

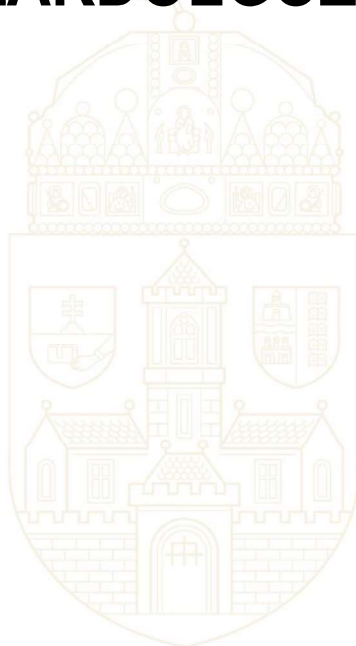


ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

KANDÓ KÁLMÁN VILLAMOSMÉRNÖKI KAR
ELEKTRONIKAI ÉS KOMMUNIKÁCIÓS RENDSZEREK INTÉZET
MŰSZERTECHNIKAI ÉS AUTOMATIZÁLÁSI TANSZÉK



SZAKDOLGOZAT



OE-KVK
2023.

Válint Zita
T009510/F112904/K



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Óbudai Egyetem
Kandó Kálmán Villamosmérnöki Kar
Műszertechnikai és Automatizálási Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve: Válint Zita

Szakedolgozat száma: SZD2309040923432867

Törzskönyvi száma: T009510/FI12904/K

Neptun kódja:

Szak: villamosmérnöki (magyar nyelven)

Specializáció: Műszer-automatika specializáció

A dolgozat címe: Evezős csapásszámmérő megvalósítása gyorsulásmérő szenzorral

A dolgozat címe angolul: Implementation of a rowing stroke rate monitor with an accelerometer sensor

A feladat részletezése: Készítsen evezős csapásszámmérésre alkalmas berendezést, amely szenzor rendszert, és hozzá tartozó felhasználói felület tervezését és szoftveres megvalósítást tartalmaz.

Intézményi konzulens neve: Sándor Tamás

Külső konzulens neve: Drexler Máté

Munkahelye: B.Braun Medical Kft.

A kiadott téma elévülési határideje: 2025. 12. 31.

Beadási határidő: 2023. 12. 15.

A szakdolgozat: Nem titkos.

Kiadva: Budapest, 2023. 11. 06.



Intézetigazgató

A dolgozatot beadásra alkalmasnak találom:

belső konzulens

külső konzulens



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

KANDÓ KÁLMÁN VILLAMOSMÉRNÖKI KAR
ELEKTRONIKAI ÉS KOMMUNIKÁCIÓS RENDSZEREK INTÉZETI
MŰSZERTECHNIKAI ÉS AUTOMATIZÁLÁSI TANSZÉK



HALLGATÓI NYILATKOZAT

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

Budapest, 2023.10.30.

.....
hallgató aláírása

Tartalom

1. Bevezetés.....	5
2. Evezős sport alapjai.....	7
3. Specifikáció	9
3.1. Hasonló termékek áttekintése	9
3.2. Működési elv meghatározása	11
3.3. A rendszer logikai terve	12
3.4. Csapásszámmérő szenzor elhelyezése	13
4. A szenzor fizikai megvalósítása	15
4.1. Fizikai rendszerterv	15
4.2. Gyorsulásmérő modul	17
4.3. Mikrovezérlő	19
4.4. Bluetooth modul.....	20
5. Szoftver logikai rendszerterv.....	21
6. Evezős mozgáshoz való igazítás	23
7. Szenzor szoftveres megvalósítása	27
7.1. A szoftveren belüli függőségek és megszakítások kezelése	27
7.2. Kommunikáció kezelése	29
8. Szenzorral való kommunikáció, OSI modellnek való megfeleltetés.....	31
8.1. BLE protokoll fizikai réteg	32
8.2. GAP (Általános elérési profil) jellemzése.....	33
8.3. GATT (Általános Attribútum Profil) jellemzése	35
8.4. A kommunikáció tesztelése.....	35
8.5. Hirdetett adatcsomag ismertetése.....	37
9. Felhasználói felület.....	39
9.1. Mérést szolgáló felhasználói felület.....	39
9.2. Adatok megjelenítését végző felhasználói felület.....	39
10. Továbbfejlesztési lehetőségek.....	42
11. Összefoglalás	43
12. Summary.....	44
13. Ábrajegyzék.....	45
14. Irodalomjegyzék	46

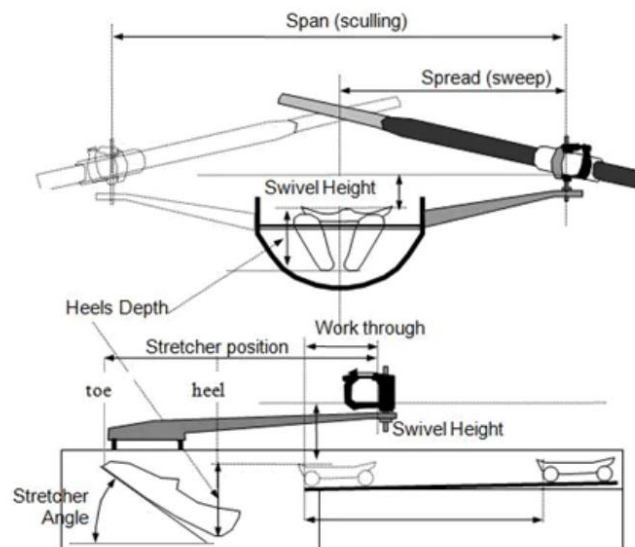
1. Bevezetés

A dolgozat fő témája egy evezős sportban használható csapásszámmérő rendszer fizikai és szoftveres megvalósítása. A versenysport minden területén előfordul a versenyzők teljesítményének növelése érdekében különböző elektronikus eszközök bevonása a felkészülés során. Az evezős sportban az egyik ilyen készülék a csapásszámmérő. Az edzések alkalmával egy csapásszámmérő nagyban hozzájárul a versenyzők szakszerű, monitorozott felkészüléséhez, ezért világszerte elterjedt edzést-támogató rendszernek számít. A Magyarországon működő vízisport-egyesületek is kivétel nélkül rendelkeznek ilyen eszközzel, azonban egyesületenként jellemzően csak 1-2 ilyen eszköz áll rendelkezésére a készülék magas ára miatt. A dolgozat elkészítése során célom az volt, hogy a piacon megtalálható, evezős csapásszámmérőkhöz hasonló rendszert valósítsak meg nagyságrendekkel olcsóbb, funkcióit tekintve továbbfejlesztett és működési elvében eltérő eszköz formájában. Az elképzelt rendszer elsődlegesen egy evezős hajó haladási sebességének (csapásszámának) mérését végzi, emellett idő és távolság mérésére, valamint adatok real-time kiértékelésére és megjelenítésére képes. Az így elkészített rendszer egy fejlett, de mégis költséghatékony edzéstámogatási rendszerként képes a versenyzők felkészüléséhez hozzájárulni. Az általam megvalósított rendszer első sorban egy gyorsulásmérő szenzorból és egy megjelenítést végző eszközből áll, ezzel is eltérve a piaci résztvevők által fejlesztett csapásszámmérőktől. A különálló szenzorral való megvalósítás előnyeit kihasználva pontosabb mérési adatokra tehetünk szert, emellett nagyobb hajóegységben akár több szenzor együttes használatára és az így mért adatok összehasonlítására is van lehetőség. A szenzor a gyűjtött adatokat Bluetooth hirdetőcsomagokba rendezve továbbítja a megjelenítést és kiértékelést végző eszköz felé, ami ebben az esetben az egyszerű fejlesztés és költséghatékonyság miatt egy mobiltelefon. A fogadott adatcsomagok egy elsődlegesen csak Androidos készülékre tervezett grafikus felhasználói felületen jeleníthetők meg. A felhasználói felület képes a szenzor által mért adatokat alkalmazási területnek megfelelő gyorsasággal, valós időben kiértékelni és megjeleníteni. Az adatok kiértékelése egy evezős hajó mozgási

paramétereinek megfelelően történik, aminek megértéséhez elengedhetetlen az evezős sport ismerete, ezért dolgozatomban ez is röviden összefoglalásra került. A kutatás ezek mellett részletesen ismerteti az adatgyűjtés és kiértékelés elvét, az elkészített szoftver logikai felépítését. Zárásként összegzésre kerülnek az eddig megvalósított eredmények, ezek mellett továbbfejlesztési lehetőségek és ötletek is említésre kerülnek.

2. Evezős sport alapjai

Az eszköz működésének és funkcióinak pontos megismeréséhez és megértéséhez szükség van az evezős sport átfogó ismeretéhez. Ebben az alponban ismertetem a fontos technikai alapokat, a működés megértéséhez szükséges alapokat. A 1. ábrán látható egy evezős hajó belső felépítése. A hajó belső részéből indulva (a kép lenti részén, balról jobbra haladva) láthatjuk: a sportoló lábának rögzítésére alkalmas lábtartót, felette a lapát tartására alkalmas villát és tulipánt, alatta az úgy nevezett kocsi, amin a sportoló ül és a sánt, amin a kocsi gurul. A hajó oldalára, a lapátok tartására szerelt villák végén, a lapátok úgy nevezett tulipánokban vannak rögzítve, amik a lapátnak bizonyos mértékben szabad elfordulást biztosítanak.

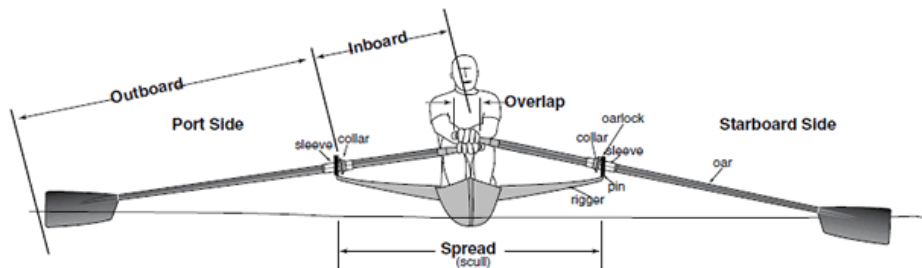


1. ábra: Evezős hajó keresztmetszete ¹ [1]

A következő ábra egy sportoló helyzetét ábrázolja a hajóban, a lapátokkal együtt, hátulról ábrázolva. Az evezős sportban a versenyző a hajóban ülve „hátrafelé” néz. Az evezős hajóban a mozgás első állapotánál a sportoló nyújtott lábával, a karjait behúzva ül a

¹ Az ábra a következő feliratokat tartalmazza felső szakaszon fentről lefelé haladva: fesztávolság (párevezős hajó esetén a két tulipán közötti távolság), a tulipánok között lévő távolság fele, a tulipán merőleges távolsága kocsihoz viszonyítva, lábtartó mélysége (a kocsi és a lábtartó sarka közötti távolság). Az ezt követő ábrarészleten látható szöveg fentről lefele majd balról jobbra: a tulipán lábtartótól való horizontális távolsága, lábtartó dőlésszöge, sarokmélység, sarokmélység, tulipán magassága a kocsihoz viszonyítva.

hajóban. A mozgás megkezdésénél először kinyújtja karjait, enyhén előre dől, majd a sportoló a lábai behúzásával elindul a lábtartó felé. A végállapot elérése után a sportoló lábaival ellöki magát, enyhén hátra dől, majd a karjait behúzza. Egyértelmű, hogy a sportoló így lényegesen nagyobb erőt tud kifejteni, mintha gurulókocsi nélkül haladna.



2. ábra: A sportoló helyzete a hajóban² [2]

Az evezős sportban a csapásszámot az alapján számoljuk, hogy a sportoló egy percen belül hány ilyen teljes csapást képes megtenni. Ez alapján az is feltételezhető lenne, hogy minél nagyobb ez a szám, annál jobban teljesít a versenyző. Ez teljes csapásokat nézve, hosszabb versenytávon nem igaz, hiszen ott fontosabb az egyenletes tempó megtartása a versenytáv nagy részén. Az evezős sport csupán rövid összefoglalása alapján is levonható az a következtetés, hogy a csapásszám mérése az edzések során fontos feladat. Egy ilyen csapásszámmérő eszköz segítségével a versenyzők hatékonyabban készülhetnek fel az általuk teljesítendő versenytávra, valamint az edzések során is jobb eredmények elérésére serkenti őket. Az evezős sportban jelenleg is előfordulnak ilyen felkészülést segítő eszközök, azonban ezek piaci ára meghaladja a legtöbb versenyző anyagi korlátait. Projektemmel az ilyen versenyzőknek is lehetővé szeretném tenni a hatékonyabb edzéseket.

² Az ábra szövege balról jobbra: a lapát külső fele; bal oldal; lapát mandzsettája; lapát bilincs; lapát belső része; átfedés, alatta a két tulipán közötti távolság; villa; bilincs; tulipán; tulipán főtengele; mandzsetta.

3. Specifikáció

Az előző pontokban nagy körvonalakban ismertettem az evezős sport dinamikáját, ez alapján ebben a pontban a felmerülő alkalmazási lehetőségek és megvalósítási opciók felvázolása következik, mivel célom az, hogy a lehető legtöbb lehetséges funkciót megvalósítsam.

3.1. Hasonló termékek áttekintése

Első sorban fontos említést tenni a kereskedelmi forgalomban megtalálható csapásszámmérő eszközökről és azok működéséről. A 3. ábrán látható egy Nielsen-Kellerman SpeedCoach, GPS használatával működő csapásszámmérő. Ez az evezős sportban az egyik legnépszerűbb csapásszám mérésre és megjelenítésére alkalmas eszköz, ami több fizikai mennyiség mérésére képes alrendszerrel kiegészíthető. Ez lehetővé teszi pontosabb adatok rögzítését és a mérés kiterjesztését. Az említett csapásszámmérő kiegészítések nélkül is több adat kijelzésére képes. Ezek az adatok a következők: 500 méterre levetített átlagidő (hajó gyorsasága), csapásszám (stroke/minute), távolság, idő. A csapásszámmérő eszköz $\pm 2,5$ méteres pontossággal rendelkezik. Külön kiegészítő funkcióként a gyártó felkínál egy Training Pack elnevezésű szolgáltatást, ami lehetővé teszi az edzések beprogramozását (tehát az edzés alatt teljesítendő távolságokat vagy időt) és ezek vezetékek nélküli adat-feltöltését is biztosítja.



3. ábra: Példa GPS alapú csapásszámmérőre [3]

A csapásszámmérő eszköz mellé a gyártó (Nielsen-Kellerman) további kiegészítő eszközöket is biztosít, például egy folyó vízben pontosabb mérési eredményeket biztosító impellert, ami a következő képen látható. Egy ilyen kiegészítő eszköz a hajó aljára rögzítve méri a propeller forgási sebességét, tehát a hajó vízhez viszonyított mozgási sebességét (a csapásszámmérő a hajó földhöz viszonyított sebességét méri, ami állóvízi közegben teljesen megfelelő, de folyóvízben pontatlan eredményeket ad).



4. ábra: Példa a csapásszámmérő kiegészítését szolgáló impellerre [4]

További kiegészítő eszközként megvásárolható a Training Pack szolgáltatás által támogatott EmPower Oarlock szenzor, aminek használatát a következő kép illusztrálja. Ez az eszköz a csapásszámmérő által nem mért értékekről szolgáltat adatokat, például a csapáshosszról és a csapás hatékonyságáról.



5. ábra: EmPower Oarlock szenzor [5]

Összesítésként megállapítható, hogy mérnöki megvalósítás tekintetében a piacon már rendelkezésre álló eszközök is kielégítik a vásárlók igényeit, azonban ezen eszközök

beszerzése költséges. Példának okáért, az először említett csapásszámmérő (SpeedCoach GPS) kereskedelmi ára 180 236 Ft, az impeller ára 12 043 Ft, az EmPower Oarlock szenzor ára 199 636 Ft (/darab).³ Ezen árak megismerése után megfogalmazódott bennem egy saját csapásszámmérő tervezésének ötlete. [3] [5] [4]

3.2. Működési elv meghatározása

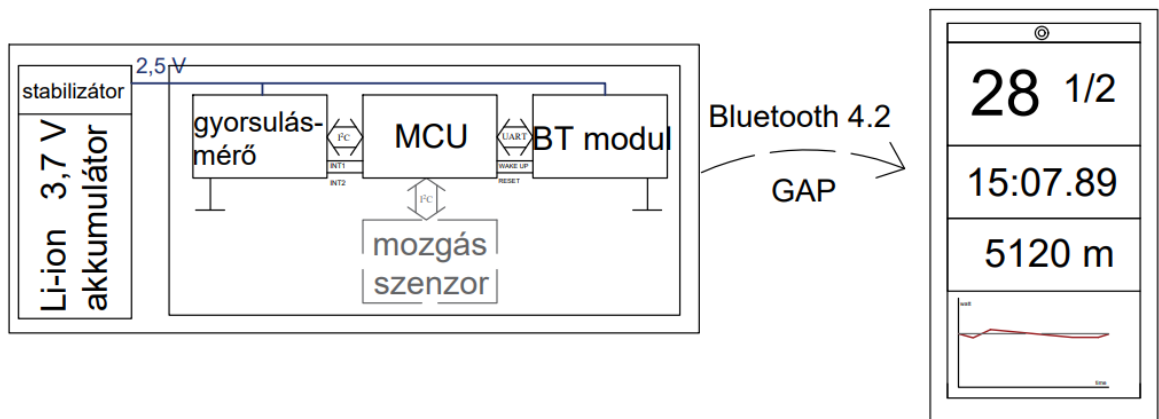
Az általam elképzelt csapásszámmérő eszköz megjelenítést tekintve jobban hasonlít a tulipánra (lapátot tartó forgó rész) szerelhető, csapáshosszt rögzítő szenzorokra, amilyen például az előző alponthan említett EmPower Oarlock edzés segítő szenzor (5.ábra). Elsődleges elképzelésként az adott szenzor egy csapásszámmérő rendszer részeként, csak adatgyűjtő egységként szerepelne. A költséghatékonyság miatt a megjelenítést végző eszköz egy mobiltelefon, mivel a mai világban már a legtöbb ember rendelkezik eggyel. Ennek köszönhetően mindenkinek használhatóvá válik egy nagyságrendekkel olcsóbb, felhasználóbarát eszköz. Ez az elrendezés lehetővé teszi, hogy nagyobb hajóegységben akár több ember is tudja egyszerre, ugyan azon szenzor adatait a saját készülékén megjeleníteni, ezzel hatékonyabbá tenni az edzéseket. A szenzorral szemben egyaránt követelmény a költséghatékonyság, a pontos mérési adatok szolgáltatása és a felhasználóbarát működés is, ezeket figyelembevéve törekedtem a lehető legjobban kivitelezni a feladatot. A feladat megvalósítását egy gyorsulásmérő szenzor által képzeltem el. Az eddig ismertetett eszköz a helyzet és csapásszám meghatározásához GPS-t használ gyorsulásmérő szenzorral kiegészítve. Annak ellenére, hogy ezzel a módszerrel is nagy pontossággal, $\pm 2,5$ méteres hibahatárral meg lehet határozni a kívánt adatokat, a feladat megoldását máshonnan közelítettem meg.

Az általam használt szenzor elvét tekintve jobban előtérbe helyezi a dinamikus mozgás és statikus gyorsulás jellemzőinek mérését. [6] Ebből kifolyólag elképzelésem szerint hasonló hatékonysággal, főleg a lapát mozgására figyelve meghatározhatóak a kívánt adatok. A következő fejezetben látható az elképzelt csapásszámmérő rendszer ábrázolása.

³ Az említett árak 2023. októberére vonatkoznak.

3.3. A rendszer logikai terve

Az elképzelt csapásszámmérő rendszer hajón belüli pontos elhelyezésének megértéséhez először a szenzor rendszerben való működését ismertetem. Az rendszer eleme egy méréseket végző szenzor és az adatok megjelenítését végző mobiltelefon. Az adott önálló energiaellátással rendelkező szenzor egy gyorsulásmérő modullal, az evezős sportoló mozgásához igazított mintavételezési frekvencia szerint rögzíti a kívánt értékeket. Ezek után egy Bluetooth protokoll-kompatibilis modul használatával továbbítja a csomagokba gyűjtött adatokat.



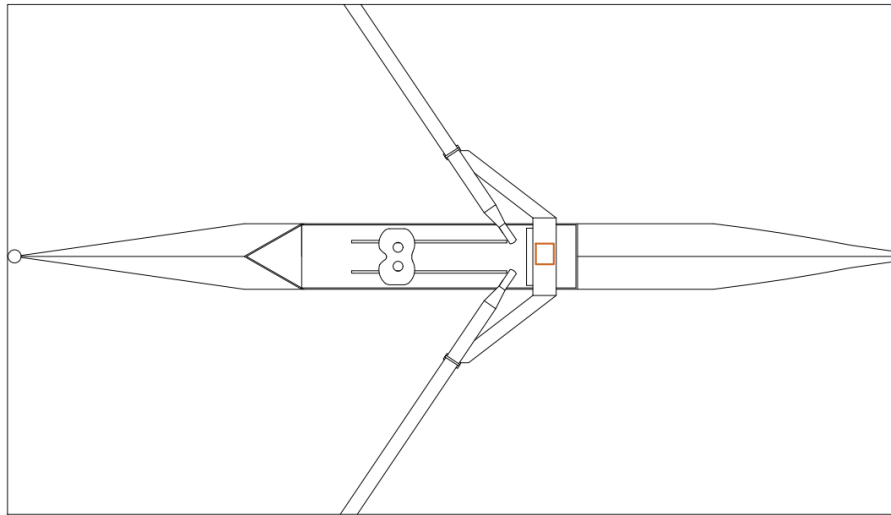
6. ábra: Az elképzelt csapásszámmérő logikai rendszerterve

A fogadott csomagok egy mobiltelefonos alkalmazásban, megfelelő szűrési és adatfeldolgozási logika alapján, kerülnek kiértékelésre. Az elrendezés használatának egyik előnye, hogy telefonos applikáció használata lehetővé teszi a FOTA, azaz firmware vezeték nélküli frissítésének lehetőségét. Ennek eredménye, hogy a szenzorok felprogramozása és a program működésének tesztelése után az egész rendszerben csak a mobiltelefonos alkalmazást kell funkcióbővítés esetén, vezeték nélkül internetes eléréssel frissíteni.

A GPS-t használó csapásszámmérővel szemben a rendszer másik előnye, hogy a GPS hálózat használatával járó bizonytalanságok és a környezetből származó akadályok nem hátráltatják a mérés folyamatát, hiszen maga a szenzort más külső rendszer a működésében nem támogatja. [7]

3.4. Csapásszámmérő szenzor elhelyezése

A mérési elvet meghatározza a szenzor elhelyezése. Az előző alpontokban leírtak szerint, a szenzor elhelyezkedése eltérő az eddig használt csapásszámmérő szenzoroktól. A piacon elérhető mérőegység (Nielsen-Kellerman SpeedCoach) GPS alapú, ezért a hajón belüli elhelyezkedés nem változtat a szenzor által szolgáltatott mérési eredményeken, bárhol elhelyezhető nagyobb hajóegység esetén is. A következő ábrán látható a GPS-es gyorsulásmérést használó csapásszámmérő egy lehetséges elhelyezése egy egypárevezős hajóegységen belül, az eszközt barna színnel jelöltem. [7]

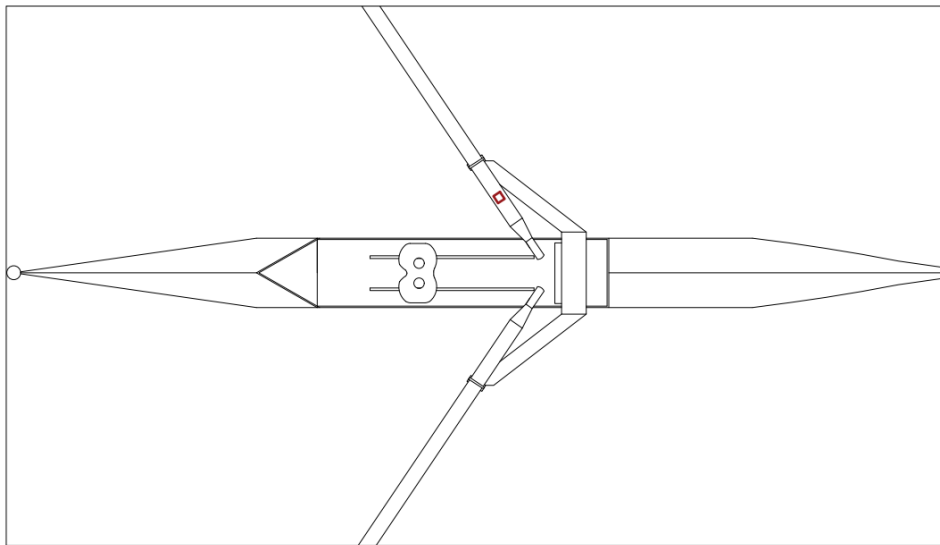


7. ábra: GPS alapú csapásszámmérő elhelyezése

Az általam tervezett szenzor elhelyezés az előző pontban is ismertetettek szerint eltérő. A szenzor erőforrásait tekintve a 7. ábra szerinti elhelyezés is megvalósítható lenne, de figyelembe véve azt, hogy a hajón belül a lapát mozgása, gyorsulásának iránya a legszemléletesebb, valamint a mozgás mintavételezése és az adatok feldolgozása is egyszerűbb feladat, ezért ettől eltérő elhelyezést választottam. Szempont volt ennek kiválasztásában, hogy mivel elektronikus eszközről beszélünk, a megfelelő, cseppálló borítás után se érje jelentős nedvesség. Mivel vízisport, ezért elkerülhetetlen, hogy a szenzor egy borulás után ne legyen vizes. Szerencsére ez gyakorlott sportolók esetén elkerülhető lehet, hiszen a vízisportok közül az evezőshajó (keskenyebb versenyhajók szempontjából is) az egyik legstabilabb. A szenzor méretéből adódó előnyöket kihasználva a lapáton való elhelyezést választottam. Azon belül, a sportolóhoz minél közelebb való elhelyezés tudja a legjobban biztosítani víz elkerülését. Ezzel ellentétben a

lapát mozgását tanulmányozva megfigyelhető, hogy a lapát vízhez közelebbi része a teszi meg az arra merőleges síkban a legnagyobb távolságot. A kettő megfigyelés alapján a szenzor számára legmegfelelőbb rögzítési pontot a lapát sportolóhoz közelebbi részén középen állapítottam meg.

Ezt az elhelyezést a következő ábra szemlélteti, ahol a szenzort piros téglalappal jelöltem. A kiválasztott rögzítési pontból a lapát mozgása során a vízre és a hajóra merőleges síkban történő elmozdulások hatásai is mérhetőek.



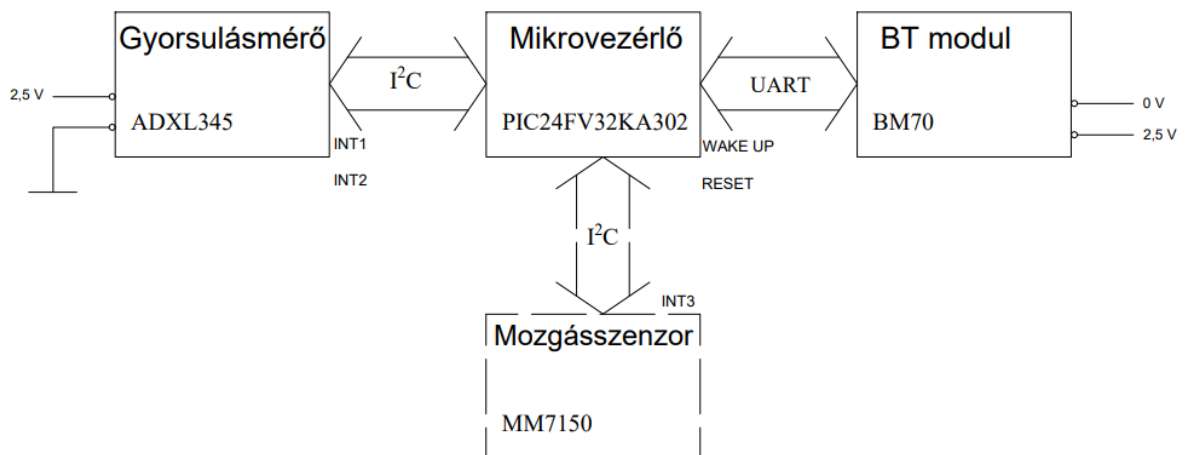
8. ábra: Gyorsulásmérő alapú csapáásszámmérő szenzor elhelyezése

A szenzor így megválasztott elhelyezése funkcióbővítés szempontjából is előnyt jelent: több szenzor által rögzített adatok utólag összehasonlíthatóak és elemezhetőek nagyobb hajóegységben történő alkalmazás esetén. Ezek a továbbfejlesztési lehetőségek a 10. fejezetben, lesznek részletesen felvázolva.

4. A szenzor fizikai megvalósítása

4.1. Fizikai rendszerterv

Ebben a fejezetben szeretném röviden ismertetni a szenzor felépítését. A fizikai rendszerterven látható, hogy a szenzornak három főbb alkatrésze van: egy ADXL345-ös gyorsulásmérő modul, ez az inercia adatok rögzítésére szolgál; egy BM70-es Bluetooth modul, ami a Bluetooth protokoll szerint való adatküldést teszi lehetővé; egy PIC24FV32KA302-es 16 bites mikrovezérlő, ami a mért valós idejű adatok adatsomagokba rendezését, kommunikációs paraméterek beállítását és ellenőrzését végzi. A rendszerterven szerepel még egy MM7150-es mozgás szenzor, amivel a jelenlegi rendszert továbbfejlesztés esetén tetszőlegesen bővíteni lehet (az áramkörtervezésnél a bővítési opció figyelembe lett véve). A jelenlegi rendszer a gyorsulásmérőre támaszkodik, mivel a követelmények teljesítése szempontjából a szenzor kibővítése nélkül is kielégíti az elképzelt felhasználáshoz szükséges pontossággal szemben támasztott elvárásokat. A szenzorral szemben a pontosság mellett követelmény az alacsony fogyasztás biztosítása: az akkumulátor élettartam meghosszabbításának céljából a szenzornak a mozgás abbamaradását követően egyből ki kell kapcsolnia. Ezt kiegészíti az a funkció, hogy csapásszámmérőként használva a sportoló felől érkező közvetlen beavatkozás nélkül visszkapcsoljon.



9. ábra: A szenzor fizikai rendszerterve

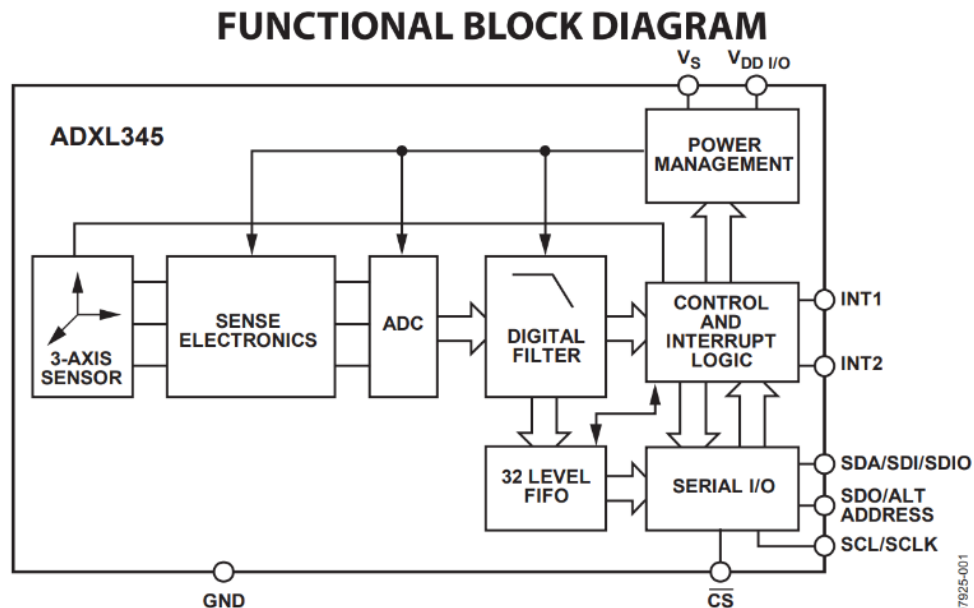
A szenzoregység ebből következtethető logikai működése alapján megköveteli a megszakítások használatát. A fizikai rendszerterven láthatóak a gyorsulásmérő modul és a mikrovezérlő (valamint a nem használt mozgás szenzor és a mikrovezérlő között) lévő INT1, INT2 (, INT3) jelölések. Ezek az áramkörön fizikailag szereplő külön a megszakítások jelzésére szolgáló vezetékeket jelölik, amik vagy kapcsolatban vannak, tehát utólagosan a megszakítást kiszolgáló programrészletben megkülönböztetés szükséges. Emellett, a mikrovezérlő és a Bluetooth modul között megtalálható meg két vezeték WAKE UP és RESET elnevezéssel. A WAKE UP vezeték a Bluetooth modul energiatakarékos alvó üzemmódból való „felébresztésének” céljából van jelen, a RESET vezetékre probléma esetén való újra konfigurálás esetén van szükség. A következő bekezdésekben az egyes modulok közötti kommunikációs paramétereket ismertetem.

A modulok közti kommunikáció is jelölésre került a fizikai rendszerterven: a gyorsulásmérő modul és a mikrovezérlő között I²C-n, a mikrovezérlő és a Bluetooth modul között UART-on. Az I²C a beágyazott rendszerek területén gyakran használt szinkron adatátvitelt megvalósító busz. A kommunikációhoz 2 vezeték áll rendelkezésre. A két vezeték közül az egyik továbbítja az adatokat, a másik az ütemezéshez és szinkronizációhoz szükséges órajelet, erre utal elnevezésük: SDA (Serial Data) és SCL (Serial Clock). [8] A másik megvalósuló kommunikációs szabvány az UART. Az UART, azaz univerzális aszinkron adóvevő egy iparban elterjedt és megbízható soros kommunikáció. A kommunikáció jellege az ismertetett I²C busztól eltérően aszinkron, ami szerint az adatátvitelhez nem szükséges az órajel, a szinkronizáció Baud Rate, start és stop bit beállításával történik. A kommunikáció megvalósítása érdekében fontos ennek pontos beállítása. [9] A kommunikációs paraméterek számszerű beállításait a 7.2. alpontban ismertetem.

4.2. Gyorsulásmérő modul

A szenzor elemeként egy ADXL345 háromtengelyes gyorsulásmérő modul méri a csapásszám megállapítása szempontjából hasznos inercia adatokat. A modul nem csak statikus gyorsulás, de dőlésszög mérésére is alkalmas eszköz. A következő ábrán látható a gyorsulásmérő modul funkcionális blokkdiagramja.

Fontosnak tartom a szenzor működési elvét röviden összefoglalni: egy poliszilikon alapú felület a felületre ható erők megváltozásával elhajlik. Az elhajlás mértéke differenciális kapacitás elven mérhető: a szenzor elhajlásra képes lemezére rögzített fegyverzet a rá ható erők megváltozásának hatására az adott mozgó réteggel kileng, ebből adódóan a kimeneten az elmozdulásnak megfelelő jelváltozás figyelhető meg. [10] [6]



10. ábra: A szenzorban szereplő gyorsulásmérő modul blokkvázlata [6]

Következő lépésben az érzékelő kimenetén mérhető jelből kell a hasznos jelet meghatározni, ez fázis érzékeny demodulációval valósítható meg, amivel meghatározható a gyorsulás iránya és nagysága. Az adatok feldolgozása után a megszakításokat és az irányítást végző részegység az adatoknak megfelelő megszakításként jelzi a mikrovezérlő felé az adott esemény bekövetkezését, ez alapján a mikrovezérlő I²C buszon adatküldést kezdeményez. A megszakítás forrás konfigurálhatósága mellett a mérési tartomány $\pm 2g$, $\pm 4g$, $\pm 8g$, $\pm 16g$ között változtatható.

Az alkalmazási területnek megfelelő méréshatár $\pm 2g$. A használat során az egyik legkritikusabb megválaszolandó kérdés a szenzorral szemben az alkalmazási területhez szükséges pontosság, tehát az, hogy csapásszámmérőként a megengedhető hibahatáron belül vannak-e a mért adatok. Ennek megállapítása során a következő paramétereket szükséges figyelembe venni: a szenzor belső koordináta-rendszer meghatározásából adódó hiba ($\pm 0,1^\circ$), a kereszt-tengelyes érzékenység ($\pm 1\%$), a kimenet felbontása ($\pm 2g$ méréstartomány esetén 10 bit), a szenzor érzékenysége (átlagosan 256 LSB/g, az átlagos érzékenységtől való eltérés $\pm 1\%$), a zajhatások (100 Hz-es kimeneti adatsebesség esetén X és Y tengelyen 0,75 LSB rms, Z tengelyen 1,1 LSB rms), az alvó állapotból való felébresztésnél fellépő késleltetés és nem utolsó sorban a hőmérsékletváltozás hatására fellépő pontatlanságok. Ezek az adatok a szenzor adatlapja szerint 0g gyorsulásra, 25°C-os közegre, 2,5V tápfeszültségre, 1,8V kimeneten megjelenő feszültségre, valamint 800 Hz-es kimeneti adatsebesség értékre vonatkoznak. A fenti jellemzőket összegezve megállapítható, hogy a szenzor alkalmazási területnek megfelelő. [6]

4.3. Mikrovezérlő

A szenzor részeként egy 16 bites PIC24FV32KA302 mikrovezérlő felelős a valós idejű adatok továbbításának ütemezéséért. A szenzor programja C nyelven íródott, ennek futtatását elősegíti, hogy a mikrovezérlő C compiler-re igazított, optimalizált utasításkészlet architektúrával rendelkezik. A használat során Flash memória és SRAM is rendelkezésre áll.

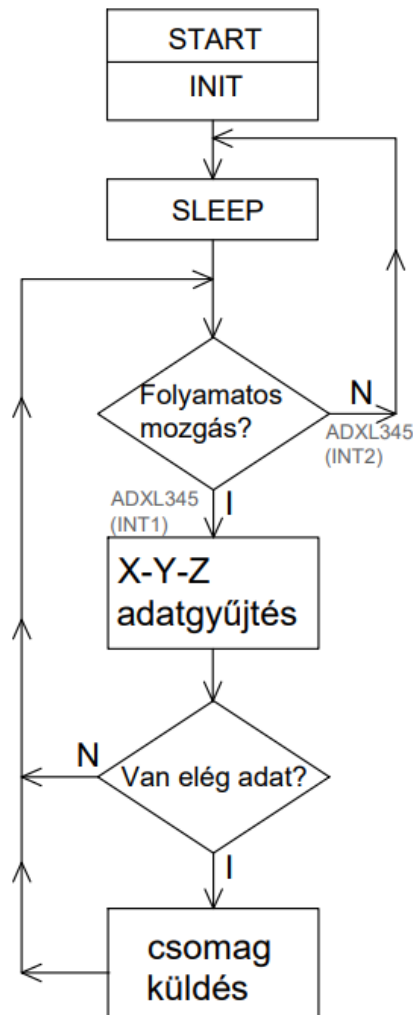
Az alkalmazási területre tekintettel fontos jellemzője, hogy különböző alvó állapot módokkal is rendelkezik. A funkciók jellemzői a következők: „Sleep” alvó üzemmódban a CPU (Központi Műveletvégző Egység), Flash és a perifériák kikapcsolt állapotban vannak, az SRAM „bekapcsolt” állapotban van. A „Deep Sleep” azaz mély alvás állapotban a CPU, Flash, SRAM és a legtöbb periféria kikapcsolt állapotban van, de ebben az esetben több különálló megszakításforrás definiálható, ami hatására az eszköz kilép ebből az állapotból. Ezek az állapotok a programfutási állapothoz lényegesen kisebb fogyasztás miatt előnyösek: a normál módban az áramfelvétel átlagosan $8\mu\text{A}$, alvó állapotban átlagosan 20 nA . A Deep Sleep állapot elérése a jelen alkalmazásban csapásszámmérőnek felhasználva előnyös, hiszen nem szükséges kikapcsolni a szenzort abban az esetben, ha a sportoló edzés közben egy rövid szünetet tart. A Deep Sleep állapot esetén egy külön Deep Sleep Brown Out Reset (DSBOR) és egy Deep Sleep Watchdog timer (DSWDT) számláló használható, ami független a rendszer Watchdog timer és Brown out reset funkciójától. A Watchdog timer egy alkalmazástól függetlenül futó időzítő, ami ebben az esetben a Deep Sleep módba lépés során nullázódik. A Brown out reset egy rendszer lefagyására való reakció, ami például feszültség ingadozás során lép fel. Mivel csapásszámmérőként a szenzor használatának célja mozgás közben történő mintavételezés, ezért az energiafogyasztás miatt előnyös, ha a mikrovezérlő nyugalmi állapotban Deep Sleep üzemmódba lép. A szenzor bekapcsolásának viszont a mozgás megkezdődésének hatására minél előbb el kell kezdenie méréseket végezni, ezért a Deep Sleep üzemmódból való felébresztés megszakítások használatával történik. [11] A Watchdog timer az alkalmazásnak megfelelően a szenzor nyugalmi helyzetbe lépését követően egyfajta „késleltetésként” használjuk a mikrovezérlő mély alvása előtt, ennek hatására javul a szenzor energiafogyasztása.

4.4. Bluetooth modul

A szenzorral való kommunikációt a csapásszámmérő rendszeren belül egy BM70-es Bluetooth modul támogatja. Ez az alkatrész Bluetooth Low Energy modulként a szenzor számára alacsony fogyasztást tesz lehetővé, amit a BLE 4.2 protokoll hatékony keretkihasználása, a szoftveres és a hardveres megvalósítás is kiegészít. Ennek eredményeképpen az akkumulátor üzemideje használat közben is jelentősen megnő. A modul több Bluetooth kommunikációs profilt támogat, mint például a GAP (Generic Access Profile), a GATT (Generic Attribute Profile), SMP (Security Manager Protocol) és az L2CAP (Logical Link Control and Adaptation Protocol). Ezek közül a szenzoron belül a GAP protokoll kerül alkalmazásra. A GAP a Bluetooth Low Energy protokoll részeként határozza meg a készülékek közötti kapcsolattartás jellemzőit és az adott eszköz betöltött szerepét a kommunikáció során. A GAP protokoll szerint az általam elképzelt csapásszámmérő rendszer részeként a szenzor slave eszközként, az adatokat hirdetőcsomagokba rendezve „hirdeti” az adatok fogadására és kiértékelésére képes mobiltelefonos alkalmazásnak. Ennek részletes leírása a 8. fejezetben található. A rádiófrekvenciás jellemzőit tekintve a modul az ISM (Industrial-Scientific-Medical) ipari, tudományos és orvosi alkalmazások számára behatárolt frekvenciasávot (2,402 GHz – 2,480 GHz) használja. A kommunikáció megvalósításához az eszköz belső antennával és egy 32 MHz-es kristály oszcillátorral rendelkezik. [12] [13]

A 4.2 és 4.3 alpontokban említettekhez hasonlóan ennél a modulnál is van lehetőség a modul elaltatására, ennek elősegítésére a modul a szenzoron fizikailag elkülönülő Wake Up vezetékekkel rendelkezik. A folyamat részeként a gyorsulásmérő mozgás érzékelését és a mikrovezérlő Deep Sleep üzemmódból való felébredését követően a Bluetooth modul az említett vezetékeken keresztül magas értéket kap, majd a mikrovezérlő által UART-on keresztül továbbított adatokat csomagokban hirdetni kezdi. A modul egészen addig van csak bekapcsolt állapotban, amíg aktívan adatokat továbbít.

5. Szoftver logikai rendszerterv



11. ábra: Szensor logikai rendszerterve

A fenti képen látható az adatgyűjtést végző szensor logikai rendszerterve, ami alapján történt a szensor programozásának kivitelezése. A szensor alapállapotban az energiatakarékosság biztosításának érdekében sleep üzemmódból indul. Csapásszámmérőként alkalmazva a mozgás szolgáltatja a hasznos információt, tehát itt erre az állapotra kell a szenzornak figyelni. Ez az információ az ADXL345 gyorsulásmérő modul lehetséges megszakításainak paraméterei között kell beállítani a programozása során. A gyorsulásmérő modul ebben az esetben két állapotban generál megszakítást: az egyik megszakítás mozgás érzékelésének hatására (INT1) az adatgyűjtés megkezdését, a

másik (INT2) hosszabb késleltetési idő lejárta után alvó üzemmódba való kapcsolását eredményezi. [6] Amennyiben egy csomag összeállítására elegendő adat gyűlik össze, a szenzor Bluetooth moduljának segítségével a mikrovezérlő által összeállított adatcsomagok továbbításra kerülnek. Az adatok mintavételezése során fontos tényező még, hogy a szenzor ne csak a mozgás megkezdésének időpontjában gyűjtsön adatot, hanem folyamatosan, egészen a mozgás teljes megszűnéséig, amíg a gyorsulásmérő újabb megszakítást nem generál. A nyugalmi állapot bekövetkezésével megszakítás generálása után először kikapcsol a gyorsulásmérő modul, majd a megszakítás hatására a mikrovezérlő is, ami ezzel egyidejűleg kikapcsolja a Bluetooth modult is. Mivel a szenzor a késleltetési idő lejárta után nyugalmi állapotban részlegesen kikapcsol, (a jelentős energiafogyasztást jelentő funkciók letiltásra kerülnek és csak a feltétlenül szükséges funkciók működnek) a szenzor külön kikapcsolására (fizikailag) nincs szükség. Ez a funkció megkönnyíti a csapásszámmérő felhasználójának, hogy erre a ne kelljen figyelmet fordítani, a szenzor automatikusan alkalmazkodik. Ezekből adódóan a csapásszámmérőt külön bekapcsolni sem szükséges. Eredetileg terveim között szerepelt, hogy az adatcsomagok hirdetése helyett közvetlen Bluetooth kapcsolat legyen a csapásszámmérő szenzor és a megjelenítést végző mobiltelefon között. Ennek elvetését a továbbfejlesztési lehetőségek mellett még az is indokolhatta, hogy hirdetőcsomagok adásával sokkal kisebb késleltetési idő szükséges kikapcsolás előtt, mint közvetlen Bluetooth kapcsolat kialakítása esetén: az első esetben nem gond, ha a szenzor inaktivitás után hirtelen kikapcsol. Példának vehetjük a következőt: a szenzort használó versenyző az edzés közben rövid pihenőt tart, majd a pihenő után tovább folytatja az edzést. A pihenő közben a hajó (a víz esetleges sodrásától eltekintve) mozdulatlan állapotban van. Közvetlen Bluetooth kapcsolat esetén a szenzornak a pihenő alatt is aktív állapotban kell maradnia, ezzel ellentétben hirdetőcsomagok esetén nem kell közvetlenül a szenzorhoz kapcsolódnia. Emiatt az energiafogyasztás is csökkenthető csupán ezzel a változtatással.

6. Evezős mozgáshoz való igazítás

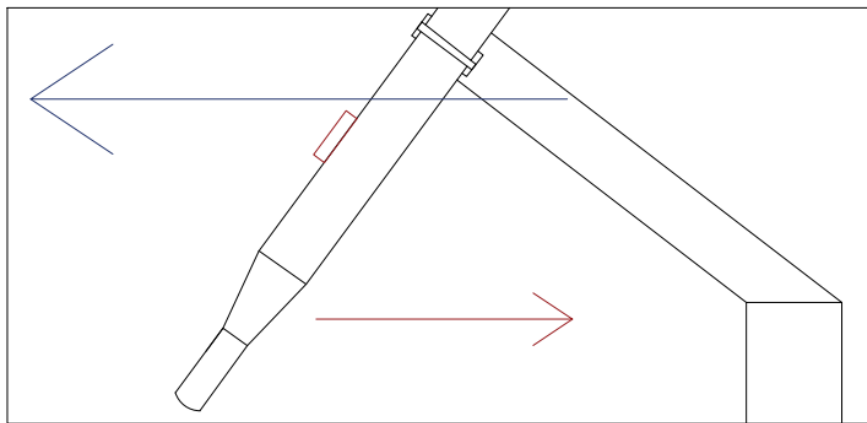
A logikai rendszerterv megismerése után felmerülhet még a mintavételezés gyakoriságának kérdése. Egy csapáson belül milyen gyakorisággal szükséges egy evezős hajó mozgásából értékeket rögzíteni, hogy az adott mintákból a sebesség és a csapásszám megfelelően nélkül kiszámítható legyen?

Ennek meghatározásához szükség van egy mérési határértéket megállapítani. Időhöz köthető mintavételezésről beszélünk-e, vagy elforduláshoz, esetleg a gyorsulásvektor irányának megváltozásához köthető? A megvalósítás szempontjából a legelőnyösebb a mintavételezést időhöz és a mozgás meglétéhez kötni. Ezek alapján jelentősen nagyobb adatfolyam rögzíthető, aminek elemzésével pontosabb értékek állapíthatók meg. A megfelelő mintavételezési frekvencia érdekében szükséges egy mérési határérték megállapítása, ami meghatározza azt a frekvenciát, amivel a számítások elvégzéséhez megfelelő mennyiségű adatot kapunk. Ehhez a lehetséges legnagyobb csapásszám meghatározására van szükség. Személyes tapasztalataim szerint a GPS alapú csapásszámmérő által mért abszolút maximális érték 60 csapás per perc. Ez egy irreálisan magas érték⁴, így a csapások „frekvenciája” körülbelül 0,83 Hz. A csapásszám értékének hiba nélküli mérése céljából a mintavételi frekvenciát úgy állítottam be, hogy a maximum értékhez viszonyítva is csapásonként legalább 7 minta készüljön. Az ez alapján kiszámított érték kibővítve 6,25 Hz.

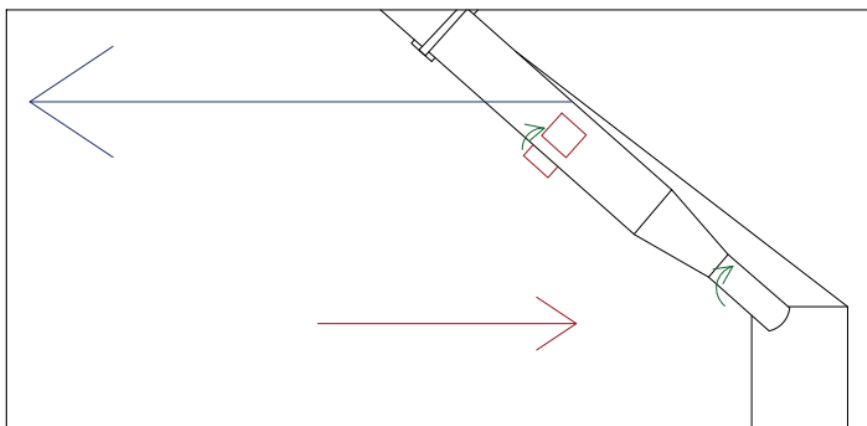
A mintavételezés módja mellett hasonlóan fontos a mintavételezett adatok feldolgozásának módja. Mivel a gyorsulásmérő szenzor alapvetően mozgás megléte alapján méri az adatokat, szükséges a hasznos adatok kinyerése végett bizonyos adatszűrési feltételeket kell meghatározni. A felhasználói felület így a megfelelően mintavételezett jelből, az említett adatszűrési feltételek felhasználásával lesz képes kiszámítani és megjeleníteni a csapásszámot. A szűrési jellemzők a szenzor elhelyezése alapján jelölhetőek ki. Ezt a 9.2. fejezetben található 24.ábra ismerteti. Az említett elhelyezés megválasztásával látványosan és egyszerűbb módon meghatározható a csapásszám. Ez a lapát által megtett mozgásnak köszönhető, mivel az evezési folyamat

⁴ A csapásszám értéke verseny közben jellemzően 28-37 közötti értéket vesz fel. Ennél nagyobb érték előfordulása esetén feltételezhető, hogy nem teljes csapásról beszélünk.

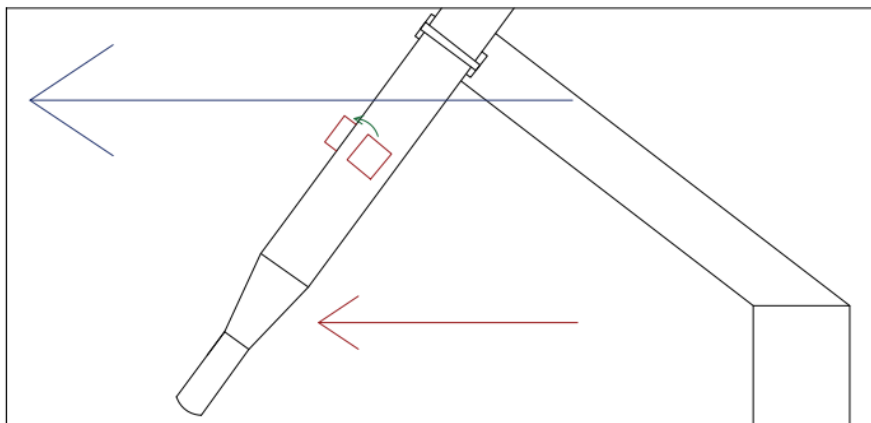
közben változik a gyorsulásvektor iránya, és a lapát orientáltsága is. A lapát orientáltságának változásából nem minden esetben lehet egyértelműen meghatározni a csapásszámot, mert az evezős sportban az evezési technika fejlesztésének céljából edzés közben előfordulhat, hogy a lapátokat a sportoló nem forgatja. Ebben az esetben is megfelelő a mérési elv, hiszen a lapátok gyorsulásának iránya ugyan úgy megváltozik. Egyelőre a jelenlegi alkalmazásban egyszerűsítés céljából feltételezzük, hogy a sportoló minden esetben a lapát forgatásával halad. A következő oldal képeiben egy evezős csapást bontottam folyamataira a haladási irány és a lapát gyorsulás vektorának feltüntetésével.



12. ábra: Lapát indulási állapota



13. ábra: A lapát vízfogáskor



14. ábra: A lapát helyzete erő kifejtés után

Az ábrákon a hajó haladási irányát kék, a lapát mozgásának irányát piros, a lapát és egyben a szenzor forgatásának irányát zöld nyilakkal jelöltem. A folyamatban az első kép (12. ábra) a jelenleg feltételezett indulási állapotot illusztrálja. Ebben az állapotban a lapát a sportoló öléből indul, a lapátot a víz felett csúsztatva, a lapátot a vízzel párhuzamosan vezetve indul meg a vízfogási állapot felé. A vízfogási állapot előtt a sportoló a lapátot 90°-kal elforgatja. A sorban következő ábrán a vízfogási állapot látható a lapát beforgatása után. A lapát beforgatásával a szenzor is elfordított állapotba kerül, aminek eredményeképpen a gyorsulásmérő modul által rögzített adatok a szenzorhoz rögzített koordináta-rendszer alapján megváltoznak. A vízfogást követően az erő kifejtés szakaszában a lapátot a sportoló a vízfogási állapotból maga felé húzza. A 14. ábra az erő kifejtés utáni állapotot mutatja, amikor a lapát forgatás után visszatér az eredeti kiindulási állapotba.

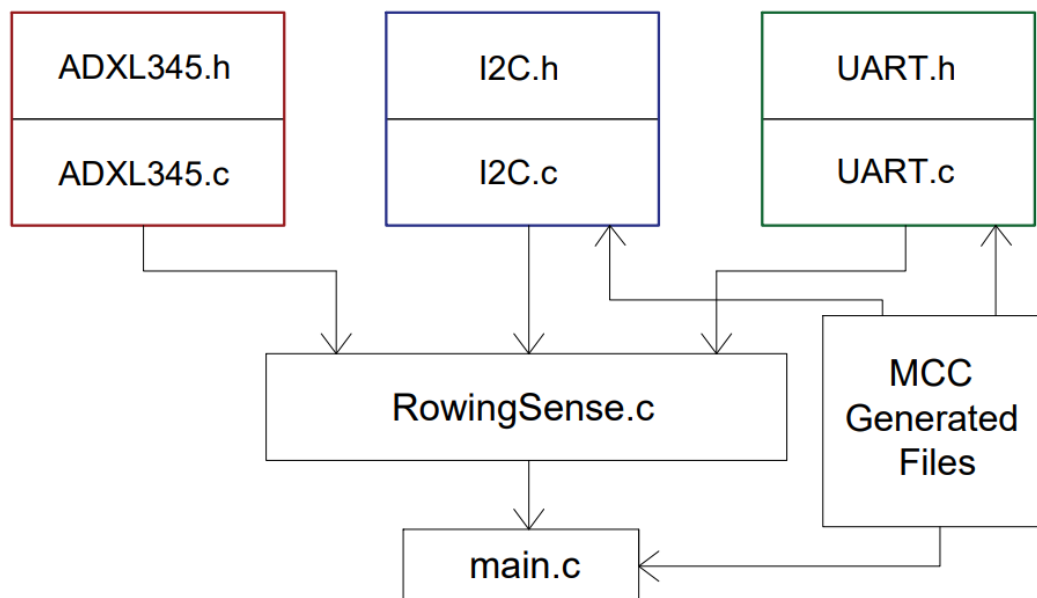
Az így meghatározott állapotokat alapul véve az adott mozgási állapotokban a lapát elfordulását tekintve a szenzoron található gyorsulásmérő modul alapján koordináta-rendszer rendelhető. A gyorsulásmérő modul koordináta-rendszere alapján rögzített mérési eredményekben az x, y, z koordináták előjelei ennek eredményeképpen a mozgás különböző állapotaiban eltérnek. Ennek ellenére a mozgás mintavételezés nem az elfordulástól függ. A szenzor az evezős sportban előforduló csapásszám értékek alapján beállított mintavételezési frekvenciával rögzíti az adatokat a mozgás minden szakaszában. Az így rögzített adatok alapján megkülönböztethetők az evezős mozgás közben fellépő állapotok, az adatokhoz a felhasználói felületet megvalósító alkalmazásban időbélyeg hozzárendelésével meghatározható a szenzor felvett helyzetei

között eltelt idő. A mintavételi frekvencia ismeretében és a fent említettek alapján meghatározható a lapát mozgásának iránya is. Ez a lapát helyzetét jelenti az adott csapáson belül, azaz a gyorsulásvektorok egymáshoz viszonyított irányát. Ennek meghatározására azért van szükség, mert csupán a lapát forgatásából nem biztos, hogy minden esetben határozható meg a csapásszám értéke. De milyen módon lehet megkülönböztetni koordináta-adatok alapján a két vektor párhuzamosságát vagy ellentétes viszonyát? A szenzor által továbbított adatsomagokat fogadás közben időbélyeghez rendeljük. Feltéve, hogy a szenzor mintavételezési frekvenciája az evezős mozgáshoz viszonyítva elég gyors, a két minta között eltelt idő és az adott pontban felvett koordináta-értékek alapján meghatározható a két gyorsulás. Az ezek alapján meghatározott mintáknál megfigyelhető, hogy a lapát gyorsulás-vektor és a hajó gyorsulás-vektor párhuzamossága esetén változatlan mintavételezési frekvenciával nagyobb a két pont között megtett távolság. Ellenkező esetben viszont a mintavételi pontokban felvett értékek távolságot tekintve sűrűsödni látszanak.

7. Szenzor szoftveres megvalósítása

7.1. A szoftveren belüli függőségek és megszakítások kezelése

A szenzor számára az 5. fejezetben ismertetésre kerültek a logikai rendszerterv alapján a megvalósítandó követelmények. Az így definiált szoftver felépítését és függőségi kapcsolatait a 15. ábra jeleníti meg. A szoftver tervezése során első lépésként a fejlesztői környezethez tartozó Microchip kód konfigurátor (továbbiakban MCC) segítségével beállításra kerültek az eszközt jellemző paraméterek és felhasználni kívánt perifériák. A beállított adatok alapján ezeket az alapbeállításokat tartalmazó C és header állományok automatikusan létrehozásra kerülnek. Ezeket az állományokat az ábrán az MCC Generated Files (MCC által generált fájlok) pont jelöli. Ezek közé tartoznak például: az órajelre és időzítésre vonatkozó beállításokat, a kommunikációs csatornák alapbeállításai (I²C és UART), külső megszakításokkal kapcsolatos beállításokat tartalmazó fájlok. Ezek a fájlok csak a mikrovezérlő periféria szintű beállításait tartalmazzák, a kommunikációt kezelő függvények külön kerültek csoportosításra.



15. ábra: Szoftver függőségi kapcsolatai

Az ábrán látható, hogy a main.c állomány MCC által generált fájlokra való hivatkozása azzal indokolható, hogy a rendszer szintű beállítások a main függvényben kerülnek meghívásra. A main.c állományban ezek mellett a konfigurációs bitek is beállításra kerülnek, például: oszcillátor kiválasztását meghatározó, watchdog timer

előosztásának beállítására szolgáló, brown out reset enable bitek. Ezen paraméterek, mint például a watchdog timer is külön funkcióval van felruházva az adott alkalmazásban. Az ábrázolt függvénykönyvtár használatával képes a szenzor hatékonyan a mozgás érzékelésére és rögzítésére. A 6. fejezetben tárgyaltak alapján ennek megfelelően a következő feladat a mintavételezés szoftveres leképezése. Az említett fejezetben mozgás hatására kiváltott, adott frekvencia szerinti mintavételezést állapítottam meg. Ennek szoftveres leképezése megszakítások definiálása mellett adott frekvenciával történő mintavételezés formájában valósul meg. Megvalósítása a következő: a main.c állományon belüli rendszer inicializálást követően a main végtelen ciklusán belül a RowingSense állomány „fő” függvénye kerül meghívásra, majd ebben a függvényben a további alapbeállításokat követően ismételten szerepel egy végtelen while ciklus. Ebben a végtelen ciklusban a 16. ábrán is látható utasítások szerepelnek. A RowingSense.c állomány main függvényében látható utasítássorozat valósítja meg a logikai rendszerterven (11. ábra) látható, indulás után felvett alvó állapotot. Az említett alvó állapotból csak bizonyos megszakítások hatására „ébred fel” a szenzor, amiket jelen esetben a gyorsulásmérő modul alapján definiálhatunk. Ennek működését a következő ábra ismerteti.

```

int main(void)
{
    // initialize the device
    SYSTEM_Initialize();

    while (1)
    {
        RowingSense_main();
    }

    return -1;
}

int RowingSense_main(void)
{
    WaitTick(); // 1 TMRL
    LED_OG_SetLow();

    BM70_inicializalas();
    Nop();

    ADXL_inicializalas();
    Nop();

    LED_OG_SetHigh();

    while (1)
    {
        Nop();
        Nop();
        Nop();
        Sleep();
        Nop();
        Nop();
        Nop();
    }
}

void Interrupt0(void)
{
    Nop();
    if (ADXL_INT2_GetValue() == FALSE) // ADXL0 I
    {
        LED_OG_SetLow();
        ADXL345_GetRegisterValue(ADXL345_INT_SOUR
        EX_INT0_InterruptionFlagClear();
    }
    if (ADXL_INT1_GetValue() == FALSE) // ADXL0 I
    {
        LED_OG_SetLow();
        VBAT_DIVIDER_GND_SetDigitalOutput();
    }

    //////////////////////////////////////
    ADC1_ChannelSelect(4);

    ADICON1 = 0x0070; // internal counter
    ADICSSL = 0;
    ADICON3 = 0x0002;
    ADICON2 = 0x8800; // 2048 mV referencia

    while (HLVDONbits.BGVST == 0);

    IFS0bits.ADIIF = 0;
}

```

16. ábra: Megszakítások folyamata

A két gyorsulásmérő modul által kijelölt megszakítást kiváltó esemény bekövetkezése után a program az ezt kiszolgáló rutinra ugrik, ami a két esemény számára azonos. Itt a forrás megkülönböztetése után valósulnak meg az ennek megfelelő, logikai rendszertervben meghatározott funkciók.

7.2. Kommunikáció kezelése

Ebben az alfejezetben a szenzorrendszer moduljai között megvalósuló kommunikációs csatornák paramétereit mutatom be és azok adottságainak kihasználását az alkalmazás szempontjából. Elsőként a gyorsulásmérő és mikrovezérlő között megtalálható soros I²C busz működését és jellemző értékeit ismertetem. A szenzor esetében egyszerű master-slave kapcsolat van jelen. A busz működési elve a következő: a buszt használó slave eszköz(ök) csak akkor továbbítanak adatot, ha a master által kezdeményezve volt az adatküldés. A kommunikáció folyamata a következőként alakulhat: a master eszköz küld adatot, a másik esetben a master kér adatot a slave modultól. A szenzor tekintetében a következő esetek fordulhatnak elő: a mikrovezérlő adatküldés után abban az esetben nem vár választ a gyorsulásmérő slave modultól, amennyiben az üzenet célja a slave energiatakarékos üzemmódba való állítása. Másik esetben a master adatkérése akkor valósul meg, ha a gyorsulásmérő megszakítást jelzett az erre a célra kitalált vezetékek egyikén, amik használata azért is előnyös, mivel erre nem vonatkozik az I²C buszra vonatkozó hierarchia, tehát ebben az alkalmazásban a slave modul is kezdeményezheti a kommunikációt.

A szenzoron belüli adatátvitel a mikrovezérlő és a Bluetooth modul között UART-on történik. Az ehhez tartozó paraméterek közül a felhasználó által a rendszerhez igazítva kalibrálható a Baud Rate. A Baud Rate nem megfelelő vagy pontatlan beállítása információvesztéshez is vezethet, ennek előre definiált szabványosított értékei a következők lehetnek: 110, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200, 128000 és 256000 bit/s. A pontos beállítást a mikrovezérlőn belül az órajelből az osztó regiszterek állítják elő. Az osztó regiszter számára a pontos értéket az érték kiszámításához szükséges képletből tudjuk származtatni, ami a következő:

$$BaudRate = \frac{F_{CY}}{4 \cdot (U2BRG + 1)} \quad (1)$$

$$U2BRG = \frac{F_{CY}}{4 \cdot BaudRate} - 1 \quad (2) [11]$$

$$\text{ahol } F_{CY} = \frac{F_{osc}}{2} \quad (3)$$

A szenzor esetében az érték hiba nélküli előállításához egy 3,6864 MHz-es kvarc oszcillátor biztosítja az órajelet. Ezzel az értékkel standard soros vonali sebesség a mikrovezérlő belső osztó regisztereivel 0%-os hibával állítható be. A beállítani kívánt Baud Rate érték 115200, tehát ez alapján szükséges ebben U2BRG regiszter értékét beállítani. A kiszámított érték a következő:

$$F_{CY} = \frac{F_{osc}}{2} = \frac{3,6864 \cdot 10^6}{2} = 1,8432 \text{ MHz} \quad (4)$$

$$U2BRG = \frac{1,8432 \cdot 10^6}{4 \cdot 115200} - 1 = 3 \quad (5)$$

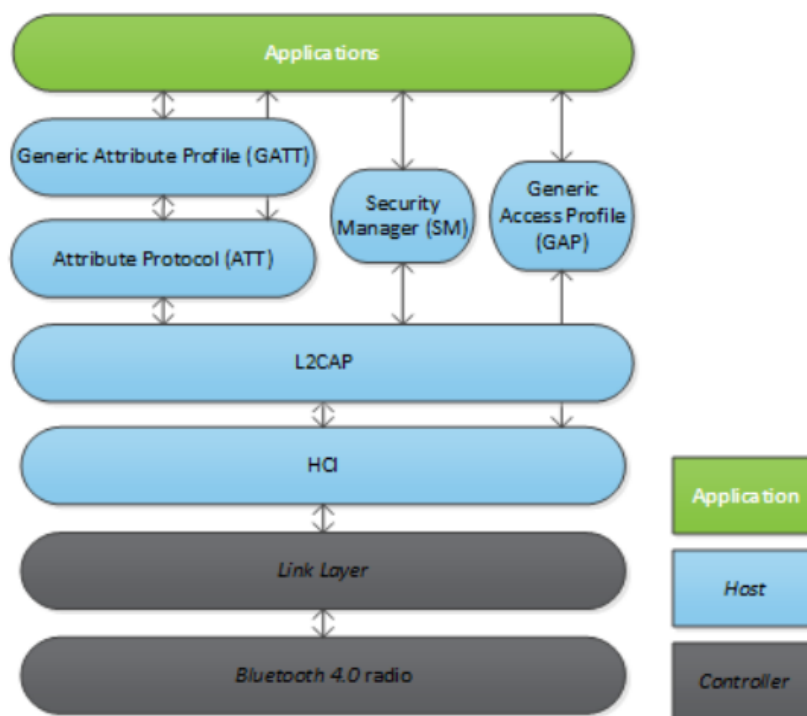
Az előző bekezdésben az I²C busz adatküldési folyamatához hasonlóan szeretném ezt az UART esetében is ismertetni. UART esetében a kommunikáció full duplex és fél duplex módon is megvalósulhat. A két megvalósítás a következőképpen hasonlítható össze: mindkettő két irányú kommunikációt jelent, azonban full duplex esetben történhet ez egyidőben is. Ebben az esetben fél duplex kommunikáció valósul meg, hiszen a master és a slave modul közötti üzenetváltások között mindig megtalálható az üzenet nyugtázása. Az adatátvitel során konfigurálható az adatkeretek felépítése is, ami ebben az esetben „8N1”, tehát az adatkeret 8adatbitből, és egy stop bitből áll, emellett nem rendelkezik paritásbittel. [14] [9]

A szenzor szoftveres leírását tartalmazó fejezet zárásaként megállapítható, hogy az eddig megvalósított funkciók nem merítik ki a rendelkezésre álló erőforrásokat. A szenzor memóriakapacitását a jelenlegi alkalmazás 7%-ig használja ki, ami a továbbfejlesztési lehetőségek figyelembevételével hasznos tulajdonság.

8. Szenzorral való kommunikáció, OSI modellnek való megfeleltetés

Ebben a fejezetben a szenzor és egy - kereskedelmi forgalomban kapható - „BM70 PICTail Plus” modul és a használt szenzor kommunikációján keresztül mutatom be a BLE protokoll OSI (Open Systems Interconnection) modell szerinti megvalósulását. A Bluetooth Low Energy protokoll stack OSI modellnek megfeleltethető ábrázolásában (17. ábra) az adott rétegek különböző, egymásra épülő absztrakciós szinteket képviselnek, melyek egy-egy definiált feladatot hajtanak végre.

Az OSI modell fizikai rétege a Bluetooth radio, felette az adatkapcsolati, vagy viszony réteg (Link Layer). Az ábrán a felső blokkban ábrázolt alkalmazási réteg alatt elágazás figyelhető meg, ami a köztes rétegeket megvalósító, GATT szolgáltatás keretrendszer és GAP profilokat jelöli. [15] [16]

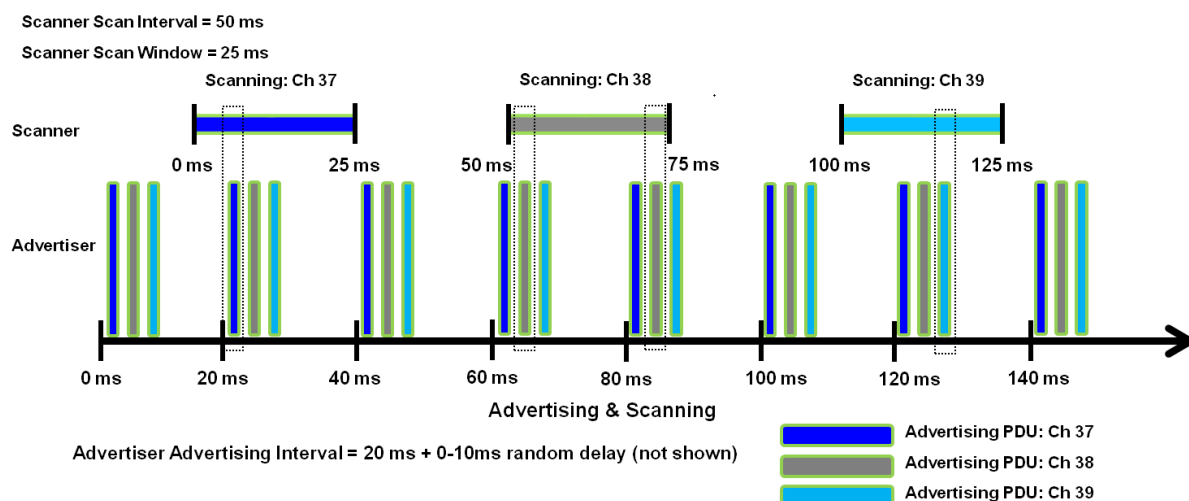


17. ábra: BLE protokoll szerint megkülönböztetett rétegek [17]

A továbbiakban csak a BLE szenzor szempontjából releváns rétegeket ismertetem.

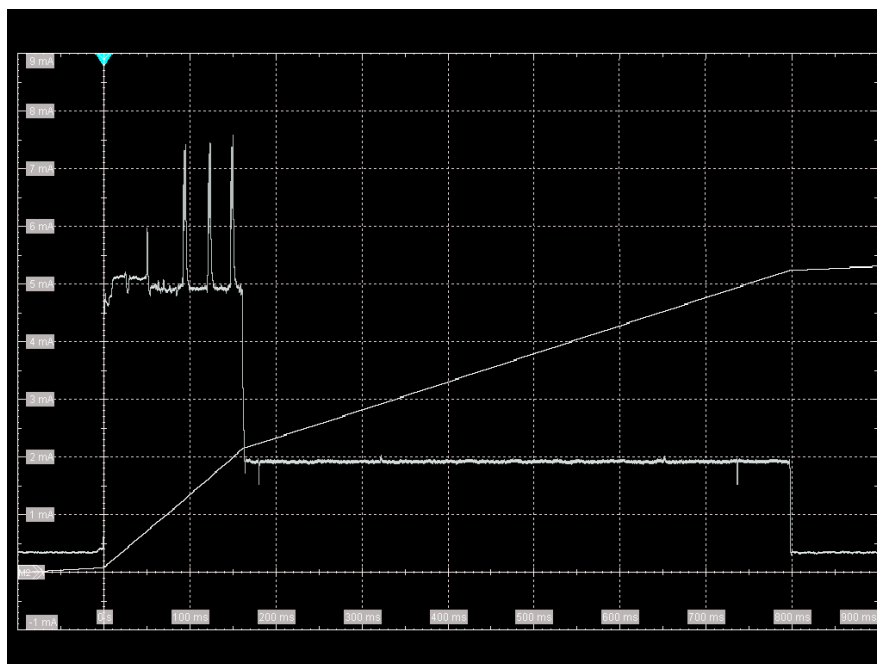
8.1. BLE protokoll fizikai réteg

Az OSI modell szerinti legalsó, „fizikai” réteg az ISM sáv egy Bluetooth kommunikáció céljára igénybe vehető 2402MHz – 2480MHz közti része, melyen osztozkodni kell más kommunikációs (WiFi – 802.11.b-g-n) és egyéb eszközökkel (ami például egy mikrosütő is lehet). A BLE protokoll a fizikai réteg kevésbé „telített” részein valósul meg az advertising csomagok küldése. Az említett hirdetőcsomagok a következő ábra szerint jelennek meg. [18]



18. ábra: Hirdetési időintervallumok [18]

A hirdetési csomagok száma (a hirdetési idő) az alábbiakban hivatkozott felhasználói programban GAP paranccsal paraméterezhető, továbbá a „scanner” eszköz scannelési ideje is paraméterezhető, mely beállításokkal „testre szabható” az adott környezetre jellemző rádiós zavarokra leginkább érzéketlen BLE hirdetés. A Bluetooth rádió kimenő teljesítménye és a hirdető csomagok hossza szintén állítható, mely paramétereket a kívánt szempontok (zavarok elleni védetség, akkumulátor kapacitás, továbbítani kívánt adatok mérete) figyelembevételével lehet meghatározni. A 19.ábrán látható a Pictail modul csomaghirdetés közben mért áramfelvétele. Ezen egyértelműen látható a hirdetőcsomagok közvetítésének időpontja.



19. ábra: Hirdetés közben mért áramfelvétel

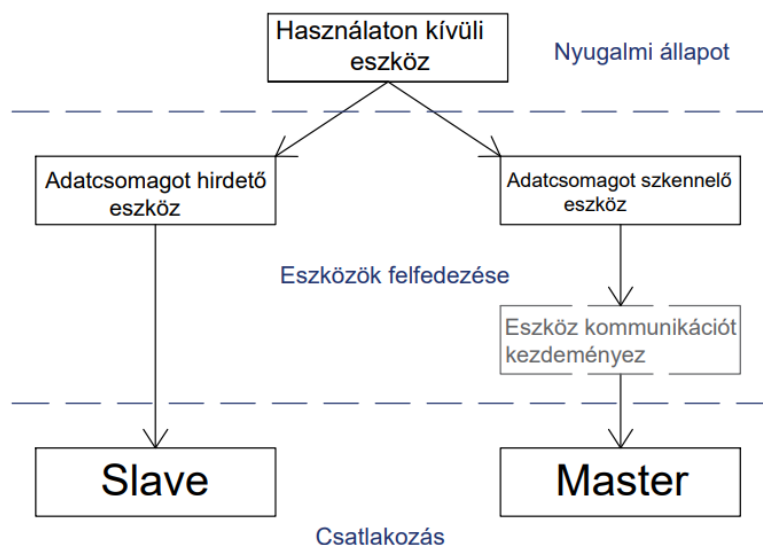
A hirdetőcsomagok elkülönülése mellett a protokoll másik tulajdonsága, hogy a BLE eszközök „connected” állapotában (a később ismertetett transparent UART üzemmódban) a fizikai rétegen a rádiós csomagok frekvenciában eltolva (frequency hopping) jelennek meg, ezzel csökkentve más BLE eszközöktől érkező csomagokkal történő ütközések valószínűségét.

8.2. GAP (Általános elérési profil) jellemzése

A GAP, azaz Általános elérési profil a Bluetooth Low Energy protokoll egy rétegét képviseli, ami a kommunikáció során felvett és felvehető szerepeket definiálja. A Bluetooth Low Energy protokoll stack-en belül a profil az L2CAP (Logical Link Control and Adaptation Layer Protocol) és a HCI (Host Controller Interface) réteg felett helyezkedik el. A kommunikációs szerepek rugalmasabb megválasztását teszi lehetővé, a résztvevőknek nem kötelező egymáshoz csatlakozniuk. A két eszköz csatlakozása esetén master és slave eszközök definiálhatók, ahol a szerepeket a kommunikáció kezdeményezése alapján rendelhetjük a részt vevő eszközökhöz. Ebben az esetben két irányú a kommunikációs folyamat, a slave egy master eszközhöz, a master több slave eszközhöz csatlakozhat. Ennek lehetőségeit a Bluetooth Low Energy 4.1 dual-mode topológiája kibővíti. A másik lehetséges konfigurálási módban hirdető (advertiser vagy

broadcaster) és a hirdetést megfigyelő (observer vagy scanner) szerepeket határozhatunk meg, ahol a kommunikáció szintén két irányban valósítható meg. [17]

A következő ábrán látható a megvalósított csapásszámmérő rendszer GAP profilnak egy lehetséges megfeleltetése alapján megrajzolható állapotdiagramja. Az eszköz nyugalmi állapotból indul, ami ebben az esetben a szenzor számára az alvó üzemmód. A kommunikációt először a szenzor oldaláról vizsgálva az adatcsomag összeállítása után azt egyből hirdetni kezdi. Ebben az alkalmazásban a szkennelő eszköz „szkennelési kérése” nélkül, publikus üzenetként szerepel az adatcsomag. Ez megengedhető, hiszen a rendszer által megvalósított funkciók nem teszik azt biztonságkritikussá. A kommunikáció során betöltött szerepek elnevezése esetében az ábrán szereplő „slave” eszköz hirdetőként (advertiser vagy broadcaster), a „master” eszköz megfigyelőként (scanner vagy observer) van jelen. A csapásszámmérő rendszer esetében a mérést végző szenzor hirdetőként, az adatok fogadását végző telefon megfigyelőként értelmezhető.



20. ábra: GAP protokoll szerint betöltött szerepek állapotdiagramja⁵

A jelenlegi alkalmazásban az említettek szerint csak publikus adatkeretről van szó, azonban ez a két eszköz között zárt kapcsolatként is kalibrálható. A leírt kapcsolat

⁵ Az ábra a jelenlegi rendszerre [12] forrás alapján lett vonatkoztatva az elképzelt rendszerre

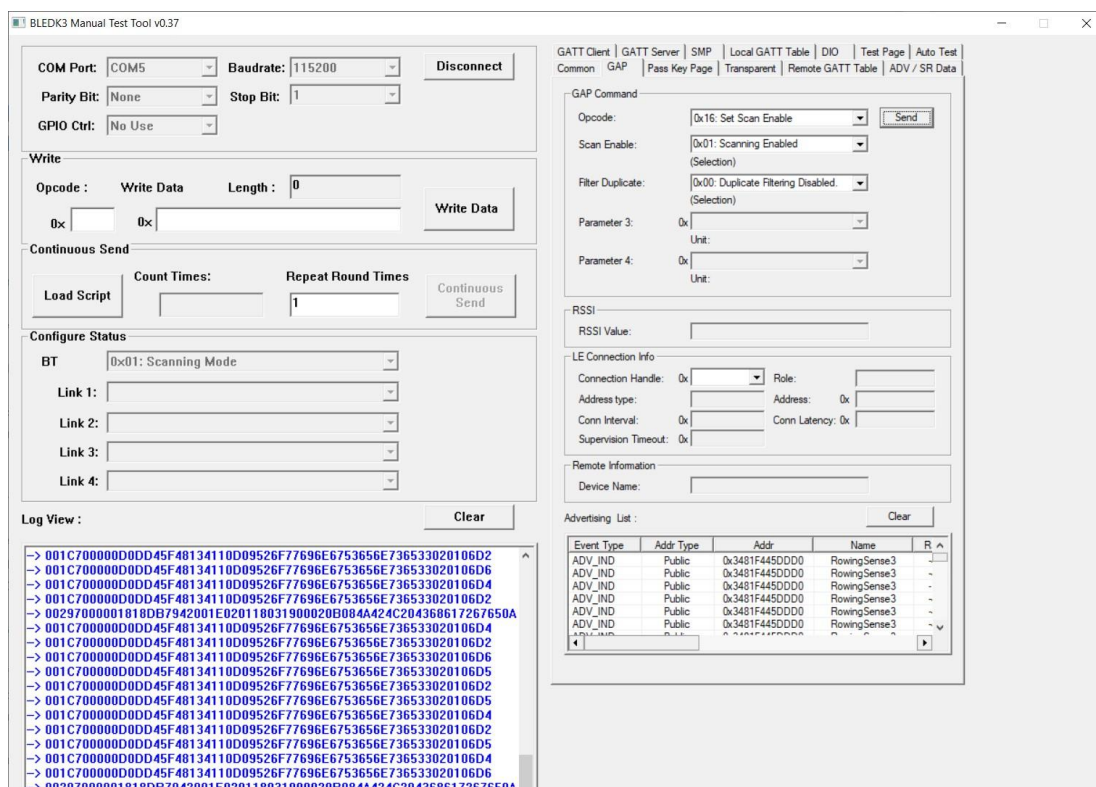
létesítéséhez már szükség van a master eszköz által való (a slave felől kommunikációt kezdeményező adatkeretre válaszolva) szkennelési kérés kiadására. [12]

8.3. GATT (Általános Attribútum Profil) jellemzése

A GATT (Generic Attribute Profile) a GAP profilhoz hasonlóan a kommunikáció résztvevőinek egymáshoz való viszonyát és a hozzájuk rendelt szerepeket írja le. Ez már tartalmazza az eszközök által használható, standard profilokat. A 17. ábrán is szemléltetve, a Bluetooth stacken belül az alkalmazási réteg alatt helyezkedik el. A GATT réteget az Attribute Protocol (ATT) támogatja, ami az eszközök közötti folyamatos kommunikációért felel. A kommunikáció résztvevői a GATT rétegben szervertként vagy kliensként definiálhatók, amik a következő feladatokat látják el: a szerver eszköz tartalmazza a jellemzőket tartalmazó adatbázist, a kliens a szerver által tartalmazott adatbázist írja vagy olvassa. A csapásszámmérő rendszeren belül a szenzor szervertként, az adatokat olvasó eszköz kliensként értelmezhető. A profilon belül az adott információcsomagok különböző tulajdonságokkal rendelkeznek, amik az értékek mellett kerülnek tárolásra. Ilyen tulajdonságra példa az UUID (Universal Unique Identifier), vagy akár az adatcsomagra vonatkozó engedélyek. [19] [17] Az említett UUID egy teljesen egyedi, eredetileg 128 bites azonosító, aminek a Bluetooth Low Energy protokoll egy lerövidített, 16 bites változatát is támogatja. [20] Ez az egyedi azonosító véletlen szám generátor használatával is generálható. Erre egy példát a [21] forrás jelöl.

8.4. A kommunikáció tesztelése

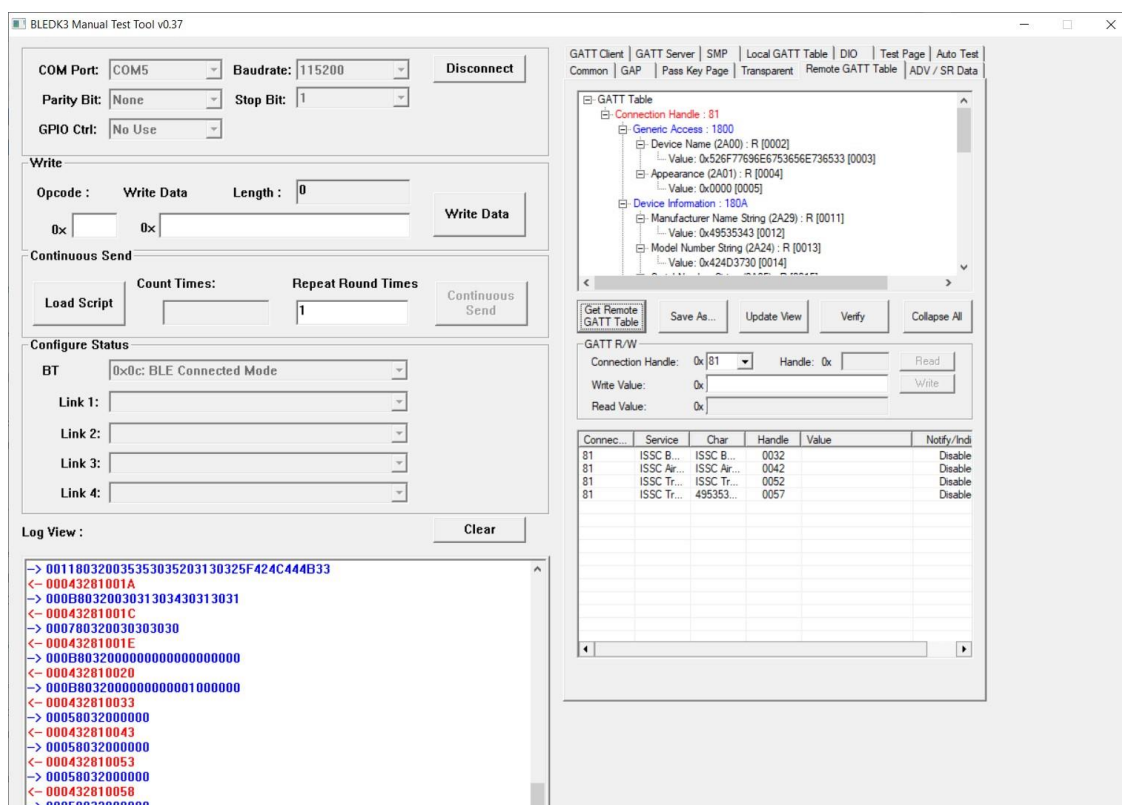
A szenzorral történő BLE kommunikáció és a gyorsulásmérő egységtől érkező adatok tesztelésére a BM70 PICTAIL modult használtam. A BLE modulok paramétereinek beállítására és a GAP parancsok továbbítására, valamint a státuszüzenetek vételére a gyártó által biztosított alkalmazásokat, a „Manual Pattern Test Tool”-t és „IS187x_102_BLEDK3_UI v100.132” programokat használtam. A szoftver segítségével beállított adatokat és parancsokat a csatlakoztatott BLE modulok UART-on kapják. [22] [23]



21. ábra: Szkenelési folyamat beállításai

A megfelelő beállítások után a Bluetooth Low Energy protokollon belüli (a 8.2. és 8.3. alfejezetekben is ismertetett) rétegek szerint megvalósítható a kommunikáció a szenzor és a PICTAIL modul között. A 19. ábrán látható egy, az említett Test Toolról készített kép, ami a GAP rétegre vonatkozó beállításokat mutatja. A 8.2. fejezetben leírtakból vonatkoztatva a rétegre érvényes kommunikációs szerepek a következők: a szenzor hirdető, a BM70-es PICTAIL modul megfigyelő szerepet tölt be. Az ábrán látható a PICTAIL modulnak az adatcsomagok szkennelésére vonatkozó beállítása a jobb felső blokkban. Az ábrán bal oldalon alul láthatóak a „Log View” ablakban a szenzor által hirdetett adatcsomagok kék színnel jelölve. Az említett ablak mellett a jobb oldalon megtalálható „Advertising List”, tehát a hirdető készülékek listája. Ebben a listában az ábrán jelenleg csak egy szenzor által hirdetett csomagok láthatóak, ahol ezeket a táblázat sorai képviselik. Minden eszközhöz a listában az „Addr Type”, azaz cím típusának meghatározása alatt a „Public” szó szerepel, ami publikusan elérhető hirdetést képvisel. [24] [25]

A GAP-hoz hasonlóan a GATT réteg által használt kommunikációs paraméterek is lekérdezhetőek. Ebben az esetben a szenzorhoz való csatlakozás szükségessége mellett a szenzor szerverként, a PICTAIL modul kliensként látja el feladatát. A felület jobb oldalán ily módon megjeleníthetőek a kívánt jellemzők. A megjelenített jellemzők lekérdezése egyesével történik, ami a bal oldali „Log View” ablakból is leolvasható. Itt piros színnel a kimenő, kékkel a bejövő „üzenetek” vannak jelölve.



22. ábra: GATT jellemzők lekérdezése

Ezek alapján megállapítható, hogy a szenzor mindkét konfigurációs módban, a kommunikációs profil használatát közelről megfigyelve megfelelően működik.

8.5. Hirdetett adatcsomag ismertetése

A csapásszámmérő megvalósítás következő lépése a gyorsulásmérő szenzortól érkező adatcsomagok tesztelése. Ebben az esetben az egyszerűség és a könnyű kezelhetőség érdekében csak a GAP protokollnak megfelelően zajlott a kommunikáció. A szenzoron belül, a tesztelési célból összeállított (BLE advert) adatcsomagot a scanner eszköz GAP

státusz üzenet formájában vette és a „Manual Test Tool” alkalmazás hexadecimális kódolással jeleníti meg és szöveges log fájlban is rögzítette. Egy ilyen teszt csomag a következőképpen épül fel:

001C700200419A32F481341101014300000180018D075CFF2A0063FFB0B8

001C: csomag hossza
70: esemény meghatározása – (7x: GAP, 70: hirdetőcsomag - advertising report)
02: esemény típusa – (02: scannable undirected advertising (ADV_SCAN_IND))
00: cím típusa - publikus (Public Device Address)

419A32F48134: BM70 modul MAC cím

11: adathossz
01014300000180018D075CFF2A0063FFB0: adat
B8: RSSI

23. ábra: Adatesomag felépítése

A log fájlban rögzített teszt adatesomagok tartalmaznak GAP protokoll szerinti „kötelező” elemeket: csomag hossz, GAP header adatok, MAC address, és az alkalmazás szempontjából releváns elemeket. A „hasznos” adatok a 3 tengelyes gyorsulásmérőtől kapott 16 bites X-Y-Z koordináták szerinti értékek, az akkumulátor feszültsége, a BM70 modul hőmérséklete, valamint a hirdetési csomag RSSI értéke. A későbbiek során a payload tartalmát (hosszát) az akkumulátoros tápellátás miatt a minimálisra kell korlátozni.

A hirdetési csomagokat a mikrovezérlő állítja össze, a csomagok felépítése kötött, a fentiekén túl a csomagok elején 0xAA, a végén 1 bájt ellenőrző összeg van (host – BM70 modul adatkapcsolati réteg), amely a Test Tool alkalmazásban nem jelenik meg.

9. Felhasználói felület

A kommunikációs paraméterek ellenőrzése után sor kerülhet a felhasználói felület megtervezésére. Ebben a fejezetben az említett feladatot két részre bontottam: első lépésként a végső cél elérése érdekében, ami egy pontos adatfeldolgozást végrehajtó felhasználói felület megtervezése volt, szükséges a mért adatok fluktuációjának jellemzésére.

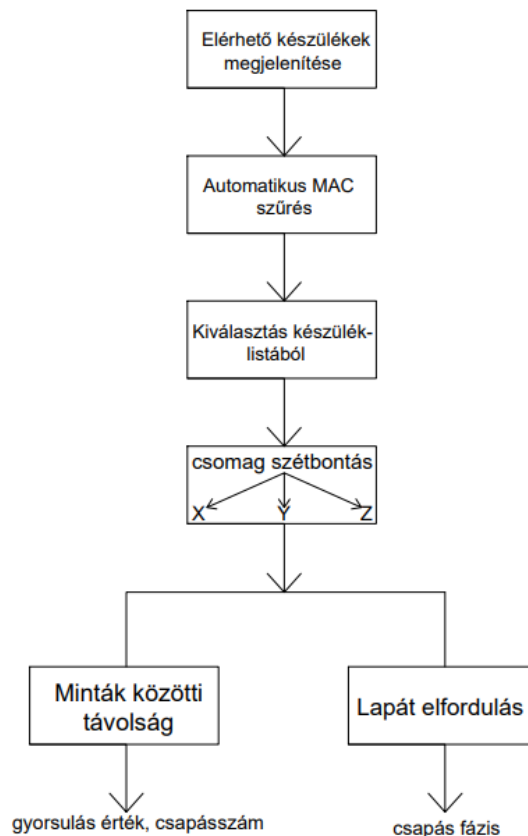
9.1. Mérést szolgáló felhasználói felület

A szenzor által továbbított adatcsomagokat fogadó oldalon a kiértékelés megtervezése előtt fontosnak találtam a fogadott adatok elemzését. Erre a feladatra egy külön „felhasználói felület” megalkotását láttam megfelelőnek, amivel szemben támasztott követelmények lényegesen egyszerűbb megvalósítást tesznek lehetővé: az alkalmazás feladata ebben az esetben az adatcsomagok hiba nélküli fogadása, esetleges felbontása, majd naplózása. Az eddig elképzelt ebben az esetben kiegészülhet egy naplózást megvalósító rendszer-elemmel is, aminek egy lehetséges megvalósítását a következőképpen képzeltem el: a lapátra rögzített szenzor a gyűjtött adatokat az adatmentést végző mobiltelefonnak továbbítja, ami a naplózási funkciót egy adatbázis szerver segítségével valósítja meg. Ennek megvalósításához internethez való csatlakozás biztosítása szükséges, aminek meglétét leegyszerűsíti, hogy a rendszer működésének tervezett tesztelése szárazföldön történik egy szárazföldi evezős gép használatával. Ennek segítségével valódi elmozdulás nélkül, csak a lapátok mozgása rögzíthető. Egy ilyen szárazföldi tesztelést elősegítő gépre példát a [26] forrás mutat.

9.2. Adatok megjelenítését végző felhasználói felület

A felhasználói felülettel szemben támasztott elvárások ebben az esetben főleg az adatok megjelenítése szempontjából kerültek megfogalmazásra. Elsődleges feladat a csapácsszám megjelenítése, emellett az elindulás óta eltelt idő és a megtett távolság. Ennek hatékony felhasználása érdekében a felhasználóval való interakció szempontjából a minimális követelmény Start, Stop és Reset gombok konfigurálása. További interakcióra a szenzor kiválasztása során számíthatunk. A 23. ábrán látható az elképzelt felhasználói felület logikai rendszerterve. Az elérhető készülékek megjelenítése során mivel ebben az

alkalmazásban a függetlenül csomagot hirdető eszközök által történnek a mérések, ezért ilyen kritériummal rendelkező szűrés szükséges. Adott egy időintervallum az eszközök keresésének megkezdése után, amikor az összes ilyen csomagot hirdető szenzor azonosítója naplózásra kerül a rendszer által, amik közül a választás megkönnyítése érdekében a szoftver automatikusan listázza a Bluetooth modul gyártójára specifikus MAC címmel rendelkező, hirdetést végző eszközöket. Ebben az intervallumban a megfelelő szenzor érzékeléséhez szükséges annak megmozgatása, hogy a használni kívánt eszköz mindenképpen a felajánlott opciók között szerepeljen.

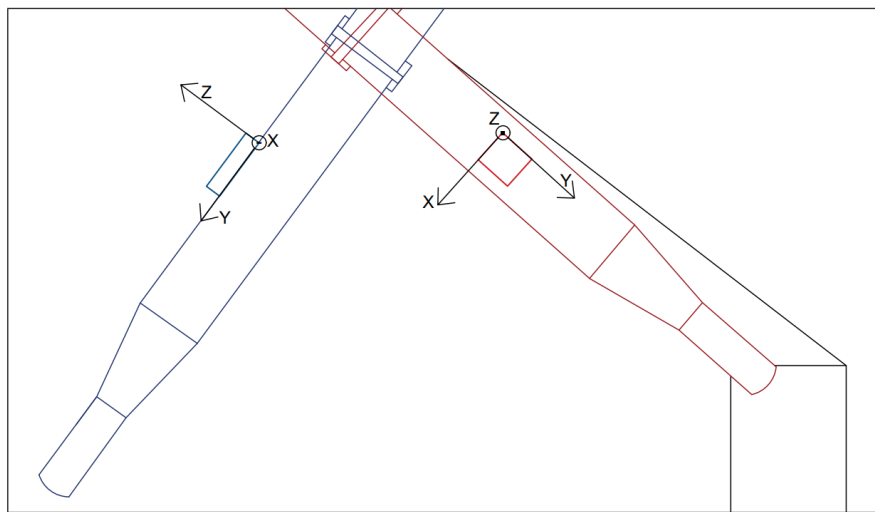


24. ábra: A felhasználói felület logikai rendszerterve

Az ez alapján kiválasztott készüléket a rendszer megjegyzi, és a következő csomag érkezésekor már csak a kiválasztott szenzor azonosítójával rendelkező hirdetőcsomagok tartalmát értelmezi. Ily módon kikerülhető a folyamatos újra-csatlakozás. Ezek után a csomag értelmezése és szeparálása következik, előtérbe helyezve a koordináta-adatokat.

A csomagban közvetített egyéb adatok is kinyerésre kerülnek, mint például a szenzor töltöttsége, amiről ezt követően a felhasználói felület tájékoztatást ad.

A kiszűrt helyzetet meghatározó adatok a szenzorhoz rendelt koordináta-rendszer alapján értelmezendők, amik a szenzort a lapátra helyezve pontosan leírják annak mozgását. Tekintve, hogy a szenzorok a 3.4. alfejezetben is ismertetett elhelyezés szerint vannak a lapátra rögzítve, a szenzor koordináta-rendszere a következő síkokban található:



25. ábra: Rögzített szenzorhoz tartozó koordináta-rendszerek

Elsődlegesen a megvalósítás és a mérési eredmények egyértelmű értékelésének egyszerűsítése miatt a szenzor felhelyezésének iránya is rögzítve van, de a fejlesztési tervek között szerepel ennek felismerésére szolgáló logikai művelet megtervezése. A 23. ábrán kékkel a szenzor indulási állapotát, pirossal a vízfogási állapotot jelöltem. A szenzorhoz rögzített koordináta-rendszerek alapján mért adatok alapján meghatározható a lapát forgatása. A gyorsulás ettől eltérő módon, a szenzor által rögzített minták között eltelt távolság alapján határozható meg. A két módszer alapján kiszámítható jellemzőkből a csapásszám egyértelműen meghatározható, az így kiszámított értékeket a felhasználói felület valós időben kijelzi. A megjelenítő felületként szolgáló alkalmazás modelljét a 3.3 fejezetben található ábra mutatja.

10. Továbbfejlesztési lehetőségek

A továbbfejlesztési ötletek között szerepelnek a rendszer használatának megkönnyítése és a szenzor méretéből adódó előnyök alapján elképzelt javítások. Elsődlegesen edzést és elemzést javító funkcióként a szenzor-mobiltelefon kapcsolatot kihasználva a telefonon bizonyos mértékig naplózási funkció is megvalósítható. A helytakarékoság miatt a Start és Stop gomb megnyomása között eltelt idő és az ez alatt mért mennyiségek átlagértéke és a megtett távolság kerülne csak naplózásra.

Ehhez hasonlóan egy elemzést elősegítő funkció is megvalósítható lenne a szenzor elektronikájának egy LoRa⁶ chippel való bővítésével. A bővítés célja a szenzor által közvetített üzenetek hatótávolságának növelése, mivel ennek használatával akár a partról vagy egy motorcsónakból a felkészülést felügyelő edző számára is láthatóvá válhatnának a mért adatok. A távolságok áthidalásával az adatok így módon egyszerre láthatóak az edző és a sportoló előtt megjelenített felhasználói felületen.

A funkcióbővítések között szerepel még a több szenzor használatával való mozgáselemzés ötlete is. Ezt a lehetőséget véleményem szerint kettes hajóegységben, két szenzor használatával lehetne a legjobban kihasználni. Kettes hajóegység felépítése a következő: a hajóegységben két sportoló egy-egy lapáttal, ellenkező oldalon evez. Ebben az esetben a sportolók lapátjaira szerelt szenzorok által mért adatok utólagosan összehasonlíthatók, ez alapján elemezhetővé válhat a csapás hatékonysága és az egység egyensúlya, amivel javíthatóvá válik a teljesítmény. A felvázolt funkciók megvalósítása és a további lehetőségek feltérképezése terveim között szerepel.

⁶ Vezetéknélküli IoT kommunikáció [27]

11. Összefoglalás

Szakdolgozatom megvalósítása során egy evezős sportban használható csapásszámmérő rendszert készítettem el. Ez az általam eddig használt eszközöktől elképzelés tekintetében és mérési elv szempontjából is eltér. A piacon elérhető csapásszámmérők GPS-es helyzetmeghatározás segítségével, az általam megvalósított rendszer egy gyorsulásmérő szenzoron alapuló helyzetmeghatározást használ. Alkalmazásomban az eddig ismert eszközök bizonytalanságait figyelembevéve határoztam meg a választott szenzor által teljesítendő követelményeket. A szenzor működés alapján való definiálásával együtt elkészült az ez alapján tervezett szoftver. A szoftverben megfelelően kezelésre kerültek a szenzor modulok közti kommunikációs formák. Az ily módon az inercia adatok mérésére és továbbítására képes szenzor az evezős mozgás és a külső, ezt befolyásoló tényezők elemzése után megfelelően elhelyezésre került a hajón belül. Az elhelyezés kiválasztásának eredményeképpen a szenzor adatok értelmezéséhez különböző módszerek kerültek meghatározásra a mérések utáni adatszűrés szempontjából. Ez után az evezős mozgásra jellemző fizikai mennyiségek nagyságának felvázolásával megállapítható volt a mintavételezés sebessége és módja. Az így gyűjtött adatokból összeállított csomag megismerése után a rendszerelemek (szenzor és megjelenítést végző eszköz) között megvalósuló, protokoll alapján felvett kommunikációs szerepek elemzése és az azok alapján levonható következtetések szerepelnek. Ezt az adatokat megjelenítő felhasználói felület logikai működésének bemutatása követi a megállapított adatszűrési feltételeket alkalmazva. Végül, de nem utolsó sorban az eddig megvalósított funkciók mellett a folyamatos fejlesztés elvét igyekeztem a témával kapcsolatban szem előtt tartani a fejlesztési lehetőségek bemutatásával.

12. Summary

During the implementation of my thesis, I developed a stroke rate measurement device usable in rowing. This system differs both in conceptualization and measurement principle from the rowing equipment I have used so far. While the stroke rate monitors available in the market utilize GPS-based position determination, the system I implemented relies on an accelerometer-based position determination. Considering the uncertainty of the known tools, I defined the requirements to be fulfilled by the sensor used in my application. Alongside defining the operation of the sensor, the software was implemented accordingly. The forms of communication were appropriately handled within the software. Consequently, the sensor capable of measuring and transmitting inertial data was placed appropriately inside the boat after analyzing rowing motion and external influencing factors. The selection of placement resulted in defining various methods for interpreting sensor data in terms of post-measurement data-filtering. Following this, by outlining the magnitudes of physical quantities characteristic of the motion of rowing, the speed and method of sampling were determined. After analyzing the package compiled from the collected data, the analysis of communication roles between the elements of the system (sensor and display device) based on protocol and the deductions drawn from them are included. This is followed by presenting the logical operation of the user interface displaying the data while applying the established data filtering conditions. Lastly, besides the functions already implemented, I aimed to keep the principle of continuous development in mind regarding the topic by presenting development possibilities.

13. Ábrajegyzék

1. ábra: Evezős hajó keresztmetszete [1]	7
2. ábra: A sportoló helyzete a hajóban [2]	8
3. ábra: Példa GPS alapú csapásszámmérőre [3]	9
4. ábra: Példa a csapásszámmérő kiegészítését szolgáló impellerre [4]	10
5. ábra: EmPower Oarlock szenzor [5]	10
6. ábra: Az elképzelt csapásszámmérő logikai rendszerterve	12
7. ábra: GPS alapú csapásszámmérő elhelyezése	13
8. ábra: Gyorsulásmérő alapú csapásszámmérő szenzor elhelyezése	14
9. ábra: A szenzor fizikai rendszerterve	15
10. ábra: A szenzoron szereplő gyorsulásmérő modul blokkvázlata [6]	17
11. ábra: Szenzor logikai rendszerterve	21
12. ábra: Lapát indulási állapota	24
13. ábra: A lapát vízfogáskor	24
14. ábra: A lapát helyzete erő kifejtés után	25
15. ábra: Szoftver függőségi kapcsolatai	27
16. ábra: Megszakítások folyamata	28
17. ábra: BLE protokoll szerint megkülönböztetett rétegek [17]	31
18. ábra: Hirdetési időintervallumok [18]	32
19. ábra: Hirdetés közben mért áramfelvétel	33
20. ábra: GAP protokoll szerint betöltött szerepek állapotdiagramja	34
21. ábra: Szkenelési folyamat beállításai	36
22. ábra: GATT jellemzők lekérdezése	37
23. ábra: Adatcsomag felépítése	38
24. ábra: A felhasználói felület logikai rendszerterve	40
25. ábra: Rögzített szenzorhoz tartozó koordináta-rendszerek	41

14. Irodalomjegyzék

- [1] „Sliding Seat Rowing Geometry,” [Online]. Available: <https://angusrowboats.com/pages/rowing-geometry>. [Hozzáférés dátuma: 15 09 2023].
- [2] [Online]. Available: <https://www.concept2.com/service/oars/setting-inboard>. [Hozzáférés dátuma: 17 9 2023].
- [3] „SpeedCoach GPS 2,” [Online]. Available: <https://nksports.com/speedcoach-gps-2>. [Hozzáférés dátuma: 31 10 2023].
- [4] „NK Impeller,” [Online]. Available: https://nksports.com/speedcoach-impeller_1. [Hozzáférés dátuma: 31 10 2023].
- [5] „EmPower Oarlock,” [Online]. Available: <https://nksports.com/empower-oarlock>. [Hozzáférés dátuma: 31 10 2023].
- [6] „ADXL345 Datasheet,” [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/adxl345.pdf>. [Hozzáférés dátuma: 8 10 2022].
- [7] „SpeedCoach Instruction Manual,” [Online]. Available: <https://nksports.com/mwdownloads/download/link/id/201>. [Hozzáférés dátuma: 5 11 2023].
- [8] „Understanding the I2C Bus,” [Online]. Available: https://www.ti.com/lit/an/slva704/slva704.pdf?ts=1699493218862&ref_url=https%253A%252F%252Fwww.google.co.kr%252F. [Hozzáférés dátuma: 5 11 2023].
- [9] „UART: A Hardware Communication Protocol,” [Online]. Available: <https://www.analog.com/media/en/analog-dialogue/volume-54/number-4/uart-a-hardware-communication-protocol.pdf>. [Hozzáférés dátuma: 30 10 2023].
- [10] „FDC1004: Basics of Capacitive Sensing and Applications,” [Online]. Available: <https://www.ti.com/lit/an/snoa927a/snoa927a.pdf?ts=1699508848907>. [Hozzáférés dátuma: 5 11 2023].
- [11] „PIC24FV32KA302,” [Online]. Available: <http://ww1.microchip.com/downloads/en/devicedoc/39995a.pdf>. [Hozzáférés dátuma: 8 10 2022].
- [12] „Generic Access Profile,” [Online]. Available: https://software-dl.ti.com/lprf/simplelink_cc2640r2_latest/docs/blestack/ble_user_guide/html/ble-stack-3.x/gap.html. [Hozzáférés dátuma: 2 11 2023].

- [13] „Bluetooth Low Energy Module,” [Online]. Available: <https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/Product Documents/DataSheets/Bluetooth-Low-Energy-BLE-Module-DS60001372K.pdf>. [Hozzáférés dátuma: 8 10 2022].
- [14] „Universal Asynchronous Receiver/Transmitter (UART),” [Online]. Available: https://www.ti.com/lit/ug/sprugp1/sprugp1.pdf?ts=1699451002110&ref_url=http%253A%252F%252Fwww.google.com%252F. [Hozzáférés dátuma: 30 10 2023].
- [15] „OSI (Open Systems Interconnection) Reference Model,” [Online]. Available: <https://docs.oracle.com/cd/E19504-01/802-5886/intro-45828/index.html>.
- [16] A. S. Tanenbaum, Számítógép-hálózatok.
- [17] C. C. Hungary, „Mindent a Bluetooth Low Energy technológiáról,” [Online]. Available: https://cdn2.hubspot.net/hubfs/281197/images/Mailings/CControls_HU/Library/Mindent_A_Bluetooth_Low_Energy_Technologiarol_eBook.pdf. [Hozzáférés dátuma: 10 11 2023].
- [18] „Resources for Bluetooth Low Energy,” [Online]. Available: <https://junqing-zhang.github.io/posts/2020/11/blog-post-bluetooth/>. [Hozzáférés dátuma: 06 12 2023].
- [19] „Generic Attribute Profile,” [Online]. Available: https://software-dl.ti.com/lprf/simplelink_cc2640r2_sdk/1.35.00.33/exports/docs/ble5stack/ble_user_guide/html/ble-stack/gatt.html. [Hozzáférés dátuma: 15 11 2023].
- [20] „Universally Unique Identifier (UUID),” Microchip, [Online]. Available: <https://microchipdeveloper.com/xwiki/bin/view/applications/ble/introduction/bluetooth-architecture/bluetooth-host-layer/bluetooth-generic-attribute-profile-gatt/UUIDs/>. [Hozzáférés dátuma: 02 12 2023].
- [21] „Online UUID Generator,” [Online]. Available: <https://www.uuidgenerator.net/>. [Hozzáférés dátuma: 02 12 2023].
- [22] „BM70 Operating Modes,” Microchip, [Online]. Available: <https://microchipdeveloper.com/xwiki/bin/view/products/wireless-connectivity/ble/bm70/operating-modes/>. [Hozzáférés dátuma: 20 11 2023].
- [23] „BM70/71 Bluetooth Low Energy Module User's Guide,” [Online]. Available: <https://ww1.microchip.com/downloads/en/DeviceDoc/50002542A.pdf>. [Hozzáférés dátuma: 20 11 2023].

- [24] „BM70 GATT Server Demo,” Microchip, [Online]. Available: <https://microchipdeveloper.com/xwiki/bin/view/products/wireless-connectivity/ble/bm70/application-examples/gatt-server-demo/>. [Hozzáférés dátuma: 5 10 2023].
- [25] „BM70 Application Example,” [Online]. Available: <https://microchipdeveloper.com/xwiki/bin/view/products/wireless-connectivity/ble/bm70/application-examples/connecting-to-a-peer-bm70/>. [Hozzáférés dátuma: 20 11 2023].
- [26] „Biorower,” [Online]. Available: <https://biorower.com/>. [Hozzáférés dátuma: 05 12 2023].
- [27] „LoRa,” [Online]. Available: <https://lora.readthedocs.io/en/latest/>. [Hozzáférés dátuma: 7 11 2023].