb hibával.

A 16 méter távolságra kiszámolt értékekből látszik, hogy bár ez a távolság is kis távolságnak minősül Wi-Fi jellel történő távolság mérés esetén, de ekkora távolságnál is már jelentősen csökkent a mért értékek szórása és relatív szórása. Az átlag ebben az esetben nagyobb, mint a tényleges érték ez az ofszet hibás méréséből következhet. A pontfelhőből leolvasható, hogy a mikrokontroller kezdetben nagyon nagy szórással mérte a távolságot, de egy idő után a mérési eredmények kis szórással beálltak a 18 méter köré.

A 16 méter méréséhez képest a 24 méter távolságra történő mérések átlagának eltérése a tényleges értéktől nagyobb, de a relatív eltérés kisebb ebben az esetben. A mérési eredmények átlaga ebben az esetben kisebb, mint a tényleges távolság a két mikrokontroller között. A szórás ebben az esetben nagyobb, mint a 16 méter mérésének esetében. A pontfelhőből látszik, hogy a mért eredmények viszonylag kis szórással a 20 és 22 méter között helyezkedik el, míg ritkán előfordulnak nagyon nagy eltérésű értékek.

A 31 méter távolságra történő mérések esetén a mért távolságok tényleges távolságtól való eltérése a mért értékek átlagának a felbontással egy nagyságrendű. A relatív szórás is nagyon kicsi érték lett. Megfigyelhetőek kiugrások a mérésekben, amik mérési hibák lehetnek, a mérések jelentős része 31 és 32 méter közé konvergál.

A mérés 43 méterre történő távolságon már jól elvégezhető és az eredmények azt mutatják elég pontos is, de a szórás nagy. A nagy szórás azért is lehet, hogy az Access point mikrokontroller már a folyosó végén helyezkedett el és ott a folyosó végi visszaverődő vagy szóródó jelek zavarhatták a mérést. A mért eredmények átlaga ekkora távolságra relatívan jó eredményt ad.

Az 50 méter távolságra történő mérés esetén az Access point mikrokontroller már a folyosón kívül, a folyosó utáni várakozó térben helyezkedett el. Ez nagyban befolyásolhatta a mérés kimenetét mert ott már nem volt jelen a folyosóban tapasztalható csőhatás és a cső végénél a jelek máshogy viselkedhetnek, mint magában a csőben. Ezt támassza alá, hogy a mérés nagyon pontatlan lett. A mérési eredmények átlaga a tényleges távolságtól nagyon eltér (11,88 m). Ettől függetlenül a mért értékek szórása nem nagy, mindössze 1,889 m, ami 50 méteren jó eredmény. A mért értékek mediánja 38,5 méter, ami nagyon távol van az 50 métertől az előző mérésekben tapasztaltakhoz képest.

Méréseket nagyobb távolságokra nem tudtam végezni a folyosó végessége miatt, így is a folyosó végi tér zavaró hatással volt a mérési eredményekre.

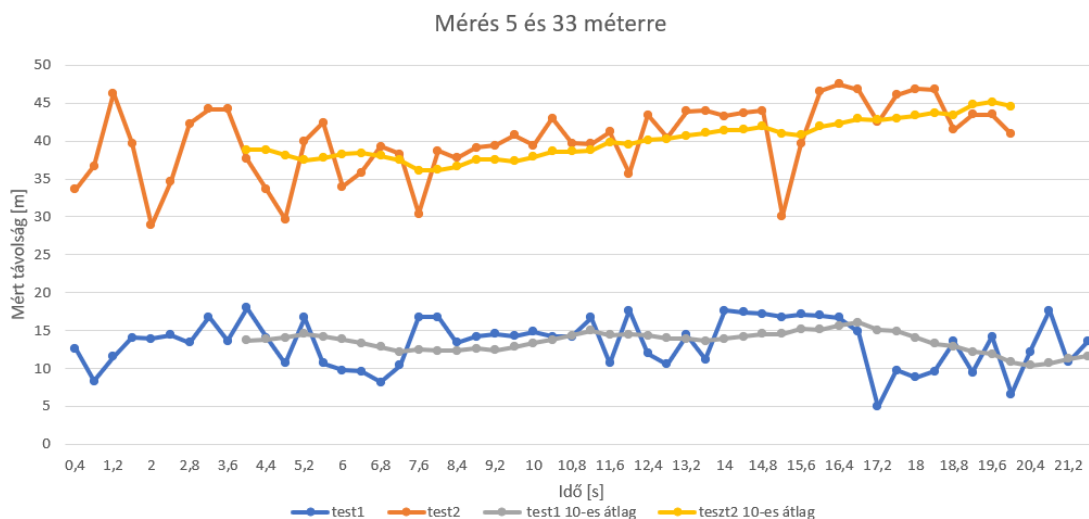
3.5 Két node-os mérések eredményeinek kiértékelése

A mérést elvégeztem oly módon is, hogy két darab Access point node-ot és egy darab mérő mikrokontrollert használtam. A két Access point node-ot egy-egy külső akkumulátorhoz csatlakoztattam és elhelyeztem őket a folyosón fix távolságra. A mérő mikrokontrollert számítógéphez csatlakoztattam és elhelyeztem egy gurulós széken a két Access point között. A mérések során csak ennek a mikrokontrollernek a helyét változtattam a két Access point között.

Az Access point mikrokontrollerek fix távolságát 38 méterre állítottam be.

3.5.1 Mérés az egyik végponttól 5 méterre a másik végponttól 33 méterre

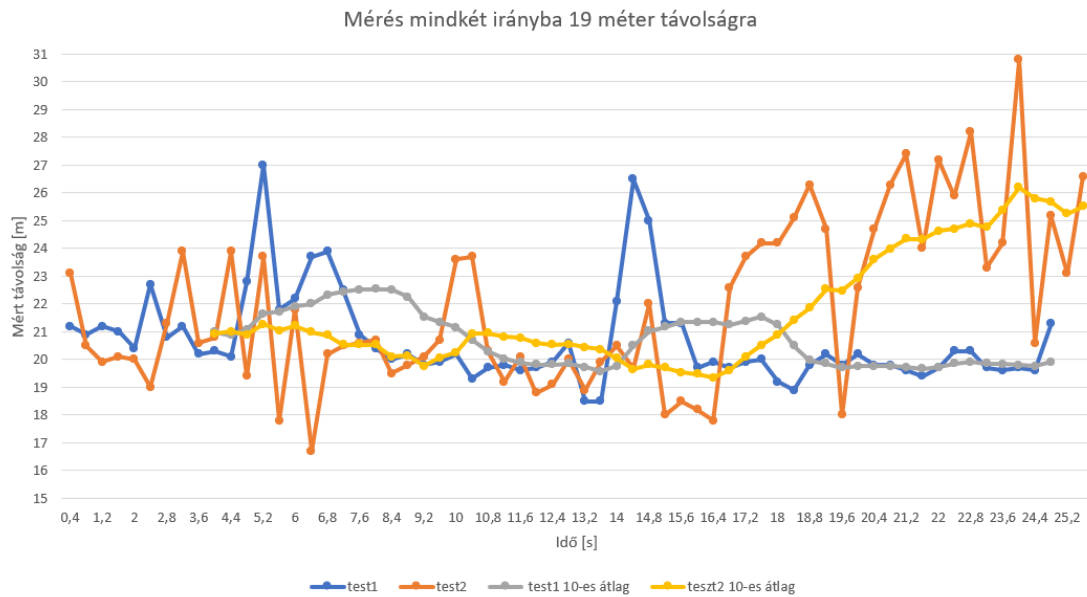
Első alkalommal az Access point mikrokontrollereket 38 méter távolságra helyeztem el egymástól a folyosón. A mérő mikrokontrollert az egyik végponttól 5 méterre a másik végponttól 33 méterre helyeztem. Az 5 méter Wi-Fi jellel még nem nagy pontossággal biztosan mérhető távolság. A 33 méter a Wi-Fi jellel való mérés szempontjából már jól mérhető távolságnak minősül.



16. ábra Két AP mérés 5 és 33 méterre

3.5.2 Mérés mindkét végponttól 19 méterre

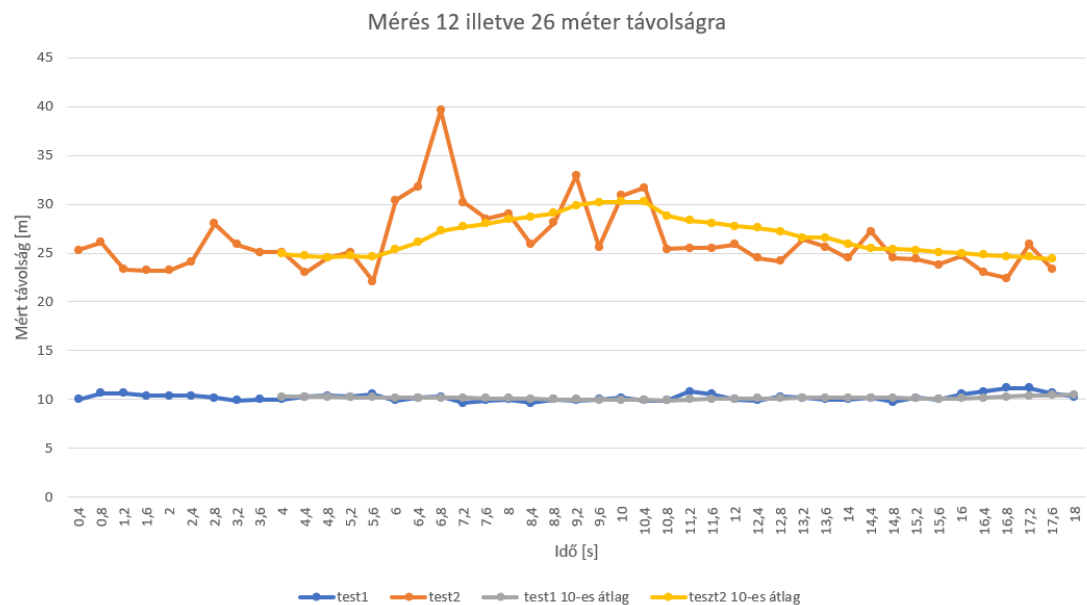
Második alkalommal az Access point mikrokontrollereket 38 méter távolságra helyeztem el egymástól a folyosón. A mérő mikrokontrollert mindkét végponttól 19 méterre helyeztem el. A 19 méter a Wi-Fi jellel való mérés szempontjából már jól mérhető távolságnak minősül.



17. ábra Két AP mérés mindkét irányba 19 méterre

3.5.3 Mérés az egyik végponttól 12 méterre a másik végponttól 26 méterre

Harmadik alkalommal az Access point mikrokontrollereket 38 méter távolságra helyeztem el egymástól a folyosón. A mérő mikrokontrollert az egyik végponttól 12 méterre a másik végponttól 26 méterre helyeztem. A 12 méter Wi-Fi jellel még nem elég nagy pontossággal biztosan mérhető távolság. A 26 méter a Wi-Fi jellel való mérés szempontjából már jól mérhető távolságnak minősül.



18. ábra Két AP mérés 12 és 26 méterre

A mérési eredményeknek kiszámoltam az olyan értékeit, amik szemléletes információkat hordoznak számunkra

	Első mérés		Második mérés		Harmadik mérés	
	test1	test2	test1	test2	test1	test2
Tényleges távolság	5 m	33 m	19 m	19 m	12 m	26 m
Átlag	13,31 m	40,22 m	20,70 m	21,90 m	10,21 m	26,26 m
Eltérés	8,31 m	7,22 m	1,70 m	2,90 m	-1,79 m	0,26 m
Relatív eltérés	1,66	0,22	0,09	0,15	-0,15	0,01
Medián	13,98 m	40,6 m	20,20 m	20,7 m	10,14 m	25,45 m
Módusz	16,8 m	33,6 m	19,7 m	20,5 m	10,01 m	25,9 m
Szórás	3,20 m	4,78 m	1,68 m	2,99 m	0,36 m	3,37 m
Relatív szórás	0,64	0,14	0,09	0,16	0,03	0,13

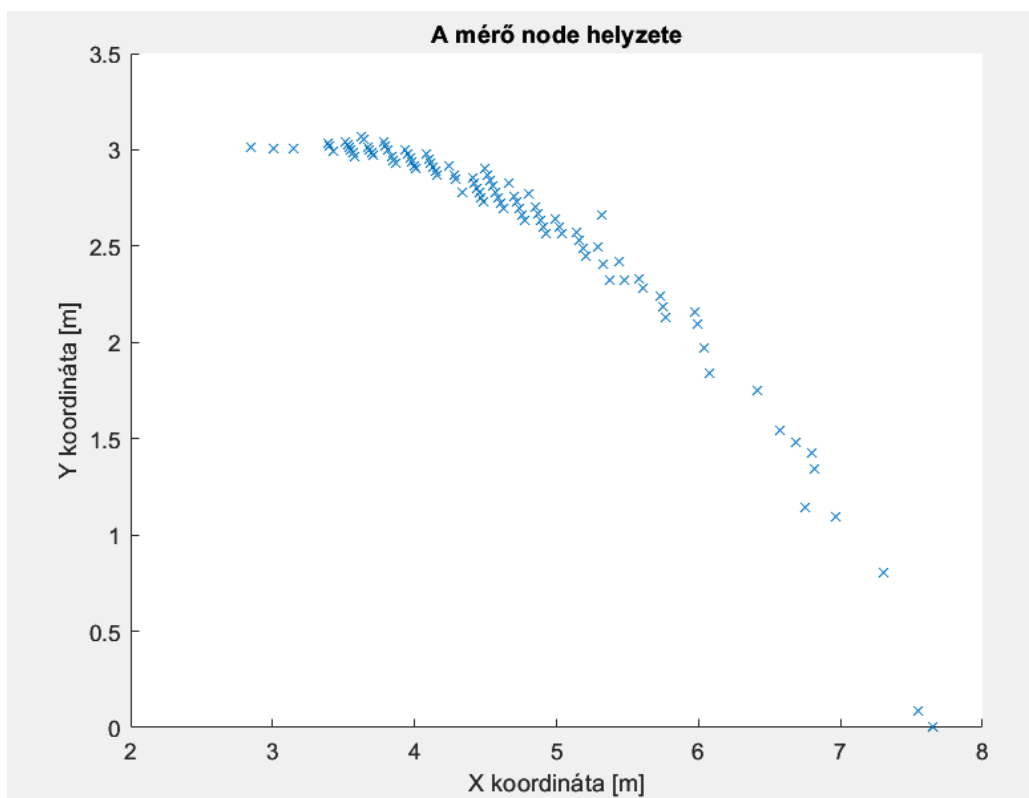
Az 5 méter és 33 méter távolságos mérés során jól látszik, hogy az egy Access point-os mérésnél meghatározott távolságon belül nem mér jól az eszköz. A 33 méteres távolság mérésénél meglepően nagy az eltérés az egy mikrokontrolleres mérés hasonló távolságához képest. Ennek oka lehet, hogy a folyosó véges mivolta miatt az Access point node-ok túl közel kerültek a folyosó végéhez, ahol a már szóródó jelek mérési hibát okozhattak.

A második mérés esetén számított eredményeken jól látszik, hogy 19 méter távolságra már egészen jó pontossággal tud mérni az eszköz. A mérési eredményekből jól látszik, hogy a két Access point mikrokontroller mérési pontosságában is van különbség, a test2-es mikrokontroller sokkal pontatlanabb, mint a test1-es.

A harmadik mérésnél is jól látszik, hogy a test2 node mérési pontossága kisebb a test1 node-hoz képest. A 12 méter távolság már mérhető távolság ezzel a módszerrel viszonylag kis eltéréssel. A 26 méter távolság ezzel a módszerrel már jól mérhető távolság.

3.6 Mérés 2 dimenzióban

A méréseim gyakorlati alkalmazására megterveztem egy kisebb rendszert mellyel 2 dimenzióban tudtam méréseket végezni. Első implementációban 3 node-ot alkalmaztam, melyek közül 1 mérő node és kettő Access point volt. Ebben az esetben ismert volt az Access point node-ok távolsága. Mérő node megmérte, hogy milyen távol van mind kettő Access point-tól, majd ezeket az értékeket elmentettem a CoolTerm szoftverrel a korábban ismertetett módon. A .txt fájlba elmentett értékeket egy erre a célra fejlesztett matlab kóddal beolvasom és feldolgozom. A mérő node pontos helyét koszinusztétellel és trigonometriával határozom meg. A kiszámított értékeket egy mátrixban tárolom el. Ezt a mátrixot egy diagrammon ábrázoltam lásd 19. ábra. A mérési eredmények feldolgozása a korábban említett matlab kóddal, majd a kiszámított értékek ábrázolása után látható a diagrammon, hogy a kapott értékek elég nagy szórásúak lettek. Az értékek túlnyomó része egymáshoz viszonylag közel egy Hold sarlót formálnak. Az értékek kiszámításánál csak a mérő node két Access point node-tól mért távolságát és a két Access point node távolságát használtam fel. A mérő node és az Access point node-ok a mérés során végig egyhelyben voltak. A szórást az FTM mérések pontatlansága és szórása okozza, mivel még a két node közötti távolságmérés nem tökéletesen megbízható.



19. ábra 2 dimenziós pozíció mérés eredménye

3.7 Mérés 3 dimenzióban

A méréseim gyakorlati alkalmazására tervezek mérést elvégezni egy kisebb rendszerrel, mellyel a pozíció meghatározást 3 dimenzióban tudom tesztelni. Ezt a mérést még nem tudtam elvégezni TDK dolgozat leadásáig. A mérést az egyetem egyik laborjában tervezem elvégezni, ahol vegyes rendszert szeretnék alkalmazni. A rendszerben legalább 5 darab Access point mikrokontrollert szeretnék használni és 1 vagy 2 mérő mikrokontrollert, hogy meg tudjam vizsgálni milyen az egyes mikrokontrollerekkel végzett pozíció meghatározás pontossága. További hasznos tanulságok, amiket le tudok majd vonni a mérésekből, hogy mennyire kompatibilisek egymással a különböző mikrokontrollerek és jó modell lehet egy későbbi valódi rendszerre.

3.8 Összefoglalás és jövőbeli tervek

A TDK dolgozatomban az első nagyobb rész fejezeteiben bemutattam a jelenleg elérhető legrelevánsabb pozíció lokalizációs technológiáit és a Wi-Fi Fine Time Measurement elméleti hátterét. Ezen technológiák felhasználásával a jövőben kellő pontosság elérése után egy költséghatékony és egyszerűen létrehozható egy beltéri pozíció lokalizációs rendszer.

A következő nagyobb logikai részben bemutattam az általam használt eszközöket, szoftvereket és implementációkat, majd értékeltem ezeket. Ezt követte a felhasznált eszközökkel és szoftverekkel végzett mérések bemutatása és a mérési eredmények kiértékelése. A méréseket logikai és időrendi sorrendben mutattam be.

A TDK dolgozatom elkészítése során elvégzett mérések eredményeiből jól látható, hogy a mérések pontossága még közel sem kielégítő. A mérések felbontása a Wi-Fi hullámhosszából kifolyólag nem megfelelő jelenleg ipari felhasználásra, mert esetleges robotok mozgása során nem tudunk nagy pontosságú pozícionálást végezni. Személyek számára a GPS-hez hasonló beltéri lokalizációs alkalmazást tudunk ennek felhasználásával létrehozni. A mérések javításával és a szórás csökkentésével jobb helymeghatározást tudunk elérni, de jelen formájában nem megfelelő a pontosság még ehhez sem.

A jövőben fontos lesz az FTM mérések pontosabb implementációjának elkészítése és újabb, a mérések elvégzésére alkalmas mikrokontroller beszerzése. Ezekkel az új implementációkkal és hardverekkel méréseket végezni és a mérési eredményeket kiértékelni, hogy még pontosabb méréseket tudjak végezni.

4. Hivatkozások

- [1] BELTÉRI LOKALIZÁCIÓS ALKALMAZÁSOK KOMMUNIKÁCIÓS ÉS ADATFELDOLGOZÁSI ARCHITEKTÚRÁJA Budapest 2018
<http://tdk.bme.hu/VIK/DownloadPaper/Belteri-lokalizacios-rendszerek-kommunikacios>
- [2] Tivadar Neuwald: Beltéri Pozicionálás, Helymeghatározás és Navigáció, LinkedIn 2023 október
<https://www.linkedin.com/pulse/belt%C3%A9ri-poz%C3%ADcion%C3%A1l%C3%A1s-helymeghat%C3%A1roz%C3%A1s-%C3%A9s-navig%C3%A1ci%C3%B3-tivadar-neuwald>
- [3] Árvai László: GPS VÉTEL MENTES KÖRNYEZETBEN IS HASZNÁLHATÓ HELYMEGHATÁROZÁSI MÓDSZEREK ÉS ALKALMAZÁSI LEHETŐSÉGEIK VIZSGÁLATA - Ph.D. értekezés, Miskolc 2022
https://geik.uni-miskolc.hu/intezetek/MAT/admin/news_files/11_0.pdf
- [4] Barát Dániel, Danyi Ádám Vilmos, Lengyel Dániel József: Felhasználók pozicionálása 4. generációs hálózatokban, Budapest 2015
<https://tdk.bme.hu/VIK/DownloadPaper/Felhasznalok-pozicionalasa-4-generacios-mobil>
- [5] AirGo indoor positioning solutions Fingerprinting, 2023 október
<http://air-go.es/technology/fingerprinting/>
- [6] Sami Huilla: Smartphone-based Indoor Positioning Using Wi-Fi Fine Timing Measurement Protocol, 2019
<https://core.ac.uk/download/pdf/233003516.pdf>
- [7] ESP32-S2 Series Datasheet, 2023
https://www.espressif.com/sites/default/files/documentation/esp32-s2_datasheet_en.pdf
- [8] ESP32-S3 Series Datasheet, 2023
https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf
- [9] Mészáros Csaba: Amit az Ultra Wide Bandről tudni kell, 2020
<https://computerworld.hu/tech/amit-az-ultra-widebandrol-tudni-kell-273775.html>

- [10] Ultra-WideBand Wikipédia, 2023 október
<https://en.wikipedia.org/wiki/Ultra-wideband>
- [11] Bluetooth Low Energy Wikipédia, 2023 október
https://en.wikipedia.org/wiki/Bluetooth_Low_Energy
- [12] A Pallas nagy lexikona online kiadás Háromszögelés, 2023 október
<https://www.arcanum.com/hu/online-kiadvanyok/Lexikonok-a-pallas-nagy-lexikona-2/h-B866/haromszogeles-BE77/>
- [13] Háromszögelés Wikipédia, 2023 október
<https://hu.wikipedia.org/wiki/H%C3%A1romsz%C3%B6gel%C3%A9s>
- [14] Network Time Protocol Wikipédia 2023 június
https://en.wikipedia.org/wiki/Network_Time_Protocol
- [15] Wi-Fi – jelentése, működése, Kovács Attila rendszergazda honlapja, 2023 június
https://kovacsattila.info/wifi_jelentese_mukodese.htm
- [16] Wi-Fi Wikipédia, 2023 június
<https://hu.wikipedia.org/wiki/Wi-Fi>
- [17] Wi-Fi for audiophiles: Alliance preps TimeSync certification program, 2023 június
https://www.theregister.com/2017/01/12/wifi_for_audiophiles_alliance_preps_timesync_certification/
- [18] Precision Time Protocol Wikipédia, 2023 június
https://en.wikipedia.org/wiki/Precision_Time_Protocol
- [19] Adafruit Metro ESP32-S2 By Kattni Rembor, 2023 június
<https://learn.adafruit.com/adafruit-metro-esp32-s2/wifi-test>
- [20] https://people.csail.mit.edu/bkph/ftmrtt_intro 2023 február