



Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Hálózati Rendszerek és Szolgáltatások Tanszék



UWB alapú beltéri helymeghatározó rendszer elemeinek fejlesztése

DIPLOMATERV

Készítette

Kovács Zoltán Márk

Konzulens

Dr. Matolcsy Balázs

2022. május 29.

Tartalomjegyzék

Kivonat	i
Abstract	ii
1. Bevezetés	1
1.1. Lehetséges technológiák	1
1.2. UWB technológiai és szabványügyi áttekintés	4
1.3. A pozíció meghatározása	12
1.3.1. ToA (Time of Arrival)	13
1.3.2. TDoA (Time Difference of Arrival)	19
1.4. A pozicionálás pontosságát befolyásoló jelenségek	24
1.5. A technológia előnyei és hátrányai	27
1.6. Rendszerterv	28
2. Hardver	30
2.1. Áramkörtervezés	30
2.1.1. Fejlesztői kártya felépítése	30
2.1.2. DWM1000 modul	32
2.1.3. ESP32 modul	35
2.1.4. Akkumulátormenedzsment	38
2.1.5. Nyomtatott áramköri terv	45
2.2. Áramkörépítés	49
2.3. Mechanikai konstrukció építése	52
3. Szoftver	57
3.1. ESP32 modul	57
3.2. Raspberry Pi	60
3.2.1. Adatbázis	60
3.3. MATLAB	63
3.3.1. Grafikus felhasználói felület	63
3.3.2. Utólagos adatfeldolgozás	66

3.3.3. Pozicionálás	70
4. Helymeghatározó rendszer működése	76
4.1. Rendszertulajdonságok	76
4.1.1. Pontosság	76
4.1.2. Pozíciófrissítési gyakoriság	77
4.1.3. Stabilitás	77
5. Összefoglalás	80
5.1. Helymeghatározó rendszer értékelése	80
5.2. Fejlesztési lehetőségek	80
Köszönetnyilvánítás	82
Ábrák jegyzéke	84
Táblázatok jegyzéke	85
Irodalomjegyzék	85

HALLGATÓI NYILATKOZAT

Alulírott *Kovács Zoltán Márk*, szigorló hallgató kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, 2022. május 29.

Kovács Zoltán Márk
hallgató

Kivonat

Az ultraszélessávú (Ultra-Wideband, továbbiakban UWB) technológia hatékonyan alkalmazható beltéri helymeghatározás esetén. Ez a beltéri helymeghatározás pontossággal szembeni igényére vezethető vissza, amely igényt a globális helymeghatározó rendszerek általában nem tudnak kellő megbízhatósággal kielégíteni. Ezzel szemben az UWB technológián alapuló módszerekkel képesek vagyunk 5...15 cm pontossággal beltéri pozíciót becsülni, ráadásul ezt adott esetben alacsony késleltetéssel. UWB esetén ezek a módszerek alapvetően beérkezési idő, illetve beérkezési időkülönbség alapján becsülik a keresett eszköz pozícióját.

Diplomatervem témáját képezi az UWB technológia alapjainak ismertetése, összehasonlítása más olyan technológiával amely helymeghatározásra alkalmas. Bemutatásra kerül, hogy milyen pontosságot befolyásoló tényezők léphetnek fel az UWB technológia alkalmazása során. Feladataim közé tartozik továbbá egy UWB tag, illetve több UWB anchor eszköz megtervezése és megépítése, amelyekkel utána demonstrálható a helymeghatározási algoritmus működése. A pozicionálás követését egy PC-re készített vizualizációs felület segíti, amelyen megjelenítésre kerül a tag, illetve az összes anchor.

Abstract

Ultra-Wideband (UWB) technology can be used effectively for indoor positioning. This is due to the need for indoor positioning accuracy, which global positioning systems are generally unable to meet with sufficient reliability. In contrast, with methods based on UWB technology, we are able to estimate the indoor position with an accuracy of 5...15 cm, and in addition, this may be done with low delay. In the case of UWB, these methods basically estimate the position of the searched device based on the time of arrival (ToA) and the time difference of arrival (TDoA).

The topic of my thesis is to describe the basics of UWB technology and to compare it with other technologies that are suitable for location. The accuracy factors that may occur when using UWB technology are presented. My responsibilities also include designing and building an UWB tag and several UWB anchor devices to demonstrate the operation of the location algorithm. Positioning is aided by a visualization interface on a PC that displays the tag and all anchors.

1. fejezet

Bevezetés

Napjainkra a helyalapú szolgáltatások iránti igény kimagaslóvá vált. Ezen szolgáltatások több kategóriában játszhatnak központi szerepet. Például a mindennapok során az okosotthon, személyre szabott kirakatok, forgalomfigyelő rendszerek említhetők meg mint helyalapú szolgáltatás. Azonban nemcsak a mindennapok során, hanem ipari felhasználásban is szót ejthetünk ilyen szolgáltatásokról. Az Ipar 4.0 fejlődésével ipari létesítményekben elengedhetetlen a beltéri pozicionálás, mely egyrészt az autonóm robotok közlekedésének alapvető feltétele, másrészt a pozíció ismerete alapján lehetőség van a logisztikai feladatok lefutási idejének optimalizálására [1]. A helyalapú szolgáltatások technológia szintjén 4 területből épülnek fel, úgy mint mobil eszközök (lényegében miattuk lett igény a szolgáltatásra), kommunikációs hálózat (mely biztosítja a szereplők közti kommunikációs csatornákat), GIS (Geographical Information System, téradatbázis, mely helyfüggő adatokat nyújt) és pozicionálási technológia. Utóbbi területhez fűződik a diplomatervem. Pozicionálás jellegű tevékenységre különböző feltételek mellett számos technológia alkalmas. Diplomatervemben az imént említett feltételeket a következők alkotják: pontosság (5...10 cm), beltéri környezet, minél nagyobb pozíciófrissítési gyakoriság. Ezen feltételeket az ipari felhasználás teszi kiemelten fontossá. Robotok vagy drónok épületen belüli közlekedésénél elengedhetetlen a pozíciójuk pontos ismerete. Ipari környezetben jellemzően épületen belül kerülnek elhelyezésre a különböző berendezések, innen ered a beltéri környezet feltétele. A nagy pozíciófrissítési gyakoriság igénye pedig a navigálhatóságot segíti, ugyanis a navigációs algoritmus hatékonyabb, ha időben és térben elegendően sűrű a mintavételezett pozíciósorozat, amely rendelkezésre áll.

1.1. Lehetséges technológiák

A következőkben számbavételezem a jelenleg elérhető technológiákat, amelyekkel képesek vagyunk pozíciót meghatározni.

- Elterjedt megoldás a GPS (Global Positioning System), GPS esetén a kezdeti szinkronizációs idő nagyságrendje az egyik probléma, de arról sem szabad megfeledkezni, hogy a szinkron beállása után is a polgári felhasználás esetén a pontosság a méter nagyságrendjébe esik, ennél fogva önmagában alkalmatlan kis (pár centiméteres) elmozdulások észlelésére. GPS esetén egy további probléma, hogy beltérben a jelentős jelerősség-csökkenés miatt alapvetően leromlik a pontosság. Beltérben az eddigiekhez hozzájön még a többutas terjedés hatása, amely a gyakorlatban használhatatlanná teszi a GPS beltéri alkalmazását [2] [3].
- Egy másik lehetőség a mobilhálózat használata helymeghatározásra. A problémát ez esetben az okozza, hogy alapvetően nem ez technológia célja, hanem a kommunikáció létrehozása, illetve adatátvitel. A mobilhálózat problémája a helymeghatározást tekintve az, hogy főként csak cella vagy szektor szintű pozicionálásra alkalmas. Emiatt csak nagyon kis méretű cellákkal lefedett területen lehet önmagában helymeghatározásra használni, általánosságban azonban kis mozgások érzékelésére alkalmatlan. Más tekintetben a cellaszintű meghatározás kifejezetten előnyös, ugyanis ilyen szinten rendkívül gyorsan megtalálható egy mobilkészülék. A kis- és középvállalatok között előfordul, hogy egy néhány száz 10 gépjárműből álló flottát tartanak fent. Ez esetben elegendő, ha a gépjárművek pozícióját cellaszinten ismerik, ezáltal hatékonyabbá tudják tenni a logisztikai folyamataikat, illetve képesek ellenőrizni az alkalmazottak által megtett útvonalakat. [4].
- A Wi-Fi (Wireless Fidelity) is felhasználható helymeghatározásra [5]. Általánosan a vett jel erősségét (RSSI-Received Signal Strength Indicator) veszi alapul. A módszer előkészülete, hogy nagy méretű táblázatban (lookup table) eltárolja az egyes pozíciókhoz tartozó RSSI értéket, majd a pozicionálás úgy történik, hogy a mért RSSI értéket összeveti a táblázattal, és a mért értékhez legközelebb eső táblázati rekord tartalmazza azt a pozíciót, amely a legvalószínűbb (ez az úgynevezett „fingerprinting” technika). Ennek a módszernek hátránya, hogy a környezeti változásokra érzékeny. A táblázat értékeit frissíteni kell, ha a rekordok felvételének idejéhez képest a belső tér objektumainak helyzete változott. Beltérben ez a hátrány jelentős is lehet, ha nagy objektumok folyamatosan mozognak, mert ez azt igényelné, hogy minden egyes elmozdulás előtt újra fel kell térképezni a területet, valamint frissíteni kell a táblázatot. Akkor célszerű használni ezt a módszert, ha követelmény az alacsony beruházási költség, mivel a Wi-Fi hálózat elterjedtsége épületeken belül szignifikáns, viszont a pontossággal szemben nem támaszt nagy igényt a megrendelő. A

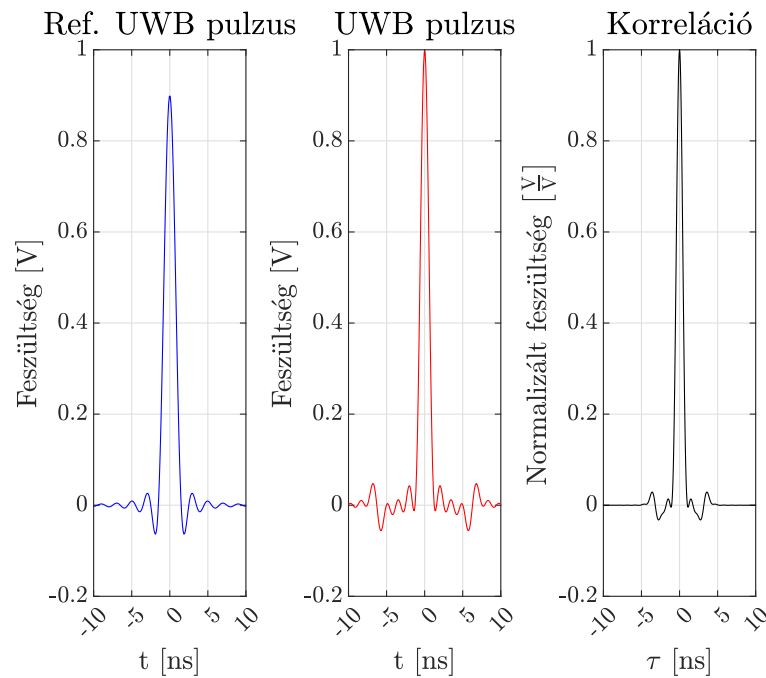
módszer nem csak beltérben üzemképes, hanem kültéren is, viszont nem nagy távolságban. Ez főként azért alakult így, mert a kültéri Wi-Fi lefedettség egyelőre jellemzően csak konkrét üzlethelyiségek, vendéglátáshoz fűződő teraszok esetén van jelen, és nem egy összefüggő kiterjedt hálózatként, így inkább mobilhálózatra csatlakoznak a mobilkészülékek.

- A Bluetooth alapvetően a Wi-Fi-hez hasonlóan adatátvitelre lett kifejlesztve, de szintén alkalmas RSSI alapon történő helymeghatározásra [6] [7]. A Bluetooth alkalmazásának előnye, hogy az épületek falain jelentős csillapítást szenved, mivel kis kitöltési tényezőt használ, így a jel erőssége elhanyagolhatóvá válik [8]. Wi-Fi-vel szemben ez előnyt jelent, mert az egyes helyiségekben kialakuló jelerősségek eloszlása kisebb valószínűséggel lesz egyenletes, így a jelerősség mintákat tartalmazó táblázat sokkal nagyobb dinamikával fog rendelkezni, ezáltal mérsékeltebb a valószínűsége annak, hogy két területhez azonos RSSI fog tartozni.
- A látható fényű kommunikációt (VLC-Visible Light Communication), infravöröst, és RFID-t (Radio-Frequency Identification) használó módszereket a diplomatervezés szempontjából egy kategóriába sorolható saját meggyőződés alapján. Ilyen tekintetben hátrányuk, hogy néhány cm-es helymeghatározásra nem használhatók, ezzel szemben nagy területű beltérben (például áruházban) statisztikus adatgyűjtést tekintve nagy potenciál van ezekben a technológiákban. Egy kivételt képez az RFID szőnyeg, amely alkalmazása esetén akár pár centiméterenként telepíthetők a padlózat felszíne alá úgynevezett RFID kártyák. Ezeknek a kártyáknak jelenlétét egy talajon mozgó objektum képes érzékelni, pozíció azonosítására alkalmas adatot fogadni, és tovább küldeni Wi-Fi, vagy Bluetooth felhasználásával a telepített helymeghatározó rendszer központjába, így egy központi adatbázis alapján az objektum orientációja, és pozíciója is lekérdezhető. Természetesen ez akkor működik, ha a keresett objektum csak 2 dimenzióban végez haladó mozgást, és ez a sík közel van a kitelepített RFID szőnyeghez.

A felsorolt technológiák és módszerek széles körben nem alkalmasak az összes említett feltételt maradéktalanul kielégíteni. Létezik azonban egy technológia, amely igen és ezt UWB-nek (Ultra-Wideband - Ultraszélessávú) nevezzük. A következőkben röviden bemutatásra kerül a technológia, illetve azok a módszerek és algoritmusok, amelyek UWB alapon realizálhatók.

1.2. UWB technológiai és szabványügyi áttekintés

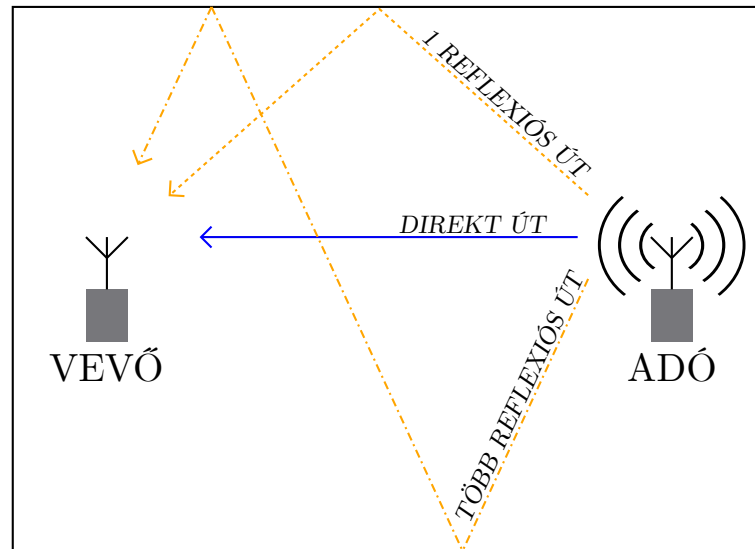
Az UWB-t használó eszközök szabályozása különleges esetként fogalmazható meg. Ennek oka, hogy az UWB technológia által lefedett spektrum átlapolódik olyan frekvenciasávokkal, amelyeket valójában más (jellemzően keskeny sávú) szolgáltatások is használnak. Érthető, hogy az UWB szabályozási folyamata érdekellentéteket von maga után, mivel számos üzemeltető jelentős összegeket fizetett a frekvenciahasználatért. Az UWB eszközökre vonatkozólag először az FCC (Federal Communications Commission - Amerikai Szövetségi Távközlési Bizottság) tett rádiószabályozási követelményeket, ezzel lehetővé téve a kis hatótávolságú (short range) UWB technológiát alkalmazó termékek forgalmazását, és külön engedély nélküli működését [9]. Ezen dokumentum definiál először széles körben alkalmazandó specifikációkat, amelyek alapján egy jel UWB kategóriába sorolható. Definíció szerint az UWB jel legalább 20 %-os relatív sáv szélességű (az elfoglalt sáv szélesség a sávközépi frekvenciának legalább 20 %-a), vagy a sáv szélessége nagyobb, mint 500 MHz.



1.1. ábra. Elemi UWB jellel való korreláció kimutatása [10]

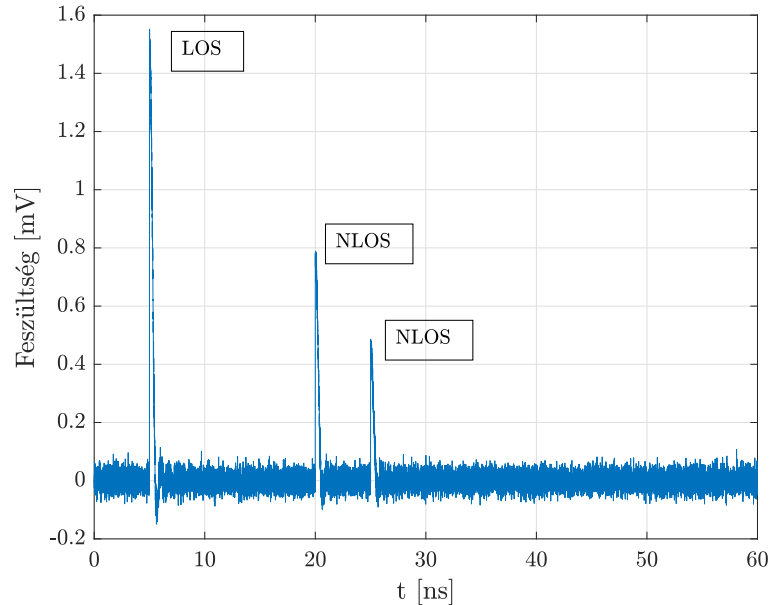
Az IEEE 802.15.4 szabvány rögzíti az UWB jelek időtartománybeli tulajdonságait, melyeknek meg kell felelni a szabvány szerinti működés érdekében. Az 1.1 ábrán látható az elemi jel definíció szerinti alakja (ábrán: Ref. UWB pulzus), közepesen pedig egy saját készítésű pulzus, jobb oldalt pedig a kettő közti korreláció, amely alapján kijelenthető, hogy a saját pulzus szabványnak megfelelő. Ezzel a módszerrel hitelesíthető elméletben egy alkalmazni kívánt pulzus, amelyet majd kisugároz egy forgalomba szánt termék. Pontosabban rögzítve van, hogy a korrelációnak milyen

tulajdonságokkal kell rendelkezni, hogy kompatibilis legyen a szabvánnyal [11]. A szabályozás, illetve a az 1.1 ábra alapján látható, hogy az UWB egy olyan rádiós technológia, mely időtartományban keskeny pulzusokat használ adattovábbításra. Az időben keskeny pulzusok spektrális szélessége szignifikáns, innen eredeztethető a technológia elnevezése.



1.2. ábra. Többutas terjedés beltérben

Az 1.2 ábrán látható az az általános jelenség, hogy beltérben nem csak egy közvetlen (direkt) útvonalon érkezik meg az adó által kisugárzott pulzus, hanem vannak reflexiós útvonalak is, amelyeken keresztül a pulzus terjedni tud. Ezt a jelenséget a villamosmérnöki gyakorlatban többutas terjedésnek nevezik. Először minden esetben a direkt útvonalon érkező pulzust detektálja a vevő (feltéve, hogy létezik a direkt útvonal), majd egyre kisebb amplitúdóval, és egyre nagyobb késleltetésekkel a nem közvetlen úton érkező komponenseket. Az 1.2 ábra egy egyszerűsítéssel él, a valóságban akár nagyságrendekkel több reflektált komponens is eljuthat a vevőbe, de nagy részük már oly mértékben csillapodik, hogy ezek elhanyagolhatóak. Az UWB technológia nagy előnye a többi technológiához képest, hogy az időben rendkívül keskeny pulzusoknak köszönhetően fokozott tűréssel rendelkezik a többutas terjedés hatása ellen. Ennek a tűrésnek a határát akkor érzjük el, ha két pulzus olyan gyorsan érkezik be egymás után, hogy a vevő autokorrelációs algoritmusai már képtelenek különbséget tenni köztük. Ez a gyakorlatban gyártótól függ, jellemzően néhányszor 10...100...1000 ps nagyságrendbe esik. Az 1.3 ábrán látható az a szemléltetés, miszerint a közvetlen rálátásból (LOS - Line Of Sight) érkező pulzustól jól elkülönödik a reflektált pulzusok (NLOS - Non Line Of Sight) érkezési időpontja, azaz a többutas terjedés ebben az esetben nem okoz számottevő problémát a jelfeldolgozásban.

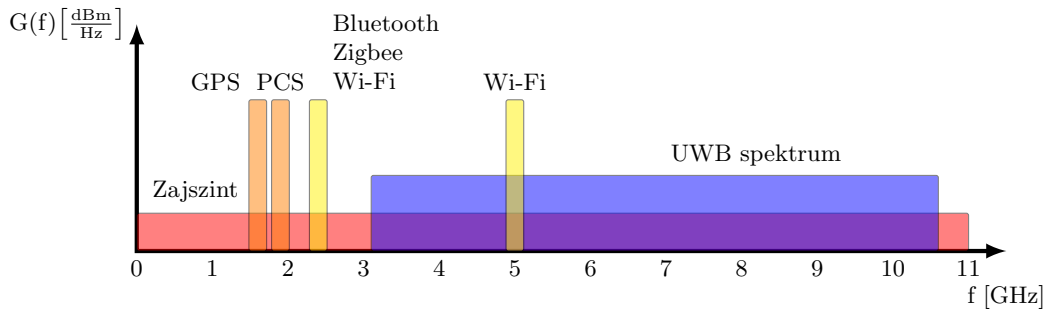


1.3. ábra. UWB jel vételi időfüggvénye [12]

Az UWB pulzusok időbeli szélessége (és a vevőben alkalmazott autokorrelátor) alapvetően meghatározza azt, hogy milyen időbeli felbontással vagyunk képesek detektálni a beérkező pulzusokat. Szabadtérben az elektromágneses hullámok terjedési sebessége jó közelítéssel fénysebesség. Ha például 1 ns pulzusszélességet tekintünk, akkor mivel ennyi idő elteltével a hullám önmagában 0,3 m-t halad, a távolságbeli felbontóképességnek felső korlátjára legrosszabb esetben 0,3 m választással szintén élhetünk. Ezt a korlátot tovább finomítja a vevőben futó autokorrelátor, amely a vett jelen utólagos jelfeldolgozást alkalmaz, ezáltal a gyakorlatban 5...10 cm pontosságra lehet általában számítani.

Az UWB jeleket különböző megoldásokkal állíthatják elő az egyes gyártók. Ez alapján lehet úgynevezett carrier-less (vivő nélküli) illetve carrier-based (vivőn alapuló) jelről beszélni [13]. Utóbbi esetben a jelgenerálás folyamata az, hogy a küldendő biteknek megfelelően pulzusokat generálnak, amelyeket utána formálásnak vetnek alá, majd egy lokális oszcillátor segítségével felkeverik egy kiválasztott szabványos vivő környékére, majd sáváteresztő szűrő után kerül antennával kisugárzásra. Előbbi (vivő nélküli) esetben elméletben kimarad a lokáloszcillátorral beállított keverőfokozat, és a pulzusokat eleve úgy hozzák létre késleltetés útján digitális kapukkal, majd formálják meg, hogy az direkt kisugározható legyen.

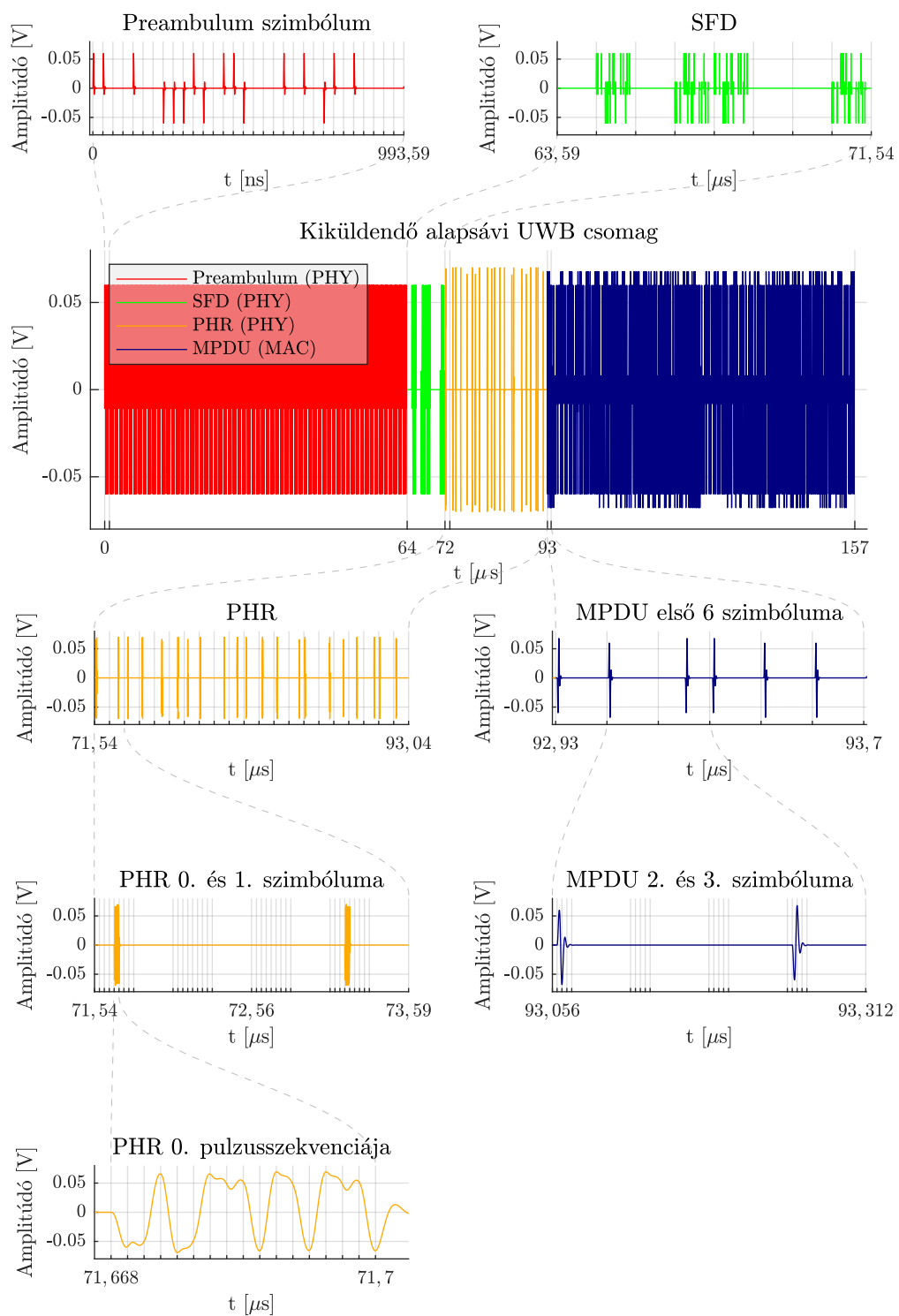
UWB-vel való kommunikálás esetén fontos, hogy alacsonyan tartsuk a kisugárzott spektrális teljesítménysűrűséget, ugyanis az UWB spektrálisan átlapolódik más keskeny sávú technológiával. Az 1.4 ábrán látható spektrális teljesítménysűrűségfüggvény ($G(f)$) szemléletesen az UWB technológia számára kijelölt frekvenciatartományt, illetve néhány meglévő keskeny sávú technológia elhelyezkedését.



1.4. ábra. UWB spektrális átlapolódása más technológiával

Magyarországon az NMHH (Nemzeti Média és Hírközlési Hatóság) honlapján található táblázatok mutatják be a Magyarországon betartandó előírásokat a kisugárzott jel teljesítményére vonatkozóan. Alkalmazástól függ, hogy adott felhasználó számára melyik tartalmazza a releváns adatokat [14]. A honlapon látható 10.2.2 táblázat alapján például az általános felhasználásra vonatkozó paraméterek olvashatók ki, melyekre akkor van szükség, ha az adott alkalmazásra egyik további speciális adatokat tartalmazó táblázat sem hivatkozik. Ha ezt az első táblázatot vesszük alapul, akkor 6...8,5 GHz-es sávban például 0 dBm (Effective Isotropic Radiated Power - Izotropikusan Sugárzott Ekvivalens Teljesítmény) a megengedett maximális kisugárzott csúcsteljesítmény (50 MHz-re vonatkoztatva). A maximális átlagos spektrális teljesítménysűrűség pedig $-41,3 \left[\frac{\text{dBm}}{\text{MHz}} \right]$. Ez utóbbi 1000 MHz sáv szélességű jel esetén $10 \cdot \log \left(10^{-\frac{41,3}{10}} \left[\frac{\text{mW}}{\text{MHz}} \right] \cdot 1000[\text{MHz}] \right) = -41,3 \left[\frac{\text{dBm}}{\text{MHz}} \right] + 10 \cdot \log (1000[\text{MHz}]) = -11,3 \text{ dBm}$ átlagos spektrális teljesítményt jelent.

A diplomatervezés keretében használt (és a hardver fejezetben bemutatott) DW1000-es integrált áramkör kommunikációs módszerét előtérbe helyezve kerül vázlatos bemutatásra az IEEE 802.15.4-2011 szabvány adatkapcsolati rétege (MAC-Media Access Control), illetve fizikai rétege (PHY-Physical). A rétegek által megvalósított funkciók bemutatása több irányból lehetséges, a saját és egyben hardver közeli szemlélet arra sarkall, hogy az áttekintés gyakorlatias formát öltse. Az 1.5 ábrán látható egy alapsávi UWB csomag felépülése, melyhez hasonló például a későbbiekben használt DW1000 IC is előállít. Ennek az ábrának az alapját a MATLAB HRP UWB IEEE 802.15.4a/z Waveform Generation példakódja adja. Ez a csomag kerül rá közvetlenül egy kétszeresen kiegyenlített keverőre, ezáltal AM-DSB/SC (Amplitude Modulation - Double-SideBand/Supressed Carrier) valósul meg, és csatornaválasztástól függően a névleges vivő változhat 3,5...6,5 GHz között. Ez a szóban forgó ábrán annyit módosít, hogy ahol pulzus (a szabványban az elemi jelet „chip”-nek illetve pulzusnak nevezik) található, ott egy pulzusnak megfelelő burkolóval rendelkező vivőfrekvenciás szinuszos jel keletkezik.



1.5. ábra. Kiküldendő alapsávi UWB csomag összetételének szemléltetése

A csomag felépülése a közeghozzáférés-vezérlő (MAC) szintjén kezdődik. Ennek megfelelően keletkezik egy MPDU (MAC Protocol Data Unit), amely magába foglalja többek között a forrás-címzett címekeket, azonosítókat, és a küldendő adatbiteket.

Ezt egy szisztematikus blokk kód (Reed-Solomon(63,55)) úgy módosítja, hogy minden 330 bithez hozzáfűz 48 paritásbitet. Ha nem tesz ki az utolsó blokk 330 bitet, akkor is 48 paritásbitet fűz hozzá. Ezt követően a fizikai rétegre kerülve az MPDU elé fűződik a 19 bitből álló PHR (Physical Header). Ebből a 19 bitből 13 bit segíti a vételi oldalon a MAC keret sikeres feldolgozását, a maradék 6 bit pedig a Hamming kódolással előállított SECDED (Single Error Correction-Double Error Detect) bitek. Az MPDU és PHR bitek (ide értve a Reed-Solomon paritásbiteket is) ezt követően egy szisztematikus konvolúciós kódoláson esnek át. Ez egy léptetőregiszterrel kivitelezett kódolás, amely minden bejövő bithez 2 bitet rendel (pozícióbit és előjelbit). A pozícióbit mindig az előző adatbittel egyenlő, az előjelbit pedig mindig az aktuális adatbit és az előző pozícióbit kizáró-vagy kapcsolata. A vevő oldalon ezt a konvolúciós kódolást az úgynevezett Viterbi-dekódolás oldja fel. A blokk kódokat alkalmazó Reed-Solomonnal ellentétben ez a dekódolás során nem csak abból a halmazból való kódszavakat képes feldolgozni, amelyből a kódszavak veszik az elemeiket (legfeljebb a törlés szimbólum megengedett), hanem tetszőleges mérési minták lehetnek. Létezik a Viterbi dekódolásnak egy jellemzője, amelyet egyeztetni kell az adóban illetve vevőben, ez a Viterbi-arány, amely pontosan azt mondja meg, hogy hány bemeneti bithez hány kimeneti bitet rendel. Ez a DW1000 esetében a 1/2, de a szabvány definiál 1/1 értéket is. A konvolúciós kódolás eredményeképp (pozícióbit, előjelbit) előáll egy BPM-BPSK (Burst Position Modulation-Binary Phase Shift Keying) modulációs séma. A koherens vevőben (frekvencia és fázishelyes vételi eljárást alkalmazó vevő) alkalmazott Viterbi-dekóder képes ezáltal +3 dB kódolási nyereséget elérni.

A BPM-BPSK feloldásában található burst szó arra utal, hogy a létrehozott pozícióbit és előjelbit nem 1 db pulzusra konvertálódik a pulzusformálás során, hanem az előző kódolási eljárásokon felül még egy úgynevezett szórókódot is alkalmaz. Ennek ismertetéséhez viszont szükséges ismerni az UWB csomag elején található preambulumot, amely a vevő szinkronizációs eljárását teszi működőképessé. A preambulum az 1.5 ábrán látható UWB csomag elején található, és minden esetben ismétlődő preambulumszimbólumokból épül fel. Egy preambulumszimbólum készülhet szabvány szerinti 31, vagy 127 hosszú ternáris (-1,0,+1 értékeket felvevő) kódok valamelyikéből. Az egyes frekvenciatartománybeli csatornákra úgy határozták meg ezeket a kódokat, hogy szomszédos csatornák esetén a keresztkorreláció minél inkább kisebb legyen, ezáltal csökkentve a szomszédos csatornák közti zavaró hatást. Ezzel szemben a kódok autokorrelációja kiemelkedő. A preambulumszimbólum időtartományi reprezentációját úgy definiálták, hogy a 31 és 127 hosszú kód névlegesen azonos időbeli kiterjedésű lehessen. Ez tekinthető egy hardver közeli megközelítéssel, miszerint a pulzusformáló maximális ismétlési frekvenciája (órajele) 499,2 MHz. Ekkora frekvenciával képes ismételni a 2 ns széles elemi pulzusokat („chip”-eket). Amennyiben

31 hosszú a kód, akkor minden kódpont mögé beékelődik 15 nullpont. Ezzel az érhető el, hogy összesen 496 kódpont legyen, és ezeket 499,2 MHz frekvenciával képezze le a pulzusformáló, így 993,59 ns hosszú preambulumszimbólum keletkezik, és az eredeti 31 kódpontot $31/993,59 \text{ ns} = 31,2 \text{ MHz}$ kódpontismétlési frekvenciával kerül a kimenetre. Figyelembe véve, hogy a 31 hosszú kódokban 15 db kódpont mindig azonosan 0, az átlagos pulzusismétlési frekvencia $16/993,59 \text{ ns} = 16,1 \text{ MHz}$. Ha 127 hosszú a kód, akkor minden kódpont mögé 3 nullpont ékelődik be, így összesen 508 kódpont (vagy chip intervallum) keletkezik, amelyet szintén 499,2 MHz frekvenciával képez le a pulzusformáló fokozat, ezáltal 1017,63 ns hosszú preambulumszimbólum keletkezik, és az eredeti 127 kódpont $127/1017,63 \text{ ns} = 124,8 \text{ MHz}$ kódpont ismétlési frekvenciával kerül a kimenetre. Hasonlóan az előző esethez itt is figyelembe vehető a kód összetétele, vagyis a 0 kódpontokat nem számítva $64/1017,63 \text{ ns} = 62,89 \text{ MHz}$ átlagos pulzusismétlési frekvencia adódik. Természetesen ezek a frekvenciák számíthatók 499,2 MHz-ből is ha a kódok szimbólummá alakulása közbeni elnyúlását nézzük, amely első esetben 16-szoros, második esetben 4-szeres. Ezekkel elosztva a maximális chipismétlési frekvenciát, adódnak a kódpontismétlési frekvenciák, és tovább szorozva a 16/31 és 64/127 arányokkal adódnak az átlagos pulzusismétlési frekvenciák.

Egy ilyen S preambulumszimbólum ismétlődéseinek száma lehet 16, 64, 1024, 4096 a szabvány alapján, viszont a DW1000 által támogatott további értékek a 128, 256, 512, 1536, és 2028. A fenti ábrán látható preambulumszimbólum 64-szer ismétlődik.

A következő egység az 1.5 ábrán az SFD (Start Frame Delimiter), amely egy meghatározott 8, vagy 64 hosszú ternáris kód szerint a preambulumszimbólumot ismétli meg. Az ábrán a 8 hosszú kóddal készített SFD látható. Akárhogyan nem választható meg az SFD hossza, a csomag későbbi részeinek adatsebességéhez kötött, hogy 8 hosszú, vagy 64 hosszú lesz az SFD által alkalmazott kód. Az SFD szerepe kulcs fontosságú a pontos beérkezési idő meghatározásának folyamatában. Egyrészt jelzi a preambulum végét, és a BPM-BPSK-ra való átállás kezdetét, másrészt a DW1000-ben található LDE (Leading Edge - „első sugár”) kereső algoritmus az SFD végére próbál illeszteni egy időbélyeget, amelyet a csomag beérkezési idejének tekint.

Ezen a ponton érdemes visszautalni a korábban említett burst kifejezésre. A preambulum és SFD egyszerű kódokból, illetve azok elnyújtott változataiból épülnek fel, alapsávi moduláció nem történik, eltekintve az SFD mechanizmusától. A PHR és MPDU viszont már alkalmazza a konvolúciós kódolást, ami felfogható BPM-BPSK sémaként. Mindemellett a konvolúciós kódolás mellett egy idővariáns szórókédot is alkalmaz, amely által egy bitből egy pulzusszekvencia (burst) keletkezik. Minden

burst a szórókód első néhány bitje által címezve ugrani is fog (time hopping) szimbólumon belül egy adott pozícióba. A szórókód egy LFSR (Linear Feedback Shift Register) kimeneteként adódik, és közvetlenül előállítható a PHR első szimbólumától kezdve az MPDU utolsó szimbólumáig a preambulumban használt kód segítségével. Ez úgy történik, hogy először a preambulumból a nullák letakarásra kerülnek, majd a mínusz pontok 0-ba állnak át. Az így kapott kódot kell tölteni az LFSR bemenetére kezdeti állapotként. Az első szórókód (és az utána következők is) aszerint a szabály szerint keletkezik, hogy az aktuális kódponthoz kell venni a 14 és 15 kódponttal ezelőtti kódpont kizáró-vagy kapcsolatát. A szórókód hossza, azaz a burstben található chipek száma több tényezőtől függ (szimbólumsebesség, adatsebesség, preambulumból kód hossz), a fenti példában 16-os értékre adódik, és minden PHR szimbólumban pontosan 1 ilyen pulzusszekvencia foglal helyet. Konvolúciós kódnak megfelelően a szimbólum első, vagy második felében (a korábban említett pozíció bit jelzi, hogy a burst a PHR szimbólum melyik felében helyezkedik el, az előjelbit pedig a burst invertálását jelzi), illetve a szórókód első 3 bitje által címezve ugrik a megfelelő helyre. A fenti példában így 0...7-ig ugorhat a burst, majd a szimbólum negyedének megfelelő időtartam védőidőnek van fenntartva, ezzel védekezve a többutas terjedés ellen. A burst ugratás pedig a többfelhasználós rendszerekben segít védekezni a sajátcsatornás-zavarás ellen. A szórókód célja pedig a spektrumfehérítés. Végül az MPDU-ban hasonló módon történik a konvolúciós kód és a szórókód együttes hatásmechanizmusa azzal a különbséggel, hogy az alkalmazott adatsebesség miatt lecsökken a pulzusok száma az egyes pulzusszekvenciákban 16-ról 2-re. A szórókód képzése folytonosan történik. Az LFSR nem törlődik a PHR-MPDU átmenetben, mindössze a szórókódok hossza lecsökken 2-re, ezáltal az ugratást is csak 0...3-ig lehet címezni.

Végezetül a DW1000 LDE algoritmusának körülményei. A beérkező csomagokat egy kétszeresen kiegyenlített keverő lekeveri alapsávra, ahol egy ADC (Analog to Digital Converter) kimenetén keletkeznek az IQ ágak megfelelő komplex minták. Minden minta 16 bit valós, 16 bit képzetes részből áll, és egy komplex minta a DW1000 akkumulátor regiszterének 4096 bájtjából 4-et foglal el. A mintavételi sebesség 499,2 MS/s-nak a kétszerese, azaz kb. 1 ns időközönként készül egy komplex minta. Az akkuból így a tisztán hardverek által képzett időbeli felbontóképesség 1 ns, amihez 30 cm távolságfelbontás társul. Az akkumulátort arra használja a DW1000, hogy meghatározza a csatorna impulzusválaszát, amelyben aztán az LDE algoritmus megkeresi azt a legvalószínűbb mintát, amihez tud illeszteni egy 16 bites mutatót. Összesen 1024 mintát tud tárolni az akku egyszerre, ezért ezek címezéséhez a 16 bit felső 10 bitjét használja, az alsó 6 bit pedig törtrész címezéshez van használva. A legkisebb egész rész címző bit 1 ns ugrást idéz elő, a 6 törtrész pedig

ezt a 64-ed részére finomítja, ami 15,625 ps. Amikor ezzel az algoritmussal végez, akkor ezt a 16 bitet felhasználva készíti el a 40 bites lokális időbélyeget, amelyet aztán a különböző üzenetváltási sémák alkalmazása során el kell küldeni. Az LDE algoritmus pontos megvalósítását nem közli a gyártó, de az valószínűsíthető, hogy 2 mozgóablakos átlagolással végig léptet az akkumulátor mintáin, és ezek alapján keres kiugró értékeket. Az LDE algoritmus ezek mellett becsléseket tesz a vett jel teljesítményére, és az első úthoz tartozó vételi teljesítményre is, valamint egy zajszinttel összefüggésben lévő értékre. Mindezeket felhasználja arra, hogy meg tudja különböztetni a zaj által létrejött nagy értékű mintákat a hasznos jel által keletkezett mintáktól. Szintén az impulzusválasz alakja által végez vivóvisszaállítást a vevőben, illetve a képes kiterjeszteni az üzemi dinamikatartományt azáltal, hogy a több úton érkező egymáshoz képest eltolt csomag által hordozott energiát összegez.

Természetesen ha megtörtént a szinkronizáció, azaz felismerte az egyik preambulumszimbólumot, akkor utána már képes feldolgozni a PHR szimbólumokat a Viterbi-dekódolással a szórókód ismeretében. Ezzel ismerté válnak a vevő számára az MPDU paraméterei, amelyet a PHR szimbólumokhoz hasonlóan képes sikeresen feldolgozni.

A technológia hátrányaként említhető, hogy például a Wi-Fi, vagy Bluetooth-hoz képest kevésbé volt ezidáig elterjedt, azonban az Apple saját fejlesztésű U1 integrált áramkörü megoldását beépítette 2019-ben az Iphone 11 készülékbe, majd utána az Iphone 12, Iphone 13 mobilokba, illetve az Apple Tag-be is [15] [16] [17]. Emellett egy másik világszerte ismert vállalat, a Samsung is integrálta a saját megoldását a Samsung Galaxy Note20 Ultra, ZFold2, és S21 készülékekbe, valamint a SmartTag+ eszközbe [18]. Jelenleg az UWB-t arra használják, hogy együttműködve a BLE (Bluetooth Low Energy) technológiával pontosabb távolságmeghatározást kínálnak az eszközök, illetve a távolságot felhasználva már meglévő funkciókat fejlesszenek tovább. Ezzel összefüggésben egy feltételezett cél a kiterjesztett valóság (Augmented Reality–AR) elterjesztése a mobil eszközök által [19]. A kiterjesztett valóság pedig nagyban könnyíti bizonyos térbeli tervezési feladatok közben fellépő problémák áthidalását, mindemellett szórakoztatóipari felhasználásban egyértelmű potenciált hordoz. Az előzőekben ismertetett információk alapján kijelenthető, hogy több jelentős világpiaci szereplő építeni kíván a jelenben és közeljövőben az UWB technológiára, más szóval a területnek kiemelt aktualitása van.

1.3. A pozíció meghatározása

A technológia által alkalmazható módszerek 2 jellemző eszközt különböztetnek meg. Az egyik amelynek a pozícióját meg szeretnénk becsülni (ennek neve tag), a másik

pedig az, amely rendelkezésünkre áll a pozicionáláshoz (ennek neve anchor). Helymeghatározás esetén beszélhetünk abszolút illetve relatív pozícióról. Előbbi akkor becsülhető, ha ki van hozzá építve egy infrastruktúra (azaz ismerjük az anchorok pontos helyzetét), utóbbi pedig akkor ha ilyen nem áll rendelkezésre. Utóbbira említhető szemléletes példa, amikor gyors reagálású katonai bevetés zajlik beltéri terepen, és nincs idő infrastruktúrát kiépíteni. Ilyenkor a katonáknál lévő készülékek ad-hoc hálózatot (készülékek közvetítő nélkül kommunikálnak) alakítanak ki, így tudják becsülni egymáshoz képesti helyzetüket. Ez azonban egy fix, ismert helyzetű eszköz nélkül csak a bevetés résztvevői számára ad információt. Ha ki van építve egy infrastruktúra, akkor a lokális koordináta-rendszer összefésülhet a globális koordinátákkal, amely matematikai értelemben véve egy lineáris leképezést takar. A továbbiakban a teljes hangsúly a lokális koordináta-rendszerben való pozicionálásra helyeződik.

Az alábbiakban bemutatásra kerülnek a gyakran használt UWB alapú, időmérést alkalmazó pozicionálási módszerek [20].

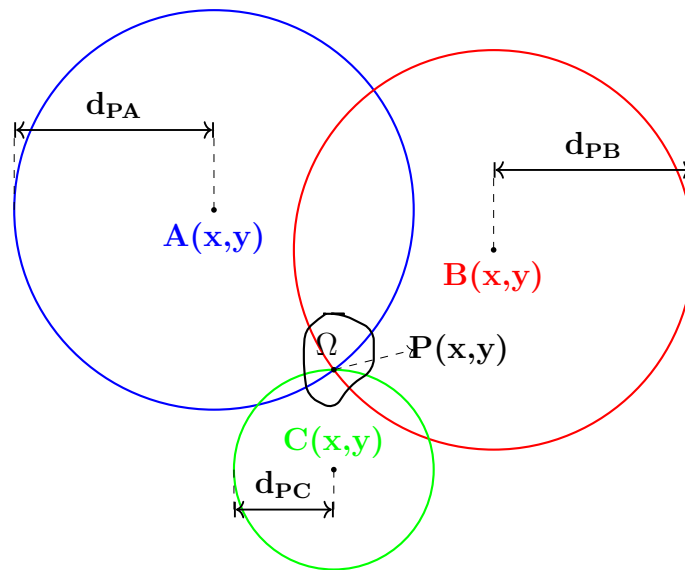
1.3.1. ToA (Time of Arrival)

A Time of Arrival egy beérkezési idő rögzítésén alapuló távolságmérés, ahol a mért paraméter küldött jel beérkezési idejének pillanata. Ha ismerjük a jel terjedési sebességét, illetve a jel kiküldésének pillanatát, akkor kiszámítható a két eszköz közötti távolság.

A pozíció meghatározása 3 dimenzióban ToA alapon a következő módon történik. Adott egy meghatározandó $P(x, y, z)$ pont, amely a tag szerepét tölti be. Kérdés, hogy hány anchor kell $P(x, y, z)$ koordinátáinak meghatározásához. 1 anchor (későbbiekben $A(x, y, z)$) esetén mindössze egy d távolságot tudunk meghatározni, amely kirajzol $A(x, y, z)$ körül egy gömböt, és ezen gömb bármely pontján lehet $P(x, y, z)$. Ha $A(x, y, z)$ mellett elhelyezünk egy másik anchor (továbbiakban $B(x, y, z)$), akkor $A(x, y, z)$ és $B(x, y, z)$ által kifeszített gömbök metszeni fogják egymást, a metszet alakja pedig egy esetet leszámítva (ha a $P(x, y, z)$ pont a két anchor között található) mindig egy szabályos kör lesz (észrevehető, hogy bár a lehetséges pontok száma továbbra is végtelen, mégis érezhetően javul a helyzet ahhoz képest mintha csak $A(x, y, z)$ működne). $P(x, y, z)$ elhelyezkedése ennek eredményeképp a gömbről egy körre korlátozódik. $A(x, y, z)$, és $B(x, y, z)$ mellé egy újabb anchor kerül elhelyezésre, amelynek neve $C(x, y, z)$. Az előbbi helyzethez képest a két gömb metszetét egy újabb gömb fogja metszeni a térben, amely azzal egyenértékű mintha egy kört metszene egy gömb. Ennek eredménye 2 metszéspont lesz és az egyikben található $P(x, y, z)$. Az eddigi gondolatmenetet folytatva a 2 metszéspont közül úgy dönthető el, hogy melyik a helyes, ha egy negyedik $D(x, y, z)$ anchor is beüzemelésre kerül. A

gondolatmenet kimenetele alapján a pozíció kiszámításához elméletben legalább 4 anchor szükséges. Erről a későbbiekben bemutatott pozicionáló algoritmus kapcsán még lesz szó.

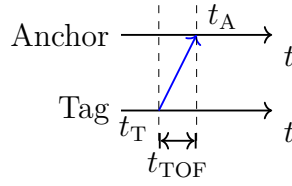
A mérési hibák következtében a kiszámított pozíció nem pontos. A beérkezési időpillanatok mérési hibája eredménye az, hogy a keresett tag ($P(x, y, z)$) az öt körülvevő Ω térrészben van. A cél a mérési hibák vagy azok hatásának csökkentése egy olyan szint alá, amely szint alatt Ω térfogata egy olyan (alkalmazástól függő) érték alá esik, amely már elfogadható. Az 1.6 ábrán látható a folyamat 2 dimenzióban. A ToA kulcsa az, hogy az anchorok és tagek közötti üzenetek küldési, és



1.6. ábra. Pozicionálás szemléltetése ToA alkalmazása esetén 2 dimenzióban

fogadási időpontját jegyezzük fel, majd ezek alapján számítjuk a jel terjedési idejét (ToF-Time of Flight). Több üzenetváltási séma létezik, melyek közül az alapvetőek felsorolásszerűen kerülnek bemutatásra [21]:

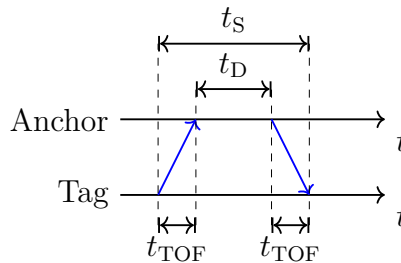
- OWR (One Way Ranging) A legegyszerűbb séma az egyutas távolságmérés, ekkor 1 db üzenetet küld a tag az anchornak, így az 1.7 ábrán, illetve az alatta található 1.1 egyenlet alapján számítható a jelterjedési idő. Egyszerűségéből következik, hogy (ToA esetén) üzenetváltásokkal ekkor lehet leghamarabb mérési eredményt kapni, ezáltal legnagyobb helyzetfrissítési frekvenciát elérni. Természetesen a módszer egyszerűsége magában hordozza a fő hátrányát is, amelyet magába rejt az 1.7 ábra. Ez pedig az időbeli szinkronizáció szükségessége a két eszköz között. Szinkronizáció nélkül a módszer használhatatlanná válik.
- TWR (Two Way Ranging) Az OWR szinkronizációs feltételét ki lehet küszöbölni egy plusz üzenet segítségével, melyet az 1.8 ábra reprezentál. Az 1.8



1.7. ábra. Egyutas üzenetváltási séma

$$t_{TOF} = t_A - t_T \quad (1.1)$$

ábra alapján világosan látható, hogy a jelterjedési időre vonatkozó összefüggésből az abszolút időpillanatok (t_T , t_A) kiesnek az 1.2 egyenletből, mindössze a késleltetések kapnak szerepet. Természetesen mindkét eszköz saját órajelének frekvenciája meg kell egyezzen, különben skálatényezőt kell alkalmazni (szoftveres úton történő kompenzáció), vagy egy bonyolultabb üzenetváltási sémát kell fontolóra venni, esetlegesen hardveres módosítást kell eszközölni (kristályoszillátor finomhangolása trimmer kondenzátorral, kisebb frekvenciahibával rendelkező oszcillátor választása). Létezik a TWR sémának egy módosított változata, amely két üzenettel kezdődik, és pozitív hatással van a mérés hibájára [22].

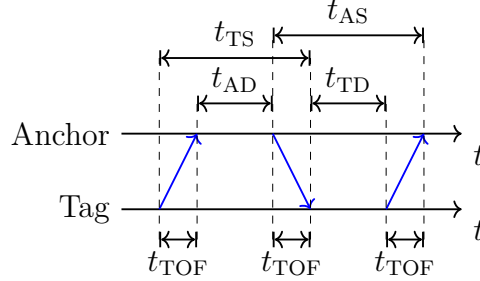


1.8. ábra. Kétutas üzenetváltási séma

$$t_{TOF} = \frac{t_S - t_D}{2} \quad (1.2)$$

- SDS-TWR (Symmetrical Double-Sided Two Way Ranging) A módszer jó tulajdonsága az, hogy a tag és anchor kristályoszillátorának frekvenciakülönbsége által okozott hibát a TWR módszerhez képest jelentősen csökkenti. Mindezt úgy teszi, hogy lényegében összefésül szimmetrikusan 2 TWR sémát, ez pedig a frekvenciahiba okozta jelterjedési idő mérésének hibáját csökkenti nagy mértékben. Az üzenetváltási séma az 1.9 ábrán látható. Hátránya, hogy cserébe ez egy hosszabb üzenetváltási séma, tehát a pozíciófrissítési gyakoriság

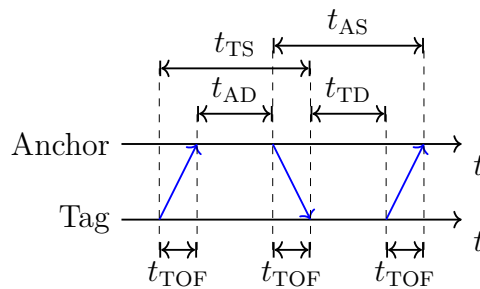
jelentősen lecsökken. Továbbá egy másik kedvezőtlen tulajdonsága, hogy két eszköz közötti üzenetváltások során a módszer feltételezi, hogy azok működési sebessége azonos (az anchor és tag általi t_D késleltetések azonosak).



1.9. ábra. Szimmetrikus kétutas üzenetváltási séma

$$t_{\text{TOF}} = \frac{1}{4} \cdot (t_{\text{TS}} + t_{\text{AS}} - t_{\text{AD}} - t_{\text{TD}}) \quad (1.3)$$

- ASDS-TWR (Alternative Symmetrical Double-Sided Two Way Ranging, szakirodalomban megtalálható AltDS-TWR néven is) Technikailag, tehát a formát tekintve azonos az SDS-TWR sémával, viszont a t_{TOF} kiszámítása más módon történik. Ennek köszönhetően a beszélgetést lebonyolító eszközök frekvencia-hibája ekkor jelenik meg legkisebb mértékben a számításban (1.15 ábra, 1.26 egyenlet), ennél fogva leginkább ennek a számítási módszernek használatát lehet javasolni. A séma vázlatja és a hozzá tartozó számítás látható az 1.10 ábrán, és az 1.4 egyenletben.

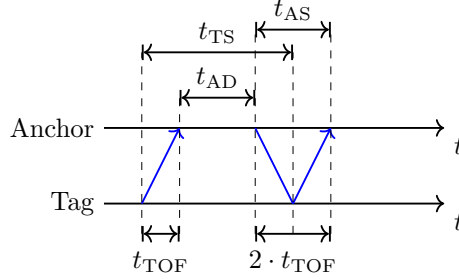


1.10. ábra. Alternatív szimmetrikus kétutas üzenetváltási séma

$$t_{\text{TOF}} = \frac{t_{\text{AS}} \cdot t_{\text{TS}} - t_{\text{AD}} \cdot t_{\text{TD}}}{t_{\text{AS}} + t_{\text{TS}} + t_{\text{AD}} + t_{\text{TD}}} \quad (1.4)$$

- ADS-TWR (Asymmetrical Double-Sided Two Way Ranging) Az SDS-TWR-hez hasonlóan abból indul ki, hogy összefésül 2 TWR sémát, viszont ezúttal

nem feltételezi, hogy az anchor és tag közel azonos késleltetéssel rendelkeznek. Innen ered az aszimmetrikus jelző. Ennek a sémának a vázlata és hozzá tartozó számítás látható az 1.11 ábrán és az 1.5 egyenletben. Az üzenetváltási sémák



1.11. ábra. Aszimmetrikus kétutas üzenetváltási séma

$$t_{\text{TOF}} = \frac{1}{4} \cdot (t_{\text{TS}} + t_{\text{AS}} - t_{\text{AD}}) \quad (1.5)$$

működésének nem feltétele az, hogy a tag kezdeményezze a folyamatot, ez csupán megállapodás kérdése. Ezzel a konvencióval élve egyszerű megvalósítani a kommunikáció sorrendjét olyan értelemben, hogy felprogramozás útján a tag ismeri az anchorok azonosítóit, így adott sorrendben képes egyesével végighaladni az anchorokon. Ez fordított esetben egy plusz üzenetkiküldést igényel az egyes anchorok között, amely definiálja a sémák levezenylésének sorrendjét. Egy másik lehetőség, hogy egy központi vezérlő egység ad utasítást az anchorok számára, így biztosítva azt, hogy 2 anchor üzenetváltási sémája ne zavarja meg egymást.

A tag és az anchorok közötti jelterjedési idők (t_{TOF}) számítása után a következő lépés a tag koordinátáinak számítása az anchorok pozíciójának ismeretében. Erre egy analitikus [23], és egy numerikus algoritmus kerül bemutatásra. Az 1.6 egyenlet fejezi ki koordinátákkal a tag távolságát az i -edik anchortól egy olyan rendszerben ahol M db anchor található,

$$d_i = \sqrt{(x_i - x_{\text{T}})^2 + (y_i - y_{\text{T}})^2 + (z_i - z_{\text{T}})^2} \quad i = 1 \dots M \quad (1.6)$$

melyet négyzetre emelve, algebrai azonosságot alkalmazva végül átrendezve jutunk az 1.7 egyenletre.

$$x_{\text{T}}^2 + y_{\text{T}}^2 + z_{\text{T}}^2 - 2x_i x_{\text{T}} - 2y_i y_{\text{T}} - 2z_i z_{\text{T}} = d_i^2 - x_i^2 - y_i^2 - z_i^2 \quad i = 1 \dots M \quad (1.7)$$

Amely mátrixos alakban a következőképp (1.8) egyenlet szerint írható fel.

$$\underbrace{\begin{bmatrix} 1 & -2x_1 & -2y_1 & -2z_1 \\ 1 & -2x_2 & -2y_2 & -2z_2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & -2x_M & -2y_M & -2z_M \end{bmatrix}}_{\underline{S}} \cdot \underbrace{\begin{bmatrix} x_T^2 + y_T^2 + z_T^2 \\ x_T \\ y_T \\ z_T \end{bmatrix}}_{\underline{P}} = \underbrace{\begin{bmatrix} d_1^2 - x_1^2 - y_1^2 - z_1^2 \\ d_2^2 - x_2^2 - y_2^2 - z_2^2 \\ \vdots \\ d_M^2 - x_M^2 - y_M^2 - z_M^2 \end{bmatrix}}_{\underline{K}} \quad (1.8)$$

Az 1.8 megoldása akkor létezik, ha \underline{K} benne van az \underline{S} együtthatómátrix oszloptejében. A gyakorlati eseteket tekintve azonban rendkívül gyakran előfordul, hogy a teljes egészében mérések által meghatározott az 1.13 mátrixegyenlet inkonzisztens, azaz ellentmondásos. Ilyen esetben egy működőképes megoldás, ha kiválasztjuk \underline{S} oszlopteréből a \underline{K} -t legjobban közelítő vektort, és azzal oldjuk meg a mátrixegyenletet. Az ilyen megoldásokat az eredeti egyenletrendszer optimális megoldásainak, vagy a legkisebb négyzetek elve szerinti megoldásainak nevezzük, amelyet az 1.9 egyenlet fejez ki.

$$\hat{\underline{P}} = \underline{S}^+ \cdot \underline{K} \quad (1.9)$$

Matematikailag a módszer működéséhez szükséges anchorok száma 4, ennél kevesebb anchor esetén a számított pozíció nem tükrözi megbízhatóan a valóságot. Említésre méltó érdekesség, hogy az algoritmus működéséhez szükséges 4 anchor koordinátája is feltételhez van kötve. Saját MATLAB szimulációk alapján akkor működik helyesen, amennyiben a 4 anchor kijelöli a térben a lokális koordináta-rendszer origóját, illetve a 3 tengelyt. Ha az origón kívül mindössze 2 tengelyt jelöl ki, akkor a számított tag pozíció egy merőleges vetítettje lesz a kijelölt két tengely által kifeszített síkra.

A numerikus algoritmus működése lényegét tekintve egy egyedileg definiált gyakran nemlineáris függvény (úgynevezett költségfüggvény) minimalizálási feladata. Különböző szoftverkörnyezetben léteznek olyan megoldók, amelynek bemenetére egy függvényt, esetleg egyéb paramétereket adva, iterációs folyamatok során megkeresi valamelyik (vagy akár egyszerre több) változó szerint a minimumhelyét a függvénynek. A minimumhely-keresés miatt célszerű úgy definiálni a költségfüggvényt, hogy az csak pozitív értékeket vegyen fel, illetve az rendelkezzen egy globális minimummal. Egy lehetséges költségfüggvényt jelent saját definiálás útján a az 1.10

egyenlet.

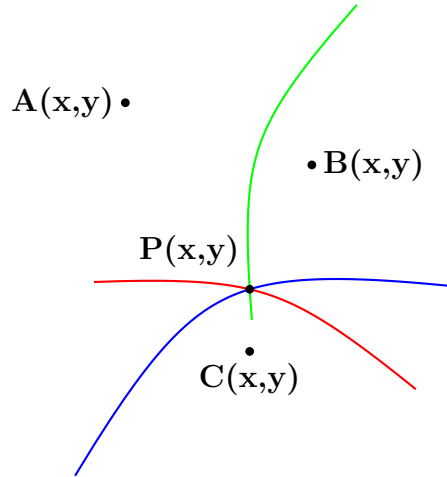
$$C(x, y, z) = \sum_{i=1}^M \left(\sqrt{(x - x_i)^2} - d_i \right)^2 + \left(\sqrt{(y - y_i)^2} - d_i \right)^2 + \left(\sqrt{(z - z_i)^2} - d_i \right)^2 \quad (1.10)$$

Az egyenlet alkalmazása során a megoldó megkeresi, majd visszaadja azt az x, y, z számhármast, ahol a C függvény értéke minimális. A visszakapott számhármast adja a keresett tag pozíciójának közelítését, ugyanis a közelítés szerint akkor van a tag egyszerre mindegyik anchortól a mért távolságra, ha a függvény értéke minimális. A szimulációban alkalmazott megoldó (`fminsearch`) a MATLAB sajátja, amely a fellelhető dokumentáció alapján a Nelder-Mead algoritmust alkalmazza [24].

1.3.2. TDoA (Time Difference of Arrival)

A módszer alapvetése, hogy az anchorok időben pontosan legyenek szinkronizálva, illetve fix helyre legyenek telepítve. Az eljárás beérkezési időkülönbség mérésén alapul. Ennek szemléletes magyarázata a következő. A tag elküld egy üzenetet minden anchor számára, azok pedig megnézik mikor érkezik meg az üzenet. Válasszunk ki két tetszőleges anchorot, és végezzük el a feljegyzett időpontok kivonását. Az így kapott érték több időpont kivonásával létrejöhethet. Ezek a kombinációk 2 dimenzióban kijelölnek a két anchor között egy hiperbolát. A tag pozíciója 2 anchor (mint megfigyelők) szempontjából bárhol lehet a hiperbolán. Az eddigiek alapján ha felveszünk (3 dimenziós esetet vizsgálva) még 2 anchorot, akkor összesen 6 forgáshiperboloid (ez esetben a hiperbola megoldást adó köpenyének megforgatása) feszül ki a térben, és ezek metszéspontja lesz a tag pozíciója [25]. A szinkronizáció pontosságára rendkívül érzékeny, 120 ps nagyságú hiba már 3 cm pozíció meghatározási hibával járhat [26]. Az 1.12 ábra mutatja be 2 dimenzió esetében a TDoA alkalmazását.

A TDoA-val kapcsolatban alapjait tekintve 3 algoritmus említhető, amelyeknek analitikus úton létezik megoldása. Mindegyik algoritmus abból indul ki, hogy az anchorok közös órajelet kapnak, ezáltal biztosított a nagy pontosságú szinkronizáció. A TDoA-n alapuló algoritmusok olyan esetekben jönnek számításba, amikor nagy pozíciófrissítési sebességre van szükség. Az üzenetváltásokon alapuló módszerekkel (ranging) egy távolságszámítás több ciklusból tevődik össze. Ezzel szemben a TDoA esetén alapvetően a tag üzenetküldésének ismétlési sebessége, illetve az anchorok mögötti adatbázis és feldolgozóegység sebessége a meghatározó. Fontos különbség ezen a ponton az, hogy míg a ranging esetén az egyes késleltetések közvetlenül összeadódnak a teljes számítási folyamatot illetően, a TDoA-t tekintve ez nincs így, mivel amíg a tag újabb üzenetküldésre készíti fel magát, addig az előző kiküldött üzenetet



1.12. ábra. Pozicionálás szemléltetése TDoA alkalmazása esetén 2 dimenzióban

az anchorok éppen feldolgozzák. A tag és az anchor-feldolgozóegység közül az egyik fog dominálni a pozíciófrissítési gyakoriság felső korlátját tekintve. A 3 algoritmus megnevezhető, mint legkisebb négyzetek módszere, Foy módszere, illetve Fang módszere. 2021 őszén egy TDK dolgozat keretében végzett munka kimerítően vizsgálja a 3 algoritmust levezetésekkel, MATLAB-ban készített saját szimulációkkal, így ez a diplomaterv ezek terjedelmének ismeretében egy-egy rövid ismertetésre szorítkozik.

A legkisebb négyzetek módszerének folyamata, hogy a tag és anchorok pozícióját koordinátáinként kezeli, és felírja az 1.11 és 1.12 egyenletet.

$$d_i = d_1 + c \cdot \tau_{1,i} = d_1 + d_{1,i} \quad (1.11)$$

$$d_i = \sqrt{(x_i - x_T)^2 + (y_i - y_T)^2 + (z_i - z_T)^2} \quad i = 1 \dots M \quad (1.12)$$

Ahol:

- $\tau_{1,i}$ jelterjedési időkülönbség, amely reprezentálja azt, hogy az 1 anchorhoz képest mennyivel több idő alatt jut el a jel az i-edik anchorhoz
- $d_{1,i}$ a $\tau_{1,i}$ -hez tartozó távolságkülönbség
- c a fénysebesség vákuumban
- d_i tag és az i-edik anchor közötti távolság
- M az anchorok száma
- x_i, y_i, z_i az anchorok, az x_T, y_T, z_T pedig a tag koordinátái

Rendezéssel, átalakítással és algebrai azonosságot felhasználva jut el egy mátrixos alakban felírható lineáris, inhomogén egyenletrendszerre (más szóval mátrixegyenletre), amely mátrixos alakban látható az 1.13 egyenletben.

$$\underbrace{\begin{bmatrix} x_1 - x_2 & y_1 - y_2 & z_1 - z_2 & -c\tau_{12} \\ x_1 - x_3 & y_1 - y_3 & z_1 - z_3 & -c\tau_{13} \\ \vdots & \vdots & \vdots & \vdots \\ x_1 - x_M & y_1 - y_M & z_1 - z_M & -c\tau_{1M} \end{bmatrix}}_{\underline{\underline{A}}} \cdot \underbrace{\begin{bmatrix} x_T \\ y_T \\ z_T \\ d_1 \end{bmatrix}}_{\underline{\underline{X}}} = \frac{1}{2} \cdot \underbrace{\begin{bmatrix} x_1^2 - x_2^2 + y_1^2 - y_2^2 + z_1^2 - z_2^2 + c^2\tau_{1,2}^2 \\ x_1^2 - x_3^2 + y_1^2 - y_3^2 + z_1^2 - z_3^2 + c^2\tau_{1,3}^2 \\ \vdots \\ x_1^2 - x_M^2 + y_1^2 - y_M^2 + z_1^2 - z_M^2 + c^2\tau_{1,M}^2 \end{bmatrix}}_{\underline{\underline{B}}} \quad (1.13)$$

Az 1.13 megoldása akkor létezik, ha $\underline{\underline{B}}$ benne van az $\underline{\underline{A}}$ együtthatómátrix oszlopterében. A gyakorlati eseteket tekintve azonban rendkívül gyakran előfordul, hogy a teljes egészében mérések által meghatározott az 1.13 mátrixegyenlet inkonzisztens, azaz ellentmondásos. Ilyen esetben egy működőképes megoldás, ha kiválasztjuk $\underline{\underline{A}}$ oszlopteréből a $\underline{\underline{B}}$ -t legjobban közelítő vektort, és azzal oldjuk meg a mátrixegyenletet. Az ilyen megoldásokat az eredeti egyenletrendszer optimális megoldásainak, vagy a legkisebb négyzetek elve szerinti megoldásainak nevezzük, amelyet az 1.14 egyenlet fejez ki. Itt megjegyzendő, hogy az matematikailag a módszer számára szükséges anchorok száma nem definit, azonban a korábbi használható szimulációs eredmények azt mutatják, hogy legalább 5 anchor alkalmazandó.

$$\widehat{\underline{\underline{X}}} = \underline{\underline{A}}^+ \cdot \underline{\underline{B}} \quad (1.14)$$

A Foy módszert Wade H. Foy publikálta először 1976-ban, mint Taylor-soros közelítő módszert TDoA pozícióbecsléshez [27]. A módszer azon alapszik, hogy lineárizálja elsőrendű Taylor-sorával a nemlineáris távolságkülönbség függvényt, amelyet az 1.15 egyenlet fejez ki.

$$d_i - d_1 = c\tau_{1,i} = \|\mathbf{p}_i - \mathbf{p}_T\| - \|\mathbf{p}_1 - \mathbf{p}_T\| = f_i(\mathbf{p}_T) \quad i = 2 \dots M \quad (1.15)$$

Ahol:

- \mathbf{p}_i az i -edik anchor pozíciója 3 dimenzióban, amely felbontva: $[x_i, y_i, z_i]^T$
- \mathbf{p}_T a tag pozíciója 3 dimenzióban, amely felbontva: $[x_T, y_T, z_T]^T$

Egyenlet átrendezés, és más formába alakítás útján jutunk az 1.16 egyenletre, amelyben \underline{C} együtthatómátrix koordinátánkénti parciális deriváltakból áll (ismeretebb nevén Jacobi-mátrix), a \underline{D} pedig a mérésekből ismert és adottnak feltételezett függvényérték különbségekből álló vektor.

$$\underbrace{\begin{bmatrix} \frac{\partial f_2(x_{T,0})}{\partial x_T} & \frac{\partial f_2(y_{T,0})}{\partial y_T} & \frac{\partial f_2(z_{T,0})}{\partial z_T} \\ \frac{\partial f_3(x_{T,0})}{\partial x_T} & \frac{\partial f_3(y_{T,0})}{\partial y_T} & \frac{\partial f_3(z_{T,0})}{\partial z_T} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_M(x_{T,0})}{\partial x_T} & \frac{\partial f_M(y_{T,0})}{\partial y_T} & \frac{\partial f_M(z_{T,0})}{\partial z_T} \end{bmatrix}}_{\underline{C}} \cdot \underbrace{\begin{bmatrix} \Delta x_T \\ \Delta y_T \\ \Delta z_T \end{bmatrix}}_{\Delta \mathbf{p}_T} \cong \underbrace{\begin{bmatrix} f_2(\mathbf{p}_T) - f_2(\mathbf{p}_{T,0}) \\ f_3(\mathbf{p}_T) - f_3(\mathbf{p}_{T,0}) \\ \vdots \\ f_M(\mathbf{p}_T) - f_M(\mathbf{p}_{T,0}) \end{bmatrix}}_{\underline{D}} \quad (1.16)$$

Lényegében ebben a formában már egy lineáris egyenletrendszerként kezelhető (eltekintve attól, hogy valójában közelítésen alapszik), amelynek megoldásával egy tag általi elmozdulás vektort ($\Delta \mathbf{p}_T$) kapunk eredményül. Az 1.16 mátrixegyenlet megoldását a legkisebb négyzetek elve szerint adja az alábbi egyenlet:

$$\Delta \hat{\mathbf{p}}_T \cong \underline{C}^+ \cdot \underline{D} \quad (1.17)$$

Az ismeretlen tag pozíció pedig a következő formában fogalmazható meg:

$$\hat{\mathbf{p}}_T = \mathbf{p}_{T,0} + \Delta \hat{\mathbf{p}}_T \quad (1.18)$$

A közelítés a tag 0. úgynevezett kezdeti pozíciója környezetében történik. Ez egyben az algoritmus hátrányos tulajdonsága, miszerint egy alkalmazott rendszerben ismereni kell a tag kezdeti pozícióját $\mathbf{p}_{T,0}$. Itt fontos megjegyezni, hogy az 1.18 egyenlet az algoritmusban egy frissítési egyenletként fogható fel, azaz az i -edik iterációban kiszámolt \mathbf{p}_T vektor a következő $(i + 1)$ -edik iterációban $\mathbf{p}_{T,0}$ szerepét tölti majd be, és ezzel egyúttal az \underline{C} együtthatómátrixot, és a mérési eredményként kapott \underline{D} vektort is frissíteni kell. Emögött az a szemléletes kép áll, miszerint egy függvényt a pontbeli Taylor-soros közelítése mindössze az a pont környezetében írja le meghatározott hibával terhelt. Minél jobban távolodunk az a ponttól, annál inkább növekszik a közelítés hibája. Az a pont esetünkben egy térbeli pontként fogható fel, így az algoritmus akkor rendelkezik kis hibával, ha az elmozdulásvektor elegendően kicsiny. Ez a követelmény pedig 2 esetben tartható be. Az egyik lehetőség, hogy tag sebesség-idő függvénye nem vesz fel nagy értékeket, a másik lehetőség pedig az, hogy az algoritmus pozíciófrissítési sebessége elegendően nagy.

Bertrand T. Fang bizonyította 1989-ben először, hogy ha a TDoA által felhasznált egyenletekben az anchorok pozícióit tekintve néhány kényszert alkalmazunk, akkor a hiperbolikus nemlineáris egyenletrendszerből átalakítások sorozatával koor-

dinátánként egymástól független másodfokú egyenletre juthatunk [28]. A módszer használhatóságának tehát feltétele, hogy az anchorok által kifeszített térbeli struktúra speciális alakú, melyet a következő 4 feltétel fogalmaz meg:

$$\mathbf{p}_1 = [0, 0, 0]^T \quad (1.19)$$

$$\mathbf{p}_2 = [x_2, 0, 0]^T \quad (1.20)$$

$$\mathbf{p}_3 = [x_3, y_3, 0]^T \quad (1.21)$$

$$\mathbf{p}_4 = [x_4, y_4, z_4]^T \quad (1.22)$$

A fenti 4 feltétel egyúttal azt is jelenti, hogy 4-nél több anchor általános esetben nem ad többletinformációt. Ennek a módszernek a levezetése és a kapott másodfokú egyenletek megoldásának dokumentumba szerkesztett alakja koordinátánként nagyjából 1 oldalt igényelne, így ennek teljes bemutatásától eltekintek, mindössze a tag x koordinátájára vonatkozó másodfokú egyenlet kerül bemutatásra:

$$\begin{aligned}
& x_T^2 \cdot \left[\frac{1}{z_4} \cdot \left(4d_{1,2}^2 \cdot \left[y_4 \cdot \left(\frac{x_3}{y_3} - \frac{x_2 d_{1,3}}{y_3 d_{1,2}} \right) - \frac{x_4}{z_4} + \frac{x_2 d_{1,4}}{x_4 d_{1,2}} \right]^2 \right) + \right. \\
& \qquad \qquad \qquad \left. + 4d_{1,2}^2 \cdot \left(\frac{x_3}{y_3} - \frac{x_2 d_{1,3}}{y_3 d_{1,2}} \right)^2 - 4x_2^2 + 4d_{1,2}^2 \right] + \\
& + x_T \cdot \left[4x_2^3 - 4x_2 d_{1,2}^2 + 8d_{2,1}^2 \cdot \left[\frac{y_4}{z_4} \cdot \left(\frac{x_3}{y_3} - \frac{x_2 d_{1,3}}{y_3 d_{1,2}} \right) - \frac{x_4}{z_4} + \frac{x_2 d_{1,4}}{x_4 d_{1,2}} \right] \cdot \right. \\
& \qquad \qquad \qquad \cdot \left[\frac{x_4^2 + y_4^2 - d_{1,4}^2 - y_3 y_4 + d_{1,2} d_{1,4}}{2z_4} + \frac{-x_3^2 y_4 + y_4 d_{1,3}^2}{2y_3 z_4} + \right. \\
& \qquad \qquad \qquad \left. + \frac{-x_2^2 d_{1,4} - y_4 d_{1,2} d_{1,3}}{2y_3 z_4} + \frac{x_2^2 y_4 d_{1,3}}{2y_3 z_4 d_{1,2}} + \frac{z_4}{2} \right] - \\
& \qquad \qquad \qquad \left. + 8d_{1,2}^2 \cdot \left(\frac{x_3}{y_3} - \frac{x_2 d_{1,3}}{y_3 d_{1,2}} \right) \cdot \left(\frac{y_3}{2} + \frac{x_3^2 - d_{1,3}^2 + d_{1,2} d_{1,3}}{2y_3} - \frac{x_2^2 d_{1,3}}{2y_3 d_{1,2}} \right) \right] + \\
& + \left[4d_{1,2}^2 \cdot \left(\frac{y_3}{2} + \frac{x_3^2 - d_{1,3}^2 + d_{1,2} d_{1,3}}{2y_3} - \frac{x_2^2 d_{1,3}}{2y_3 d_{1,2}} \right)^2 - x_2^4 - d_{1,2}^4 + \right. \\
& \qquad \qquad \qquad + 4d_{1,2}^2 \cdot \left(\frac{x_4^2 + y_4^2 - d_{1,4}^2 - y_3 y_4 + d_{1,2} d_{1,4}}{2z_4} + \frac{-x_3^2 y_4 + y_4 d_{1,3}^2}{2y_3 z_4} + \right. \\
& \qquad \qquad \qquad \left. + \frac{-x_2^2 d_{1,4} - y_4 d_{1,2} d_{1,3}}{2y_3 z_4} + \frac{x_2^2 y_4 d_{1,3}}{2y_3 z_4 d_{1,2}} + \frac{z_4}{2} \right) + 2x_2^2 d_{1,2}^2 \left. \right] = 0 \quad (1.23)
\end{aligned}$$

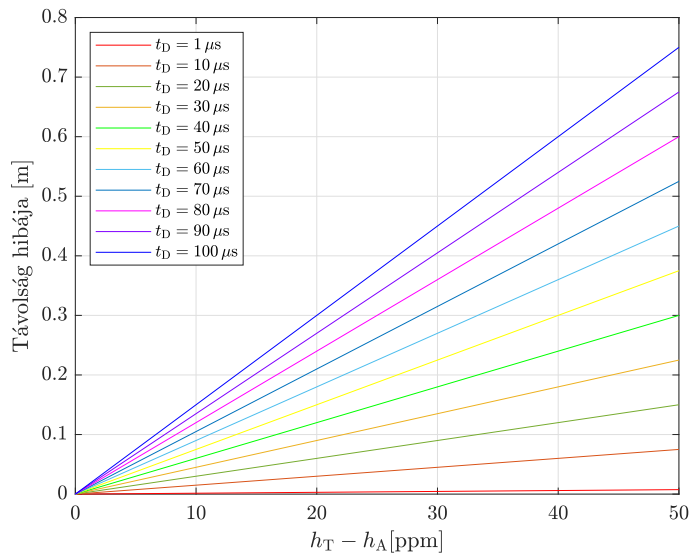
1.4. A pozicionálás pontosságát befolyásoló jelenségek

A bemutatott módszerek időmérésen alapulnak. Az egyes anchorok és tag időmérési pontosságát pedig a saját órajelük pontossága határozza meg. A modulokat általában kristályoszillátor működteti, melynek frekvenciájának a hibáját ppm-ben (part per million) szokás megadni, és a névleges frekvenciától maximum ekkora hibával tér el. Ez függhet tápfeszültségtől, hőmérséklettől, de akár áramkörü elrendezéstől is. Amennyiben két anchor órajele különbözik, úgy a mérési eredményben is megjelenik a különbség mértékével kapcsolatban álló hiba. A kapcsolat jellege attól függ, hogy milyen az alkalmazott pozicionáló módszer. TWR esetén levezetés nélkül az 1.24

összefüggés kapható [21]:

$$t_{\text{hiba}} \approx \frac{1}{2} t_D (h_T - h_A) \quad (1.24)$$

Ahol t_D az anchor késleltetése, h_T és h_A pedig a tag és anchor frekvenciájának hibája ppm-ben. Az 1.24 egyenlet alapján készített 1.13 ábrán látható az, hogy néhány konkrét t_D érték mellett mekkora a távolságmérés hibája annak függvényében, hogy mennyire különböznek a ppm-ben mért frekvenciahibák. Jól látható, hogy például ha a hibák különbsége 20 ppm, akkor 100 μs késleltetés esetén a távolságmérés hibája 0,3m. Ennél nagyságrendekkel jobb eredmény kapható abban az esetben, ha az



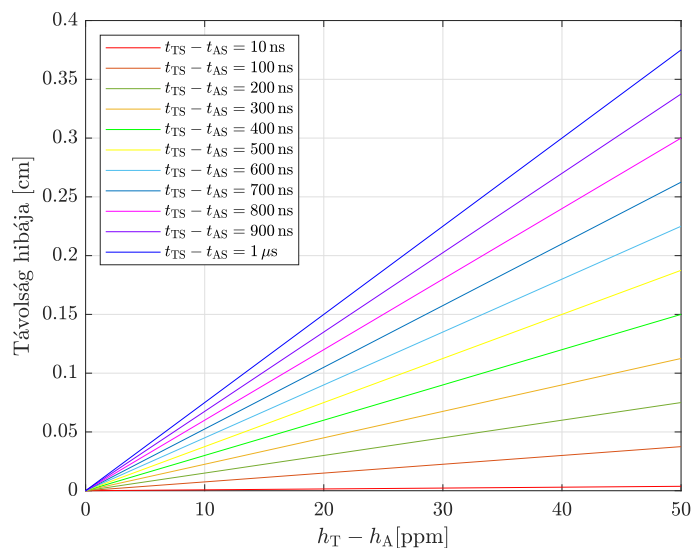
1.13. ábra. Mért távolság hibája a tag és anchor közti frekvenciahiba-különbség függvényében TWR esetén

alkalmazott módszer SDS-TWR. Levezetés nélkül ekkor a következő összefüggés adódik [21]:

$$t_{\text{hiba}} \approx \frac{1}{4} (t_{\text{TS}} - t_{\text{AS}}) (h_T - h_A) \quad (1.25)$$

Ahol h_T és h_A továbbra is a tag és anchor frekvenciájának hibája ppm-ben, t_{TS} pedig az az időtartam, amíg a tag várakozik az anchor válaszára. t_{AS} pedig azt az időtartamot jelöli, amíg az anchor várakozik a tag válaszára. Az 1.14 ábra szemlélteti távolságmérés hibáját. Láthatóan sokkal jobb a helyzet, mivel feltételezhetően t_{AS} és t_{TS} egymáshoz közel álló értékek. Ennek szemléletes magyarázata az, hogy a tag és az anchor által várt időtartamok nagyságrendileg ms közeli értékek. A frekvenciakülönbség eleve ppm-ben értelmezett, amely frekvenciakülönbséget ha ms nagyságrendű időtartamig tekintünk, akkor gyakorlatilag a hiba hibája képződik le.

Az 1.13, és az 1.14 ábrákból kivehető, hogy legjobban akkor lehet csökkenteni a



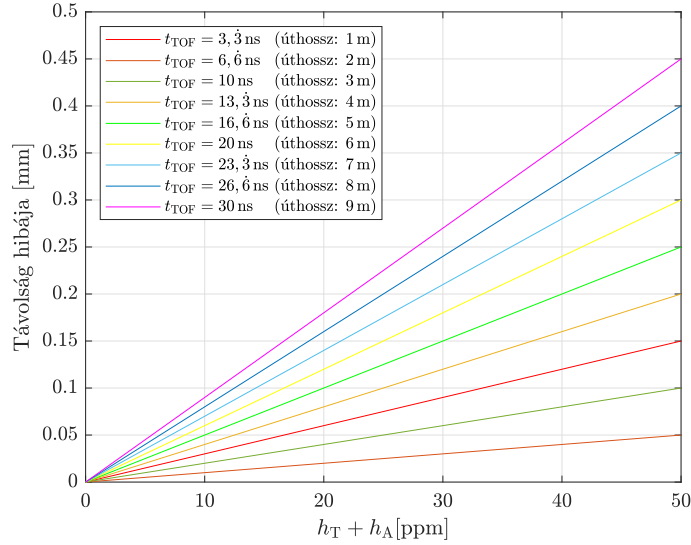
1.14. ábra. Mért távolság hibája a tag és anchor közötti frekvenciahiba-különbség függvényében SDS-TWR esetén

mérési hibát (illetve ez a leginkább kézenfekvő megoldás), ha a frekvenciakülönbség minimális. Ez elérhető a korábban említett kristályoszillátor finomhangolással, melyet egy trimmer kondenzátorral tudunk elérni. Egy másik megoldás a TCXO (Temperature Compensated Crystal Oscillator - hőmérséklet kompenzált kristályoszillátor). Az ilyen oszcillátorok tartalmaznak egy kis kiegészítő áramkört, amellyel akár 1 ppm alá is csökkenhet a frekvenciahiba. Sajnos ez jelentősen megnövelheti a modul árát, így olcsóbb termékekben nem jellemző a TCXO használata. Ha nincs lehetőség TCXO használatára, és a modul belső kristályoszillátora nem trimmelhető, akkor szoftveres úton kompenzálható bizonyos mértékben a frekvenciahiba.

Az 1.15 ábra, és a hozzá tartozó 1.26 egyenlet kifejezi azt az esetet, amely esetben a jel terjedési idejétől is függ az időmérési (és ezzel párhuzamosan a távolságmérési) hiba. Az 1.26 egyenlet egy durva felső becslést alkalmaz annak érdekében, hogy egyszerűen ábrázolható lehessen az összefüggés. A durva felső becslés ellenére az ASDS-TWR sémának a korábban említett használatára vonatkozó ajánlás helyénvaló.

A terjedési időn kívül az időpont feljegyzéséig eltelik néhány 100 ps, amely az antenna késleltetéséből adódik (noha ebbe bele kell érteni az analóg illetve digitális feldolgozási időt is). Ez a késleltetés szoftveres úton kompenzálható oly módon, hogy a minden feljegyzett időpont egy konstans értékkel inkrementálódik, amelynek értéke referenciamérések útján kapható meg.

A közvetlen rálátás (Line Of Sight - LOS) esetében olyan útról van szó, ahol a jel reflexió nélkül megérkezik az adótól a vevőig. A jelterjedési sebesség konstans és



1.15. ábra. Mért távolság hibája a tag és anchor közti frekvenciahiba-összeg függvényében ASDS-TWR esetén

$$t_{\text{hiba}} \approx t_{\text{TOF}} \cdot (h_T + h_A) \quad (1.26)$$

értéke közelítőleg a fény sebességével egyenlő. Abban az esetben ha ez nem teljesül, akkor beszélünk NLOS-ról (Non Line Of Sight). Ekkor a jel közvetlen úton nem érkezik be a vevőhöz, és csak egy másik útvonalon érkező üzenet lesz csak feldolgozható. Az UWB kulcsfontosságú előnye éppen abban rejlik, hogy időtartományban a közvetlen úton érkező pulzustól a nem közvetlen úton érkező pulzusokat meg tudjuk különböztetni, ahogy az 1.3 ábrán látható. Ez természetesen nem jelenti azt, hogy nem szükséges közvetlen rálátás UWB esetén, mindössze annyit jelent, hogy amennyiben vannak NLOS komponensek is a beltéri környezet miatt, akkor azok nem érzékeltek zavaró hatást.

1.5. A technológia előnyei és hátrányai

Az UWB technológia kulcs elemei pozicionálást figyelembe véve a következők:

- Nagy pontosságú pozíciómeghatározás (cm...dm nagyságrend), melynek köszönhetően például akár drónok térbeli útvonalának nyomon követésére nyílik lehetőség, illetve beltéri pozicionálás esetén a helyalapú szolgáltatások finomítása lehetséges.
- Nem zavar más technológiákat a spektrális átlapolódás (Wi-Fi, Bluetooth, stb.), illetve a zajtűrő képessége is kiemelkedő.

- Az időben keskeny pulzusok következtében a többutas terjedés hatása ellen fokozott tűréssel rendelkezik.
- Kis teljesítményű adók segítségével működtethető a technológia, ezáltal lehetőség nyílik mobilis eszközökben való alkalmazásra.
- Az UWB modulok előállítási költsége alapján gazdaságilag kedvező megoldást kínál pozicionálásra.

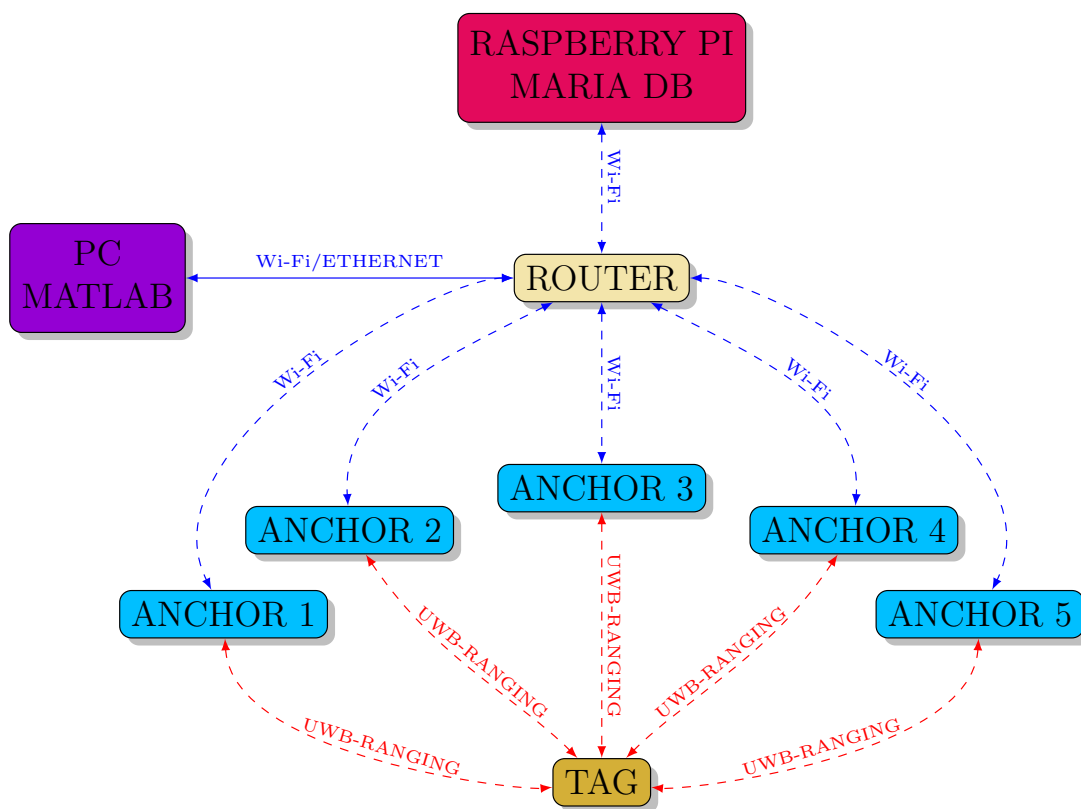
Az UWB technológia kedvezőtlen tulajdonságai pozicionálást figyelembe véve az alábbiak:

- Széles spektrumon történő működés miatt átlapolódik más technológiákkal, ezért szigorú előírások vonatkoznak a kibocsátott maximális teljesítménysűrűségre.
- Léteznek módszerek a közvetlen rálátás hiánya által okozott negatív hatás csökkentésére [29], viszont az UWB közvetlen rálátás esetén hatékony igazán [25]. Ugyanakkor a gyakorlatban ez elmondható minden olyan technológiáról amely nagy pontosságú helymeghatározást kínál.

1.6. Rendszerterv

Az 1.16 ábrán látható ezen diplomatervhez tartozó rendszerterv, amely vázlatosan megjeleníti az egyes fizikailag különálló objektumokat, illetve azok kapcsolatainak jellegét. A folyamat működése a következő. A tag üzenetváltást kezdeményez az első anchorral, melynek eredménye az anchor oldalán a köztük lévő távolság. Ezt követően a következő anchorokkal hasonló operáció zajlik le. Egy anchor miután kiszámolta a távolságot, azt Wi-Fi-n keresztül eljuttatja a Raspberry Pi-n futó adatbázisba. Végezetül egy PC-n futó MATLAB kód hozzáfér a Raspberry Pi adatbázisához, ahonnan lekérdezi a meghatározott távolságokat, végül egy grafikus kezelői felületen található ablakban megjeleníti egy lokális koordináta-rendszerben a pozicionálás eredményét.

A következő fejezetekben a rendszerterven látható objektumokhoz tartozó tervezési és építési lépések kerülnek bemutatásra.



1.16. ábra. Rendszerterv

2. fejezet

Hardver

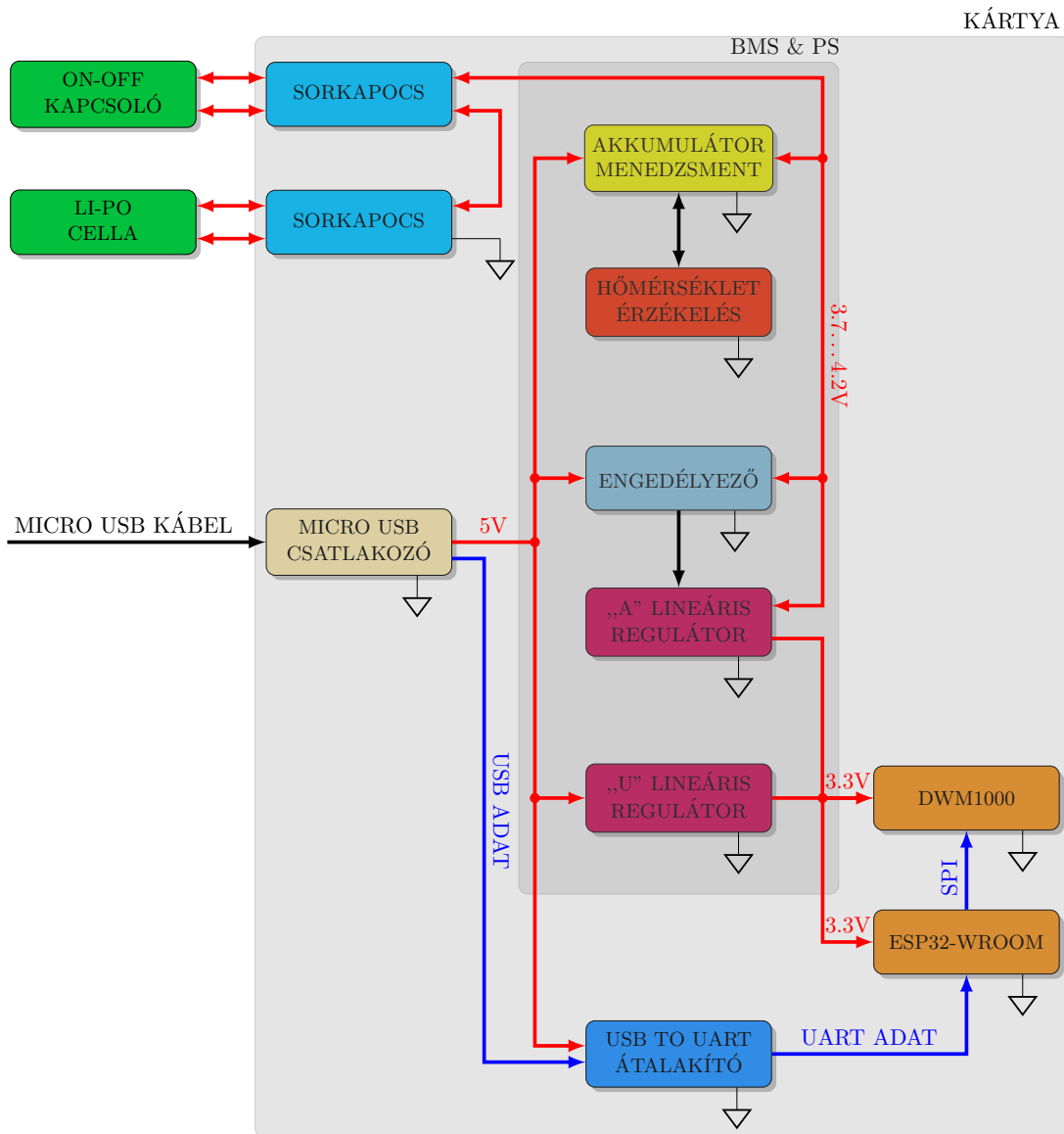
Ebben a fejezetben a diplomatervhez fűződő hardvertervezés és építés kerül bemutatásra. A tervezés teljes körű bemutatása sajnálatos módon egyértelműen meghaladja a diplomaterv által nyújtott kereteket az oldalszámot tekintve, ezért a dokumentációt alkotó egységek saját belátás szerint súlyozásra kerülnek. Az alkatrész választási szempontok, a kapcsolási rajzok, tervezési lépések bemutatásának sorrendje az alább látható 2.1 ábrán bemutatott blokkvázlat elemeinek megfelelően csoportosítva alakul. Az áramkörtervezéshez a [KiCad](#) 6.02 verziószámú ingyenes áramkörtervező szoftvert használtam, a kapcsolási rajzok illetve nyomtatott huzalozási tervek vektorgrafikus megjelenítéséhez a szintén ingyenes [Inkscape](#) grafikus szoftverre esett a választásom, korábban szerzett rendkívül kedvező tapasztalatok miatt [30] [31]. A fejezetben található többi ábrát, diagramot TikZ vektor grafikát leíró nyelv segítségével készítettem, annak $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ kompatibilitása, és minőségével összefüggő személyes véleményem okán.

2.1. Áramkörtervezés

2.1.1. Fejlesztői kártya felépítése

A diplomaterv írásának időintervalluma alatt elhúzódó alkatrészhiány globális jellege okán minden áramkörrel kapcsolatos tervezési lépés, gondolat magában hordozza a költség, alkatrész mennyiség, nyomtatott áramkör mennyiség minimalizálását. Ebből kifolyólag a diplomaterv keretében megtervezett anchor és tag ugyanazon saját tervezésű fejlesztői kártyára épül. Az utólag beültetett alkatrészek határozzák meg, hogy melyik fejlesztői kártya működhet kedvező feltételekkel úgy mint tag illetve anchor. A 2.1 ábrán látható szürke háttér jelöli ki a kártya határvonalát, illetve a kártyán belül egy újabb szürke blokk foglalja magába az akkumulátor menedzsmenthez és a tápellátáshoz szorosan fűződő elemeket. A kártya határán kívül látható

diagram elemek (On-Off kapcsoló, Li-Po cella) fizikailag kapcsolódnak majd a kártyához, és a működés bemutatása szempontjából előnyös itt már feltüntetni, de a kapcsolási rajzokon nem szerepelnek. Későbbi alfejezetben látható lesz a tényleges kapcsolódási mód, de ez már az építés kategóriájába tartozik. A 2.1 ábra által rep-



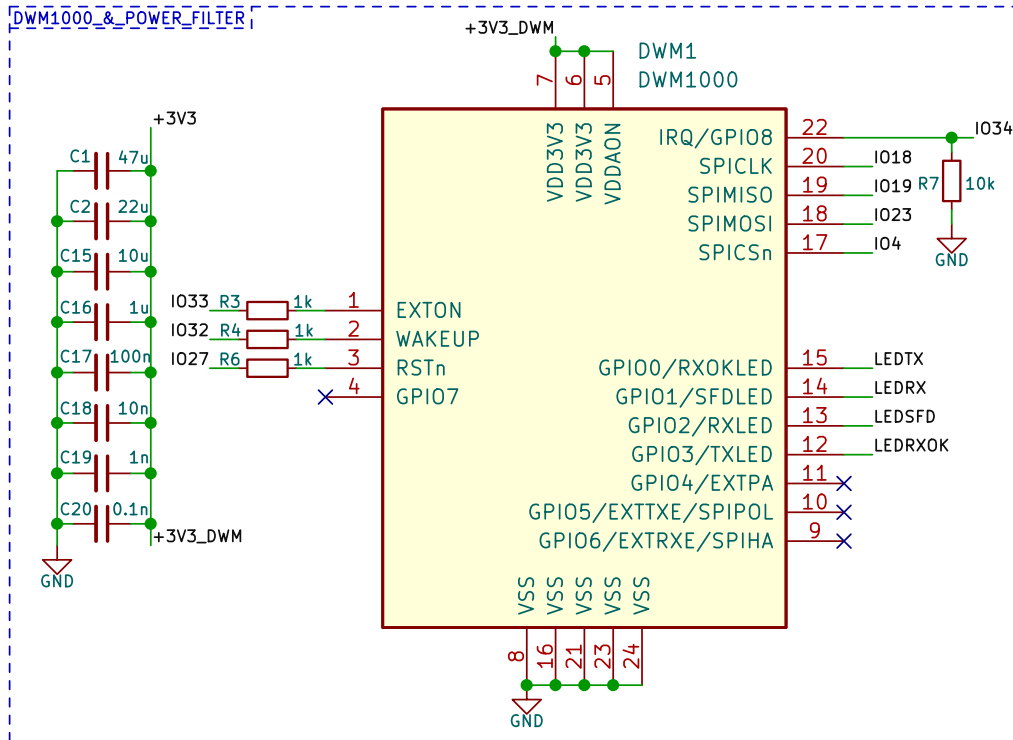
2.1. ábra. Fejlesztői kártya blokkvázlata

rezentált kártya működése a következő. A kártya tápellátása alapvetően egy Li-Po cella, amely névlegesen 3,7V-os feszültséget szolgáltat. A negatív kivezetés közös pontra kerül, a pozitív kivezetés rögtön egy billenő kapcsolón keresztül érkezik meg az akkumulátormenedzsmen és tápfeszültség-előállító rendszerhez. Ennek oka, hogy az akkumulátort teljesen le lehessen választani a többi elektromos alkatrésztől anélkül, hogy a sorkapcsoknál bontani kellene az összeköttetést. Ezt követően a működési mechanizmust meghatározza, hogy egy micro USB-n keresztül csatlakozó tápforrás

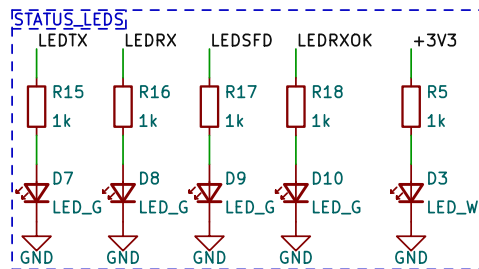
jelen van-e vagy sem. Ha nincs jelen, akkor a teljes blokkvázlat jelentősen egyszerűsödik, ugyanis ekkor az „A” lineáris regulátort az engedélyező elem bekapcsolja, és így előáll az ESP32 és DWM1000 modulok működéséhez szükséges 3,3 V-os feszültség. Ez az üzemi állapot az amely a tag esetében a Li-Po celláról való üzemelés esetén áll fent. Amennyiben jelen van egy külső tápforrás a micro USB csatlakozón keresztül, akkor az „U” lineáris regulátor fog a 3,3 V-os feszültségű vonalra táplálni, az „A” lineáris regulátor pedig az engedélyező által letiltásra kerül. Ezzel a módszerrel biztosítva van az, hogy egyszerre a két lineáris regulátor nincs bekapcsolt állapotban. A második funkciója a külső tápforrásnak, hogy ekkor az akkumulátormenedzsment blokk is működésbe lép. Ez biztosítja az akkumulátor számára megfelelő töltési karakterisztikát, illetve túlmelegedés védelmet. A harmadik funkciója a micro USB csatlakozónak az, hogy az ESP32 modult programozni ezen (és az átalakítón) keresztül is lehessen.

2.1.2. DWM1000 modul

A tag az anchorokkal UWB alapon kell, hogy kommunikáljon. Ehhez szükséges egy erre alkalmas alkatrész, a választás pedig a fentebb leírt gazdasági okból kifolyólag egy komplett modulra esett, amelynek üzemeltetéséhez mindössze a tápfeszültség és a vezérlő utasítások szükségesek. Korábbi önálló laboratórium keretében végzett munka nyomán tapasztalatot szereztem UWB modul tervezésében és építésében, most azonban a idő hiányában a Qorvo (korábban Decawave) által gyártott DWM1000-es modulra esett a választás, amelynek a 2.2 ábrán látható a kapcsolási rajza. Ez a modul adatlap alapján kompatibilis az IEEE 802.15.4-2011 UWB szabvánnyal, alkalmas 2 utas üzenetváltási algoritmusok megvalósítására (ranging), valamint SPI interfészen tud kommunikálni az őt vezérlő egységgel. A modul leírásához bizonyos szinten szükséges ismerni a modulban található DW1000 integrált áramkör 2.4 ábrán látható állapotdiagramját, ezáltal a következőkben előforduló állapotok megnevezése közérthetőbb lesz. Tápfeszültség igénye 3,3 V, az adatlap alapján minimum 2,8 V szükséges neki, viszont korábbi tapasztalatokra támaszkodva stabilan akkor üzemel a modulban található integrált áramkör (DW1000), ha a 3,3 V-ot stabilan kapja. A stabilitásban egyrészt a később ismertetett lineáris regulátor is fontos szerepet tölt be, viszont a modulhoz fizikailag közel helyezett tápszűrő kondenzátorok is lényegesek a hirtelen áramfelvételek illetve zavarok szempontjából. A 12...15 kivezetések kis méretű zöld színű LED fényforrásokat hajtanak meg, a 2.3 ábrán látható áramkorlátozó ellenállásokon keresztül. A 2.3 ábra jobb oldalán egy plusz fehér LED jelzi vissza a 3,3 V-os tápfeszültség létrejöttét. A konkrét SMD LED-ek nagy fényerejűek, ezáltal a komfortos látható fény beállításához jóval kevesebb áram szükséges. A 17...20. kivezetéseken keresztül valósul meg a modul vezérlé-

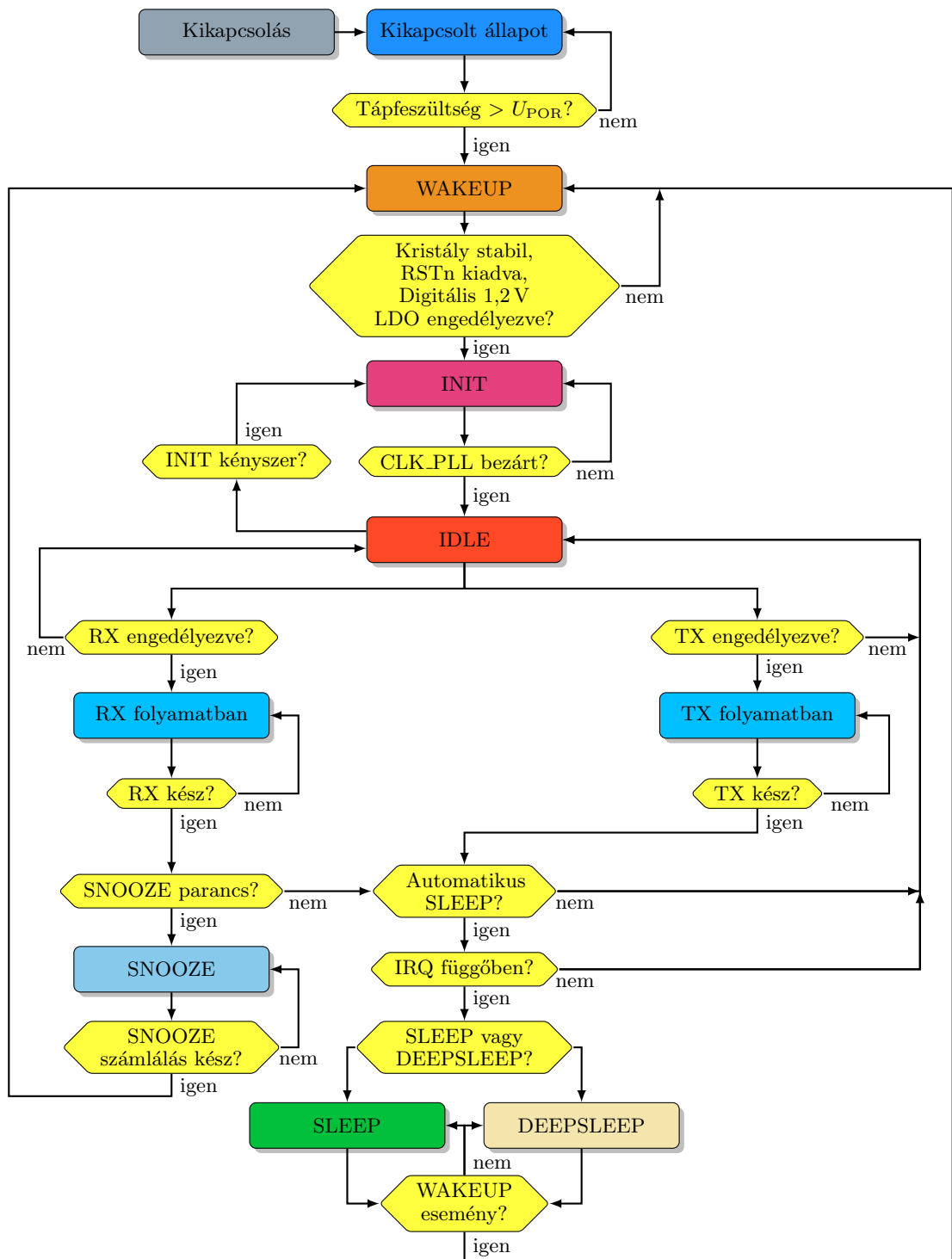


2.2. ábra. DWM1000 kapcsolási rajz



2.3. ábra. Állapot visszajelző LED kapcsolási rajz

se SPI interfészen. Hozzá tartozik még az SPI interfészhez opcionálisan a 9,10-es kivezetés. Az SPI órajel polaritás és fázis értelmezését befolyásolja. Alapértelmezésben belsőleg közös pontra (kapcsolási rajzokon GND jelöli a közös pontot) vannak húzva, ezáltal a 0. SPI mód lép életbe, vagyis a vezérlő (master) által létrehozott órajel felfutó élére történik az aktuális adat mintavételezés, és az ezt követően érkező órajel lefutó élénél történik az új adat kiküldés. A modul EXTON kivezetése külső eszköz engedélyezésére alkalmazható. Amikor a modul SLEEP vagy DEEPSLEEP üzemmódból az ébredési folyamaton keresztül halad, akkor az EXTON az első kimeneti logika amely logikai magas szintre vált (kb. 300 μ s-al a tápfeszültség megjelenése után), ezáltal például külső DCDC konvertert tud engedélyezni. Ez a kivezetés szintén össze van kötve a vezérlő ESP32 modullal. A WAKEUP bemeneti logika használható arra, hogy a modul SLEEP illetve DEEPSLEEP üzemmódból normál műveleti módba váltson, a logikai magas szint érkezése után 3 ms időbe telik



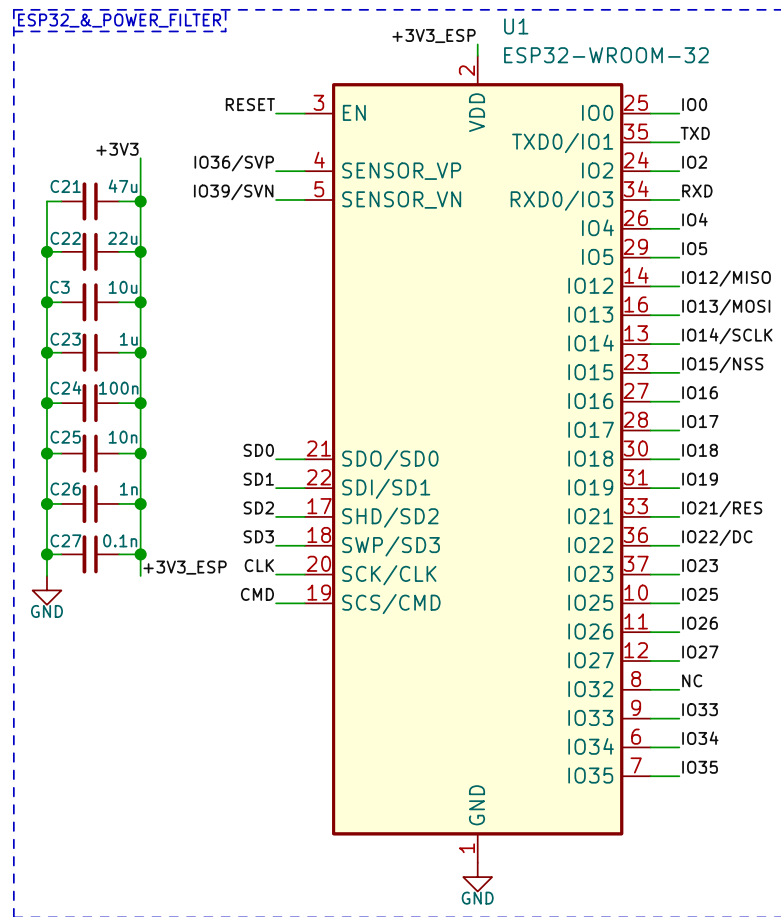
2.4. ábra. DW1000 integrált áramkör állapotdiagram

míg a modul készen áll az adásra, illetve vételre (hasonlóan a teljesen kikapcsolt állapothoz). Az RSTn aszinkron ki és bemeneti logika alapvetően a modul bekapcsolt állapotában (a bekapcsolási folyamat végétől kezdve) logikai magas szinten van. Kimenatként használható, hogy külső áramköröknek jelezze a modul a bekap-

csolási folyamat végét illetve, hogy jelezze a vezérlő irányába, hogy a modul INIT üzemmódba lépett. Bemenetként használható arra, hogy a vezérlő egy reset funkciót indítson el a modul működésében. Az IRQ kivezetés egy általános megszakítás kérés kimenete a modulnak a vezérlő felé. Alapértelmezetten magas aktív logikával működik, de ez konfigurálható. SLEEP illetve DEEPSLEEP üzemmódban ez a kivezetés lebeg, és hamis megszakításokat okozhat, emiatt került beiktatásra az R_7 lehúzó ellenállás.

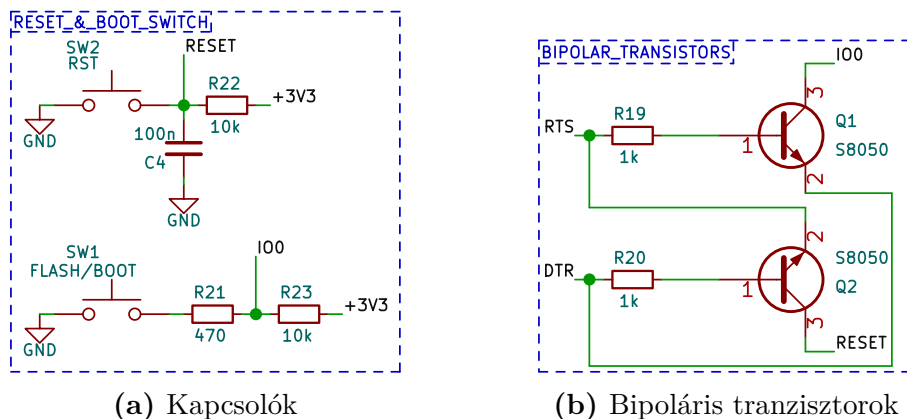
2.1.3. ESP32 modul

A DWM1000 modult vezérlő egység szintén egy komplett modul, amelynek kapcsolási rajza a 2.5 ábrán látható, melynek neve az Espressif Systems által gyártott ESP32-WROOM (továbbiakban ESP32 modul). A választást az indokolta, hogy az



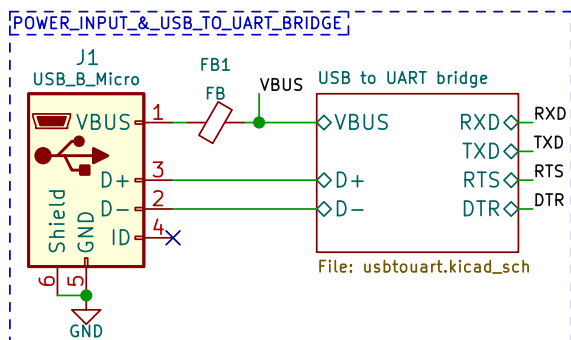
2.5. ábra. ESP32 modul kapcsolási rajz

anchoroknak Wi-Fi kapcsolatra képesnek kell lenniük, lehetőség szerint illeszkedjen a 3,3 V-os tápfeszültség rendszerbe, mindenképp rendelkezzen 1 SPI busszal, és könnyedén programozható legyen. Nem mellesleg ez a modul elérhetőségét tekintve kedvezőbb, árát tekintve pedig gazdaságosabb, mint a DWM1000 modulhoz aján-



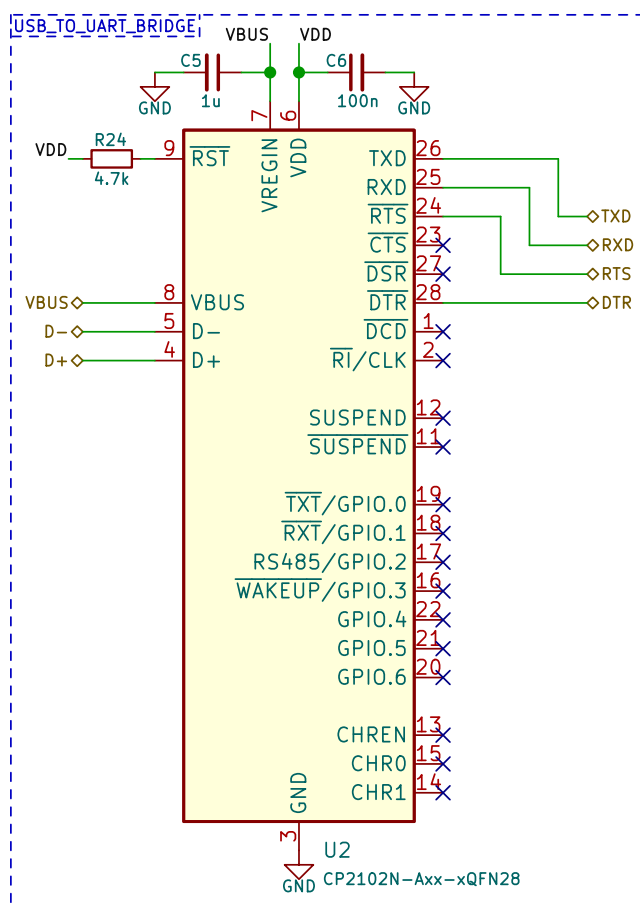
2.6. ábra. Kapcsolási rajzok

lott STM32 család tagjai. Az ESP32 modult az előzőekben ismertetett kivezetések kötik össze a DWM1000 modullal (SPI: IO18, IO19, IO23, IO4, EXTION: IO33, WAKEUP: IO32, RSTn: IO27, IRQ: IO34). Az ESP32 modul további kivezetései közül kiemelt szerepet kap a TXD és RXD, amelyeken keresztül a soros adatkommunikáció keretén belül felprogramozható a modul. Továbbá az IO0 és EN kivezetés, amelyek a 2.6a és a 2.6b ábrán látható módon kerültek felhasználásra. Lényegében az ESP32 felprogramozásához az IO0 és EN (RESET) kivezetést kell logikai alacsony szintre húzni. Erre az imént említett 2 ábra 2 lehetőséget ad. Az egyik az, hogy ha az RST és FLASH/BOOT kapcsolót egyszerre lenyomjuk. Ekkor a tűkesorra kivezetésre került (2.18a ábra) TX és RX tűsken keresztül egy külső adapter segítségével közvetlenül programozható a modul. A másik módszer, amikor a fejlesztői kártyára ültetett USB-UART (Universal Asynchronous Receiver-Transmitter) átalakító által kiadott alacsony aktív RTS és DTR jelek segítségével történik az ESP32 modul megfelelő kivezetéseinek (IO0, EN (RESET)) alacsony szintre húzása. Ez az összeállítás elterjedt megoldásnak számít az ESP32 családban, ez indokolta a használatát a fejlesztői kártya tervezése közben. A 2.7 és 2.8 ábrán látható az USB-UART átalakító. A KiCad lehetőséget nyújt hierarchikus kapcsolási rajzok alkalmazására, amellyel egyes jól elkülöníthető áramköri részeket igényesen elrejtethetők, ezáltal a gyökér lapfelületen átláthatóbb rajz születik. A 2.7 ábrán látható továbbá a fejlesztői kártyára ültetendő micro USB kapcsolási rajza, amelyről a tápfeszültség rögtön egy ferritgyöngyön van keresztül vezetve, amely hatékonyan csillapítja az USB kábelben keresztül becsatoló 9...200 MHz frekvenciatartományban jelenlévő zavarjeleket. Fontos szempont, hogy a ferritgyöngyön áthalad a fejlesztői kártya által felvett teljes áram amennyiben az akkumulátor töltődik, és még a két modul is működötve van. Ennek megfelelően a kiválasztott ferritgyöngy nagy áramot képes elviselni, a konkrét típus esetén ez pedig 1000 mA. A 2.8 ábrán látható kapcsolási rajz a hierarchikus rendszerben egy szinttel lejjebb kerül az imént említett lehető-



2.7. ábra. USB csatlakozó és USB-UART konverter blokk

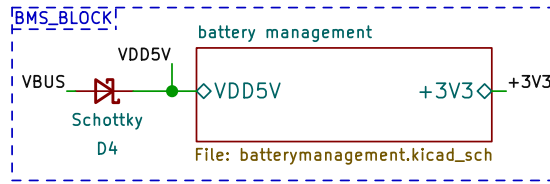
séggel élve. A Silicon Labs által gyártott CP2101 típusú integrált áramkör széles körben elterjedt, több controller család alkalmazza egyszerűsége okán. A tervezés



2.8. ábra. USB-UART konverter IC kapcsolási rajz

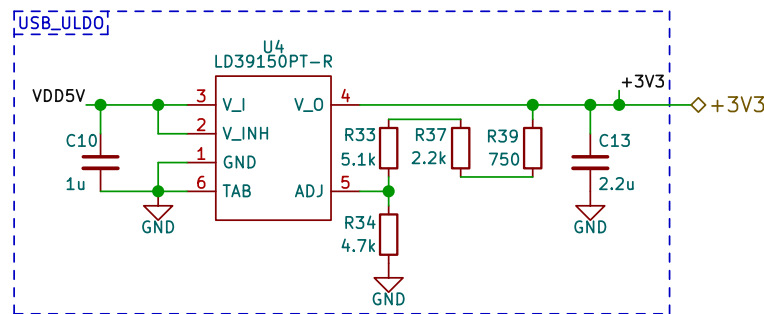
utáni alkatrészrendelési fázisban sajnálatos módon kiderült, hogy az USB-UART átalakítót gazdasági okokból kifolyólag nem beszerezhető, így ennek részletes ismeretetésétől eltekintek. Kihagyása a korábbi tervezési módszernek köszönhetően nem okoz problémát, mert rendelkezésre áll olyan külső eszköz, amellyel közvetlenül tükörön keresztül felprogramozható az ESP32 modul (erről a folyamatról egy későbbi fejezetben még lesz szó).

2.1.4. Akkumulátormenedzsment



2.9. ábra. Akkumenedzsment blokk

A 2.9 ábrán látható egy másik hierarchikus blokk, amelybe a következőkben bemutatott kapcsolási rajzok kerültek. Ez a blokk egy Micro Commercial Components (MCC) által gyártott B5819WS-TP típusú Schottky diódán keresztül közvetlenül megkapja a micro USB-n keresztül érkező tápfeszültséget. A Schottky dióda jó tulajdonsága, hogy a nyitóirányú felszültségesés amelyet okoz, az (ennél a konkrét B5819WS-TP típusnál) nem nagyobb mint 0,4...0,5 V abban az esetben ha a nyitóirányú árama elérné az 1 A-t. Feladata, hogy az USB-n kifelé ne folyjon áram, ezért blokkoló diódának is nevezik. A 2.10 ábrán látható a fejezet elején „U” lineáris regulátornak nevezett elem kapcsolási rajza. Az elnevezést motiválja, hogy a bemeneti feszültsége a micro USB felől érkezik közvetlenül. A fejlesztői kártya tervezése során



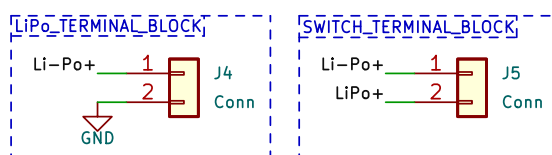
2.10. ábra. USB lineáris regulátor kapcsolási rajz

készlethiány miatt a lineáris regulátor kiválasztása volt a legnehezebb feladat. Így esett a választás végül a DL39150 típusra, amelynek (jelen tervezés szempontjából) legfontosabb tulajdonsága, hogy rendkívül alacsony a dropout feszültsége. A dropout feszültség az a feszültség amely minimálisan megjelenik a szabályozatlan bemenet és a szabályzott kimenet között ahhoz, hogy a szabályozás megfelelően működjön. Ez főként akkor fontos, amikor a 3,7...4,2 V-os celláról kell táplálni a 3,3 V-os vonalat, azaz worstcase (leginkább kedvezőtlen) esetben maximum 0,4 V-os feszültségesés a megengedett. Ehhez képest ennek a regulátornak 1 A kimeneti áram esetén a dropout feszültsége 125 mV környékén van. A bemeneten és kimeneten található C_{10} , C_{13} kapacitások mérete az adatlap alapján lett kiválasztva, de a kimeneten található kapacitás eredője jóval nagyobb, mivel a 2 modul tápszűrő kondenzátorok kapacitása

additív módon megjelenik. Működése a 2.1 ábrához írtak szerint történik, azaz amikor a micro USB csatlakozón keresztül tápfeszültséget kap a fejlesztői kártya, akkor ez a lineáris regulátor bekapcsol és a kimenetén előállítja a 3,3 V-os tápfeszültséget. Ez a konkrét típus állítható kimeneti feszültséggel rendelkezik, létezik belőle fix 3,3 V-os verzió is, viszont abból raktáron nagyon kevés raktárkészlet állt rendelkezésre a tervezés során, és a több hetes tervezési periódus alatt egy saját megfigyelés volt, hogy az állítható kimeneti feszültségű lineáris regulátorok sokkal kevésbé voltak keresett alkatrészek. A kimeneti feszültség beállítása a kimeneten található R_{34} , R_{33} , R_{37} , R_{39} ellenállások segítségével történik. Mivel ezeknek az ellenállásoknak kis tűrésűnek kell lennie, vékony réteg ellenállásokra esett a választás. Azonban ezekből lényegesebb kevesebb érték állt rendelkezésre, így soros kapcsolással került kialakításra a kívánt érték. A 2.1 egyenlet segítségével számítható a kimeneti feszültség a kapcsolási rajzban szereplő változónevekkel.

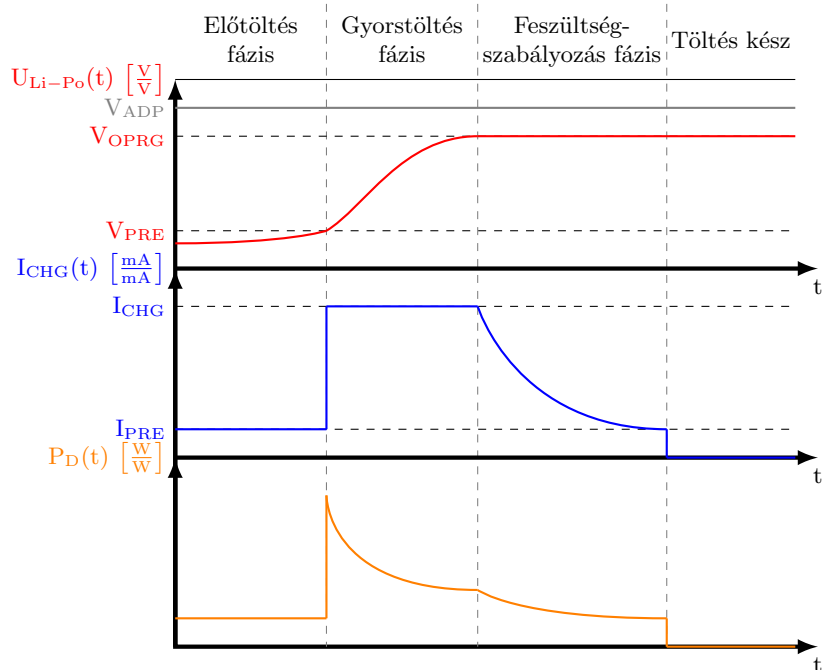
$$U_{ki} = U_{REF} \cdot \left(1 + \frac{R_{33} + R_{37} + R_{39}}{R_{34}} \right) = 3,3096 \text{ V} \quad \text{ahol : } U_{REF} = 1,22 \text{ V} \quad (2.1)$$

A 2.1 alapján kellő pontossággal megközelíti a kívánt feszültséget. Korábban leírtak szerint a micro USB csatlakozón érkező tápfeszültség egyúttal az akkumulátort is tölteni kezdi annak állapotától függően. A 2.11 ábrán látható a kapcsolási rajz lényegében a fejezet elején található blokkvázlattal egyenértékű információt ad, viszont a további kapcsolási rajzokon lévő jelölések érthetősége érdekében került feltüntetésre. A 2.13 ábrán bemutatott kapcsolási rajzon szerepel az akkumulátor menedzsmen-



2.11. ábra. Sorkapocs kapcsolási rajz

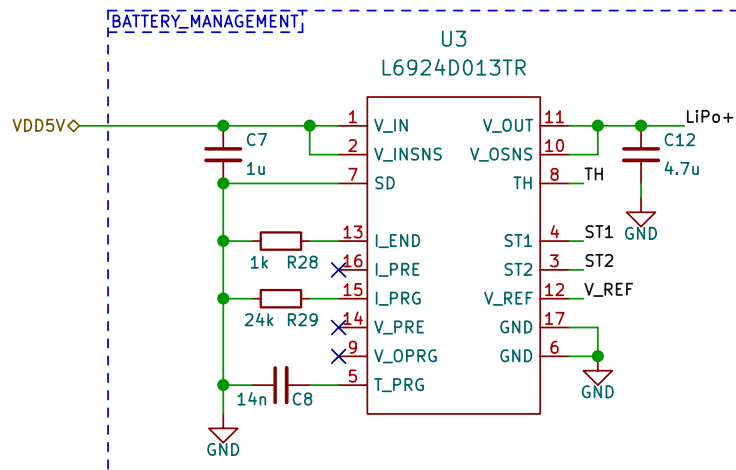
tért felelős STMicroelectronics által gyártott L6924D típusú integrált áramkör és a működéséhez szükséges alkatrészek egy része. Ehhez még hozzátartozik a 2.14a, és a 2.14b ábra, ahol a hőmérséklet érzékeléshez tartozó alkatrészek, illetve akkumulátor töltési státusz LED-ek foglalnak helyet. Az L6924D egy jó választás ezen fejlesztői kártyához, mert önállóan képes külső vezérlő nélkül megvalósítani egy 1 Li-Po cellás rendszer töltését. Rendelkezik hőmérséklet kontrollal, valamint ellenállással és kondenzátorral kontrollálható a töltési karakterisztika egyes fázisai. A 2.12 ábrán látható a töltőáramkör normalizált feszültség-, áram-, disszipáció-idő függvényekkel reprezentált töltési karakterisztikája, amely az egyes töltési fázisokban történő viszonylagos mechanizmusokat mutatja be. A 2.13 kapcsolási rajzon látható V_{IN} kivezetésen veszi fel a töltési áramot, és a V_{INSNS} kivezetésen méri a beme-



2.12. ábra. L6924D akkumulátortöltési karakterisztika

neti feszültség értékét, ahol ha 4 V fölé emelkedő értéket érzékel, akkor bekapcsol az eszköz. A kapcsolási rajzon tovább haladva az SD (ShutDown-Lekapcsolás) kivezetés ha közös pontra (GND) kerül, akkor az eszköz folyamatosan engedélyezve van amennyiben a bemeneten megjelenik a szükséges feszültség. Ebben az esetben azért volt célszerű közös pontra húzni, mert a fejlesztői kártya használati körülményei kompatibilisek a lépéssel. Nem 0-24 órás működésre történik a tervezés, azaz nem egy külső vezérlő kapcsolja majd ki és be a töltő áramkört, hanem az a hatás, hogy egy micro USB csatlakozón keresztül tölteni szeretné a felhasználó. Külső vezérlő esetében a töltő IC kikapcsolása (Integrated Circuit-Integrált Áramkör) az SD kivezetés lebegve hagyásával lehetséges. A kapcsolási rajzon soron következő 3 kivezetés I_{END} , I_{PRE} , I_{PRG} a töltési áramok értékeire vannak számszerű hatással. Az I_{PRG} kivezetést $24\text{ k}\Omega$ értékű ellenállással (kapcsolási rajzon R_{28}) GND-re húzva 7%-os pontossággal 490 mA-es maximális töltőáramot lehet beállítani. Az alkalmazott Li-Po cella adatlapján 1200 mAh olvasható, ez alapján a kb. 0,5 A áramerősség alkalmazása maximális töltőáramként egy cellakímélő beállítás. 2-szer ekkora maximális töltőáram elérése érdekében $12\text{ k}\Omega$ -os ellenállás használata javasolt. Az I_{PRE} kivezetést GND-vel összekötő ellenállás az előtöltési áram nagyságát szabályozza. Amennyiben lebegve hagyott állapotban van, úgy az előtöltési áram a maximális töltési áram 10%-a (ebben az esetben ez 49 mA). Ezt lehet kis mértékben befolyásolni mindkét irányban azáltal, hogy $62\text{ k}\Omega$ -al közös pontra van húzva a kivezetés, és ekkor 67 mA előtöltési áramot kap a cella, illetve $39\text{ k}\Omega$ -al a V_{REF} kivezetéshez

kapcsolódhat és ez esetben 35 mA előtöltési áramot kap a cella. Az I_{END} kivezetést ellenállással szintén közös pontra húzva beállítható az végső töltési áram, amelyet a töltés utolsó fázisában, a feszültségszabályozási fázisban el fog érni az exponenciális jelleggel csökkenő töltési áram, majd kikapcsol töltés. Ez 1 k Ω -os ellenállás esetén 52,5 mA értékre adódik. A V_{PRE} láb lebegve hagyásával elérhető, hogy az

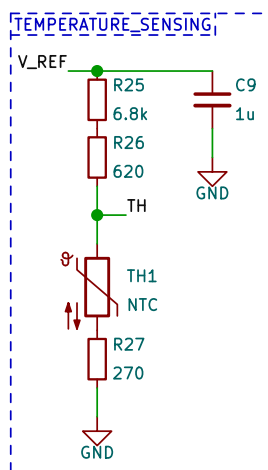


2.13. ábra. Akkutöltő IC kapcsolási rajz

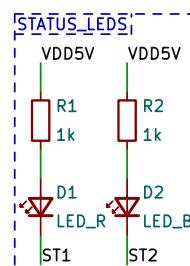
alapértelmezett feszültségküszöb 2,8 V legyen, amelynél ha a töltőáramkör V_{OSNS} kivezetésén mért cella kapocsfeszültség kisebb, akkor elindul az előtöltési fázis, az előzőekben ismertetett előtöltési árammal. Amennyiben ennél a küszöbfeszültségnél magasabb a mért cella kapocsfeszültség, úgy néhány másodperc alatt növekszik fel a töltőáram a maximális értékre, és ezzel a gyorsított fázis áll fent. A V_{OPRG} kivezetés a cella feltöltött állapotban mért kapocsfeszültségével kapcsolatos. Lebegve hagyva $V_{OUT} = 4,1 V$ a töltési fázis befejeztével a mérhető cella kapocsfeszültség, közös pontra húzva 4,2 V. Hasonlóan az eddigiekhez, a cellakímélő töltés okán a választás a 4,1 V-ra esett. A T_{PRG} kivezetést a C_8 kondenzátorral közös pontra kötve állítható a maximális töltési idő, ezen a ponton szintén a cellakímélő töltést szem előtt tartva került meghatározásra a kondenzátor értéke. Kb. 4 óra maximális töltési idő esetén 14...15 nF értékű kondenzátor szükséges, a kapcsolási rajzba 14 nF került, természetesen a számolt értékhez legközelebb eső szabványos érték a tényleges, amely 15 nF. Ez a max. töltési időt számottevően nem befolyásolja, 2,5%-os hibahatáron belüli eltérést okozhat, amely elfogadható, mivel nincs definiálva töltési idő a tervezésben. Az ST_1 , és ST_2 kivezetések felelősek a státusz visszajelző LED-ek meghajtásáért, 30 mA maximális áramot viselnek el, viszont az áramfogyasztás lehetőség szerinti minimalizálása érdekében mindössze néhány mA-es áramra lesz állítva a nyitóirányú áram. A 2.14b ábrán látható a státusz visszajelző LED-ek kapcsolási rajza, amely adatlapon található ajánlás szerint került összekötésre. Az ábrán a bal oldali LED_R kifejezés a piros LED-re utal, a jobb oldali pedig a kék-

Töltési állapot	Leírás	LED _R	LED _B
Töltés alatt	Előtöltés vagy gyorsöltési fázis áll fent.	BE	KI
Töltés kész	A töltőáram a végső töltési áram alá esik.	KI	BE
Készenlét	A bemeneti feszültség a cellafeszültség alá esik 50 mV-al.	KI	KI
Cellahőmérséklet nem megfelelő	A TH kivezetés feszültsége kívül esik a beállított feszültség ablakon	BE	BE
Hiányzó cella	A cella eltávolításra kerül	BE	BE
Időtúllépés	A töltési folyamat túllépi beállított maximális töltési időt	BE	BE

2.1. táblázat. Státusz LED állapot táblázat



(a) Termisztoros hőmérséklet-érzékelő

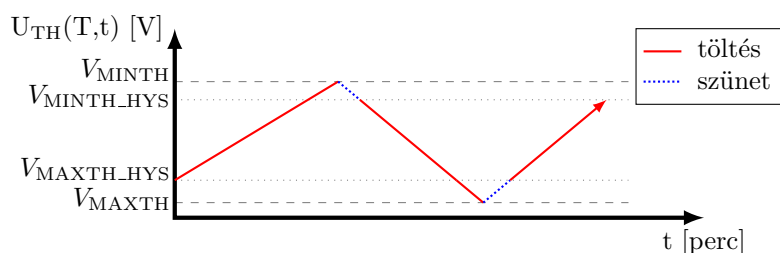


(b) Állapot-visszajelző LED

2.14. ábra. Kapcsolási rajzok

re. A 2.1 táblázat mutatja be az egyes kombinációkat, amely kombinációban ha világítanak a LED-ek az milyen szituációt jelenthet. A 2.14a ábrán látható termisztoros hőmérséklet-érzékelő kapcsolási rajza. Ennek üzemeltetéséhez használandó a töltőáramkor V_{REF} , és TH kivezetése. Röviden megfogalmazva a működési elvet: a termisztor hővezető paszta (és szivacsos öntapadó szalag) segítségével a Li-Po cellára fog rögzülni, és a termisztor hőmérséklet függő ellenállásának köszönhetően a 2.14a kapcsolási rajznak megfelelően egy hőmérséklet függő feszültséget oszt vissza a V_{REF} lábról a TH lábra. Amennyiben egy adott, szűkebb feszültség ablakban tartózkodik a visszaosztott feszültség, akkor (ha csak a hőmérsékletfüggő szabályozást tekintjük) mindenképp töltési fázisban lesz a töltőáramkör. Viszont van egy tágabb feszültség ablak aszimmetrikusan ráhelyezve a szűkebb feszültség ablakra, amely egy hiszterézis elvű működést feltételez. Lényegét tekintve arról van

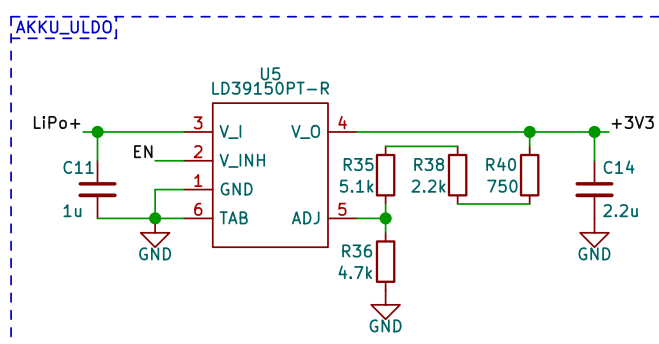
szó, hogy definiálva van egy alsó és egy felső küszöbfeszültség (ezek jelentik a tágabb feszültség ablak határait) ahol abbahagyja a a töltőáramkör a töltést. Ennek hatására természetszerű reakció lesz a Li-Po cella részéről, hogy a töltés hatására létrejövő melegedés megszűnik, és lassan egy idő után elkezd hűlni a cella. Azonban nem az elért küszöbfeszültségnél kapcsol vissza a töltés, hanem a korábban említett szűkebb ablak aktuális küszöbfeszültségénél. A 2.15 ábrán látható az imént megfogalmazott elv. A konkrét termisztor és ellenállások megválasztása a következő módon



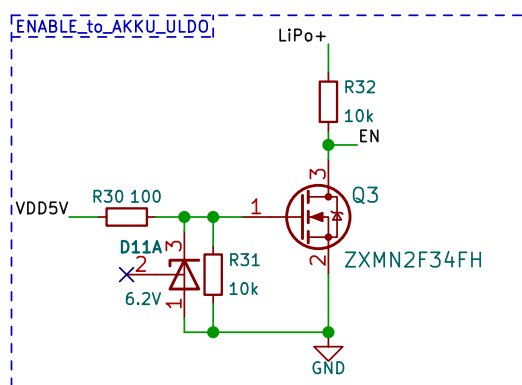
2.15. ábra. Hőmérsékletfüggő visszaosztott feszültség-idő függvény

történt. Az adatlap javaslata alapján a töltési hőmérséklet tartomány $0...50^{\circ}\text{C}$. Első lépésben kiválasztásra került a Vishay által gyártott NTCLE100E3222 termisztor, melynek 2 kulcsfontosságú paramétere van, ez a 2 paraméter a felvett ellenállás értéke 0°C illetve 50°C -on. Ezek az $R_{0^{\circ}\text{C}} = 7162\ \Omega$, és $R_{50^{\circ}\text{C}} = 793,2\ \Omega$ értékek lesznek. A következő lépésben a töltőáramkör adatlapja alapján kideríthető, hogy a 2.15 ábrán szereplő paraméterek ismertek $V_{\text{MINTH}} = 0,9\ \text{V}$, $V_{\text{MINTH_HYS}} = 0,78\ \text{V}$, $V_{\text{MAXTH_HYS}} = 0,225\ \text{V}$, $V_{\text{MAXTH}} = 0,248\ \text{V}$. Ezek alapján a feszültségosztás túlhűlés esetén pont felezi a referencia feszültséget ($V_{\text{REF}} = 1,8\ \text{V}$), tehát a TH kivezetés és a V_{REF} kivezetés közé pont $R_{0^{\circ}\text{C}} = 7162\ \Omega$ értékű ellenállás tűnik elsőre jó választásnak. Viszont vissza kell számolni, hogy túlmelegedés esetén azaz, amikor az NTC ellenállása $R_{50^{\circ}\text{C}} = 793,2\ \Omega$, akkor mekkora feszültség jut a TH láb-
ra. Ez $1,8\ \text{V} \cdot \left(\frac{793,2\ \Omega}{793,2\ \Omega + 7162\ \Omega} \right) = 0,1795\ \text{V}$ -ra adódik, amely jóval kevesebb mint az alsó feszültséghatárt jelentő $0,225\ \text{V}$. Ezáltal világos, hogy a feszültségosztó mindkét oldalához egy-egy ugyanakkora értékű sorba kötött ellenállást kell hozzáadni, hogy ezáltal emelkedjen a visszaosztott feszültség túlmelegedés esetén, viszont mivel egyenlő arányban növekszik mindkét oldal, így a túlhűlés esetére végeredményben érvényes marad a számítás. Ennek a sorba kötött ellenállásnak az értékét például egy iterációs folyamat során lehet meghatározni, amelynek eredményeképp a keresett, de forgalomban kapható ellenállás értéke $270\ \Omega$. Ekkor az NTC-vel összesen $270\ \Omega$ értékű ellenállás van sorba kötve (2.14a ábrán R_{27}), a másik oldalon pedig $7432\ \Omega$. Ez utóbbi pontosan nem kivitelezhető gazdaságos úton fix ellenállásokból, ezért két ellenállásból (2.14a ábrán R_{25} , és R_{26}) került megvalósításra $7420\ \Omega$. Ezzel

a nagyon kis mértékben módosul a feszültségosztás, de ez elhanyagolható mértékű mert a termisztor ellenállás-hőmérséklet táblázata alapján túlmelegedés esetén 50 °C környékén $\pm 5^\circ\text{C}$ változáshoz kb. 300 Ω ellenállás változás tartozik, amely nagyságrendileg nagyobb mint a tervezés során természetesen bevitte ellenállás változás. Erre egy alkalmas ellenőrzési pont az adatlap alapján, hogy a termisztor ellenállása és a vele sorba kötött ellenállás összegét véve a két szélső hőmérsékleten, és ezek hányadosát véve 7-et kell kapni. A módosult értékekkel számítva ez $\frac{7150\Omega + 270\Omega}{793,2\Omega + 270\Omega} = 6,98$. A tápfeszültség megjelenése micro USB csatlakozón keresztül magával hordozza azt a hatást, hogy egy lineáris regulátornak ki kell kapcsolnia, hogy párhuzamosan ne működjön. Erre a célra készült a 2.17 ábrán látható engedélyező kapcsolás. Üzem-

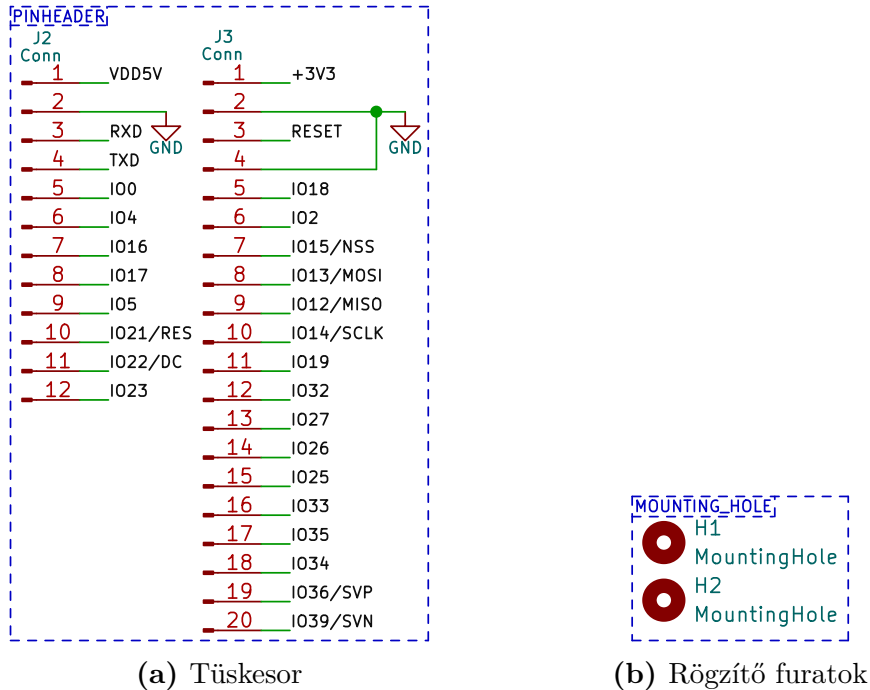


2.16. ábra. Akku lineáris regulátor kapcsolási rajz



2.17. ábra. Engedélyező áramkör kapcsolási rajz

szerűen a kártyára kötött Li-Po cella feszültsége az R_{32} ellenálláson keresztül felhúzza a EN nevezetű csomópontot, amely közvetlen összeköttetésben áll a 2.16 ábrán látható lineáris regulátor engedélyező bemenetével. Amennyiben az USB csatlakozón megjelenik a tápfeszültség, úgy az előzőekben ismertetett módon megindul egy töltőáram a cella felé, illetve az engedélyező áramkör U_{GS} feszültségét maximum 6,2 V-ra emeli, ez által a Q_3 MOSFET kinyit, így az EN csomópont közel közös pontra kerül. Ekkor pedig az akkumulátorról üzemelő lineáris regulátor kikapcsol. A 6,2 V-os maximumot a Gate-Source kivezetés közé záróirányban kapcsolt Zener dióda biztosítja,



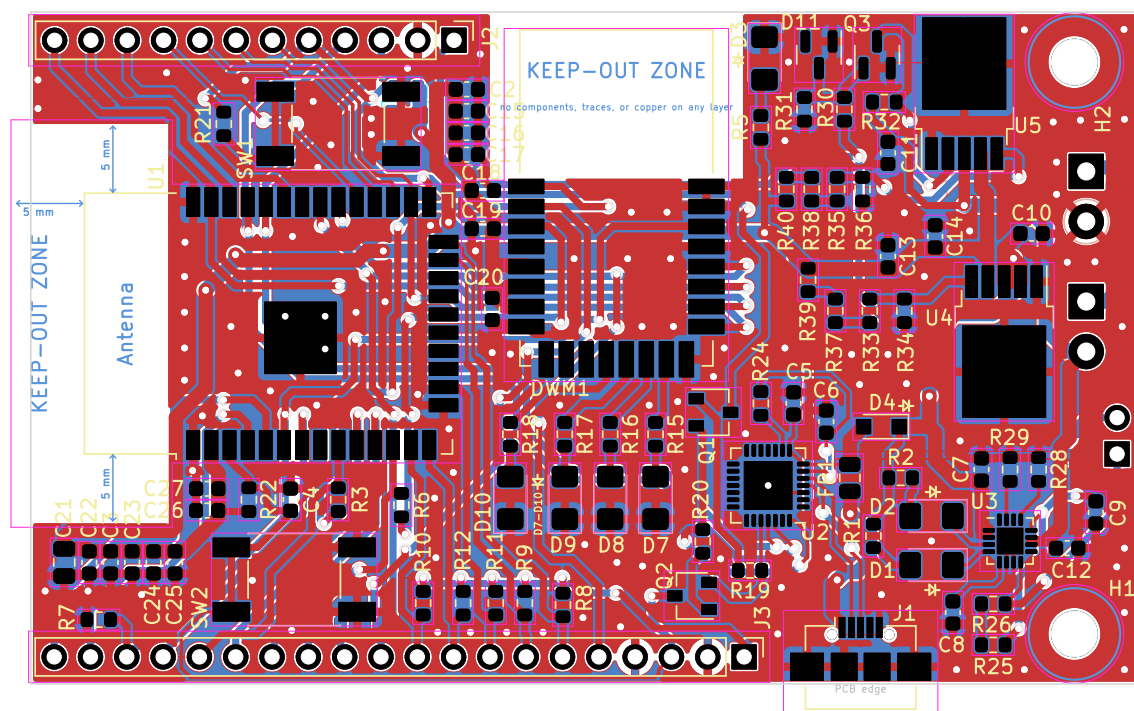
2.18. ábra. Kapcsolási rajzok

ezzel védve a térvezérlésű tranzisztort, illetve megvalósít az USB buszon egy túlfeszültség védelmet, melyhez még egy áramkorlátozó R_{30} ellenállás is hozzátartozik. Az R_{31} ellenállás pedig abban játszik biztosító szerepet, hogy a feltöltött Gate-Source kapacitás hamar elveszítse töltöttségét. A tranzisztor kiválasztásánál fontos szerep volt, hogy nagyon alacsony U_{GS} küszöbfeszültség esetén kinyisson. Ez a konkrét típus egy korábbi tervezésből gyári csomagolásban rendelkezésre állt, így kézenfekvő volt a jelenlegi helyzetben újra felhasználni. Végezetül a 2.18a és a 2.18b ábrán látható kapcsolási rajzok a tüskesorra került ESP32 modul kivezetéseket mutatja be, illetve a fejlesztői kártya rögzítő furatait jeleníti meg kapcsolási rajz szintjén. A tüskesorra való kivezetés alapvetően nem létfontosságú tervezési lépés, viszont ha valami probléma merül fel, aminek a megoldása egy-egy használt, és nem használt ESP32 láb átkötésével orvosolható, illetve ha külső eszközzel kell programozni a modult, akkor előnyös ha könnyen hozzáférhető.

2.1.5. Nyomatott áramköri terv

A nyomatott huzalozási tervben első lépések között van annak eldöntése, hogy mekkora méretű legyen a fejlesztői kártya. Ez az anchor pontoknál nem számottevő kérdés, mert azok telepített, üzemszerűen egy helyen maradó egységek. Ezzel szemben a tag egy mobilis eszköz, lehetőleg kis méretű kell legyen. Másrésztől egy fejlesztői kártyáról van szó, amely kereskedelmi forgalomban kapható több modul példák alapján nem a legkisebb áramkörök. Egy következő megközelítésből, a

tag egy műanyag (ABS) dobozba beszerelhető kell legyen. Jelenleg nem állt rendelkezésre gazdaságos úton 3D nyomtatási lehetőség, hogy saját tervezésű dobozolás történhessen, így piackutatás folytatása után a Gainta Industries Ltd. által gyártott G1020B azonosítójú, UL94-HB lángállósági osztályba tartozó, külső méretei tekintve 54x83x30 mm nagyságú dobozra esett a választás. Az fejlesztői kártya, amelynek nyomtatott áramköri terve a 2.19, és a 2.20 ábrákon látható, ennek a doboznak a belső méreteit figyelembe véve tervezetten 77,5 mm hosszú és 47 mm széles. Ezáltal lehetőség lesz arra, hogy néhány mechanikai módosítás után metrikus csavarokkal rögzíthető legyen a fejlesztői kártya a dobozon belül. A 2.20 ábrán jobb oldalon ta-

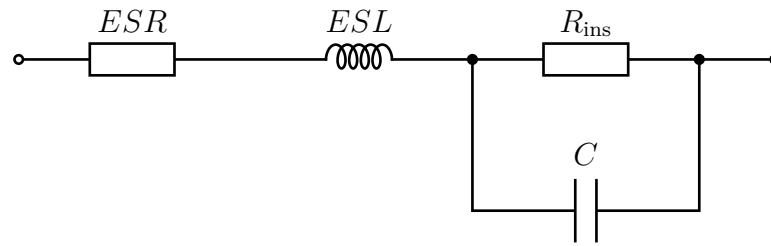


2.19. ábra. Nyomtatott áramköri terv felső oldal

lálható a két sorkapocs, valamint a termisztor számára kijelölt ültetési keret. Ezen az oldalon fog helyezkedni majd el az akkumulátor is, amely fektetve fel tud feküdni a fejlesztői kártyára, és a korábban már említett szivacsos öntapadó szalag segítségével rögzülni tud. Jobb oldalon az ábrák szélein látható a rögzítési furat is, amelyek elhelyezésénél aktívan szerepet játszott az ABS doboz adatlapjában található méretpontos felülnézeti szerkezeti rajz. Ennek megfelelően a furatok pontosan illeszkednek a doboz belsejében található műanyagból felépülő, közepén furatolt hengeres anyákhoz, amelyekbe majd egy M3 metrikus csavarral rögzíthető a kész fejlesztői kártya.

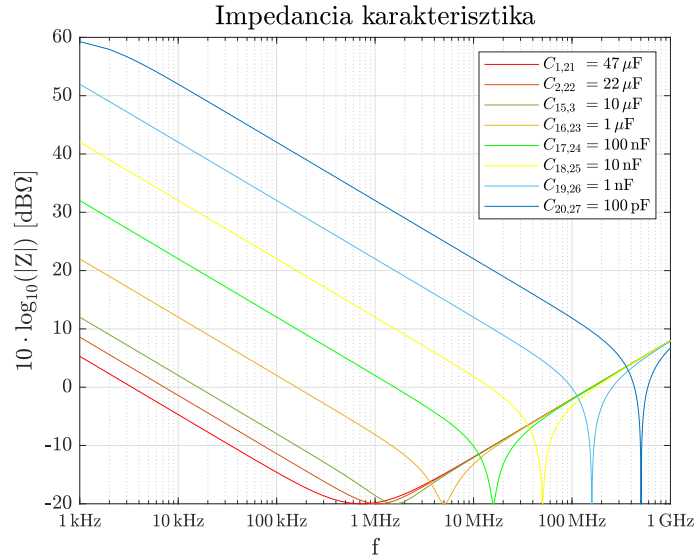
Balra haladva a nyomtatott áramköri terv felső oldalán a következő csoportot az akkumulátormenedzsmenethez tartozó áramköri elemek alkotják. Ebben a csoportban a nagy áramú huzalozási vonalak 1 mm szélesek, ezáltal növelve a keresztmetszetet és csökkentve a huzalok ellenállását. Ez azért lényeges, mert egyrészt kevesebb

ábrán rajzolt módon kell elképzelni, amely ábrán a kondenzátor áramköri helyettesítőképe látható. A C kapacitás adja a hasznos, „lényegi” funkciót. Az R_{ins} egy



2.21. ábra. Kapacitás áramköri modell

jellemzője a fegyverzetek között található dielektrikumnak, melynek a vezetőképessége zérusnál minimálisan nagyobb. Ennélfogva egy feszültségtől függő szivárgási áram állandóan folyni fog, amely egy $M\Omega$ nagyságrendű. Ellenállással jellemezhető. Az ESL (Equivalent Series Inductance) pedig az alkatrész hozzávezetésein keresztül megjelenő parazita (azaz nem kívánatos) induktivitást reprezentáló elem. Az ESL befolyásolható a beforrasztás módszerével, illetve a hozzávezetések hosszúságával, és a nyomtatott áramköri terven kialakított ültetési felülettel (pad) is mivel egy nagyon kis értékű induktitásról (nH nagyságrend) van szó. Az ESR (Equivalent Series Resistance) a kondenzátort mint soros rezgőkört jellemzi. Impedancia diagramon tekintve az ESR az a lokális minimum ameddig a kondenzátor impedanciája fizikailag le tud csökkenni. Ez tehát nem egy szivárgási áramot okozó modell paraméter, mint az R_{ins} , hanem a kondenzátor fegyverzetének, hozzávezetésének, felépítésének jóságát jellemző mennyiség, amely $m\Omega \dots \Omega$ nagyságrendű. Összességében tehát széles frekvenciatartományban $0 \dots \text{GHz}$ tartományban a modell jól alkalmazható) impedancia karakterisztikát tekintve egy soros, veszteséges rezgőkörnek felel meg minden egyes tápszűrő kondenzátor. Rezgőkörre jellemző, hogy létezik egy olyan rezonancia frekvencia, ahol a kapacitás, és induktívitas reaktanciájának abszolút értéke egyenlő (rezonancia frekvencia), itt egy lokális minimumhely található az impedanciagörbén, és ez a minimumhely ESR értéket közelíti meg bizonyos frekvencia tartományban. A 2.22 ábrán 8 kondenzátor impedancia diagramja látható. Jellegre mind a 8 kondenzátor hasonló tulajdonságot mutat, jól látható mindegyik kondenzátor rezonáns frekvenciáját, amely ahogy a kondenzátor kapacitása csökken, úgy tolódik el a nagyobb frekvenciák irányába. A korábban említett veszteségi ellenállások hatása is egyértelműen azonosítható, mivel ezen a saját készítésű MATLAB szimulációval készített ábrán az $ESR=10 \text{ m}\Omega$, és jól láthatóan mindegyik kondenzátor impedanciájának ekkora a lokális minimuma (logaritmizálva a $10 \text{ m}\Omega$ -ot). A R_{ins} a legkisebb kapacitású kondenzátor esetén kezd láthatóvá válni, ugyanis 0 Hz felé haladva az ábrán már elkezdi telítésbe menni, amely valid mivel $R_{\text{ins}}=1 \text{ M}\Omega$. A hatékony zavarjel elnyomó képesség pedig azáltal jön létre, hogy frekvenciatar-



2.22. ábra. Kapacitás impedancia diagram

tományban minél több, és minél sűrűbben egymás mellett elhelyezett impedancia minimumhelyek jöjjenek létre, így létrehozva azt a kényszert, hogy az adott frekvenciájú zavarjel, amely beleesik valamelyik minimumhely környezetébe, ott egy ESR értékű (nagyságrend) ellenállást látva elvezetődjön a közös pont felé. A szimulációs modellparaméterek (ahol rendelkezésre állt gyártói adat) összhangban vannak a felhasznált kondenzátorok paramétereivel, viszont mivel ez nem teljes körű, így az ábra főleg szemléltető célra alkalmas.

A kártya egész területén elhelyezkednek fehér színű kis kör formájú átvezetések (via), amelyek a két oldal közös pontjának galvanikus kapcsolatát erősítik. Azon átvezetések viszont amelyek szemmel láthatóan egy huzalra esnek, ott egy jel átvezetés történik a túoldalra.

A sárga színnel látható feliratok a beültetés során nagy segítséget nyújtanak, így azok átlátható és logikus elhelyezésére a nyomtatott áramköri tervezés végső fázisában külön figyelmet érdemes fordítani. Egy plusz könnyítés a beültetés folyamatára, ha az olyan alkatrészekenél, ahol fontos a megfelelő orientáció (pl.: dióda, szimmetrikus tokozású IC) ott érdemes egy alkalmas jelző feliratot kialakítani, amely egyértelműsíti IC-k esetében tipikusan az 1-es lábat, diódák esetében pedig a kapcsolástechnikailag helyes irányt.

2.2. Áramkörépítés

Az áramkörépítés, és élesztés során használt eszközök listája a következő:

- ATTEN 850D hőlégfúvó
- Weller WSD 81 forrasztóállomás

- Weller WT 1 forrasztóállomás
- NYÁK tartó állvány
- Andonstar ADSM302 digitális mikroszkóp
- Erem SWISS 102ACA csipesz
- ChipQuik SMD29130CC folyasztószer
- Loctite 60/40 370 5C 0,38mm forrasztóon
- Loctite 60EN 309 5C 0,7mm forrasztóon
- XURO-WICK W4015-1 ónszívó szalag
- FLU01L áramköri tisztító folyadék
- ESD karszalag
- Maxwell 25304 multiméter
- UNI-T UT58D multiméter
- BASETech BT-305 labortáp

Folyamatát tekintve először az anchor funkciót betöltő fejlesztői kártya került megépítésre, azon belül is kezdve az összes ellenállás, kondenzátor, dióda, tranzisztor beforrasztásával. Ezt követően a LED-ek, illetve a lineáris regulátorok beforrasztása következett. Ezt a sorrendet motiválta az élesztési metódus.

Célszerű ismerni az üzemszerű tápáram felvételeket még a modulok nélkül, ezáltal azok fogyasztása egyszerűen számítható a későbbiekben, illetve egy esetleges hiba hatása már ekkor kiderülhet, megkönnyítve a hibakeresési folyamatot. A kártya áramfelvétele közvetlenül leolvasható a fentebbi felsorolásban látható labortáp kijelzőjéről, ha az 10 mA vagy afölötti érték, viszont a modulok nélküli kártya áramfelvétele ennél jóval kevesebb. A 2.2 táblázat foglalja össze az élesztéshez tartozó mérési eredményeket. Emiatt a kártya tápbemenetének pozitív ágán egy sorosan kapcsolt néhány Ω értékű ellenállás két kivezetésén mért feszültségből számítható ki a felvett áram. Ez a módszer pár mA áramfelvétel esetén könnyen alkalmazható, mert a néhány Ω -os ellenálláson a felvett áram csupán mV-os nagyságrendű feszültségesést produkál, ezért nem zavarja meg számottevően a beállított tápfeszültség értékét. Ahogy növekszik a felvett áram, ez a hatás is egyre jelentékenyebb lesz, így előfordulhat, hogy a labortáp kimeneti feszültségét növelni kell, hogy a sorba kötött ellenálláson eső feszültség után a kártya bemenetén pont üzemszerű feszültség legyen.

Paraméter	Mért érték	Leírás
Kártya áramfelvétele	1,61 mA	Modulok nélküli fogyasztás
Kártya áramfelvétele	14,42 mA	DWM1000 beforrasztva
Kártya áramfelvétele	96,55 mA	ESP32 beforrasztva
Kártya áramfelvétele	111,03 mA	ESP32 és DW1000 beforrasztva
Fehér LED-ek áramai	0,7...0,92 mA	GND-hez képest az áramkorlátozó ellenállás (R_5) két pólusán mért feszültségek alapján.
Kék LED árama	2,02 mA	GND-hez képest az áramkorlátozó ellenállás (R_2) két pólusán mért feszültségek alapján.
Piros LED árama	2,97 mA	GND-hez képest az áramkorlátozó ellenállás (R_1) két pólusán mért feszültségek alapján.
Zöld LED-ek áramai	2,71...2,97 mA	GND-hez képest az áramkorlátozó ellenállás ($R_{15...18}$) két pólusán mért feszültségek alapján.
Li-Po cella töltőárama	472 mA	3,8 V kapocsfeszültség mellett
Schottky dióda feszültsége	210 mV	5,06 V bemeneti feszültség mellett, ESP32. DWM1000 modulok beforrasztva.
Lineáris regulátorok kimeneti feszültségei	3,28...3,30 V	DWM1000 és ESP32 modul beforrasztva

2.2. táblázat. Élesztéshez tartozó lényeges mérési eredmények

A mérési eredmények, és az élesztés körülményei alapján a fejlesztői kártyák programozásra készen állnak.

2.3. Mechanikai konstrukció építése

Az áramkörök dobozba ültetése során használt saját eszközök a következők voltak:

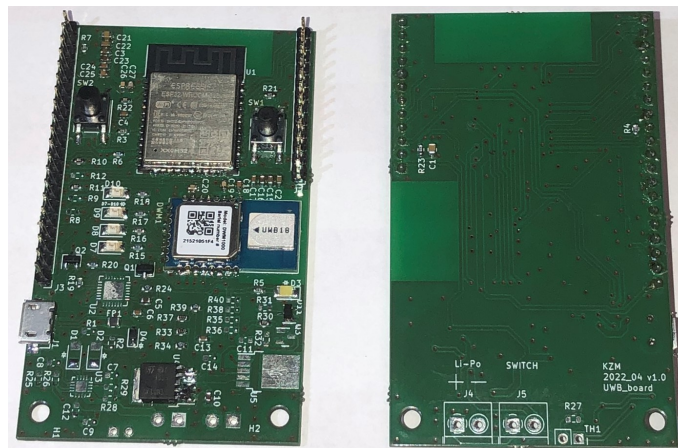
- DeWALT DCD791D2 akkumulátoros fúró-csavarozó
- DREMEL 200 multifunkcionális szerszám (200-5)
- OLFA X-DESIGN XH-1 tapétavágó kés
- HOLEX digitális tolómérő
- Fortum műszerész oldalcsípő fogó (125 mm)
- YATO YT-2028 hajlított rádió fogó (200 mm)
- Extol Prémium érvéghüvelyfogó (0,08...6 mm² ,175 mm)
- Handy kábelblankoló fogó
- YATO YT-7361 védőszemüveg
- Portwest A120 PU tenyérmártott kesztyű (fehér)

A dobozolás folyamatának célja, hogy a megfelelő mechanikai módosítások útján a műanyag dobozokba rögzíthetőek legyenek az elkészült áramkörök. A tag számára némiképp összetettebb a szituáció, mivel a dobozon egy billenőkapcsolónak is el kell készíteni a beépítési ablakot. Első lépés annak meghatározása volt, hogy a doboz belsejében található hengeres rögzítési pontok magasságából egyik illetve másik oldalon körülbelül hány mm-et kell eltávolítani. Ez azért különbözik a két oldalon, mert a kártya ahogy felfekszik a 4 rövidített rögzítési pontra, csak az egyik oldalon jön létre menetes rögzítés. A másik oldalon a tűkesorok túllógásain fog támaszkodni, és emiatt ezen az oldalon több műanyagot kell eltávolítani a kártya közelítőleg vízszintes helyzeté érdekében. Az eltávolítandó mennyiség hosszára 13,1 mm és 14,4 mm adódott. Második lépésként a tolómérő mélységmérőjével, illetve a rádiófogóval tartott pengedarabbal megtörtént egy bejelölési folyamat, ezáltal apró bemetszések jelelték ki azt a pozíciót, amelynél az felesleges műanyagmennyiséget le kell választani. A leválasztáshoz egy alkalmas szerszám a DREMEL, amelybe egy kis méretű, műanyag vágására alkalmas korongot illesztve a legkisebb fordulatszámmon egyenletes vágás végezhető el. Precíz kézmozdulatokat igénylő feladat, célszerű úgy elvégezni,

hogy a szerszámgépet az egyik kéz egy masszív munkafelülethez tartja, a másik kéz pedig a munkadarabot (műanyag doboz) lassan, körültekintően mozgatja a megfelelő irányban. Ez a művelet mind a 6 doboz esetén hasonlóan nagy pontossággal sikerült. Egy ellenőrzőmérés következett, hogy a megmaradt rögzítési pontok mérete azonos-e. A munkadarabok átmentek ezen az ellenőrző mérésen, néhány tized mm pontossággal azonos méretűek maradtak a megfelelő rögzítési pontok. Harmadik lépésként a dobozok belső oldalfalán található erősítések benyúlásából való visszavágás következett. Ezt szerencsésen el lehetett végezni szintén a DREMEL szerszámgéppel egy nagyobb korongra váltást követően. A korong vastagsága nagyságrendileg azonos volt azzal a névleges mérettel, amelyet vissza kell vágni, így mindössze az volt a lényeg, hogy a korong síkja maradjon a benyúlás síkjában, így pont a megfelelő mennyiséget távolítja el. Negyedik lépésben egy próba következett, hogy a kártyák fel tudnak-e feküdni megfelelő módon a rögzítési pontokra. Kivétel nélkül mindegyik pontosan a helyére tudott kerülni, így következett a rögzítési furat megmunkálása. Ehhez az akkumulátoros fúró-csavarozó használata célszerű, és mivel a metrikus csavarok 3 mm névleges átmérővel rendelkeznek, egy 2,5 mm-es kétélű HSS (High Speed Steel) fúrószár alkalmazása a javasolt. Ha az imént említett fúrószár a rögzítési pontokban található 1,5 mm-es előfuratokat kitágítja 2,5 mm-re, akkor a metrikus csavarok képesek saját maguknak menetet vágva létrehozni egy menetes rögzítést. Természetesen ez nem tekinthető mechanikailag azonosnak egy fém-fém kombinációval, de a kártyák kis tömege miatt (illetve mert üzemszerűen nem lépnek fel nagy energiájú rezgések) elegendő a kötés mechanikai szilárdsága. Utolsó mechanikai módosításként a tag dobozára szánt billenőkapcsolónak, illetve az összes dobozra szánt USB csatlakozó ablakának a kialakítása történt. Ez több méretvétel után kezdődhetett meg, ugyanis nagy pontossággal kell eltalálni a furathelyek középpontjait az esztétikus ablakkialakításhoz. A kétélű HSS fúrószár helyett ezúttal egy szintén kétélű, de központosítóval ellátott 3 mm-es HSS fúrószár alkalmazása vált szükségessé. Ennek oka, az, hogy műanyag fúrásakor a kézzel végzett furatok pontossága lecsökken az anyagszerkezet miatt. A dobozokon előre bejelölt pontokon kialakított furatok után DREMEL szerszám segítségével lehet megformálni a végleges ablakformákat. Ehhez egy speciális kialakítású 1,5 mm-es marószár befogatása javasolt. Az elkészült ablakok kialakítása után egy próbasorozat történt, hogy a kártyákat a dobozba rögzítve üzemszerűen csatlakoztatható-e a microUSB kábel minden kártyához, amely sikeres volt az összes kártya esetén.

Következő lépésben a taghoz tartozó akkumulátor a saját vezetékai végére (blankolás után) 1-1 érvéghüvely roppantásával került kártyához csatlakoztatható állapotba. A billenőkapcsolóra technikailag helyes csatlakoztatás egy-egy méretben megfelelő csúszósaruvallal lehetséges, amelyekre érintésvédelmi (és elektromos zárlat ellen

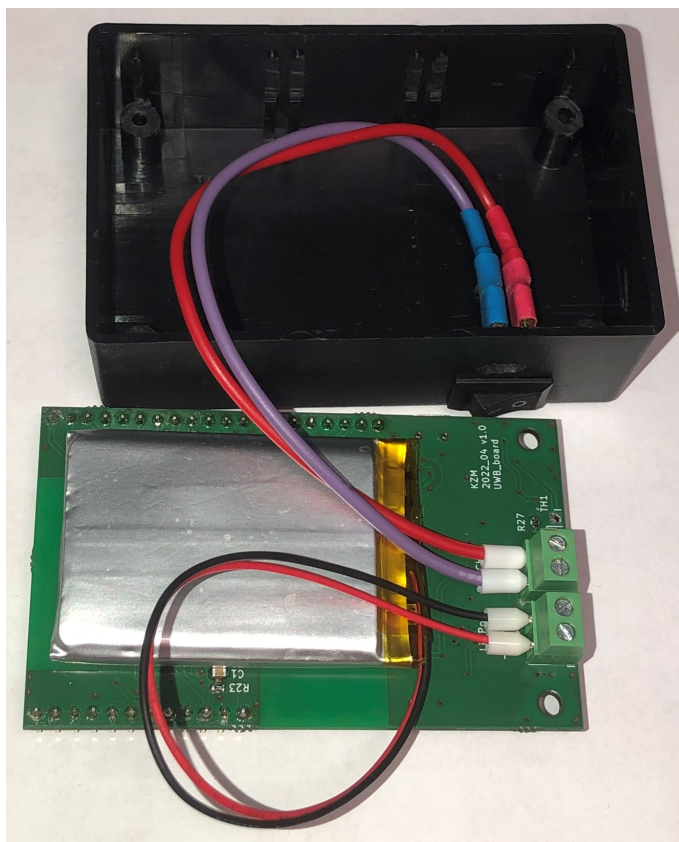
való védekezési) okból kifolyólag célszerű zsigorcsovet felhelyezni, és melegítéssel rászorítani. Mindez azután történt, hogy a csúszósarukra megfelelő krimpelőfogót alkalmazva rögzítésre kerültek a méretre vágott vezetékdarabok. Az imént említett vezetékdarabok másik végét blankolva, majd érvéghüvellyel ellátva csatlakoztatni lehet a taghez. Az elkészült áramköröket, és az építési munkálatok eredményeit szemlélteti a 2.23, 2.24, 2.25, 2.26 és a 2.27 ábra.



2.23. ábra. Elkészült fejlesztői kártyák felső, és alsó nézetben



2.24. ábra. Dobozban rögzített fejlesztői kártya zárt tetővel, illetve nyitott tetővel



2.25. ábra. Akkumulátor és billenőkapcsoló bekötésének megjelenítése



2.26. ábra. Dobozba helyezett tag bekapcsolt állapotban



2.27. ábra. A 6 db dobozolt állapotban lévő elkészült fejlesztői kártya

3. fejezet

Szoftver

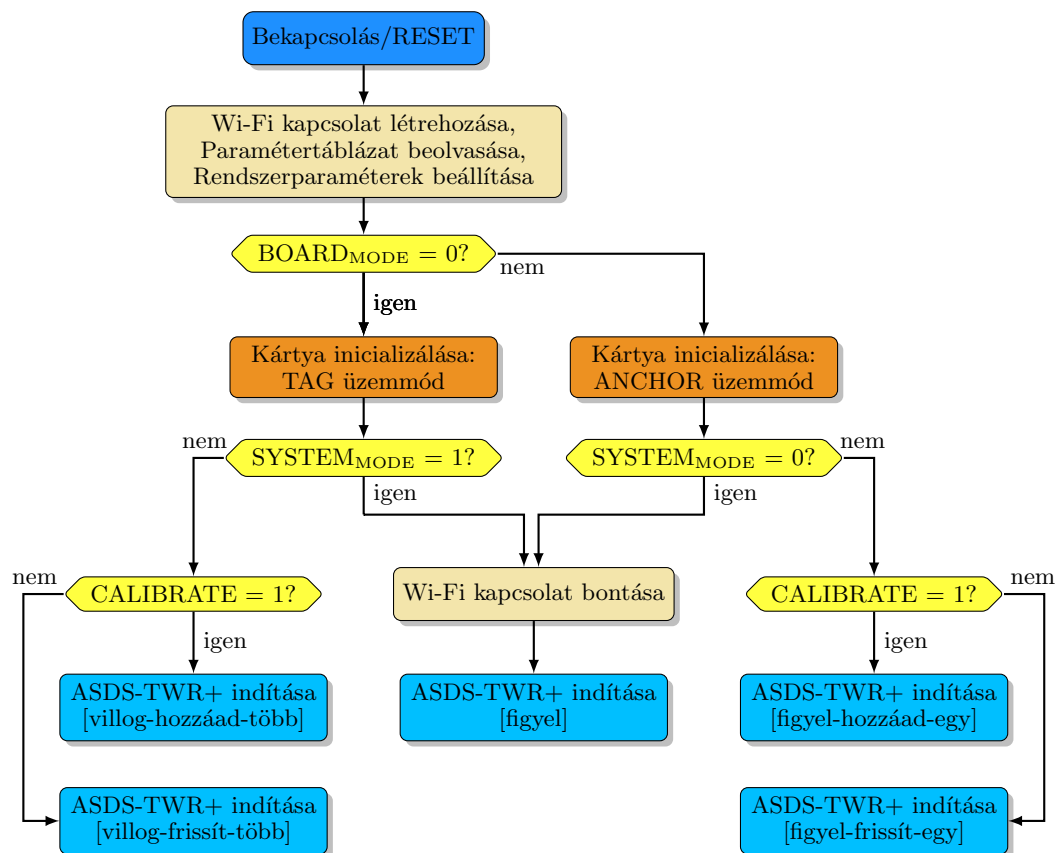
Ebben a fejezetben a diplomatervhez kapcsolódó szoftveres környezet kerül bemutatásra. Ez 3 főbb csoportra osztható, amelyeken az egyes alfejezetek haladnak végig. A pozicionálás alapját képező anchor-tag távolságok rendszeren való keresztülhaladását tekintve az alfejezetek sorrendje üzemszerű működésének megfelelően alakul. Az első főbb csoport a fejlesztői kártyán helyet foglaló ESP32 modulhoz tartozik, a második a Raspberry Pi segítségével megvalósított adatbázisszervert mutatja be, végül a PC-n futtatott MATLAB applikáció funkcióit a 3. csoport ismerteti.

3.1. ESP32 modul

A modul programozásához használt ingyenes szoftver az [Arduino IDE](#) 1.8.19. Kedvező tulajdonsága, hogy felhasználóbarát környezetet teremt a programozási feladatokhoz, illetve egyszerűen használható könyvtárkezelője lehetővé teszi a különböző modulokhoz már elérhető programkönyvtárak telepítését. Ennek köszönhetően lehetővé válik például egy forrasztási folyamaton és élesztésen túljutó fejlesztői kártya gyors felprogramozása egy példakóddal, amelynek működése soros porton keresztül monitorozható. A soros porthoz való csatlakozáshoz a már korábban említett külső eszköz ([Digilent Pmod USBUART](#)) használata feltétlenül szükséges.

Az ESP32 modulon futó programkód egyik fő feladata, hogy vezérelje a DWM1000-es modult. Ehhez rendelkezésre álló programkönyvtár a DW1000 0.9.0. Ez tartalmaz több példaprojektet (.ino kiterjesztésű fájlok), amelyet az Arduino IDE nyit meg, illetve az Arduino IDE telepítése közben létrehozott könyvtárszerkezet lekötése után megtalálhatóak a .ino fájlok alapját képező .cpp és .h fájlok is. Az ESP32 modulnak képesnek kell lennie ezen kívül felcsatlakozni Wi-Fi hálózatra is, valamint a Raspberry Pi-n futó adatbázis szerverre is be kell tudnia jelentkezni. Ehhez egy további könyvtár letöltése szükséges, amelynek neve MySQL_MariaDB_Generic 1.7.2.

Az ESP32 modulon futó kódok az alapját a korábban említett könyvtárakban található példaprojektek adják. Ezek a projektek rendszerint úgy készültek, hogy felhasználói szinten nyomon lehessen követni például a Wi-Fi hálózatra való felcsatlakozás folyamatát a soros port monitorozásán keresztül. Természetesen ezen projektek mögött lévő kódok alapos megértése szükséges ahhoz, hogy kisebb-nagyobb módosítások után is üzemképes maradjon a modul. A példaprojektek alapját képező .cpp és .h fájlok visszafejtése után a diplomaterv szempontjából helyes működés elérése érdekében számos módosítás történt. Az alapvető indulási folyamata egy fejlesztői kártyának bekapcsolás/RESET hatására az, hogy felkapcsolódik a Wi-Fi hálózatra, beolvasson egy erre a célra létrehozott táblát az adatbázisszerverről, majd a kártya sorszámának megfelelő dedikált sorban található paramétereknek megfelelően üzembe áll. A szoftverfejlesztési folyamat során ez a módszer nagyban könnyíti a paraméterváltoztatás érvényesítését, ugyanis az adatbázis egy tábláján „távolról” is akár, de közvetlenül a szerveroldalon is könnyű megváltoztatni néhány értéket. Ezt követően a kártyán található RESET gomb megnyomásának hatására újraindul a paraméterbeolvasási folyamat, így az Arduino IDE ismételt használata nem szükséges mindegyik kártyára egymás után. A 3.1 ábra a fejlesztői kártyán található program-



3.1. ábra. Fejlesztői kártyán futó program folyamatábrája

kód működését szemlélteti egy folyamatábrával. Ennek ismertetése nagy vonalakban az, hogy a kártya elindulás, paraméterbeolvasás után üzemállapottól függően vagy meghagyja a Wi-Fi kapcsolatot, vagy bontja azt. Továbbá attól függően, hogy antennakésleltetést kalibráló folyamatról van-e szó, vagy már pozicionálásról más-más MySQL kéréseket generál le. Az antennakésleltetés kalibrálásáról bővebben a 3.2.1. Adatbázis fejezetben lesz még szó. Wi-Fi kapcsolatokat illetően két lehetőség áll rendelkezésre a programkód által. Az egyik, ami a korábban vázolt rendszertervnek megfelelő, azaz a tag nem csatlakozik a hálózatra, és az összes anchor igen, és egymás után bonyolítják le a távolság meghatározását. A másik lehetőség az, hogy egyik anchor se marad hálózaton, viszont a tag igen, és egy plusz UWB csomagküldéssel az anchorok által számított távolságot visszaküldik a tagnek, amely aztán egy MySQL kéréssel beszúrja azokat az adatbázis megfelelő táblájába (illetve frissíti a megfelelő sorokat). Ezzel a módszerrel csökkenteni lehet az adatbázisszerver leterheltségét, illetve egy korábban készített saját MATLAB szkript ilyen táblaszerkezetre készült el, így kézenfekvő volt ennek használata. A 3.1 ábra alsó állapotaiban történik a távolság meghatározása. Ennek alapja az 1.3.1. ToA (Time of Arrival) fejezetben ismertetett ASDS-TWR üzenetváltási séma. A + jel arra utal, hogy a tényleges séma kibővül 3 üzenettel. Minden esetben a tag kezdeményezi a folyamatot, egy „blink” üzenettel, amely egy broadcast típusú üzenet, mindegyik anchor fogadja az üzenetet, és amint tud válaszol egy „init” üzenettel. A folyamat ezen szakaszán az anchorok feljegyzik a tag MAC címének első 8 bitjét, ez alapján tudnak esetlegesen több tag-et figyelembe venni a későbbiek folyamán. Természetesen a tag a válaszul kapott üzenetek forráscímét szintén feljegyzi, és az anchorhoz hasonlóan elmenti. Egy UWB csomag időtartományban mért kiterjedése miatt gyakorlatban biztosra vehető, hogy több csomag ütközni fog, mivel az anchorok ezen a ponton nem veszik figyelembe, hogy esetlegesen más is válaszol a broadcast típusú üzenetre. Az anchorok késleltetés nélküli válasza 10...100 μ s közé tehető, ezzel szemben az alkalmazott preambulum hossza kb. 2 ms, 10 m esetén a jelkiterjedési idő kb. 33 ns, az antennakésleltetés pedig névlegesen 200...300 ns. Ezeket az értékeket összevetve garantált a csomagok időbeli átlapolódása hasonló távolságokon. A gyakorlatban tapasztaltak szerint az a csomag jut el a dekódolásig, amelyik a legközelebbi anchortól érkezett, vagy amelyet leginkább gyorsan válaszoló anchor küldte. Miután megvolt a „blink” üzenet, és a ráadott „init” válasz, a tag egy „poll” üzenetet küld ki, amely szintén broadcast típusú, viszont ezúttal tartalmaz egy válaszkésleltetést paraméterező értéksorozatot, amelyet minden olyan anchor tud értelmezni, amely már egy „blink” üzenetre válaszolt. Ezáltal a „poll” üzenetre reagáló anchorok által fogalmazott „pollack” üzenet rendre nagyobb késleltetéssel kerül kiküldésre. Ezzel a módszerrel kerül el a csomagütközést. A tag dekódolja minden anchor „pollack”

válaszát, majd erre reagálva küldi ki a „range” üzenetet amely változatlanul broadcast típusú. Ezzel az üzenettel az összes anchorban (amelyek a „blink” üzenetre reagálva felkerültek a regisztrált anchorok közé) számíthatóvá válik az ASDS-TWR séma által visszaadott jelterjedési idő, majd abból a két eszköz közötti távolság. Végezetül történik még egy „report” üzenet az anchorok részéről, amelyet szintén késleltetve küldenek ki. Ez tartalmazza a számított távolságot, így lehetővé válik a tag alkalmazása is a MySQL kérések kiküldésére Wi-Fi-n keresztül. A tag egy belső számlálót tart fent, amely egy adott idő után elér egy beállítható limitet, és ekkor automatikusan kiküld egy „blink” üzenetet. Ennek lényege, hogy jól definiált időn belül az elérhető anchorokról a tag tudomást szerezzen, és regisztrálni tudja a jelenlétüket. Amennyiben egy anchortól, vagy a tagtól bizonyos idő elteltével nem érkezik üzenet, úgy az eszköz regisztrációját törli a programkód.

3.2. Raspberry Pi

A diplomatervhez rendelkezésemre állt egy Raspberry Pi 3 Model B+, amelyen egy Linux alapú operációs rendszer fut. Egyszerűsített használat érdekében egy VNC szerver került beüzemelésre, így Wi-Fi-n keresztül „távoli használat” üzemmódban vezérelhető.

3.2.1. Adatbázis

Linux alapú operációs rendszerek esetén gyakori a terminál használata, mivel ekkor lehetőség nyílik a bash fájlok egyszerű futtatására. Lényegét tekintve egy terminálban kiadott utasítás sorozat foglalható össze egyetlen fájlban, így egy összetettebb művelet sor is elvégezhető egyetlen fájl futtatásával. A saját készítésű test.sh fájl tartalmaz néhány egyszer elvégzendő utasítást, amely frissíti a rendszert (ezt érdemes gyakran elvégezni), majd mindent letöröl ami a MariaDB kezelőhöz tartozik, majd újrategyíti. Ezzel elkerülhető az, hogy a korábban esetlegesen eszközölt rejtett beállítás módosítások miatt probléma lépjen fel. Végül megnyit egy szerkesztési módban egy MariaDB által használt belső fájlt, ahol a „bind access” IP címet célszerű átírni 0.0.0.0-ra, ezáltal az adatbázis szerverhez nem csak a Raspberry Pi saját felületein lehet hozzáférni (localhost), hanem egy másik eszköztől is. Ennek megkönnyítéséhez még javasolt a használatban lévő router admin felületén a DHCP (Dynamic Host Configuration Protocol) beállításokban a Raspberry Pi MAC címéhez egy fix IP címet fenntartani, így nem kell minden egyes távoli bejelentkezésnél leolvasni a megfelelő IP címet. A test.sh fájl folytatásként egy szintén saját készítésű datab.sql szkriptet vezet rá a terminálon belül a MariaDB bejelentkezési utasítására. A datab.sql létrehoz egy uwb adatbázist, amelyhez dedikál egy felhasználót a fejlesztői

kártyáknak, valamint a MATLAB számára, illetve minden jogosultságot meg ad a 2 felhasználónak az uwb kezelésére. Ezt követően az adatbázison belül összesen 6 táblát hoz létre. Ebből 1 tábla a rendszerparaméterekkel töltődik fel későbbi INSERT utasítások hatására, egy másik tábla a MATLAB számára egy sikeres bejelentkezést ellenőrző paramétert tartalmaz, 2 táblát adat nélkül hagy, mert azokba az antennakélesztetés kalibrálása során dinamikusan kerülnek majd be a számított távolságok, és 2 táblát a pozicionálás folyamata fog alkalmazni. A rendszerparamétereket tartalmazó, illetve a bejelentkezéshez használt táblával szemben nincsenek nagy igények támasztva, ezért szokatlan beállítást nem szükséges hozzáadni létrehozáskor. Ezzel szemben a kalibráláshoz, illetve pozicionáláshoz használt táblák esetén nagyon fontos, hogy gyorsan történjen a parancs végrehajtása. Alapértelmezett beállításokkal egy sor beszúrása egy táblába a Raspberry Pi-n bejelentkezve nagyon változó időbe telik. A művelet alatt eltelt időt ms felbontással mutatja az adatbázisszerver felülete, és egy sor beszúrása a használt adatoktól függetlenül 8 ms...1,5 s-ig tart. Ehhez még hozzájön az ESP32 modulon futó példakódokban beállított kélesztetés, így kedvezőtlen esetben akár 3...5 másodpercbe is telhet egyetlen sor beszúrása. A sebesség növelése érdekében mindkét oldalon egy optimalizálási folyamat kezdődött, amelynek célja az iménti időtartamok lecsökkentése. Adatbázisszerver oldalon számos optimalizálási javaslat létezik a sebesség növelésére, a felhasználási kör viszont abból a szempontból megengedő, hogy a beszúrt, frissített adatokat nem kell évekig nagy biztonságban tárolni, hanem mindössze addig amíg azt ki nem olvassa a MATLAB applikáció. Azaz például kedvező esetben párszor 10...30 ms-ig. Emiatt a kalibrációs és pozicionálásra használt táblának a tárhelykezelő motorja az alapértelmezett InnoDB-ről MEMORY-ra lett átállítva. Ennek a beállításnak az a következménye, hogy nem a memóriakártyán tárolja majd a tábla tartalmát, hanem RAM-ban (Random Access Memory). A RAM hozzáférési ideje pedig lényegesen kevesebb, mint a memóriakártya elérési ideje. A tapasztalat az volt, hogy 1 sor beszúrás ilyen esetben mindenképp 1 ms alatti, több adat beszúrásával körvonalazódott, hogy jó közelítéssel 150...200 μ s-ra csökkent egy sor beszúrása. Az eredményt összevetve a kiinduló értékkel adatbázisszerver oldali optimalizálás sikeresnek tekinthető. ESP32 oldalon első lépésként az összes kélesztetés, és soros port monitorozást biztosító kódrészlet eltávolításra került, így sikerült kb. 300 ms-ra lecsökkenteni egy adat beszúrását Wi-Fi-t használva. Mivel ennek az időnek a nagyságrendje még nem esik elég közel az adatbázisszerveren elért sebességhez, viszont az Arduino IDE projektkódjában már nem volt elérhető újabb eszköz a sebesség növelésére, így következőként az alatta dolgozó .cpp fájl függvényei kerültek átvizsgálásra. Ebben az esetben érdemes lekövetni, hogy az Arduino IDE szinten meghívott .cpp függvény pontosan milyen műveleteket végez el, illetve esetleg milyen egyéb függvényeket hív meg. Ezen a gon-

dolatmeneten haladva kiderült, hogy egy MySQL kérés elküldésekor a kérés típusától függetlenül visszavár egy választ a szerveroldaltól. A kártyák által használt MySQL kérések 3 nagyobb típusa röviden megnevezve az INSERT, UPDATE és a SELECT. Ebből a SELECT az, amelyik üzemszerűen egy válasszal zárul, és ez a kérés minden kártya induláskor kerül elküldésre, utána vagy csak UPDATE, vagy csak INSERT kérések jönnek létre. Ez motiválta azt, hogy a .cpp fájlokban a kérés elküldését végrehajtó függvényből legyen elkészítve egy másodpéldány azzal a módosítással, hogy a választ váró kódrészlet törlésre kerül, és a neve kis mértékben módosul. Ezáltal az Arduino IDE szintjén 2 különböző függvényt lehet meghívni, amely aztán elküldi a kérést, viszont a kérés típusától függően eldönthető, hogy melyik függvény legyen meghívva. Természetesen ez a megoldás hordozza azt a kockázatot, hogy a kérés elküldése után nem érkezik visszajelzés, hogy az sikeres volt-e. Ezt a módszert szükségképp egy tesztelés követi, hogy kiderüljön a módszer megbízhatóságának mértéke. A tesztelés során kiderült, hogy egy beszúrás idejének nagyságrendje lecsökkent az adatbázisszerver szintjére, egyszerű időmérő függvényekkel nehezen eldönthető, hogy az ESP32 lett gyorsabb, vagy az adatbázisszerver. Az ESP32 soros porton történő monitorozásával igazolható, hogy 300 ms helyett 150...200 μ s-ba telik egy kérés, és teszt alapján legalább 1 millió beszúrás egymás után hibátlanul elvégez. A tesztelésben tapasztaltak alapján kijelenthető, hogy az optimalizálás sikeresnek értékelhető mind adatbázisszerver, illetve ESP32 oldalon is, mindemellett az ESP32 oldali kód-módosítás megbízhatósága a feladatra valószínűleg elegendő.

A kalibrációs táblák esetén alkalmaznak a kártyák INSERT jellegű kérést, mert a tábla tartalmából pár ezer mérés után gyakoriság diagram készíthető. A mérési adatokra könnyen illeszthető egy várható érték, így könnyedén lehet összevetni a kártyák által számított, és a valós lemerő távolságot, amelyek különbségét felhasználva közvetlenül kiszámítható az az érték, amely egészre kerekítés után direkt felhasználható az antennakésleltetés paraméterének módosítására a megfelelő tábla megfelelő sorában. A korábban említett antennakésleltetés kalibrációs folyamata abból áll, hogy beüzemelésre kerül 1 anchor-tag távolságmérési folyamat, amely dinamikusan bővíti egy erre létrehozott táblát. Ezt a táblát, ha már elegendő mérés áll rendelkezésre akkor kiolvassa egy erre alkalmas MATLAB szkript, és meghatározza a mérésekből a távolság tapasztalati várható értékét. Amennyiben a várható érték nem tükrözi a valóságban mérhető távolságot, akkor kalibrálni kell az antennakésleltetés értékét az adott anchorhoz tartozó soron belül a rendszerparaméterek táblában. Az antennakésleltetés a nevével ellentétben az az időtartam, amíg a vett UWB csomag feldolgozás szintjén odáig eljut, hogy a DW1000 IC ráillessze a 15,65 ps felbontású időbélyegét az SFD (1.5 ábrán látható) végére. A tapasztalat azt mutatja, hogy névlegesen ez kb. 256 ns. Ennyi idő alatt a beérkező csomag keresztül halad

a chip antennán, balun (balanced-unbalanced) transzformátoron, rádiófrekvenciás kapcsolón ami az adási és vételi ágat kapcsolja, analóg rádiófrekvenciás erősítőn, kétszeresen kiegyenlített keverőn fokozaton (illetve fokozatokon, a gyártó nem közli a keverőfokozatok számát), egy hangolható alapsávi analóg szűrőn, egy hangolható alapsávi analóg erősítőn, illetve egy névlegesen 1 GSample/s mintavételi sebességgel rendelkező analóg digitális átalakítón. Illetve még beletartozik az az időtartam is, amely alatt a preambulumon illetve SFD-n az LDE algoritmus meghatározza az időbélyeget. A kalibrálás gyakorlati megvalósítása a következő matematikai összefüggés szerint végezhető el

$$\Delta_{\text{INT}} = \frac{d_{\text{valós}} [\text{m}] - d_{\text{számított}} [\text{m}] \cdot 10^{12}}{3 \cdot 10^8 \left[\frac{\text{m}}{\text{s}} \right] \cdot 15,65 [\text{ps}]} \quad (3.1)$$

A 3.1 egyenlet által kapható Δ_{INT} értéke előjelhelyesen jelenti azt a számot, amelyet hozzá kell adni a jelenlegi antennaképletetés értékéhez. Ez egy előjel nélküli egész típusként van tárolva program szintjén, ezért beírás előtt egészre kell kerekíteni.

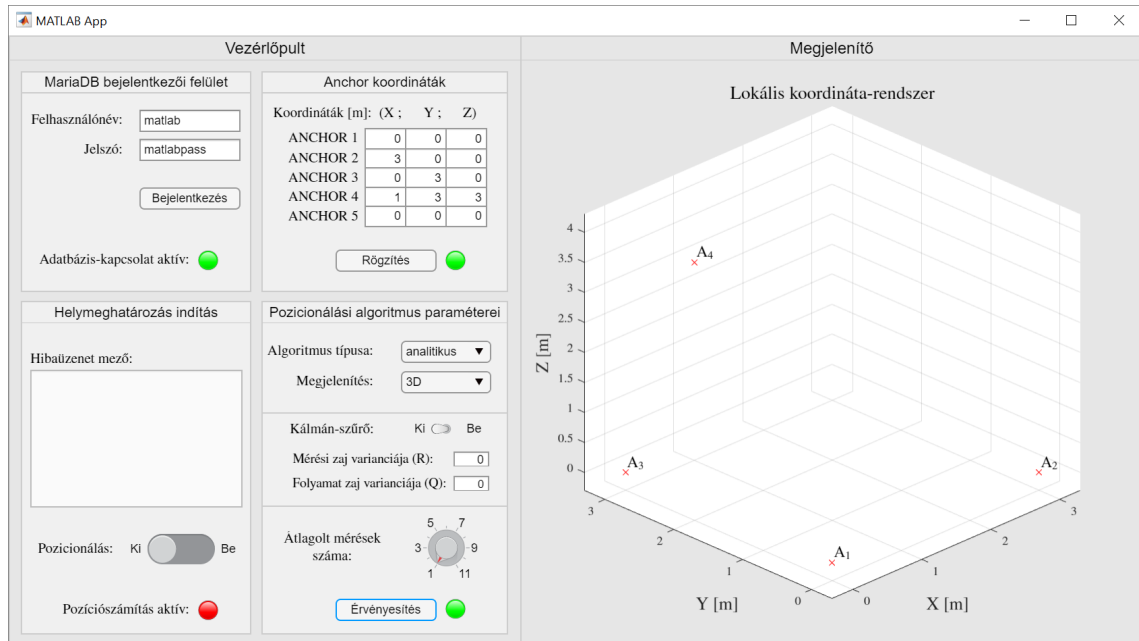
Azok a táblák, amelyeket a pozicionálás folyamata használ nem üresen kerülnek létrehozásra a szerver oldali datab.sql hatására, hanem 20 sorral feltöltve. Ezek a táblák gyűrűs bufferként vannak alkalmazva, ez pedig azért lényeges, hogy egy hosszabb üzem esetén ne kerüljön túl sok memóriába a növekvő tábla mérete. Ez egyszerűen kivitelezhető úgy, hogy minden UPDATE kérés sztringek összefűzéséből áll, ahol egy-egy sztringet egy integer-sztring konvertálással kapunk. Ezáltal tud változni a kérés bármelyik eleme, amelynek változnia kell küldésről küldésre (ilyen például a számított távolság, vagy a sor azonosítója amelyet aktuálisan frissíteni kell a frissen számított távolsággal).

3.3. MATLAB

Korábbi tanulmányok során folyamatosan központi szerepet betöltő volt a MATLAB aktuális verziója, illetve felhasználóbarát keretrendszert ad a grafikus felhasználói felület tervezéséhez, így kézenfekvő volt az alkalmazása.

3.3.1. Grafikus felhasználói felület

A MATLAB applikáció grafikus felhasználói felülete látható a 3.2, és a 3.3 ábrán. A tervezés során előtérbe helyezett szempont volt az átláthatóság, illetve egy alapvető szintű hibakezelési metódus. Ennek lényege, hogy nem helyénvaló adatbevitel esetén megjelenjen egy intuitív hibaüzenet, amely a felhasználót a helyes műveleti sorrendre ösztönzi. Első lépésként be kell jelentkezni a MariaDB adatbázisba az ott létrehozott



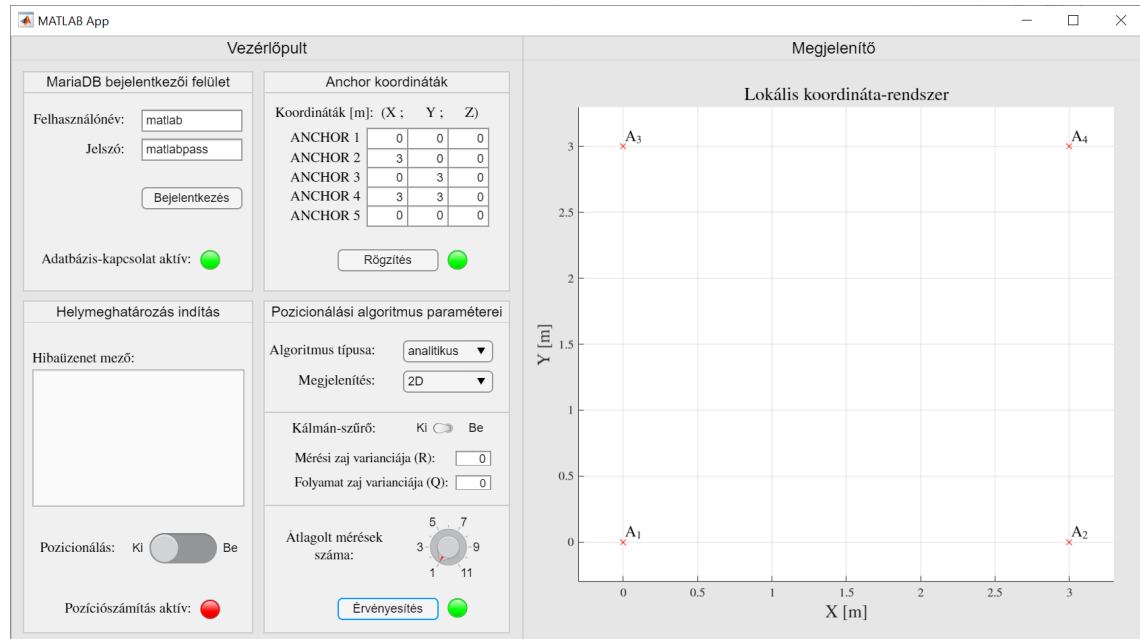
3.2. ábra. MATLAB applikáció grafikus felhasználói felülete 3D nézetben

felhasználónévvel é jelszóval. A bejelentkezés gombra kattintva a gomb alatti piros lámpa zöldre vált, ha a kapcsolat létrejött. Ez a folyamat tartalmazza annak a táblának a beolvasását, amely egy ismert paramétert tartalmaz. Így ha a lámpa zöldre vált, akkor biztosan megtörtént a bejelentkezés, és helyesen üzemel a kiolvasási folyamat.

Ezt követően kell megadni az egyes anchorok koordinátáit méterben. Célszerű úgy felvenni a koordinátákat, hogy az első anchor legyen az origó, majd utána a következő anchorok jelölik ki az egyes tengelyeket. Ez nem kötelező, viszont tesztelési tapasztalatok azt mutatják, hogy ebben az esetben üzemel a legkisebb hibával a rendszer. 5 anchorig van felkészítve jelenleg az applikáció, de már 4 anchorral mindkét algoritmus üzemképes. Az applikáció belső működése lehetővé teszi, hogy felismerje hány anchorot kíván alkalmazni a felhasználó, ennek viszont az a feltétele, hogy egymás után legyenek megadva az anchorok. Origó után az első (0,0,0) koordináta kombinációig számolja az anchorokat, a többit figyelmen kívül hagyja. A rögzítés gomb lenyomásával lehet rögzíteni a koordinátákat, amennyiben már sikeresen bejelentkezett a felhasználó.

Majd rögzítés után fel kell paraméterezni a pozicionálást. Ehhez először ki kell választani az algoritmus típusát, amely lehet numerikus, vagy analitikus. Ez a két módszer került bemutatásra az 1.3.1. ToA (Time of Arrival) fejezetben. Ha a felhasználó kiválasztotta az algoritmus típusát, akkor utána a megjelenítés módját kell megválasztani. Az applikáció 2D illetve 3D módot kínál a megjelenítésre, mindkét esetben saját azonosítóval jeleníti meg az anchorokat egy lokális koordináta-

rendszerben, a tag becsült pozícióját pedig egy kék csillag jelenti majd. Az utólagos



3.3. ábra. MATLAB applikáció grafikus felhasználói felülete 2D nézetben

adatfeldolgozást végző két módszer 4 kombinációra ad lehetőséget. Értelemszerűen lehet csak az egyiket, csak a másikat, mindkettőt használni, illetve figyelmen kívül hagyni mindkettőt. A Kálmán-szűrő hangolására 2 paraméter áll rendelkezésre, ezek a mérési zaj variancia (R), illetve folyamat zaj variancia (Q). Hibaüzenetet kap a felhasználó, ha negatív értékeket próbál érvényesíteni. Ezekről a paraméterekről bővebben egy másik alfejezetben lesz szó. Az átlagolt mérések száma pedig a másik utólagos adatfeldolgozási módszer hangolási paramétere. Ez egy tekerőgomb beállításával adható meg, amely ugyan közvetlenül állítható nem egész értékre, azonban az érvényesítés gomb megnyomásával az applikáció beállítja a legközelebbi egész értékre. Amennyiben sikeresen érvényesülnek a beállított paraméterek, úgy az applikáció felépíti a megjelenítő felületet, amelyen rögzülnek a tengelyfeliratok, cím, anchorok, illetve a kamerapozíció, és kameracélpont. Végezetül a pozicionálás elindítása egy kétállású tolókapcsolóra való kattintással történhet. Az anchoroknak, tagnék ebben a pillanatban már üzemszerű állapotban kell lenniük, és az anchor-tag távolságokkal már frissülnie kell a megfelelő táblának, különben természetesen nem lesz beolvasható adat. Üzemszerű működés esetén minden új megjelenített pont létrejöttékor az előző törlődik, illetve a Kálmán-szűrő és az átlagolás funkció ki-be kapcsolható. A pozicionálás megállításával az összes lámpa pirosra vált, viszont a bevitt értékek nem törlődnek, így egy gyors kattintás sorozattal újra lehet paraméterezni a rendszert.

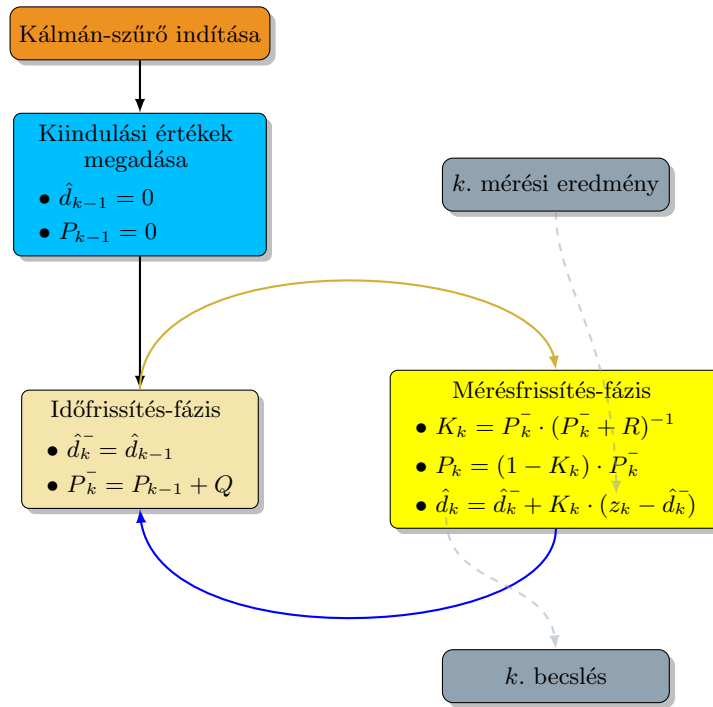
3.3.2. Utólagos adatfeldolgozás

Utólagos adatfeldolgozásként 2 módszerre esett a választás. Az első a mozgóablakos átlagolás, amely működési elvét tekintve abból áll, hogy a kapott távolságokból az applikáció folyamatosan átlagol egy ablakszélességbe eső mennyiséget, és az eredményül kapott átlag kerül a megjelenítő bemenetére. A megoldás hátránya, hogy nagyobb ablakszélesség esetén komoly memóriahatás lép fel, azaz egy esetleges hirtelen „oda-vissza” helyváltoztatás esetén annak mértéke alig érzékelhető módon jelenik meg.

Második a Kálmán-szűrő, amely egy dinamikus rendszer állapotát képes becsülni zajjal terhelt esetben. Ennek általános leírása ezúttal nem feltétlenül szükséges, mivel most jelen esetben nagy mértékben egyszerűsödnek a Kálmán-szűrő általános egyenletei. Lényegében az applikáción belül egy zajjal terhelt skalár becslését fogja elvégezni, amely skalár pedig éppen az anchor-tag távolság.

A Kálmán-szűrő modellezés szintjén azt feltételezi, hogy a mérési eredményeket is, illetve a távolságbecslési folyamatot, mint rendszert is normális eloszlású zaj terheli, amelyek várható értéke 0, varianciája pedig rendre R és Q .

A 3.4 ábrán olvasható egyenletekből világosan látszik, hogy a Kálmán-szűrő memória és számításgénye is alacsony, mivel 1 becslési folyamathoz mindössze néhány változó tárolására van szüksége az előző ütemből, a becsléshez szükséges számítások pedig mindössze néhány összeadás, kivonás, szorzás és osztás. A működési folyamatot szemléltető 3.4 ábrán láthatóan a szűrő indulását követően meg kell adni a szűrő kiindulási paramétereit. Ezek megválasztása a gyakorlatban tipikusan 0 érték, ezért az applikáción belül is 0 értékkel vannak ellátva. Ez a választás megfeleltethető annak az állapotnak, amely szerint a Kálmán-szűrő energiamentes kezdeti állapotból indul. Ezt követően indul az időfrissítés-fázis, amelyben felhasználja a szűrő a kiindulási értékeket, mint „fiktív” előző ütemből származó eredményeket. Ekkor történik meg az *a priori* ismeretekből (előző ütemből származó távolságbecslés) származtatott *a posteriori* távolságbecslés, amely felhasználásra kerül a szűrő második fázisában. Emellett még az *a priori* ismeretekből származó becslési hiba variancián és a felhasználó által bevitt Q folyamat variancián alapuló *a posteriori* becslési hiba variancia is kiszámításra kerül. A szűrő második fázisa a mérésfrissítés-fázis, amely a Kálmán-erősítés (esetenként keverési tényezőként is nevezik) számításával kezdődik. Ennek meghatározása egy olyan matematikai eljárás, amellyel minimalizálja P_k -t, amelyet aztán a következő ütemben fog hasznosítani. Végezetül \hat{d}_k egy lineáris kombinációként áll elő az *a priori* távolságbecslés és egy Kálmán-erősítéssel súlyozott különbségből. A különbség képzés tartalmazza az új mérési eredményt illetve az *a priori* távolságbecslést. Ezen a ponton érthető meg a Kálmán-szűrő lényege ebben a felhasználásban. Amennyiben a mérések pontosak, azaz R egyre kisebb értéket vesz

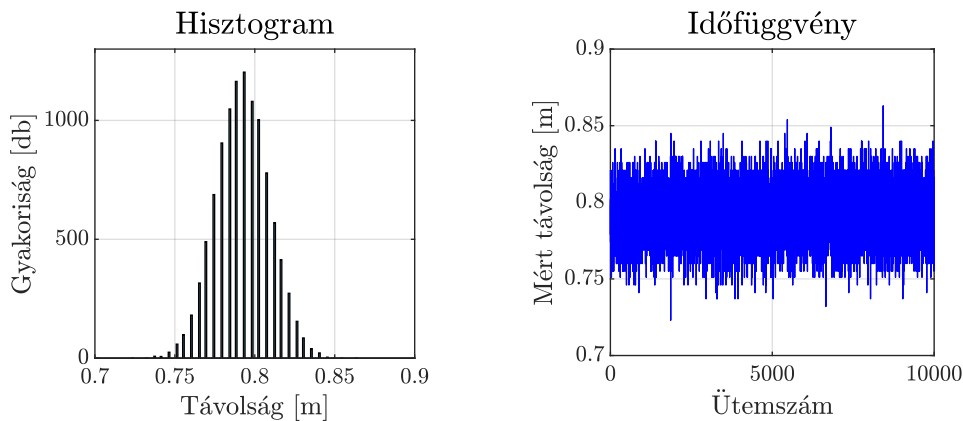


3.4. ábra. Kálmán-szűrő állapotdiagramja

Ahol:

- \hat{d}_k^- az *a priori* távolságbecslés a k . ütemben
- \hat{d}_k az *a posteriori* távolságbecslés a k . ütemben
- P_k^- az *a priori* becslési hiba varianciája a k . ütemben
- P_k az *a posteriori* becslési hiba varianciája a k . ütemben
- K_k a Kálmán-erősítés a k . ütemben
- Q a folyamat zaj variancia
- R a mérési zaj variancia

fel, úgy a Kálmán-erősítés 1-hez fog tartani. Ez azt okozza, hogy a szűrő egyre nagyobb súllyal veszi figyelembe a mérési eredményt tartalmazó különbségi tényezőt, illetve ezzel szemben visszaszorítja P_k -t, tetszőleges folyamat zaj variancia esetén. Összefoglalva a pontos méréseknek köszönhetően a szűrő inkább a mérésekre támaszkodik, mintsem a folyamat modellezéséből eredő becslésekre. Ezzel szemben, ha a P_k^- nullához tart, mivel kis Q -val rendelkezik a folyamat, akkor a Kálmán-erősítés 0-hoz tart, amely azt eredményezi, hogy a szűrő egyre kisebb súllyal veszi figyelembe a mérési eredményt tartalmazó különbségi tényezőt, és a folyamat modelljére támaszkodik inkább. Fontos megemlíteni, hogy az applikációban R és Q mint konstans

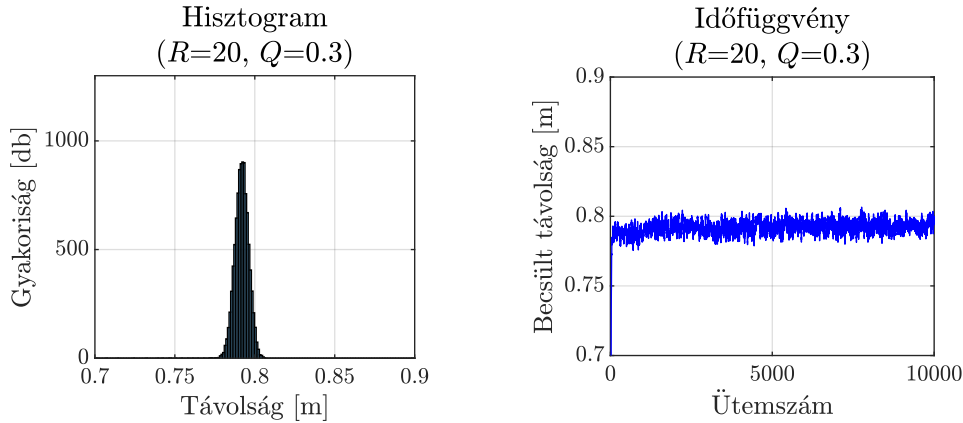


(a) Hisztogram a nyers mérési eredményekből (b) Időfüggvény a nyers mérési eredményekből

3.5. ábra. Nyers mérési eredmények 10000 mérésből

paraméterek szerepelnek, ez viszont modellezés szempontjából egyfajta hiba, mert a mérési értékekhez hozzáadódó zaj időben (ütemről ütemre) változhat, illetve ez ugyanúgy helyes állítás a folyamat zajára is. Emiatt a szűrő hangolását attól függően kell elvégezni, hogy a mérésekhez adódó zaj milyen szórást okoz a tapasztalati valószínűségi eloszláson (R beállítása), illetve mennyire gyorsan változik a távolság amelyet becsülni kívánunk (Q állítása: kisebb érték a nem mozgó tag esetére optimumot nyújt, nagyobb érték pedig a mozgó tag esetén biztosít pontosabb üzemelést). Nem megfelelő Q beállítás (például a tag gyors, ennek ellenére Q értéke 0 közeli) esetén a szűrő nem képes megbízhatóan, elég gyorsan lekövetni a tag mozgását, amely egyfajta tehetetlenségi jelenségként fogható fel.

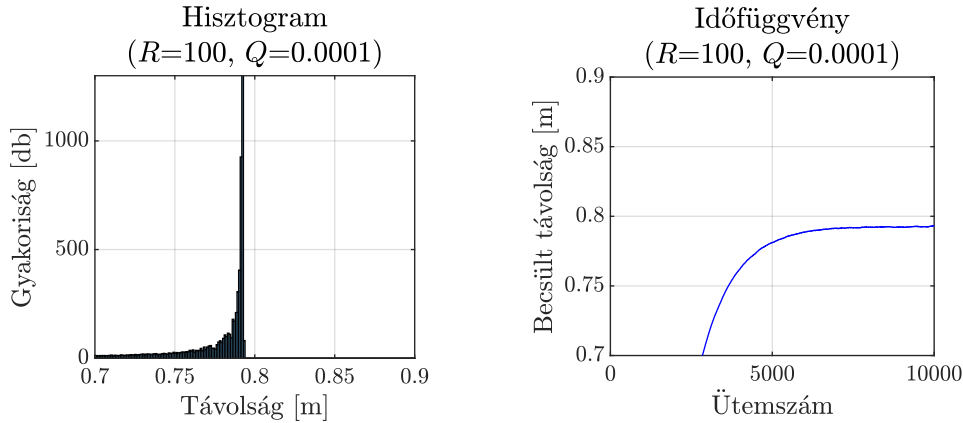
Az R illetve Q állítása teljes mértékben a felhasználóra van bízva. A két paraméter közül az R beállítása a korábbiak alapján egy előismeretet feltételez a mért távolságok eloszlását illetően. Célszerű emiatt egy mérési eredmény sorozatot megismerni, amelyen keresztül szemléltetni lehet a Kálmán-szűrő hatását. Ezek szerepelnek a 3.5, 3.6, és a 3.7 ábrákon. Számos érdekes jelenség figyelhető meg mindhárom ábrán. Ehhez hozzátartozik az, hogy ismert legyen a hisztogram felbontása, amely 1 mm. Ez azt jelenti, hogy az ábrákon látható távolság tartományt 0,7 m-től 1 mm-es lépésközzel osztja fel mind a 3 hisztogram. A 3.5 ábrán jól láthatóan nincsenek szorosán egymás mellett a gyakorisági sűrűségek, amelyet az okoz, hogy a DW1000 IC egy korábbi információnak megfelelően 15,65 ps-os időbeli felbontással rendelkezik. Ez távolságban kb. 4,7 mm-t jelent, és az ábrát közelről vizsgálva igazolható, hogy átlagosan valóban 4...5 mm-nek megfelelő távolságra vannak egymástól az egyes oszlopok. A DWM1000 modult vezérlő kódban eredetileg cm-es pontosságú távolságszámítás volt beállítva, ezt saját belátás okán mm-es pontosságra lett növelve (ezáltal a mérésekből kapott eloszlási kép reprezentációs képessége növekszik). A hisztogra-



(a) Hisztogram a Kálmán-szűrő kimene- (b) Időfüggvény a Kálmán-szűrő kime-
tén tén

3.6. ábra. Kálmán-szűrő hatása

mot szemrevételezve egy helyénvaló állítás az, hogy normális eloszlásra emlékeztető formával rendelkeznek. Ezt nagy biztossággal állítani különböző statisztikai tesztek elvégzésével lehet, ezúttal nem kerül végrehajtásra, mivel ha azokon el is bukik a mérési adatsor, a Kálmán-szűrő modellezés szintjén mindenképp normális eloszlású hibát feltételez, így el kell fogadni az ábrán látott mértékű hasonlóságot. Jelen esetben az időfüggvénynek még nincs akkora jelentősége, viszont a későbbiekben látható lesz egy lényeges jelenség. A 3.6 ábrán látható beállítások mellett a Kálmán-szűrő látványos hatást gyakorol a mérési eredmények eloszlására. Jelentősen mérséklődött a szórás, ezzel szemben látszólag a várható érték nem módosult. Ez a két tapasztalat pedig együttesen azt jelzi, hogy a célnak megfelelő szűrés valósult meg. Ebben az esetben viszont az időfüggvényen már megfigyelhető egy jelenség, amely egyelőre nem szignifikáns, csak akkor vehető észre, ha jobban ráfókuszálna az ábra a mérési eredmények elejére. A következő (3.7) ábra egy elfajuló esetet szemléltet, amikor túlságosan nagy hangsúlyt érvényesít a felhasználó a folyamat modelljére, és egyúttal túllövi a mérésekhez hozzáadódó zajt. Mindkét szűrőt alkalmazó ábrában 0 kezdeti értékkel vannak ellátva a szűrők. Ennélfogva a szűrő beállási idején múlik, hogy a mérések tapasztalati várható értékét milyen hamar közelíti meg (illetve milyen hamar követi le annak változását), és mekkora szórással teszi azt a távolságbecslést végző Kálmán-szűrő. A 3.7 ábra extrém módon redukálja a mérési eredmények szórását, cserébe viszont a szűrő beállási ideje túlságosan megnőtt. Még 5000 távolságszámítás után sem állt be az állandósult állapot, amikor rááll a szűrő a konstans távolságra. Plusz információ, hogy ez a mérés egyedileg 10 Hz-es sebességgel készült, azaz 5000 mérés alatt több mint 8 perc telt el. A Kálmán-szűrő egy pozitív hatása, hogy tovább finomítja a felbontást, mert az egyenleteken áthaladó mérési eredmé-



(a) Hisztogram a Kálmán-szűrő kimene- (b) Időfüggvény a Kálmán-szűrő kime-
tén netén

3.7. ábra. Elfajuló Kálmán-szűrő hatása

nyek olyan értékekkel kombinálódnak, hogy lényegében elvész az első hisztogramon tapasztalt „szétterülés”.

A 3.4 ábrán látható Kálmán-szűrőből anchorok számával megegyező működik az applikációban párhuzamosan, mindegyik anchor-tag távolságra külön-külön 1 db.

Amennyiben a Kálmán-szűrőt és az átlagolást egyszerre használja a felhasználó, úgy először a Kálmán-szűrőn halad keresztül a mérési adatsor, majd utána történik az átlagolás.

3.3.3. Pozicionálás

Az applikációban használt analitikus alapokon készült algoritmus műveleti igénye alapvetően nem számottevő. Viszont néhány megfontolás tehető, amelynek következtében tovább csökkenhet az egy ciklusra vonatkozó számításigény. Az 1.3.1. ToA (Time of Arrival) fejezetben ismertetett algoritmusban szereplő \underline{S} mátrix, illetve annak pszeudoinverze is számítható, mivel abban mindössze az anchorok koordinátái foglalnak helyet. Emellett a \underline{P} vektorban látható anchor koordinátás tagok szintén előre számíthatók. Ezáltal minden pozicionálási ciklusban az elvégzendő műveletek közé tartozik egy \underline{P} vektor frissítés az aktuális távolságokkal amelyekből le kell vonni az előzetesen rögzített értékeket, illetve egy szorzás művelet a pszeudoinverz és a frissített vektor között. MATLAB időmérő függvényét alkalmazva ez a művelet ms nagyságrendbe esik, amely elfogadható sebességet jelent.

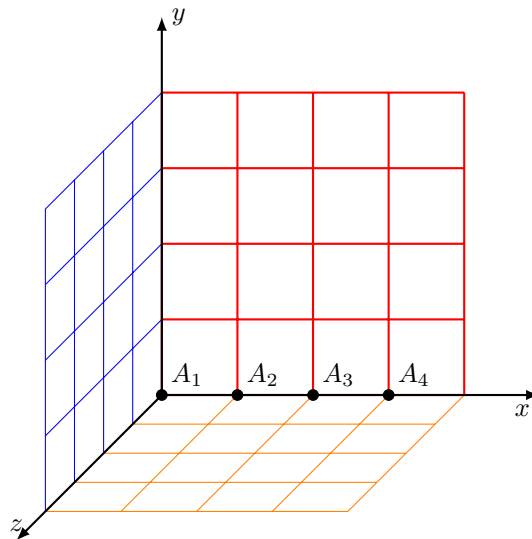
A numerikus alapokra készült pozicionálási algoritmus ezzel szemben előre nem paraméterezhető fel, azonban egy kezdeti értéket biztosítani kell minden pozicionálási ciklus elején a költségfüggvény minimumát kereső algoritmusnak. Ez az első ciklus elején egy (0,0,0) koordináta, majd utána mindig az előző becsült tag pozíció kerül a kezdeti érték helyére. Ez feltételezhetően egy alkalmas választás, mivel a tag

koordinátáinak változása nem túl nagy két pozicionálási ciklus között. Minél inkább gyorsabb a tag, ez annál kevésbé lesz helytálló, de kompenzálható a pozicionálási ciklusidő csökkentésével.

A pozicionálás vizsgálata 3 anchor elrendezésben történt meg. A 3 elrendezés merőben eltér egymástól, ennek az az oka, hogy megjelenjen rendszer esetleges érzékenysége az anchorok elhelyezkedésére.

Kiindulásként egy saját készítésű MATLAB szimulációs szkript használata segítségével történik a vizsgálat, amelyben meg lehet adni a tag tényleges pozícióját, illetve az anchorok koordinátáit. Ezt követően a szimuláció részeként kiszámításra kerülnek a tényleges távolságok, amelyeket additív hibával lehet terhelni. Ezzel egyszerűen lehet próbák alá vetni az alkalmazott algoritmus működésének sajátosságait.

Első elrendezésben az anchorok egymás mellett helyezkednek el, jó közelítéssel egy vonalban, amely a lokális koordináta-rendszer x tengelyét alkotja. Ezt szemlélteti egy lehetséges példával a 3.8 ábra. Analitikus algoritmus használata esetén a

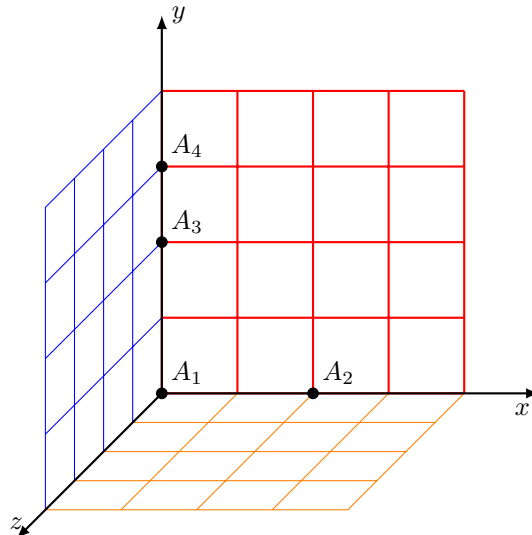


3.8. ábra. Elrendezés szemléltetése

szimulációs eredmények tanúsága szerint ideális esetben (additív hiba nélkül) a tag x koordinátája elhanyagolható (mindössze numerikus eredetű) hibával rendelkezik, ezzel szemben az y és z koordináták azonosan 0-ra értékelődnek. Ez valószínűleg az algoritmus sajátosságát tükrözi, azaz az algoritmus akkor tud visszaadni értékelhető koordinátát egy adott tengely mentén, ha azon a tengelyen található anchor koordinátája. A távolságokhoz hibákat hozzáadva ennek megfelelő eredmények keletkeznek. Ezt kifejtve csak az x koordináta értékelődik ki a hibák mértékétől és előjelétől függetlenül, természetesen az x koordinátának hibájára egyértelmű hatást gyakorol az additív távolságelrontás. Numerikus algoritmus használata esetén minden esetben a

tag összes koordinátája kiértékelődik, amely elvárható a költségfüggvény minimalizálás kimeneteleként. A szimulációs tapasztalatok ezúttal abban eltérnek az analitikus algoritmusnál látottaktól, hogy a tényleges távolságokat nem befolyásolva a numerikus eredetű hibák számottevőek, és összemérhetőek lehetnek a tag koordinátáinak nagyságrendjével. Ezen elsődlegesen az anchorok elhelyezkedése segít. Az egy tengely mentén történő elhelyezkedés ilyen szempontból egy kifejezetten kedvezőtlen választás. Amennyiben még a hibák hozzáadása is megtörténik a számított távolságokhoz, akkor az algoritmus kimeneteként kapott koordinátahármas a rá jellemző hibamérték miatt gyakorlatban nem használható.

Második elrendezésben az anchorok elrendezése módosul, így már x és y tengely mentén is lesz kiterjedése az anchorok által kifeszített alakzatnak a lokális koordináta-rendszerben. Ezt szemlélteti egy lehetséges példával a 3.9 ábra. Az ana-

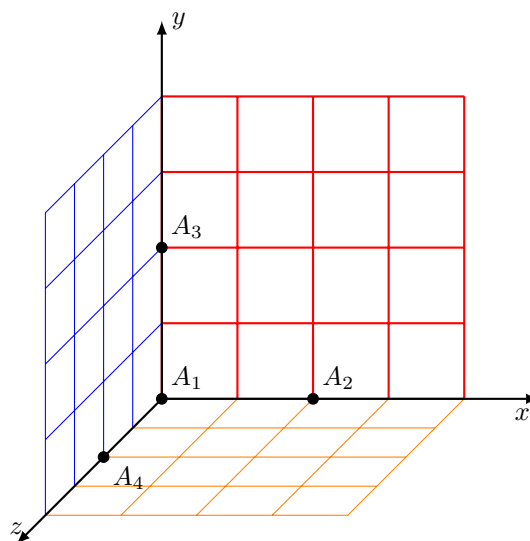


3.9. ábra. Elrendezés szemléltetése

litikus algoritmus az előző esetben levont következtetéseket a megújult elrendezésben az előző elrendezéshez hasonlóan igazolja. Ez azt jelenti, hogy immáron a tag x és y koordinátája mindenképp kiértékelődik, és csak a z koordináta marad azonosan 0. Egy másik szimulációs tapasztalat az, hogy a amennyiben van kiterjedése egy újabb tengelyirányban valamelyik anchornak (ezáltal az algoritmus képes kiértékelni a tag ezzel azonos koordinátáját), úgy annak a kiterjedésnek egy legalább egy nagyságrenddel kell meghaladni a számított távolságok abszolút hibáját. Ennek köszönhetően nagy valószínűséggel a pozicionálási hiba összevethető lesz azzal a hibával amelyek a távolságokat terhelik. További eredmény megfigyeléseken alapul az a következtetés, hogy ilyen elrendezésben lényegét tekintve a 3D helymeghatározás egy 2D-ra való levetítése történik. A tag x,y koordinátája ismerté válik, a z

koordináta pedig eliminálásra kerül az algoritmus által. Numerikus algoritmus működésén a 2 tengely menti anchorok hatása egyértelműen pozitív. Hiba hozzáadás nélküli esetben tapasztalható pozicionálási hiba eltörpül a tag koordinátáinak nagyságrendjéhez képest. Természetesen továbbra is igaz, hogy mindhárom koordináta kiértékelésre kerül. Viszont a hibaterjedési kép, amelyet a számított távolságokhoz szándékosan hozzáadott értékek befolyásolnak, érdekes következtetésekre vezet. Az eredmények azt mutatják, hogy a tag x és y koordinátájára ez az elrendezés meglehetősen kis erősítéssel gyakorol pozicionálási hibát, szemben a z koordinátával, ahol viszont a tag koordináta értékével összemérhető, illetve azt meghaladó hiba képződik le. Ez valószínűleg a költségfüggvény definiálásából adódhat, egyszerűen a minimalizálás eredményét így befolyásolja az anchorok elrendezése.

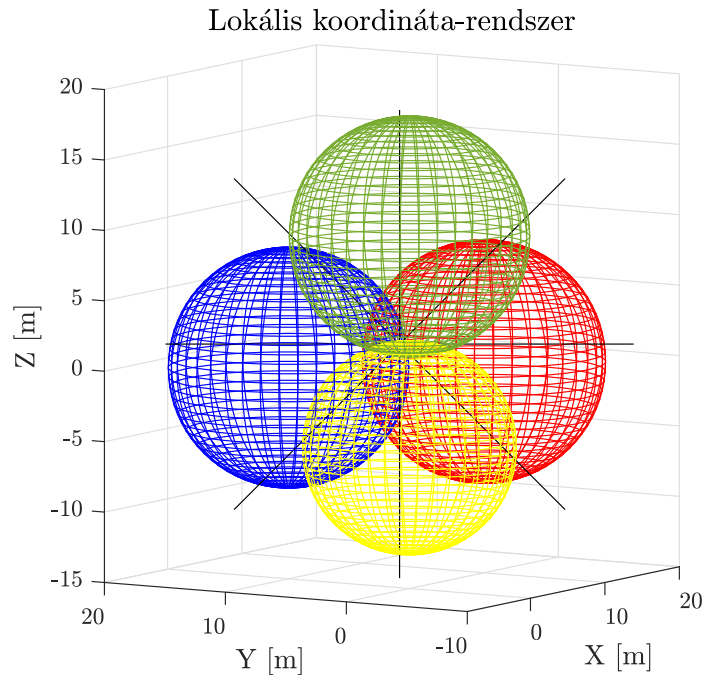
Harmadik elrendezésben már minden tengelyirányban van kiterjedése az anchorok által kifizített alakzatnak, az előzőekben ismertetett tapasztalatokat figyelembe véve ez lehet a leginkább hatékony formáció. Ezt szemlélteti egy lehetséges példával a 3.10 ábra. Analitikus algoritmus alkalmazásával elérhető, hogy a tag mindhárom



3.10. ábra. Elrendezés szemléltetése

koordinátája ki legyen értékelve, és hibamentes számított távolságok esetén numerikus számítási hiba sem terheli az eredményül kapott pozíciót. A számított távolságokat hibával terhelve az a tapasztalat, hogy a hibák nagyságrendjét általában össze lehet vetni azzal a hibával, amellyel a tag pozíciója terhelve lesz. Előfordulhatnak kiugró értékek, ezek valószínűleg a konkrét anchor elrendezés következtében lépnek fel, az algoritmusban alkalmazott mátrix pszeudoinverz számítás eredményezheti. Ezt a gondolatot az támasztja alá, hogy kiugróan nagy hibaterjedésnél az anchorok pozícióját minimálisan változtatva esetenként szignifikáns hibaterjedés

változás lép fel, amely kedvezően befolyásolja a tag pozíciójának számítási hibáját. Numerikus algoritmus használata esetén hibamentes távolságokat feltételezve a 3D-



3.11. ábra. Numerikus algoritmus működésében kirajzolható gömbök

s anchor elrendezés nagy mértékben enyhíti a numerikus számításból eredő hibákat. Amennyiben viszont megjelennek a távolságokat terhelő hibafaktorok, úgy az analitikus algoritmussal ellentétben sokkal inkább változatos a hibaterjedési kép. Ez valószínűleg onnan ered, hogy a numerikus algoritmus realizálását tekintve valóban az anchorok körül térben kirajzolható gömbök metszéspontját keresi, amelyek ha a távolságok hibái miatt nem metszik egymást, akkor egy olyan pont értékelődik ki, amely mindegyik gömbhöz a legközelebb található. A 3.11 ábrán látható egy példa annak megjelenítésére, amikor az anchorok körül kirajzolódó gömbfelületek találkoznak, és metszéspontjuk környezetébe egy nyújtott ágú csillagpont kerül, amely a tag számított pozíciója.

A szimulációt követően a működő rendszer vizsgálata következett, amely esetben a hiba hozzáadása már a DWM1000 modulok működéséből fakadóan megtörténik. A hibák mértékén például az előzőekben bemutatott Kálmán-szűrő képes csillagpítani. A vizsgálat során a szimulációnak megfelelő anchor pozíciók kerültek rögzítésre, illetve megvalósult mind a numerikus, mind az analitikus algoritmus alkalmazása. A mérési eredmények kielégítő lefedettségben állnak szimulációkon keresztül megismert tulajdonságokkal. Összességében az analitikus megoldás viselkedése jobbnak mondható, ez főként abban nyilvánul meg, hogy a grafikus megjelenítőn látott

tag pozíciójának szórása ugyan nem nagyobb, de az eloszlási kép sokkal inkább inhomogén. Az analitikus algoritmus használatával észrevehetőbb módon kirajzolódik egy egyre sűrűsödő gömbszimmetrikus felhő, ahol a tag pozíciójának valószínűsége a sűrűsödéssel összefüggésben van. Ezzel szemben a numerikus módszer alkalmazása esetén ez a felhő nem feltétlenül emlékeztet gömbszimmetrikus alakzatra, és a sűrűsége sem változik látványosan a rendszer üzemelése alatt.

4. fejezet

Helymeghatározó rendszer működése

4.1. Rendszertulajdonságok

Az alábbiakban ismertetésre kerül 3 lényeges tulajdonsága a helymeghatározó rendszernek, amelyek alapvetően meghatározzák a működését.

4.1.1. Pontosság

A pontosságot alapjában véve meghatározza az egyes anchor-tag távolságokhoz hozzáadódó mérési hiba, amely elsősorban a DW1000 IC-ben található LDE algoritmus sajátossága, így a diplomaterv keretében ez a pontosság az elméletileg elérhető maximum. A mérési eredmények azt mutatják, hogy ez a hiba anchor koordinátáktól függően terjed a pozicionálást végző algoritmus kimenetére. Előzetes megfontolásoknak megfelelően, amennyiben az anchorok minél inkább „kockára” hasonlító térbeli alakzatot feszítenek ki, általában annál kisebb a hibaterjedési együttható. Ennek az állításnak matematikai alátámasztása rendkívül bonyolult feladat, mivel az anchorok elhelyezkedéseinek száma elméletileg korlátlan, persze a gyakorlati megvalósíthatóság ezt jelentősen visszaszorítja.

Állandó helyzetű tag esetén az utólagos adatfeldolgozással elérhető pontosság 2D és 3D esetben is egyaránt átlagosan 8...12 cm. Vannak kiugró értékek, amelyek azonban szoftveres úton hatékonyan kezelhetők. Ilyenek azok a periódusok, amikor 0 m távolságot határoz meg egy anchor-tag páros, és ezt jelenleg nem szűri ki a rendszer.

Az egyik leginkább meghatározó pontosságot rontó tényező a DWM1000 modulon található chip antenna iránykarakterisztikája. Emiatt forgatás hatására $\pm 10...15$ cm eltolódás is létre tud jönni az egyes mért távolságokban, amely ked-

vezőtlen anchor elrendezés esetén akár 20...30 cm pozícióhibát is képes okozni. A jelenség azért állhat kapcsolatban az antenna iránykarakterisztikájával, mert van tapasztalati összefüggés a vételi jelteljesítmény, és a beérkezési időbélyeg létrehozása között. A gyártó ezt „range bias”-nek nevezi, amelyet arra az esetre mutat be, amikor a távolság növekedése miatt csökken a vételi jelteljesítmény. Viszont a jelenség akkor is fellép, ha az antenna sajátossága miatt csökken a vételi jelteljesítmény. Ezen javítani saját meglátás szerint 2 módon lehetséges. Az első, hogy más irányba, helyzetbe fordítani a kártyán belül az UWB modul, ezáltal az antenna környezeti tere esetleg kevésbé lenne perturbálva az egyes áramköri elemek által. A második, hogy a fejlesztői kártya alkalmazzon egy olyan szenzormodult (vagy többet), amely tartósan és pontosan tudja lekövetni az orientáció változásokat. Ezáltal szoftveres úton lehetővé válik egy időbélyeg kompenzáció az orientáció függvényében. Ez utóbbi egy terjedelmes kutatást, és fejlesztést kíván, illetve kész hardver esetén egy mérési metódus kidolgozását, amely során egy alkalmas táblázatba érdemes kigyűjteni az elforduláshoz tartozó távolsághibákat, amelyeket aztán időbélyeg módosításba lehet konvertálni.

4.1.2. Pozíciófrissítési gyakoriság

A pozíciófrissítési gyakoriság jelenleg időben változó mennyiség, átlagosan 5...7 Hz. Ennek fő oka a MATLAB által elérhető adatbáziselérési sebesség mértéke, illetve az az időtartam amíg megérkezik a megfelelő válasz a MySQL kérésre. Ennek a műveletnek periódusideje emellett összemérhető a jelenlegi UWB alapú üzenetváltások periódusidejével, amely periódusidő alatt az összes alkalmazott anchor meghatározza a kérdéses távolságot. A pozíciófrissítési gyakoriság növeléséhez további fejlesztési ötletként felmerül, hogy egyszerre 2 területen is érdemes megfontolni egy további optimalizációt.

4.1.3. Stabilitás

A stabilitás a rendszer bizonyos hatásokkal szemben tapasztalható tűrési képességét jelenti, illetve annak a körvonalazódása, hogy milyen mértékű behatásoktól válik szemmel láthatóan zavartnak a grafikus felület megjelenítőjén látható pozícióbecslés. Az imént említett behatás lehet hardveres, illetve szoftvertechnikai eredetű is. Ezt több módon lehet megközelíteni. Egy példa az idegen objektumok okozta direkt jelterjedési útvonalak megszűnése. A szélessávú UWB csomagok terjedésének leírása alapvetően más megközelítést igényel, mint egy keskenysávú technológia. Ezt többek között a csatorna frekvenciafüggő átviteli tulajdonsága befolyásolja. Ennek megfelelően előállhat olyan környezeti összeállítás, amikor a terjedési sajátosságok

miatt az UWB csomagok dekódolása után megnövekszik a bithibaarány, esetleg csomagvesztés lép fel. Egy-egy ilyen csomagvesztésből rendkívül nehéz nagy megbízottsággal következtetni a hiba okára, azonban ha a tesztelési, mérési elrendezés alakításával szándékosan előidézhető hibáról van szó, akkor már el lehet indítani egy következetes hibakeresési folyamatot. Egy ilyen folyamat eredménye az, hogy bizonyos orientáció esetén, amikor a tag egy anchort adott szög mellett lát, akkor ugrásszerűen megnövekszik a nem megfelelően dekódolt csomagok mennyisége.

Másik példa a tápellátás minőségével szemben támasztott követelmény. Üzemi körülmények mellett, azaz megfelelő tápellátás esetén, amely lehet külső akkumulátor, hálózati adapteres megoldás is elmondható, hogy a rendszer üzemelése stabil. Tesztelési fázis alatt nem volt tapasztalható rázkódás, kismértékű mechanikai behatás hatására szokatlanul nagy kitérés a felhasználói felület megjelenítőjén. Abban az esetben, ha viszont a külső akkumulátor töltöttsége jelentősen lecsökkent, akkor tapasztalható a távolságokhoz hozzáadódó zaj növekedése. Ez ésszerű reakció a rendszer működésében, mivel ha nincs biztosítva a megfelelő üzemi tápfeszültség, illetve az impulzus jellegű áramfelvételi igény fedezetlen, akkor a DWM1000 modul belső órajelei is nagymértékű kilengéseket produkálhatnak. Végző soron ez pedig a beérkezési időbélyeg illesztésének szórásának növekedését okozhatja. Megjegyzendő, hogy tipikus problémaforrás USB-n keresztül kapott tápfeszültség esetén a csatlakozó minősége okozta kontakt hiba. Ebben az esetben ilyen probléma nem jelentkezett, ehhez valószínűleg hozzájárul a jó minőségű microUSB csatlakozó. Stabil, masszív a csatlakozás minősége, nem érzékelhető a legkisebb mozgás, kotyogás a kézzel való megfelelő erő kifejtés hatására sem.

Egy harmadik példa a folyamatos üzem esetén fellépő rendszertúlterheltségi jelenségek fellépése esetén történő válaszreakció. A tesztelési fázis magába foglalt egy folyamatos, 8 órán át tartó üzemeltetést, amely alatt nem történt rendszerszintű hiba. Nem volt szükség egyik fejlesztői kártya újraindítására, az adatbázis szerver is megfelelően teljesített, illetve a PC-n futó MATLAB applikáció is gond nélkül viselte a több órás folyamatot. Eközben egy erőteljes melegedés jelentkezett az ESP modulokon, illetve DWM1000 modulokon egyaránt, viszont ez kivétel nélkül mindig beállt néhány perc elteltével egy állandósult értékre. Ezt egy célzólézerrel ellátott infrahőmérő által mért értékek támasztják alá. Egy MATLAB applikációs sajátosságára azonban érdemes felhívni a figyelmet. A pozícióbecslés indítása végző soron egy while ciklus indulását eredményezi. Abban az esetben, ha nem kerül beiktatásra a while ciklus elejére egy néhány ms értékű késleltetés, akkor hajlamos az applikáció a pozicionálás leállítása után is még 10...15 s-ig bent maradni a while ciklusban. Attól függetlenül, hogy a gomb megnyomása egy megszakítási műveletet indít el. Késleltető használatával ez a probléma teljes mértékben megszűnik.

Mindezek alapján kijelenthető, hogy a rendszer alkalmazhatósága, stabilitása a tesztelési folyamatokkal alátámasztott.

5. fejezet

Összefoglalás

5.1. Helymeghatározó rendszer értékelése

A diplomaterv kiírásban megfogalmazott feladatokat tekintve helymeghatározó rendszer működése megfelelő, képes meghatározni az anchorok által képviselt szenzorhálózat a tag pozícióját, illetve ezen utólagos adatfeldolgozással további javítás érhető el. Összességében a rendszer beüzemelése kívánja a legtöbb figyelmet, ugyanis az anchorok koordinátáinak rögzítése, és azok fix pozícióban tartása egy körültekintést igénylő feladat. A tervezési folyamatok során alkalmazott módszerek, gondolatmenetek produktuma utólag az élesztési, tesztelési fázisokban egyértelműen jelen voltak, mint ok-okozati összefüggések, ezáltal egyfajta fejlesztési siker könnyelhető el.

A járvány miatt kialakult gazdasági helyzet mellett ennek a rendszernek a megtervezése és megépítése hosszú, és rendkívül alapos, precíz munkát és odafigyelést igénylő feladat volt. Ennek dokumentálása a saját elvárásoknak megfelelő színvonalon tükrözi magának a rendszernek a kvalitását.

Összegezve saját munka értékelésének végkifejlete meglehetősen pozitív.

5.2. Fejlesztési lehetőségek

Hardver tekintetben megfontolást igényel az UWB modul egy másik pozícióban való elhelyezése, amivel esetleg kisebb érzékenységet mutathat a fejlesztői kártya a forgatásra. Ennek motivációja az, hogy a modulon található chip antenna közelterének szabadon hagyása méginkább hangsúlyos legyen.

Egy másik, akár utólagosan is elvégezhető módosítás, az akkumulátor maximális töltőáramának növelése. Ezt egyedül az akkumulátortöltő IC korlátozhatja, amelynek van egy maximális disszipációsteljesítmény-korlátja, amely fölé kúszva az IC nagy valószínűséggel meghibásodik. Célszerű egy termisztoros hőmérővel mérési

sorozatot végezni, hogy kiderülön a jelenlegi maximális hőmérséklet, ebből pedig (illetve a méréshez tartozó áramgrafikkal, és feszültséggrafikkal együtt) következtetni lehet, hogy a töltőáram növelésével mekkora hőmérséklet emelkedésre lehet még számítani. Amennyiben így valószínűsíthetően nem melegszik az IC az adatlapban szereplő maximum közeléig sem, akkor elvégezhető a módosítás.

Szoftveres úton egy nagyobb volumenű fejlesztési lehetőség, az anchorok által kialakított koordináta-rendszer automatikus kalibrációjának támogatása. Ez természetesen egy feltételrendszerben lehetséges csak, de annak használatával jó eséllyel kivitelezhető egy olyan megoldás, amelyben az anchorok egymás közti távolságukat meghatározva felállítanak egy térbeli alakzatot, amelynek egyes kitüntetett pontjait ők maguk alkotják.

További szoftvermódosítást igénylő fejlesztési feladat lehet az anchor-tag távolságok meghatározásának finomítása, illetve további optimalizálása, más preambulumok használata annak érdekében, hogy a jelenlegi pontosságot tartani tudja a rendszer, viszont kevesebb időre legyen szüksége egy távolság meghatározásához.

Végül de nem utolsó sorban a grafikus felhasználói felület mögötti tartalom fejlesztetőséget tekintve szintén nagy potenciált hordoz magában. Ilyen például a magasabb szintű hibakezelés megvalósítása, további paramétermódosítási lehetőségek, illetve egy párhuzamosan kapcsolt, önmagát hangoló Kálmán-szűrő implementálása. Ez utóbbi egy olyan sémát követ, amely szerint több Kálmán-szűrő egymással párhuzamosan dolgozik különböző R és Q varianciákkal, és a tag mozgási sebességétől függően mindig annak a szűrőnek a kimenetét veszi alapul a rendszer, amelyik a legpontosabb becslést adja.

Köszönetnyilvánítás

Köszönetet mondok Dr. Matolcsy Baláznak, hogy munkám teljes folyamata közben tanácsokkal, javaslatokkal látott el. Köszönöm, hogy lehetőséget biztosított a munkám elvégzéséhez, és a diplomatervem sikeres megírásához.

Mindemellett szerencsésnek mondhatom magam, hogy az egyetemi tanulmányaim alatt alapképzésen a témalabortól kezdve, mesterképzés végéig folyamatosan kísérték a szemléletformáló konzultációk, eszmecserék, problémafelvetések és megoldások, amelyekben konzulensként és emberként is példás hozzáállásával járult hozzá szakmai fejlődésemhez.

Szeretném megköszönni Gerhátné Dr. Udvary Eszternek a több éves fejlődési lehetőséget amelyet az általa vezetett laboratóriumban tölthettem. Továbbá köszönöm, hogy figyelemmel kísérte a laboratóriumban végzett munkáimat, és számtalanszor támogatta az előrehaladásom. Végző soron köszönöm Dr. Schranz Ágoston Kristóf értékes észrevételeit, amelyekkel hozzájárult a diplomatervem szerkeszttségének színvonalához. A laborban töltött idő mindig jó hangulatban, közvetlen magatartás mellett telt el, amelynek munkára gyakorolt pozitív hatását hatalmas élmény volt megtapasztalni.

Valamint tekintélyes tartozom szüleimnek, barátaimnak, akik hosszas egyetemi féléveim során támogattak, megértőek voltak hozzám.

Ábrák jegyzéke

1.1. Elemi UWB jellel való korreláció kimutatása [10]	4
1.2. Többutas terjedés beltérben	5
1.3. UWB jel vételi időfüggvénye [12]	6
1.4. UWB spektrális átlapolódása más technológiával	7
1.5. Kiküldendő alapsávi UWB csomag összetételének szemléltetése	8
1.6. Pozicionálás szemléltetése ToA alkalmazása esetén 2 dimenzióban . . .	14
1.7. Egyutas üzenetváltási séma	15
1.8. Kétutas üzenetváltási séma	15
1.9. Szimmetrikus kétutas üzenetváltási séma	16
1.10. Alternatív szimmetrikus kétutas üzenetváltási séma	16
1.11. Aszimmetrikus kétutas üzenetváltási séma	17
1.12. Pozicionálás szemléltetése TDoA alkalmazása esetén 2 dimenzióban .	20
1.13. Mért távolság hibája a tag és anchor közti frekvenciahiba-különbség függvényében TWR esetén	25
1.14. Mért távolság hibája a tag és anchor közti frekvenciahiba-különbség függvényében SDS-TWR esetén	26
1.15. Mért távolság hibája a tag és anchor közti frekvenciahiba-összeg függ- vényében ASDS-TWR esetén	27
1.16. Rendszerterv	29
2.1. Fejlesztői kártya blokkvázlata	31
2.2. DWM1000 kapcsolási rajz	33
2.3. Állapot visszajelző LED kapcsolási rajz	33
2.4. DW1000 integrált áramkör állapotdiagram	34
2.5. ESP32 modul kapcsolási rajz	35
2.6. Kapcsolási rajzok	36
2.7. USB csatlakozó és USB-UART konverter blokk	37
2.8. USB-UART konverter IC kapcsolási rajz	37
2.9. Akkumenedzsment blokk	38
2.10. USB lineáris regulátor kapcsolási rajz	38
2.11. Sorkapocs kapcsolási rajz	39

2.12. L6924D akkumulátortöltési karakterisztika	40
2.13. Akkutöltő IC kapcsolási rajz	41
2.14. Kapcsolási rajzok	42
2.15. Hőmérsékletfüggő visszaosztott feszültség-idő függvény	43
2.16. Akku lineáris regulátor kapcsolási rajz	44
2.17. Engedélyező áramkör kapcsolási rajz	44
2.18. Kapcsolási rajzok	45
2.19. Nyomatott áramköri terv felső oldal	46
2.20. Nyomatott áramköri terv alsó oldal	47
2.21. Kapacitás áramköri modell	48
2.22. Kapacitás impedancia diagram	49
2.23. Elkészült fejlesztői kártyák felső, és alsó nézetben	54
2.24. Dobozban rögzített fejlesztői kártya zárt tetővel, illetve nyitott tetővel	54
2.25. Akkumulátor és billenőkapcsoló bekötésének megjelenítése	55
2.26. Dobozba helyezett tag bekapcsolt állapotban	55
2.27. A 6 db dobozolt állapotban lévő elkészült fejlesztői kártya	56
3.1. Fejlesztői kártyán futó program folyamatábrája	58
3.2. MATLAB applikáció grafikus felhasználói felülete 3D nézetben	64
3.3. MATLAB applikáció grafikus felhasználói felülete 2D nézetben	65
3.4. Kálmán-szűrő állapotdiagramja	67
3.5. Nyers mérési eredmények 10000 mérésből	68
3.6. Kálmán-szűrő hatása	69
3.7. Elfajuló Kálmán-szűrő hatása	70
3.8. Elrendezés szemléltetése	71
3.9. Elrendezés szemléltetése	72
3.10. Elrendezés szemléltetése	73
3.11. Numerikus algoritmus működésében kirajzolható gömbök	74

Táblázatok jegyzéke

2.1. Státusz LED állapot táblázat	42
2.2. Élesztéshez tartozó lényeges mérési eredmények	51

Irodalomjegyzék

- [1] Gyártástrend weblapja: <https://gyartastrend.hu>, Utolsó elérés dátuma: 2022.03.06.
- [2] P. Puricer and P. Kovar, „Technical limitations of gnss receivers in indoor positioning,” in *2007 17th International Conference Radioelektronika*, pp. 1–5, 2007.
- [3] M. Piras and A. Cina, „Indoor positioning using low cost gps receivers: Tests and statistical analyses,” in *2010 International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–7, 2010.
- [4] IT business cikk: <https://www.it-business.hu>, Utolsó elérés dátuma: 2022.03.06.
- [5] Wifi alapú helymeghatározás <https://www.iotforall.com/>, Utolsó elérés dátuma: 2022.03.06.
- [6] M. Ture and A. Hatipoglu, „Indoor location finding of the transmitter based on bluetooth received signal strength,” in *2019 International Symposium on Networks, Computers and Communications (ISNCC)*, pp. 1–5, 2019.
- [7] K. Phutcharoen, M. Chamchoy, and P. Supanakoon, „Accuracy study of indoor positioning with bluetooth low energy beacons,” in *2020 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT NCON)*, pp. 24–27, 2020.
- [8] V. Grinyak, A. Shurygin, and A. Devyatisilnyi, „Accuracy of indoor navigation with bluetooth beacons,” in *2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, pp. 1–4, 2019.
- [9] *Revision of Part 15 of the Commission’s Rules Regarding Ultra WideBand Transmission Systems*. First Report and Order, (First R&O), FCC ET Docket 98-153,, Apr. 2002 Apr. 22.

- [10] *IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs): Amendment 1: Add Alternate PHYs*, 2007.
- [11] „Ieee standard for low-rate wireless networks,” *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)*, pp. 1–800, 2020.
- [12] Proxy NV weblapja: <https://www.pozyx.io/>, Utolsó elérés dátuma: 2022.03.06.
- [13] W. Wu, W. Wang, H. Yin, and D. Wang, „Carrier-less, single and multi-carrier uwb radio technology,” in *2004 International Workshop on Ultra Wideband Systems Joint with Conference on Ultra Wideband Systems and Technologies. Joint UWBST IWUWBS 2004 (IEEE Cat. No.04EX812)*, pp. 192–196, 2004.
- [14] Sávhasználati feltételek és frekvenciagazdálkodási követelmények az NMHH honlapján: <https://stir.nmhh.hu/>, Utolsó elérés dátuma: 2022.03.06.
- [15] UWB elérhetősége különböző országokban az Apple készülékein: <https://www.apple.com/uwb/>, Utolsó elérés dátuma: 2022.03.06.
- [16] Iphone 13 adatok az Apple honlapján: <https://www.apple.com/iphone13/>, Utolsó elérés dátuma: 2022.03.06.
- [17] AirTag adatok az Apple honlapján: <https://www.apple.com/airtag/>, Utolsó elérés dátuma: 2022.03.06.
- [18] Tech2 cikk: <https://www.tech2.hu/>, Utolsó elérés dátuma: 2022.03.06.
- [19] Computerworld cikk: <https://www.computerworld.com/>, Utolsó elérés dátuma: 2022.03.06.
- [20] Q. Zhou, C. Shen, X. Chen, and G. Feng, „Uwb wireless positioning technology in the application,” in *2016 IEEE Conference on Wireless Sensors (ICWiSE)*, pp. 106–109, 2016.
- [21] Y. Jiang and V. C. M. Leung, „An asymmetric double sided two-way ranging for crystal offset,” in *2007 International Symposium on Signals, Systems and Electronics*, pp. 525–528, 2007.
- [22] M. Kwak and J. Chong, „A new double two-way ranging algorithm for ranging system,” in *2010 2nd IEEE International Conference on Network Infrastructure and Digital Content*, pp. 470–473, 2010.
- [23] A. Norrdine, „An algebraic solution to the multilateration problem,” 04 2015.

- [24] J. Lagarias, J. Reeds, M. Wright, and P. Wright, „Convergence properties of the nelder–mead simplex method in low dimensions,” *SIAM Journal on Optimization*, vol. 9, pp. 112–147, 12 1998.
- [25] Y. Cheng and T. Zhou, „Uwb indoor positioning algorithm based on tdoa technology,” in *2019 10th International Conference on Information Technology in Medicine and Education (ITME)*, pp. 777–782, 2019.
- [26] B. Choi, K. La, and S. Lee, „Uwb tdoa/toa measurement system with wireless time synchronization and simultaneous tag and anchor positioning,” in *2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pp. 1–6, 2018.
- [27] W. H. FOY, „Position-location solutions by taylor-series estimation,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-12, no. 2, pp. 187–194, 1976.
- [28] B. Fang, „Simple solutions for hyperbolic and related position fixes,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 5, pp. 748–753, 1990.
- [29] X. Li and T. Zhang, „Research on improved uwb localization algorithm in nlos environment,” in *2018 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS)*, pp. 707–711, 2018.
- [30] KiCad weblapja: <https://www.kicad.org>, Utolsó elérés dátuma: 2022.04.19.
- [31] Inkscape weblapja: <https://inkscape.org/>, Utolsó elérés dátuma: 2022.04.19.