

**Ipari felhasználásra tervezett, gyorsulásmérő, távolságmérő
és optikai szenzorokkal felszerelt, üzemcsarnokban telepített
ipari daruk folyamatos darupálya ellenőrzésre szolgáló
monitoring rendszer – DaMoR**

<Fejlesztési dokumentum>

GINOP-2.1.7-15-2016-02352

This is IT Kft.

<Fejlesztési dokumentum>

GINOP-2.1.7-15 – Prototípus, termék-, technológia- és szolgáltatásfejlesztés

Ipari felhasználásra tervezett, gyorsulásmérő, távolságmérő és optikai szenzorokkal felszerelt, üzemcsarnokokban telepített ipari daruk folyamatos darupálya ellenőrzésre szolgáló monitoring rendszer – DaMoR

Kapcsolódó pályázat azonosítója: **GINOP-2.1.7-15-2016-02352**

Kedvezményezett neve: **This is IT Kft.**

Dátum: **2021.02.26.**

Tartalom

1.	Projekt bemutatása	4
2.	Daru működésének, szerkezetének ismertetése, darupályavizsgálat típusai, konkurenciaelemzés	10
2.1.	Híddaru elemei	10
2.1.1.	A futódaru változatai és elemei:	10
2.1.2.	A futódaru főbb gépészeti egységei:	12
2.1.3.	A bakdaru változatai és elemei:	13
2.1.4.	A bakdaru főbb gépészeti egységei:	15
2.2.	Darupályavizsgálatok	17
2.2.1.	Hagyományos geodéziai felmérés	18
2.2.1.1.	Síntengely vízszintes kígyozásának mérése, párhuzamossági vizsgálat	19
2.2.1.2.	A sínszálak magassági kígyozásának mérése	20
2.2.2.	Modern geodéziai felmérés	21
2.2.2.1.	Lézerszkenner alapú felmérés	21
2.2.2.2.	Robotmérőállomás alapú felmérés	22
2.3.	Konkurenciaelemzés, nemzetközi kitekintés	24
2.3.1.	TruConnect – Daru-távfelülgylet	24
2.3.2.	RailQ – Darupálya-vizsgálat	24
2.3.3.	Összehasonlítás	24
3.	Prototípus - Hardver	25
3.1.	Koncepció, rendszerterv	25
3.2.	Kritériumok, specifikációk	26
3.3.	Szenzorok	26
3.3.1.	Távmérő szenzorok	26
3.3.1.1.	Ultrahang alapú méréstechnika	26
3.3.1.2.	Távolság mérése elektronikus módszerekkel	28
3.3.1.3.	Távolságmérés Bluetooth segítségével	29
3.3.1.4.	ToF (Time of flight) módszer	30

3.3.1.5. LIDAR	32
3.3.2. Gyorsulásmérő szenzorok	35
3.3.2.1. Kapacitív gyorsulásmérő	40
3.3.2.2. Piezzo elektromos gyorsulásmérők	53
3.3.2.3. Gyorsulásmérők nyúlásmérő bélyegek alkalmazásával	56
3.3.2.4. Termodinamikai gyorsulásmérők	58
3.4. Feldolgozó egységek	62
3.5. Egyéb hardverelemek	74
3.5.1. Wifi ESP32	74
3.5.2. Hűtés	74
3.5.3. Tápellátás - PKCELL ICR18650 6600mAh 3.7V	75
3.5.4. Prototípus hardverdoboz elkészítése	76
3.5.5. Hálózati, vezérlő és adattároló eszközök	90
3.5.5.1. Vezeték nélküli hálózat kiépítése	90
3.5.5.2. Hálózati tároló és merevlemezek	91
3.5.5.3. Feldolgozó és vezérlő mini PC	93
3.5.5.4. Vezérlő tablet	94
3.6. Hardver elemek összeépítése	95
4. Prototípus - Szoftver	96
4.1. Szenzorfúziót megvalósító szoftver	96
4.1.1. Tervezés	96
4.2. Felügyeleti szoftver interface	97
4.2.1. Felhasználói felület kód:	97
4.2.2. Felhasználói felület:	103
4.3. Szenzorfúziós elemek firmware tervezése	108
4.3.1. Kód	111
4.3.1.1. Main code:	111
4.3.1.2. Server code:	113
4.3.1.3. Wifi code:	124
4.3.1.4. AX code:	131
4.3.1.5. RF Hal code:	144

4.3.1.6. Filterezés	159
5. Tesztkörnyezet kiválasztása, tesztelés, jegyzőkönyvek	160
5.1. Tesztkörnyezet	160
5.2. Referenciamérés	160
5.2.1. Teszt1	170
5.2.2. Teszt2	176
6. Prototípus - Telepítési és használati útmutató	183

1. Projekt bemutatása

Világszerte és Magyarországon is számos gyárban és üzemben használnak a nagy terhek emeléséhez, mozgatásához ún. futómacskás futódarut. A daru hídszerkezete általában a földre támaszkodó oszlopokon elhelyezett, a csarnok megerősített falaira fektetett vagy a födémszerkezethez erősített síneken mozog, annak hossztengelye mentén. A hídszerkezeten fut (az előbbi mozgásra merőlegesen) az úgynevezett futómacska. Ez lényegében egy sínen mozgó kocsi, melyre a kötéldobot, az azt működtető villanymotort és fogaskerekes hajtóművet szerelnek. A hídszerkezet és a futómacska segítségével a függő teher hossz- és keresztirányban is mozgatható, amely emelési magassága az emelőmű segítségével szabályozható. A futódaru működése és hosszú távú, üzembiztos használhatósága érdekében fontos annak elemei megfelelő illeszkedése, geometriai jellemzőik és egymáshoz való elhelyezkedésük adott tűrésen belül kell legyenek. A futódaruk, azok elemeinek állapotának vizsgálata állapotának időszakonként, rendszeresen vagy alkalmanként történik meg. A darupálya geometrijának az ideálistól, a megkívánttól való eltérése az extrém torzulásokat kivéve csak precíziós vizsgálatokkal mutatható ki. Ugyanakkor a folyamatos monitorozás a lehető leghamarabb képes jelezni azon kockázatokat, amelyek a darupálya vagy a daruhíd valamely elemének akár már fennálló, akár a használat során előálló, elhasználódásból, nem megfelelő vezérlésből, túlzott teher emeléséből, mechanikai behatásából stb. származnak. A darupálya elemei geometrijának és egymáshoz képesti elhelyezkedésének ideálistól való eltérése számos forrásból fakadhat. Működési problémát eredményezhet a futósín bármilyen elmozdulása, egyenetlensége, hajlása, csavarodása, irányultságának eltérése az egyenestől, a felszínének folytonossági hibája, hullámzása, az

elemeinek tökéletlen egymáshoz illesztése, hiányzó rögzítőelem vagy kopott támasztótalp. minden eltérés minimum a híd egyenes vonalú mozgásának zavarát, a daru görgőinek, sírjainak és egyéb komponenseinek abnormális, idő előtti kopását, hibáját, költséges kiesést és javításokat eredményezi. Nem utolsó sorban aláássa a biztonságos és hatékony működést.

A prototípus megoldás vizsgálata nemcsak az egyes síneket önállóan, hanem a sínek egymáshoz képesti eltérését is érinti – pl. ha adott pozícióban egymáshoz képest máshogy helyezkednek el, vagy az állásuk nem megfelelően illeszkedik egymáshoz. Emellett jelzi a forgó alkatrészek (pl. görgők) deformációját, kopását is, de a híd kétoldali meghajtás közötti szinkronizációs eltérést is mutatja. Jelen projektjavaslat célja az üzembiztoság, az üzemzavar megelőzése, a nagyobb meghibásodások megelőzése egy olyan folyamatos üzemű darupálya monitoring prototípus fejlesztése, amely alkalmas a darupálya geometriájának folyamatos figyelésére, az esetleges eltérések regisztrálása, vészelyzet (beállított értékeknél nagyobb eltérés) esetén vészleállításra és azonnali értesítésre. A prototípus újdonságtartalma a folyamatos működésben, az extrém értékekfellépések kor küldött azonnali riasztásban, az „éles” használat közbeni monitorozásban és a megoldáshoz felhasznált érzékelők (gyorsulásmérők) e célra való alkalmazásában van. Fontos kiemelni, hogy olyan prototípust kívánunk készíteni, amely alkalmas vagyalkalmassá tehető minden szélsőséges környezeti viszonyok között működni, amelyek között a futódaruk alkalmazásra kerülhetnek (pl. hűtőkamrában vagy öntőüzemben való használat), valamint hogy nemcsak új telepítésű daruknál alkalmazható, hanem bármely már használatban lévőn is telepíthető utólag. A fejlesztésre kerülő megoldás komplett monitorozási rendszert képez: nemcsak azérzékelők kerülnek elhelyezésre és a jelük összegyűjtésre, hanem egy központi feldolgozó egység folyamatosan fogadja és értékeli is azokat, és a határértékek túllépések kor riasztást küld a megfelelő email címre, vagy akár a daru leállítását is kiválthatja.

Prototípus fejlesztés előkészítése, kutatások és elemzések megvalósítása Az előkészítés keretében felmérjük, kutatjuk és elemezzük azon építőelemeket, amelyek a kísérleti fejlesztés megvalósításához szükségesek. Ennek során vizsgáljuk az különböző gyorsulásmérő egységeket (IMU – Inertial Measurement Unit, VRU – Vertical Reference Unit, AHRS – Attitude and Heading Reference System), az ezek jelének folyamatos vételére, tárolására és továbbítására alkalmas egységekkel együtt. A vizsgálat kitér arra is, hogy a fejlesztés eredményének olyannak kell lennie, amely az üzemcsarnokok speciális jellemzőit és adottságait – pl. működési hőmérséklet tartomány és egyéb környezeti paraméterek, adattovábbítás használható technikája, energiaellátás megoldása – kezelní

tudja. A gyorsulásmérő egységeken túl az anomális helyének meghatározásához az is megoldandó, hogy a daruhíd pontos helyét, a teljes pályán a kezdeti vagy végálláshoz képesti pozícióját ismerjük. Ehhez, erre a cérla való alkalmazásra lézeres vagy ultrahangos távmérőket esetleg odometriás eszközökkel vizsgálunk, amelyek eredménye szintén folyamatosan kiolvasható, szüksége esetén további megoldási módozatokat is figyelembe veszünk. A központi adatelemző megoldáshoz az adatgyűjtő és tároló komponensen túl azt is ki kell választani, hogy a beérkező gyorsulási információk elemzését milyen környezetben célszerű kifejleszteni. A kiindulásként használandó keretrendszernek támogatnia kell mind az egyes szenzorok, mind azok összessége jelének vizsgálatát, valamint a szükséges kimutatások és grafikonok leképezését is. Mindezeket annak érdekében tesszük, hogy a fejlesztéshez azon elemek kerüljenek beszerzésre, amelyek a leginkább szolgálják a kitűzött célok megvalósítását.

A kísérleti fejlesztés során egy egysége építjük a hídszerkezeten elhelyezendő lokális vezérlő egységet, a gyorsulásmérőt, az adattároló és az adattovábbító elemeket. Szintén ide kerül a futósínban való pozíció meghatározásáért felelős távolságmérő, amely számára viszont gondoskodni kell arról – a távmérés technikájából fakadóan –, hogy folyamatosan vizuálisan „rálásson” a futósín végére. Gondoskodunk arról, hogy az összeépített egység az ipari környezetnek megfelelően megfelelő védelmet nyújtó dobozolást és felerősítési lehetőséget kapjon. A fenti egységek a doboz belsejében kerülnek elhelyezésre, közülük csak az kerül mérlegelésre, hogy a kommunikációhoz – amennyiben ilyen módon kerül megvalósításra – megfelelő-e a dobozba integrált antenna használata, vagy azt a dobozon kívülre kell vezetni. Meg kell oldani a dobozban elhelyezett elemek folyamatos energiaellátását, valamint a doboznak a hídgerenda működéséből fakadó mozgásának ellenálló felerősítését is. Megoldandó, hogy a lokális vezérlő egység folyamatosan kiolvassa a gyorsulásmérő által mért értékeket, és azokat az adattárolón elhelyezi. Ugyanezzel szinkronban veszi és tárolja a távolságmérő/ útmérő által mért értékeket. A lokális vezérlő egység gondoskodik arról is, hogy az adatokat a beállított gyakorisággal elküldje a központi feldolgozó egység felé. A lokális egység inicializálásához a prototípus feladatának megfelelően be kell állítani a gyorsulásmérő tengelyeit: az egyiket a futósínnel párhuzamosan, a másikat arra és a sín felületére merőlegesen. A híd két végállásában el kell végezni a távolságmérő vagy útmérő mérési tartományának inicializálását is. A lokális egység vezérlője gondoskodik arról is, hogy az azonos időben mért értékek egymással párosíthatók legyenek. A prototípus kialakításához megfelelő, ha a lokális egység

működésének vezérlése az egységre ideiglenesen csatlakoztatott külső egységen keresztül kerül beállításra. Ehhez a lokális egység dobozán megfelelő csatlakozási lehetőséget kell kiépíteni. A központi egység rendszeres időközönként veszi/lekéri a lokális egységek mérési adatait és azokat elhelyezi a naplóállományban. Gondoskodik arról, hogy minden, lokálisan letárolt érték bekerüljön a naplóba. A lokális értékek a fejlesztésre kerülő program felületén bármikor megtekinthetők, kereshetők, szűrhetők. Műszakonként összesítés készül az eredményekről. A mért értékeket adatelemzésnek is alávetjük, az operátori/karbantartói figyelmet érdemlő értékeket külön meg is jelöli a program. Másrészt kritikus határértéket elérő eredmények esetén, bármikor a napi használat során riasztást küld, mert ez esetben a darunál azonnali beavatkozásra lehet szükség a nagyobb károk megelőzése érdekében.

A prototípus kifejlesztését és a tesztelési környezetben való implementálását követően annak kiterjedt tesztelése szükséges, mind a daru terheletlen, mind a terhelt állapotát vizsgálva. Hangolni kell a lokális egység elemeinek inicializálási folyamatát a tesztelési környezetnek megfelelően. Ki kell alakítani a határértékek és a kritikus határértékek megállapítási, beállítási folyamatát. Tesztelni kell a sínpálya vagy a híd egyes jellemző, potenciális rendellenességeinek a mérésre gyakorolt hatását, hogy lehetőleg már a mért értékekből következtetni lehessen azok okára. A prototípus rendszer működését vizsgálni kell különböző hídsebességeknél. Tartós tesztet is végezni kell a hosszú távú megbízható működés ellenőrzése érdekében, amikor a valós vagy ahhoz közelítő körülmények között vizsgáljuk a megoldás elemeinek önálló, valamint az egymáshoz képesti működését. Tesztelni kell a kommunikációt a lokális és a központi egység között, ki kell alakítani az adatküldési periodicitást.

Prototípus rendszer hardver és szoftver elemeinek beszerzése: A szükséges hardver és szoftver elemek beszerzése a fejlesztés, a prototípus megoldás és a kapcsolódó alkalmazások megvalósítása érdekében. A beszerzéseket követően megtörténik a fizikai fejlesztési környezet és a szoftver környezet (infrastruktúra) kialakítása. Piacra vitelt megalapozó tevékenységek: - Szolgáltatási modell koncepciójának kialakítása, piacon meglévő hasonló szolgáltatások kutatása, elemzése, szellemi oltalom alá helyezhetőségre vonatkozóan jogi elemzés elvégzése.

Kutatásunk szerint jelenleg nincsen olyan önálló, a futódarukra utólag is telepíthető monitorozási megoldás, amely a darupálya, a futósín geometriai rendellenességeit folyamatosan követi. Ennek segítségével egy olyan megoldás adható a daruk üzemeltetői vagy karbantartói kezébe, amely útján a lehető leghamarabb (már a minden nap működés

közben) tudomást szerezhetnek a daru működési problémáiról, és intézkedhetnek a javításról még a költségesebb, hosszabb termeléskiesést is eredményező meghibásodásokat megelőzően. Emellett a futódaru hosszanti pályájának vagy a görgőknek a szabálytalanságai, egyenetlenségei nemcsak idő előtti kopást vagy meghibásodás eredményezhetnek, hanem komoly veszélyt jelenthetnek munkavédelmi szempontból is, amennyiben a szállítás alatt lévő teher ezekből fakadóan kileng vagy a darupálya nem tervezett mértékben berezeg, ezzel olyan erőket ébresztve, amelyekre a darut nem terveztek, így a szerkezet jelentős deformációját vagy a teher leesését kockázatva. A fejlesztés hozzájárul ahhoz, hogy az Emelőgép Biztonsági Szabályzat kiadásáról szóló 47/1999. (VIII. 4.) GM rendeletben foglaltaknak megfelelően az emelőgép üzemeltetője gondoskodhasson az emelőgép (jelen esetben a futódaru) biztonságos állapotának megőrzéséről. Az elemzett, kiértékelt gyorsulásmérési adatok, az abból levonható következtetések részét képezhetik az emelőgép napló/darukönyv bejegyzéseinek. Ezen túl indikálhatják további, specifikus mérőműszeres (az emelőgép mérőeszközökkel nem mérhető geometriai méreteinek és működési helyének pontos geometriai jellemzőire – pl. a darupálya lejtése, szintkülönbség, párhuzamosság, kígyózás – irányuló) vizsgálat elvégzésének, karbantartásnak, javításnak a szükségességét. A gyorsulásmérők és a kapcsolódó adatgyűjtő-elemző elemek önálló, e célra való közvetlen alkalmazása, mint a futódaru pályájának monitorozására utólag is beépíthető megoldás, újdonság ezen a területen. Szellemi hozzáadott értéket jelent a kifejlesztésre kerülő prototípusnál az egyes tipikus rendellenességek – pl. darupálya függőleges egyenetlensége, a sín elcsavarodása, görgő kopása – előfordulása esetén annak megjelenési mintái a gyorsulásmérők által mért értékekben, valamint az ezeken túl tapasztalt kiugró értékek, anomáliák okának felderítése. Szintén hozzáadott értéket jelent a különböző, jellemzően a híd két oldalán elhelyezett gyorsulásmérők által mért értékek összevetése, az ezek eltérési mintáiból való következtetések levonása. Hozzáadott érték még azon módszertan kialakítása is, hogy a megoldás egy-egy implementációja során hogyan kell azt inicializálni, és mi az eljárásrendje a „normális”, a „határérték” és a „kritikus határérték” gyorsulásértékek meghatározásának. Ennek során az is figyelembe vételre kell kerülni, hogy az adott darunak és használatának mik a sajátosságai, mely értékek és gyorsulási minták velejárói a normál működésének, és melyek érdemelnek figyelmet. E) A projektjavaslat önálló technológiai fejlesztést valósít meg, mivel a daru minden nap használatának részévé teszi a darupálya állapotának folyamatos monitorozását. Így az nem csak az időszaki ellenőrzések vagy szervizek alapján kerül sorra, hanem folyamatosan

megtörténik. Fontos kiemelni, hogy a fejlesztés által a pálya geometriájának vizsgálata nem csak szemrevételezésen vagy lokális műszeres mérésen alapul, hanem a pálya egészének állapotát vizsgálja a daru dinamikus működése közben, így az üzemeltető a lehető leghamarabb tudomást szerez a daru pályán való haladásának anomáliáiról, az esetleges nem előírások szerinti használatról.

2. Daru működésének, szerkezetének ismertetése, darupályavizsgálat típusai, konkurenciaelemzés

2.1. Híddaru elemei

A daru olyan emelőgép, ami a terhek emelésére és kisebb távolságra történő szállításra szolgál. Fő alkotóelemei az emelőmű, a mozgatómű és az azokat kiegészítő tartószerkezet.

Ipari környezetben, üzemcsarnokokban vagy a csarnokok mellé telepítve két fő híddarufajtával találkozhatunk:

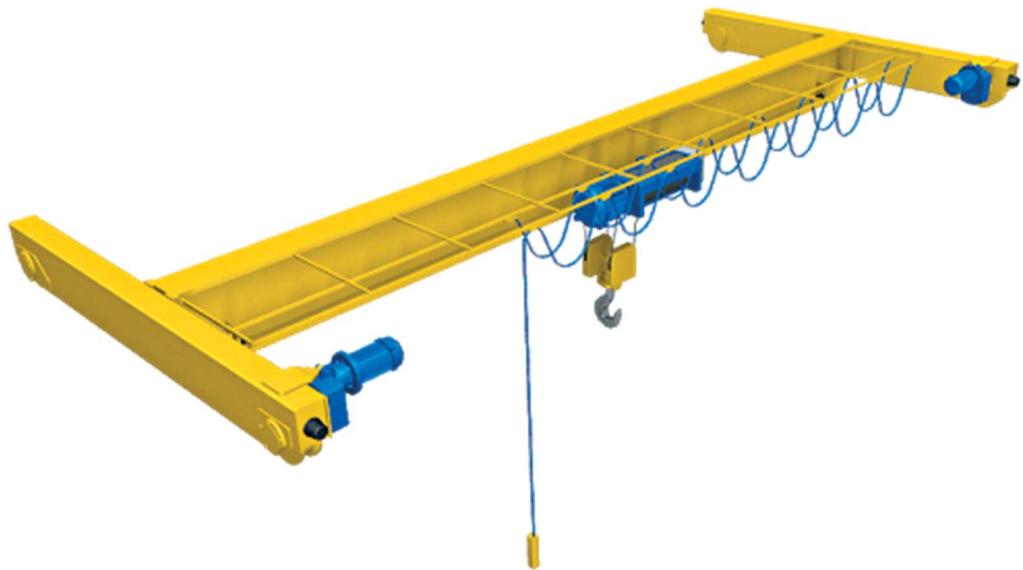
a, Futódaru

b, Bakdaru

Mindkét típus közös sajátossága a vasbeton- vagy acél szerkezetre rögzített sínpályán történő mozgás, valamint a híd szerkezetén szintén sínpáron mozgó, az emelésért és a csarnoktengelyre merőleges mozgásért felelős futómacska.

2.1.1. A futódaru változatai és elemei:

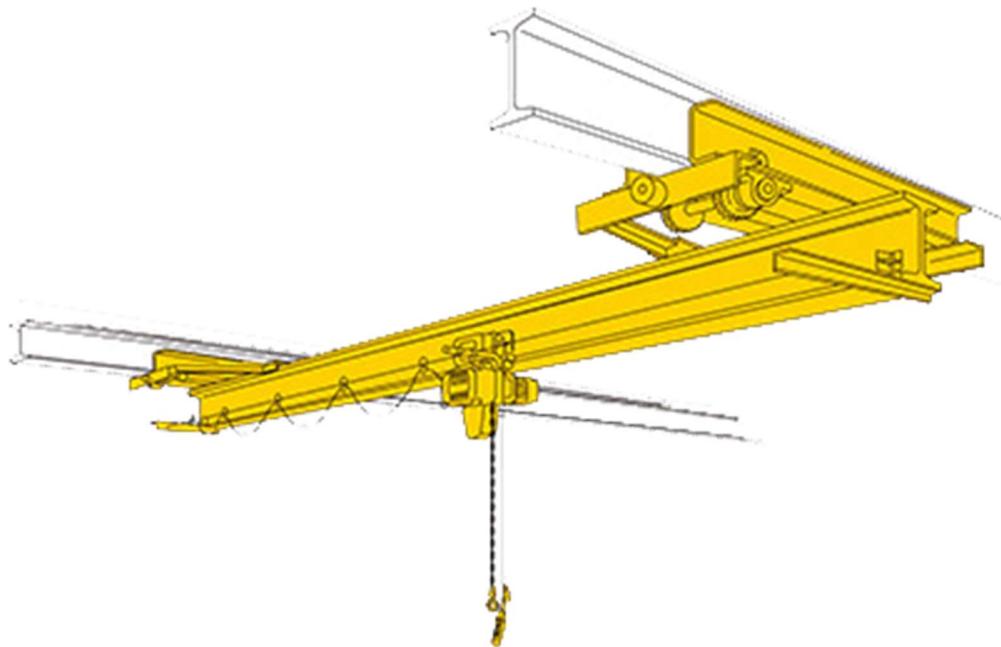
A futódaruk között megkülönböztetünk a főtartók számától és a futómacska elhelyezkedésétől függően **egyfőtartós-** és **kétfőtartós** illetve **alulfutó** és **felülfutó** darukat. Az egyfőtartós darukat jellemzően kisebb fesztávú csarnokokba építik be, A kétfőtartós darukat nagyobb csarnokokba telepítik. Az épület vasbeton vagy acél oszlopain elhelyezett konzolra kerül a daru sínpályája. Egy sínpályán a felhasználói igényektől függően akár több daru is mozoghat, az emelők számát a felhasználók száma határozza meg. A felhasználás intenzitásától függően villamos sodronyköteles emelővel illetve kézi vagy villamos láncos emelővel készülhetnek.



1. ábra Egyfőtartós hiddaru



2. ábra Kétfőtartós hiddaru



3. ábra Alulfutó daru

2.1.2. A futódaru főbb gépészeti egységei:

1. Futómacska, haladómű: A futómacska más néven haladómű a daru főtartóján a csarnoktengelyre merőleges irányba mozogó, csapágyazott tengelyekkel ellátott acélkeret. Az elhelyezkedésétől függően alulfutó vagy felülfutó verziói ismertek, alulfutó esetben a gerendán fügesztve, felülfutó esetben a gerenára ültetett sínpályán mozogva.
2. Emelőmű: A futómacskán elhelyezett, általában villamos motort és az áttételeket tartalmazó szerkezet. Célja a súly vertikális irányú mozgatása.
3. Kötéldob: A futómacskán elhelyezett szerkezet, ami az emelőkötél megvezetését és tárolását látja el. A dob tengelye az emelőmű tengelyével rugalmas, rezgéselnyelő kötéssel csatlakozik.
4. Macska-mozgatómű: a futómacska főtartó irányú mozgását szolgálja. Általában villamos motorral, lánchajtás segítségével a futómacska egyik vagy minden tengelyét hajtja. Macskapályasínnek általában négyszögkeresztmetszetű darusínt használnak.

5. Főtartó: Kiviteltől függően egy I tartó is lehet főtartó, de nem ritkák a nagyobb feszttávot kisebb önsúllyal átívelő rácsostartós szerkezetek sem. Méretezéstől függően készülhet csőből és lehet héjszerkezetű is. Kéttartós darunál lehetséges a két tartó különböző szerkezete, fő- és melléktartó rendszerben történő használata is.

6. Kerékszekrény: A főtartó két végén elhelyezett, a főtartó és a darupálya kapcsolatát megeremtő kerekekkel, tengelyeket magában foglaló szerkezeti elem. Külső felületén a főtengely irányában helyezkedik el a hídütköző. Ez a kemény gumiból készült ütközőfelület az esetleges túlfutásaok csillapítására szolgál.

7. Híd- mozgatómű: A teljes hídszerkezetet a darupályán a csarnok tengelyével megegyező irányban mozgatja villamos motorok segítségével. Az egyfőtartós daruk esetében egy pár (bal- és jobboldali) motor felelős a mozgatásért, a kétfőtartós daruknál ezek duplázva vannak és oldalanként két-két motor hajtja a kerékszerkényben lévő tengelyeket. A motorokat közös kapcsolóval működtetik, ezzel biztosítják az egyidőbe való indítást és leállást. A hídmozgató szerkezet az áramot a darupálya mentén szigetelt alátámasztásokra szerelt csupasz munkavezetékről vagy úszókábelről kapja. Utóbbi megoldást olyan esetekben alkalmazzák, amikor a helyi körülmények állandó elhelyezésű vezeték használatát nem teszik lehetővé.

8. Darupálya -tartó: A darupályapálya alátámasztását biztosító, a csarnok szerkezetéhez erősített (acélszerkezet) vagy a csarnok építésekor monolit szerkezetként (vasbeton) elkészített konzol.

9. Darupálya: A darupálya a daru hosszengely irányú megvezetését biztosítja. Anyaga acél, profilja lehet egyszerű négyzetes, de használhatnak sínprofilt a kisebb kopás és az alacsonyabb gördülési ellenállás érdekében. A darupálya ideális állapotától való függőleges és-vagy vízszintes eltérése veszélyezteti a daru működését, idő előtti elhasználódáshoz vagy nem tervezett leálláshoz vezethet.

2.1.3. A bakdaru változatai és elemei:

A bakdaru a híddaru olyan változata, amely lábakkal támaszkodik a padlószinten elhelyezett darupályára. A bakdarut a futódarutól a darupálya elhelyezése és magassága különbözteti meg. Olyan esetekben használják ezt a megoldást, amikor nincs lehetőség a meglévő csarnokszerkezetre rögzíteni a darupályát, vagy szabad téren egyáltalán nincs csarnok az anyagmozgatás területén.

A bakdarukat a járószerkezetük szerint **kötöttpályás** és **szabadpályás** csoportra oszthatjuk. A kötöttpályás csoportba tartozók előre telepített darupálya sínpáron gördülnek, a szabadpályás csoport darui gumiabroncos vagy szélestalpú acélkerekes futófelületet használnak. Szerkezeti kialakításukat tekintve létezik **egyszerű-** és **konzolos** kivitelű. Az egyszerű kivitel esetén a futómacska csak a két láb (bak) között mozog. A konzolos kivitelnél a futómacska egy vagy minden két oldalon a lábakon túlnyúlva kijárhat. Az egy oldalon lévő lábak száma szerint az egyszerű kivitel általában **egyoszlopos**, a konzolos kivitelű rendszerek általában **kétoszlopos** szerkezeti kialakításúak. A futódaruhoz hasonlóan a híd főtartónak száma alapján egyfőtartós és kétfőtartós bakdaru lehetséges.



4. ábra Egyfőtartós bakdaru



5. ábra Konzolos bakdaru

2.1.4. A bakdaru főbb gépészeti egységei:

1. Futómacska, haladómű: A futómacska más néven haladómű a daru főtartóján a csarnoktengelyre merőleges irányba mozogó, csapágyazott tengelyekkel ellátott acélkeret. Az elhelyezkedésétől függően alulfutó vagy felülfutó verziói ismertek, alulfutó esetben a gerendán függesztve, felülfutó esetben a gerenára ültetett sínpályán mozogva. Konzolos kivitel esetén a futómacska képes a lábakon túlnyúlvá is mozogni.
2. Emelőmű: A futómacskán elhelyezett, általában villamos motort és az áttételeket tartalmazó szerkezet. Célja a súly vertikális irányú mozgatása.
3. Kötéldob: A futómacskán elhelyezett szerkezet, ami az emelőkötél megvezetését és tárolását látja el. A dob tengelye az emelőmű tengelyével rugalmas, rezgéselnyelő kötéssel csatlakozik.
4. Macska-mozgatómű: a futómacska főtartó irányú mozgását szolgálja. Általában villamos motorral, lánchajtás segítségével a futómacska egyik vagy minden két tengelyét hajtja. Macskapályasínnek általában négyszögkeresztmetszetű darusínt használnak.

5. Főtartó: Kiviteltől függően egy I tartó is lehet főtartó, de nem ritkák a nagyobb feszttávot kisebb önsúllyal átívelő rácstartós szerkezetek sem. Méretezéstől függően készülhet csőből és lehet héjszerkezetű is. Kéttartós darunál lehetséges a két tartó különböző szerkezete, fő- és melléktartó rendszerben történő használata is.

6. Lábak: A lábak a főtartó(k)hoz csatlakozva támasztják alá a híd szerkezetét. A lábak talppontjánál található a kapcsolat a daruszerkezet és a darupálya között. A lábak lehetnek vízszintes rácsozott - négyzetkérészetűek, rácosszerkezetűek – változó keresztmetszettel, csőből készült szerkezetek, héjszerkezetű úgynevezett szekrénytartós oszloppal szerelt is.

7. Kerékszekrény: A főtartó két végén elhelyezett, a főtartó és a darupálya kapcsolatát megteremtő kerekekkel, tengelyekkel magában foglaló szerkezeti elem. Külső felületén a főtengely irányában helyezkedik el a hídütköző. Ez a kemény gumiból készült ütközőfelület az esetleges túlfutásaok csillapítására szolgál.

8. Híd- mozgatómű: A teljes hídszerkezetet és lábakat a darupályán a hossztengellyel megegyező irányban mozgatja villamos motorok segítségével. Az egyfőtartós daruk esetében egy pár (bal- és jobboldali) motor felelős a mozgatásért, a kétfőtartós daruknál ezek duplázva vannak és oldalanként két-két motor hajtaja a kerékszerkényben lévő tengelyeket. A motorokat közös kapcsolóval működtetik, ezzel biztosítják az egyidőbe való indítást és leállást. A hídmozgató szerkezet az áramot általában úszókábelről kapja, hiszen legtöbbször a helyi körülmények állandó elhelyezésű vezeték használatát nem teszik lehetővé.

9. Darupálya -tartó: A darupályapálya alátámasztását biztosító vasbeton alapozás. Csak kötöttpályás bakdaruknál található meg.

10. Darupálya: Anyaga acél, profilja lehet egyszerű négyzetes, de használhatnak sínprofilt a kisebb kopás és az alacsonyabb gördülési ellenállás érdekében. A darupálya ideális állapotától való függőleges és-vagy vízszintes eltérése veszélyezteti a daru működését, idő előtti elhasználódáshoz vagy nem tervezett leálláshoz vezethet.

2.2. Darupályavizsgálatok

Erők hatására a testek elmozdulnak, vagy ha nem tudnak elmozdulni, akkor alakváltozást szenvednek. **Elmozdulásról** akkor beszélünk, amikor a test két tetszőleges pontja változatlan marad. Eltolódás alatt értjük, ha a test minden pontja azonos mértékben és ugyanabban az irányban mozdul el. Elfordulás esetén a pontok elmozdulásának nagysága különböző. minden elmozdulás hat összetevő (három egymásra merőleges tengely mentén történő eltolódás és három az előbb leírt tengelyek menti elfordulás) eredőjeként írható le. Ha van a testnek két tetszőleges pontja, ami között a távolság megváltozik akkor **alakváltozásról** beszélünk. Az elmozdulás és az alakváltozás a műszaki gyakorlatban együtt lép fel, de külön kell őket kezelni. Az elmozdulás és a deformáció az önsúly, a hasznos teher és külső járulékos terhek együttes hatására következik be. Meg kell különböztetnünk statikus és dinamikus terheket. Akár elmozdulást, akár alakváltozást vizsgálunk, mindenéppen pontok térbeli helyzetét kell különböző időpillanatokban meghatározni.

A hagyományos geodéziai módszerrel elvégezhető darupálya mozgásvizsgálatokat a következő felsorolás jellemzi:

1. A vizsgálni kívánt darupályát annak néhány kijelölt pontjával azonosítjuk. A pontok számát úgy kell megválasztani, hogy az egy adott mérési idő alatt elvégezhető legyen. A szerkezetet ezalatt az idő alatt mozdulatlannak kell, hogy tekintsük. Hagyományos módszerek mellett nem lehet végtelen számú pontunk, ezért azokat a várható elmozdulás vagy alakváltozás szempontjából mértékadó helyen kell rögzíteni.
2. A vizsgálatkor használt módszert úgy kell megválasztani, hogy a vizsgálati pontok elmozdulása megfelelő pontossággal kimutatható legyen. Ha ismert a várható elmozdulás iranya, akkor a pontok helyzetét ennek az iránynak megfelelően kell pontosabban meghatározni.
3. A vizsgálni kívánt pontjainak helyzetét egy rögzített koordináta-rendszerhez viszonyítva kell megmérni. A viszonyítási rendszer pontjait mozdulatlannak kell tekinteni, és az állandósításukat is úgy kell elvégezni, hogy a legkisebb elmozdulást szenvedjék. Az eredmények kiértékelésekor olyan feldolgozási módszert kell alkalmazni, ami a viszonyítási pontok egy részének esetleges elmozdulását is kimutatja. A viszonyítási pontok számának többnek kell lenni, mint a tájékozáshoz szükséges pontok számának, ezzel biztosítva a fölös mérést a kiegyenlítéshez.

4. A vizsgálati pontok elmozdulását vizsgálhatjuk egy ideális állapothoz (darupálya tökéletes függőleges és vízszintes helyzete) vagy összevethetjük egy időben máskor készült méréssel és az ahhoz képesti eltérést vizsgáljuk. Így két időpont közötti elmozdulást és az idealizált állapottól való elmozdulást is szemléltetni tudjuk.
5. Fontos az eredmények helyes értelmezése is. Egy vertikális irányú eltérést okozhat a darupálya alátámasztásának hibája, de lehet egy- a darupálya felszínén a vizsgálati pontál keletkezett kopás is.

2.2.1. Hagyományos geodéziai felmérés

Darupályák zavartalan működését a terv szerinti pontos szerelés teszi lehetővé. Üzembe helyezés után azonban a darupályák a különböző erőhatások következtében - még rendeltetésszerű működtetés mellett is - kisebb-nagyobb mértékű deformációkat szenvednek. Fontos tehát, hogy a szerkezet üzembe helyezésétől fogva rendszeres vizsgálat alatt álljon.

Három esetben történik az ellenőrző mérések végrehajtása:

1. tervezett karbantartás során
2. üzemzavar esetén
3. rekonstrukció alatt.

A darupálya felmérése során két típusú felmérés jellemzi a darupálya tökéletestől való eltérését:

1. Sínszálak vízszintes kígyózása
2. Sínszálak párhuzamossági vizsgálata
3. Sínszálak függőleges kígyózása

A két értékből megfelelő súlyozás mellett egy általános 3D eltérés számítható, ami jó jellemzést ad a pálya állapotáról, a daru használati módjáról.



6. ábra Hagyományos geodéziai felmérés előkészítése

2.2.1.1. Síntengely vízszintes kígyózásának mérése, párhuzamossági vizsgálat

A feladat a síntengelyek egyeneshez viszonyított helyzetének ellenőrzése, a fesztávolság ellenőrzése, a síntengelyek párhuzamosságának kimutatása.

A feladat megoldása az úgynevezett **egyenesre méréssel** történik. Egyenesre méréssel valamely tervezett egyenesen vagy függőleges síkban ellenőrizhetjük a pontok helyzetét. Az egyenesre mérés nem hagyományos teodolittal vagy mérőállomással történik, jellemzően az

erre használt műszeren nincs beosztott kör, távcsökük nagyítása 50-70x szögnagyítású (a mérőállomásé 20-30x), a távcső nem hajtható át a fekvőtengely körül, a fekvőtengely vízszintessé tételeit nagy pontosságú csöves libella (max. 20" állandó) segíti.

A műszer felállításának pontjai (a darupálya méreteitől függően) lehetnek egy-egy önálló egyenes (alapvonali) pontjai, vagy egy szabatosan meghatározott hálózat alappontjai. Az egyenes valamelyik végpontján felállított műszerrel, a másik végpont irányzása történik. Ezután a sín mentén a darupályán az előre meghatározott távolságokon, mérőeszközöt illesztve a síntengelyre, a műszeren leolvasásra kerül az irányonal t_1 távolsága a lécosztások kezdőpontjától. Mérni kell a két sínszál közötti távolságot (T_1), így minden síntengely helyzetét a kitűzött egyenesre vonatkoztathatjuk, így számíthatjuk minden sínszál vízszintes kígyozását. Hosszú, és nagy fesztávolságú darupályák esetén az ellenőrzés sok és nehézkes mérési munkát igényel. Több pontból álló, célszerűen derékszögű alaphálózatot kell kialakítani (lehetőleg a sínszálak közelében). Az ellenőrző mérések előtt ki kell számítani a hálózati pontok helyi rendszerű koordinátáit. Az így meghatározott pontok által alkotott hálózati oldalak lesznek azok az egyenesek, melyekhez képest mérhető az egyes sínszálak vízszintes kígyozásának értéke.

2.2.1.2. A sínszálak magassági kígyozásának mérése

A darusín korona magassági kígyozása szintezéssel határozható meg. Álláspontként választható a daruhíd, ha épületcsarnokon belül vagyunk az épület egy magas pontja, vagy felállítható a szintezőműszer a padlószinten is. Padlószinten végzett mérésnél a magassági kitűzésnél említett segédeszközöket kell használni. Biztosítani kell, hogy a szalag vagy szintezőléc függőlegesen álljon, a kezdővonalak a síncorona élének magasságában legyenek.

Mind a vízszintes, mind a magasságmérés eredményeit alkalmasan választott méretarányú alaprajzban, illetve hossz-szelvényen grafikusan is ábrázoljuk. A kígyozás értékek rendszerint az alaprajz, ill. a hossz-szelvény méretaránytól eltérően, 1:1 méretarányban kerülnek felszerkesztésre.

2.2.2. Modern geodéziai felmérés

A darupálya mérés a futódaruk esetében mindenkorban magasban végzett munkának minősül. Ipari környezetben ez fokozottan veszélyes, ezért új felmérési módszerek kidolgozásával igyekeztek csökkenteni a magasban töltött időt. Két fő irányt különböztetünk meg a modern felmérések közül:

1. Lézerszkenner alapú felmérés
2. Robotmérőállomás alapú felmérés

A két felmérési módszer eltérő, az eredmény pontosság és részletezettség tekintetében nagyon hasonló.

2.2.2.1. Lézerszkenner alapú felmérés

A modern geodézia legújabb eszköze a lézerszkenner. A lézerszkenner olyan automatizált mérőállomás, amin nincs távcső, helyette egy forgó tükrő található. A műszer az állótengely körül körbe fordulva a forgó tükrő segítségével egy távmérő jelét folyamatosan a külvilág felé küldi. A jel segítségével milliméter pontos távolságmérés történik, az állótengely és a tükrő állását rögzítve pedig két szögérték is rendelkezésre áll a távolsághoz, hasonlóan, mint egy hagyományos mérőállomásnál. A különbség az, hogy amíg a mérőállomásnál egy irányzott pont 3D koordinátája kerül rögzítésre, a szkenner esetén ez egy random pont a térben. Cserébe a szkennerek akár 1 millió pontot is képesek rögzíteni másodpercenként, így letapogatva a környezetüket.

A lézerszkennert a darupálya egy pontjára állítva, az a darupálya egy szakaszát térben rögzíti, a csarnok tartószerkezetével együtt. Egy ilyen mérés 3-6 percert vesz igénybe, a mérés távirányítható, ezáltal nem kell fent tartózkodni a mérőcsapatnak. A különböző mérési állományok közös koordináta-rendszerbe történő transzformációját a csarnokban ideiglenesen vagy permanens módon kihelyezett jeltárcsák vagy kapcsológömbök biztosítják.



7. ábra Z+F lézerszkenner - <https://www.zf-laser.com/>

A több összeillesztett mérés eredménye egy pontfelhő, ami több millió 3D pontból áll. Ezeket a pontokat használva visszamodellezhető a darupálya tengelye. Ezek mellett a lézerszkennelés sajátossága, hogy a teljes környezet felmérésre kerül, beleértve a daru és a csarnok teljes szerkezetét. Ezáltal nem csak a darupálya vizsgálható, hanem a darupálya deformált szakaszánál az alátámasztó elemek geometriája is elemezhető, pontos információval szolgálhat például a csavarkötések meglétéről.

A felmérés előnye a gyorsaság és a darupálya mellett annak környezetének felmérése. Hátránya, hogy nagyobb darupálya esetén több állaspontot kell létesíteni.

2.2.2. Robotmérőállomás alapú felmérés

Robotmérőállomásnak az olyan geodéziai eszközöket nevezük, amik motorizált álló- és fekvőtengellyel rendelkeznek, képesek folyamatos távmérésre, ezzel biztosítva a prizma automatikus követését. A prizmakövetés és a mérőállomás távirányíthatósága segítségével nincs szükség arra, hogy a felmérést végző közvetlenül a mérőállomás mellett álljon. A prizmát azonban a hagyományos méréshez hasonlóan végig kell vinni a darupálya mentén. Olyan esetekben, amikor nincs megfelelően kialakított járófelület a pálya mentén, ez veszélyes feladat.

Vannak olyan gyártók, akik távirányítható prizmakocsit használnak a felmérésre, ezzel kiváltva az emberi munkát.



8. ábra Távirányítható prizmakocsi - www.konecranesusa.com

A felmérés eredménye egy térbeli trajektória, ami egy konstans külpontossági értékkal (primzaállandó, prizma külpontosság) jellemzi a darupályát. A párhuzamosság vizsgálatához a megfelelő összelátás mellett elég lehet csupán egyetlen álláspont felállítása.

2.3. Konkurenciaelemzés, nemzetközi kitekintés

2.3.1. TruConnect – Daru-távfelügyelet

A TruConnect egy olyan távfelügyeleti rendszer, ami a daru üzemeltetési és használati adatait rögzíti és továbbítja, ezzel elősegítve a karbantartás és leállások ütemezését. Az adatok segítségével becsülhető az alkatrészek fennmaradó élettartama. A daruhaználatból kímputatható, milyen területeken kell a kezelők továbbképzését biztosítani a munkahelyi biztonság és a hatékonyság növelése érdekében. A riasztási szolgáltatás elektronikus úton, e-mail vagy SMS formájában értesít az olyan rendkívüli eseményekről, mint a túlmelegedés, a túlterhelés vagy a vészleállítás. A rendszer rögzíti a biztosági kockázatot jelentő problémákat, és ezekből jelentést készít. A távfelügyeleti adatközpontban összeállítják a berendezés használatával kapcsolatos üzemi adatokat tartalmazó jelentéseket. Ezek szöveges vagy könnyen értelmezhető grafikus formában is elérhetők. Új vagy régebbi darukba is beépíthető utólagos bővítésként.

2.3.2. RailQ – Darupálya-vizsgálat

Automatizált darupálya sínelemző vizsgálat. Pontos információkkal szolgál a pálya jelenlegi állapotáról. A nem megfelelő haladás miatti kopások és elhasználódások előzhetők meg a használatával. A hagyományos geodéziai darupályavizsgálathoz képest gyorsabb, több mérési pontot eredményező monitormódszer. Az eredmények használhatók illesztésméréshez, a deformáció vizualizálásához, két-háromtengelyes megjelenítéshez, egyenességgel, távolsággal és emelkedéssel kapcsolatos problémák javításához.

2.3.3. Összehasonlítás

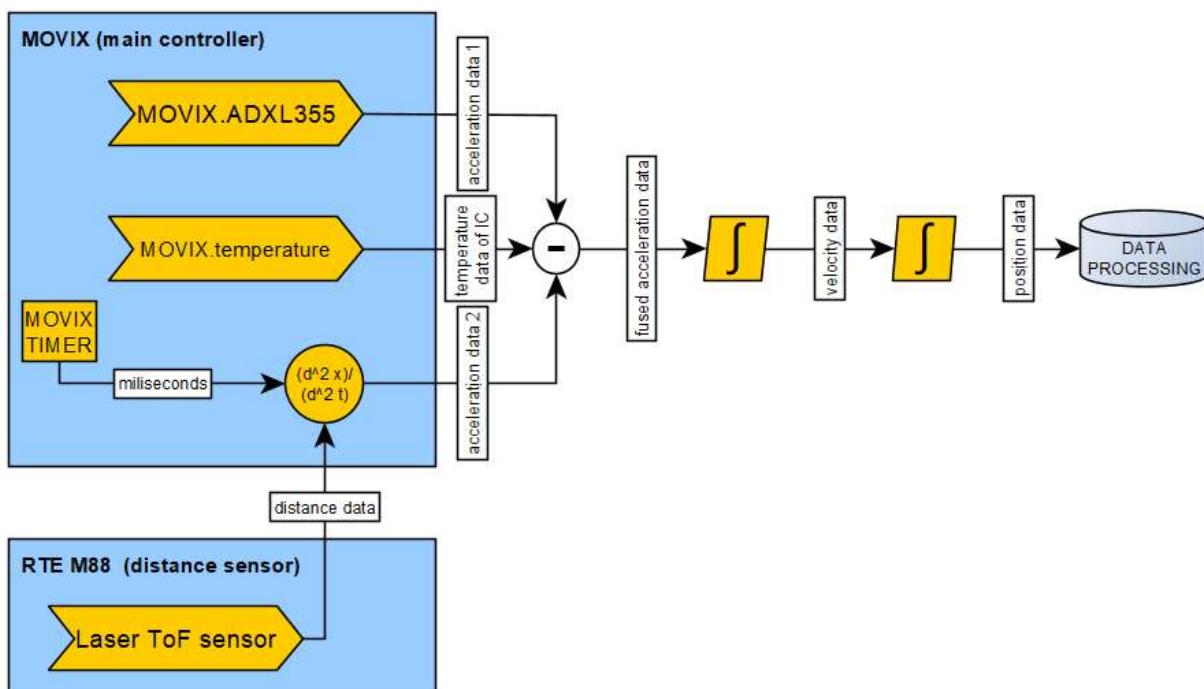
	DAMOR	TruConnect	RailQ
Szenzoros figyelés	●	●	-
Túlterhelés figyelés	-	●	-
Túlmelegedés figyelés	-	●	-
Vészleállítás figyelése	●	●	-
Darupálya hibák	●	-	●
Pályahiba lokalizálása	●	-	●
Naplózás	●	●	●
Új darura telepíthető	●	●	●
Régi darura telepíthető	●	●	●
SMS/E-mail jelentés	●	●	-
Adatelemzés	●	●	-

--	--	--	--

3. Prototípus - Hardver

3.1. Koncepció, rendszerterv

A rendszer alapvetően, két főszenzorból, két mellékszenzorból és ezeket vezérlő központi vezérlő egységből áll. A két főszenzor egy nagy pontosságú gyorsulásmérő kiegészítve egy távolságmérő szenzorral. A gyorsulásmérőt az AIRobotix-tól tervezük beszerezni, mert ez a szenzorcsomag nagy pontosságú alap integrált áramkört használ, továbbá belső szenzorházon belüli hőmérsékleti kompenzációval rendelkezik, ez a harmadik szenzor, egy kiegészítő belső hőszenzor. A központi vezérlőt is az AIRobotix-tól tervezük beszerezni, a MOVIX modellt. Ebben, a WiFi és BT kapcsolatra alkalmas mikrovezérlőhöz illesztve, egy nagy pontosságú 3-tengelyes gyorsulásmérő található, mely -40 és +125 °C fokban képes működni +/- 8G gyorsulási tartományban. Zaj és ofszet összetevői kicsik: a zaj 20 µG/√Hz minden irányban, míg az elmászás X és Y irányban 9, Z irányban pedig 13 µm/sec/√Hr.



9. ábra

3.2. Kritériumok, specifikációk

Fontos megszabás, hogy az eszköz három irányban legyen képes elmozdulást mérni. További elvárások voltak, hogy az alkatrészek működőképesek legyenek ipari hőmérsékleti tartományban (-10 és +40 °C között) és nagy pontosságú és alacsony zajú/ofszetű szenzorok legyenek alkalmazva.

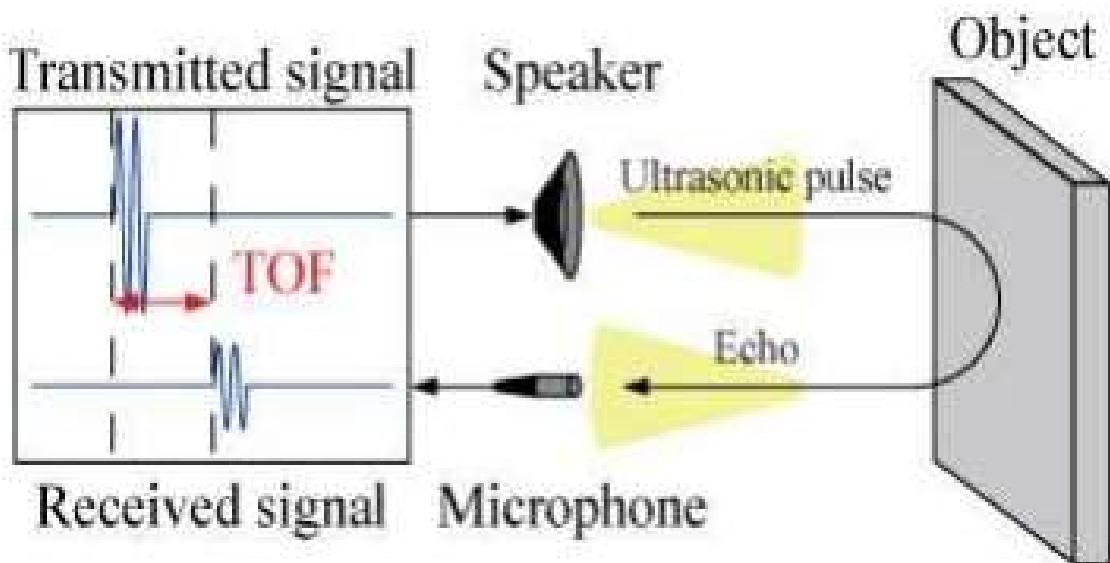
3.3. Szenzorok

3.3.1. Távmérő szenzorok

3.3.1.1. Ultrahang alapú méréstechnika

Az ultrahang használata távolságmérésre nem új felfedezés, mert az állatvilágban már régóta használják a delfinek kommunikációról, sőt olyan nagy erejű, hogy esetenként még a zsákmányállatok megbénítására is alkalmas, vagy ami a dolgozat témájához közelebb áll, a denevérek tájékozódása. A kis emlősök látása elég rossz ezért repülés közben ultrahangot bocsájtanak ki - aminek a frekvenciája fajonként különböző 35-105 KHz közé esik - és a visszaverődött hanghullámokból fel tudják térképezni környezetüket, vagy az áldozat pontos helyét is meghatározhatják. Lényegében a denevért egy biológiai ultrahangos távolságmérőnek is felfoghatjuk. Az ultrahang az elektronikában, a navigációs technikában, az iparban és a biztonsági alkalmazások területén széles körben használatos.

Az ultrahangos távolságmérés, a pulse-echo módszer: A pulse-echo módszer nagyon egyszerű elven működik: van egy adóegység, amely általában egy 40KHz-es többnyire 8 impulzusból álló ultrahangcsomagot sugároz ki (pulse) - ez berendezésekkel változhat - és ha a hang útjába valami akadály kerül, akkor arról egy része a kisugárzott burst-nek visszaverődik (reflektálódik) és a vevő egység ezt érzékelheti (echo). A 4. ábra a pulse echo módszert szemlélteti. A kisugárzás és a vétel közötti eltelt idő pontos mérésével (Time of Flight, TOF) és ismerve a közeget, amiben az ultrahang terjedt kiszámíthatjuk az objektum távolságát.



10. ábra Echo ultrahang módszer

A pontos távolságméréshez két érték precíz ismeretére van szükségünk, a hang terjedési sebessége az adott közegben és a kisugárzás és vétel között eltelt pontos idő.

3.3.1.2. Távolság mérése elektronikus módszerekkel

Minden olyan mérési elvet, megoldást elektronikus mérésnek fogok tekinteni, amely előállításához elektromosság szükséges, ezért az optikai elven működő lézeres távolságmérést is itt fogom meg említeni, mivel a lézerfény előállítása elektronikus úton történik. Távolságot mérhetünk kapacitásváltozás mérésével, ilyenkor a kapacitást meghatározó három komponens (fegyverzetek közötti távolság, a fegyverzetek területe és a dielektrikum permittivitása) bármelyikének változása határozza meg az elmozdulást. A fegyverzetek távolságának elmozdulásával csak 2-3mm-es távolságot tudunk mérni. Az induktivitás változás alapján is tudunk távolságot, pozícióváltozást mérni, ilyenkor a saját, vagy kölcsönös (mutuális) induktivitás változását figyeljük. Az induktivitás a vasmag elmozdulása miatt változhat, ebben az esetben 5-30mm-es elmozdulás érzékelhető. Láthatjuk, hogy az előző esetekben igen csekély változásokat tudunk mérni, melyek egy robot működésében nem jelentős változások, bár lehetnek olyan esetek, amikor ilyen kis távolságok mérése szükséges, azonban nem ez az általános. A másik csoportja a méréseknek a lézeres távolságmérés. Amikor a távolságmérés technológiája még nem volt igazán kifinomult intenzitásképeket használtak a távolság meghatározására. Az egyes pontok fényessége attól függ, hogy a környezet adott pontjáról mennyi fény jut az érzékelőbe. A fényforrás és az érzékelő modellezése nagyon bonyolult, és az objektumról semmilyen információ nem áll rendelkezésünkre. A magas bizonytalansági faktor miatt egyre szélesebb körben kezdtek el alkalmazni olyan eszközöket, amelyek közvetlenül alkalmasak a távolság mérésére. Ezek e vizsgált térrészről távolságképet alkotnak, melyek pontmártixában a távolságmérések értékei szerepelnek.

3.3.1.3. Távolságmérés Bluetooth segítségével

Egy zárt téren belül egy jeladó pozíciója akár méteres pontossággal is meghatározható 3(TOA) vagy 4(DTOA) fixen telepített antenna segítségével. A TOA (Time of Arrival) eljárás lényege, pl.: villámlás helyének detektálására, a nagyon pontos időmérés. Villámlás esetén eltárolják a pontos időpontot, amit a rendszerben lévő állomások esetén GPS segítségével szinkronizálnak. Mivel a villámlás az egyes állomásoktól különböző távolságra következik be, azokra eltérő időpontban érzékelik a villám által kisugárzott jel. A jel fénysebességgel terjed, így az időkülönbség nagyon csekély, ezért kiemelten fontos a nagyon precíz időmeghatározás. Az időkülönbségek két-két állomás között egy-egy hiperbolával írhatók le. A hiperbolák metszéspontja pedig megadja a villámlás helyét. A Bluetooth távolságmérés is erre az elvre épül. A három fixen telepített antennát rendszerbe kötik, és pontosan egymáshoz szinkronizálják. Amikor a jeladó küldi az előre meghatározott időközönként a jeleket, az antennák érzékelik ezeket és visszaküldik a jeladó felé. A jel terjedési sebességének a fele a megtett út így kiszámítható. A három antenna pontos pozíciójának ismeretében a jeladó helyzete pontosan meghatározható. A DTOA vagyis a 4 antennás eljárás alkalmával a rendszerbe beillesztünk egy negyedik antennát is, a 3 antenna síkjától eltérő síkban, ezzel is növelte a mérési pontosságot. Ez a távolságmérési módszer kísértetiesen hasonlít a GPS és a differenciális GPS módszerek alkalmazásához.

3.3.1.4. ToF (Time of flight) módszer

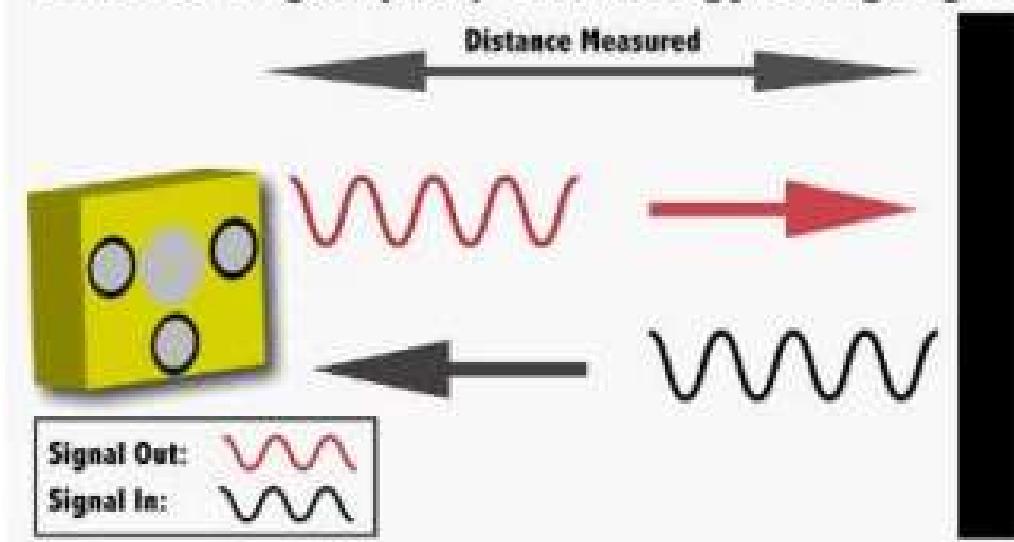
A repülési idő (TOF, Time of flight) lényegében egy test, részecske, akusztikus vagy elektromágneses hullám (összefoglaló néven objektum) tulajdonsága. Ez az az idő, amely mutatja, mennyi ideig mozog az objektum a közegben. Az idő mérésére a standard időmértekegység használható, amiből kiszámítható a közegen keresztül megtett út, vagy az objektum sebessége. Az előzőekben megismert mennyiségek (sebesség, megtett út) segítségével jobban megismerhetjük az adott objektumot, vagy közeget. A mozgó objektum észlelhető direkt (pl: ion detektor, tömegspektrométer) vagy indirekt módon (pl. a testről megtört fény lézer doppler sebességméréssel). Az ultrahangos áramlásmérő akusztikus szenzorok által méri a csőben lévő folyadék vagy gáz sebességét. Előnye a többi mérési technikával szemben, hogy az eredményeket csak kis mértékben befolyásolja a hőmérséklet, sűrűség vagy vezetőképesség. A technológia karbantartása egyszerű, mert nincsenek mozgó alkatrészek.

Ultrahangos áramlásmérők 3 változatban léteznek:

1. átvitel (ellenirányú tranzit idő) áramlásmérők: megmérik az időkülönbséget a kiküldött ultrahang impulzus az áramlási irányba és az azzal ellentétesen küldött impulzus között.
2. visszaverődési (doppler) áramlásmérők: doppler-váltást mérik eredményezve egy visszavert ultrahangot. Ez a reflexió történhet részecskékről folyadékban, légbuborékokról a folyadékban vagy az áramló folyadék turbulenciájáról
3. nyitott-csatornás áramlásmérők: az árok előtti ellenirányú áramlás szintjét mérik.

Optikai ToF szenzorok 2 fénycsót bocsájtanak ki a vizsgált tárgyra. A sebességet úgy tudjuk kiszámolni, ha ismerjük a két sugár elhelyezkedését (egymástól való távolságát). Amennyiben csak egy detektor van, akkor az időkülönbséget autokorreláció által számoljuk ki. Ha 2 detektor van (minden sugárhoz egy-egy), akkor az elmozdulás iránya is meghatározható. Mivel a sugarak helyét viszonylag könnyű meghatározni, a mérés pontossága főleg azon múlik, hogy a szenzort (mérőeszközöt stb.) mennyire kicsiben lehet kivitelezni.

Time-of-Flight (ToF) Technology Using Light



11. ábra ToF módszeren alapuló távolságmérés

Az alábbi ábrán látható a két végső versenyben maradt ultrahang és a ToF metodika összehasonlítása egy táblázatban.

	Ultrahang	ToF
Magas olvasási frekvencia	✗	✓
Nagy távolság	✗	✓
Kis súly	✓	✓
Kis méret	✓	✓
Szem védelem	✓	✓
Többszörös szenzorok	✗	✓

12. ábra ToF vs. UH

3.3.1.5. LIDAR

A LIDAR (Light Detection and Ranging) aktív távérzékelési rendszerekhez tartozik, ami azt jelenti, hogy a felvező berendezés saját energiaforrással rendelkezik – így sugárforrás és érzékelő egyaránt. A működési elve legjobban a radar működési elvével hasonlítható össze, azzal az alapvető különbséggel, hogy a LIDAR által kibocsátott és fogadott elektromágneses sugárzásnak magasabb a frekvenciája. A LIDAR az ultraviola, a látható és az infravörös tartományban működik. A LIDAR működésének alapelve megegyezik az elektromágneses energia alapú (Electronic Distance Measuring Instrument) távolságmérés elvével. A lézer által kibocsátott energia-impulzus (vagy folytonos hullám) kölcsönhatásba kerül a terjedés irányában lévő objektumokkal, melynek következményeként visszaverődik. Az egyes impulzusok kibocsátása és visszaverődése között eltelt idő rögzítésre kerül. Mivel az elektromágneses energiaterjedési sebessége ismert, a tereptárgynak a lézertől való távolsága meghatározható a tárgy által visszavert sugár kibocsátásának és visszaérkezés idejének különbségéből. A terepi tárgyak (mérési pontok) pontos térbeli elhelyezkedésének (koordinátáinak) meghatározásához azonban szükséges a szenzor helyének és helyzetének pontos ismerete.

Távolságmérés során, mint fentebb látható felmerült szenzor opciók voltak az ultrahang érzékelők és a lézeres távolságmérő szenzorok. Az ultrahang szenzor opciót elvetettük, mivel nem képes teljesíteni a minimális elvárt 30 méteres távolságot. Felmerült a GARMIN által gyártott két LIDAR is. Az egyik a LIDAR LITE V3 volt, ez képes 5cm és 4000cm között mérni, de +/-25mm pontossággal rendelkezett, a másik a LIDAR-Lite v3HP volt, amely szintén 5cm és 4000cm között képes mérni, de ez is 25mm-es pontossággal dolgozik. A szintén GARMIN LIDAR-Lite v4 LED eszköz, csak 10 méteres távolságig üzemképes. Az előbb felsoroltakat elvetettük az RTE Measure által készített M88-as szenzorhoz képest, melynek specifikációját az alábbi ábra tartalmazza.

Table 2-1 key parameters	
Accuracy	± 1 mm
Measuring Unit	millimeter
Measuring Range	0.03-40m
Measuring Time	0.4~4 seconds
Laser Class	Class II
Laser Type	635nm, <1mW
Size	25*48*12mm
Weight	About 9g
Voltage	DC 2.5~3.3V
Operating Temperature	0~40 °C (32~104 °F)
Storage Temperature	-25~60 °C (-13~140 °F)

1.Under bad measure condition,like environment with strong light or the diffuse reflectance of measuring point over-high or low the accuracy would have bigger amount of error:±3mm+40PPM.

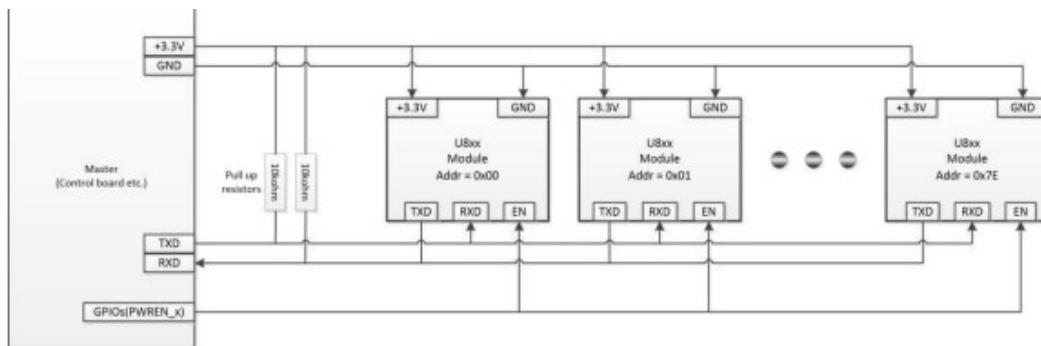
2.Under strong light or bad diffusere flectance of target, please use a reflector.

3.Operating temperature -10°C~50°C can be customized.

13. ábra Lidar specifikáció

Látható, hogy a pontosság bőven 1mm alatt marad, amivel mesze a versenytársak feletti teljesítményt nyújt a szenzor, miközben a mérési tartomány is kedvezőbb 3cm-től, 40m-ig terjed. A működési hőmérséklet -10°C és +50°C fok közé is kiterjeszthető.

További előny a lenti ábrán látható, mely alapján az eszköz, meghaladva a hagyományos USART (RS232) lehetőségeit és alkalmas egy master eszköz több slave eszközzel is kommunikálni ugyanazon a buszon, gyárilag az összes szenzor a 0x00 hexadecimális címmel rendelkezik, ezt egyenként módosítanunk kell, mivel mi négy szenzort használunk, így lesz egy 0x00, egy 0x01, egy 0x02 és egy 0x03 címzésű érzékelőnk. A kívánt slave címet USART protokollon keresztül tudjuk beleégetni az összeszerelés során még a beüzemelés előtt az egyes szenzorokba, tehát ahogy a vezérlő egység felprogramozása úgy a szenzorok egyenkénti (!) felkonfigurálása is a gyártási folyamat része lesz.



14. ábra Master és slave kommunikáció

3.3.2. Gyorsulásmérő szenzorok

A nyomásszenzorok mellett a mikromechanikai szenzorok másik nagy csoportját a gyorsulás mérő szenzorok alkotják. Ezeknek a szenzoroknak a nagy többségét a járműipar használja, mégpedig a következő rendszereknél:

1. benzinmotorok kopogásszabályozása,
2. járműgyorsulások meghatározása a blokkolásgátló (ABS) vagy a stabilitási (ESP) rendszerek számára,
3. légszák és övfeszítő biztonsági rendszerek működtetésére ütközéskor,
4. karosszériagyorsulások kiértékelésére a felfüggesztési rendszerek szabályozásához.

A gyorsulásszenzoroknál a mérési tartományt a nehézségi gyorsulásra szokás vonatkoztatni, ennek mértékegysége a g ($g = 9,81 \text{ m/s}^2$), de használják ennek ezredrészét is, ez a mg (nem tévesztendő össze a milligrammal).

A különböző feladatoknál használatos méréstartományokat a táblázat mutatja:

alkalmazás	mérési tartomány (gyorsulás)
kopogás-szabályozás	40 g
utasvédelem: – légszák, biztonsági öv – oldal- és frontális ütközés érzékelése – borulás érzékelés	35...100 g 100...400 g 3...7 g
ESP, HHC, ABS	0.8...1,8 g
futómű szabályozás (rugózás): – felépítés – tengelyek, csillapítás	1 g 10...20 g
gépkocsi riasztás	1 g

15. ábra Gépjárművekben alkalmazott gyorsulásmérők mérési tartománya - forrás: Bosch

A gyorsulásszenzorok elméleti alapjait a következőkben foglaljuk össze.

A mérési elvek közül a leggyakoribb a Newton 2. törvénye alapján működő elv, amelynek jellemzője a „szeizmikus tömeg”, amelyre érvényes az

$$F=m \cdot a \\ F=m \cdot a = m \cdot a$$

összefüggés és a szenzor az m tömegre ható F erőt méri. Az erőmérést vagy útmérésre, vagy mechanikai feszültség mérésére vezetjük vissza (lásd később). Állandó gyorsulásnál a gyorsulási erő az x kitérésű és c rugóállandójú rugó ellenerejével van egyensúlyban.

$$F=m \cdot a=c \cdot x \\ F=m \cdot a=c \cdot x$$

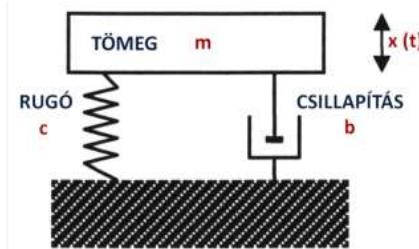
A rendszer mérési érzékenysége (S, sensitivity):

$$S=xa=mc \\ S=xa=mc$$

Nagy érzékenységet tehát nagy szeizmikus tömeggel, és kis rugómerevséggel lehet elérni.

Változó gyorsulásnál, azaz dinamikus esetben (ez a gyakoribb eset) a gyorsulásszenzor működését legegyszerűbben állandó együtthatós másodrendű differenciálegyenlettel lehet leírni.

$$F=m \cdot a=c \cdot x+b \cdot x' +m \cdot x'' \\ F=m \cdot a=c \cdot x+b \cdot x' +m \cdot x''$$



16. ábra A gyorsulásmérők egyszerűsített rendszerteknikai modellje

A szeizmikus tömeg (m) és a rugóállandó (c) adja meg a rendszer csillapítás nélküli ω_0 sajátfrekvenciáját, amelyet a rendszer csillapítási tényezője kismértékben csökkent ugyan, azonban ez a csökkenés nem jelentős mértékű.

$$\omega_0=\sqrt{cm} \\ \omega_0=\sqrt{cm}$$

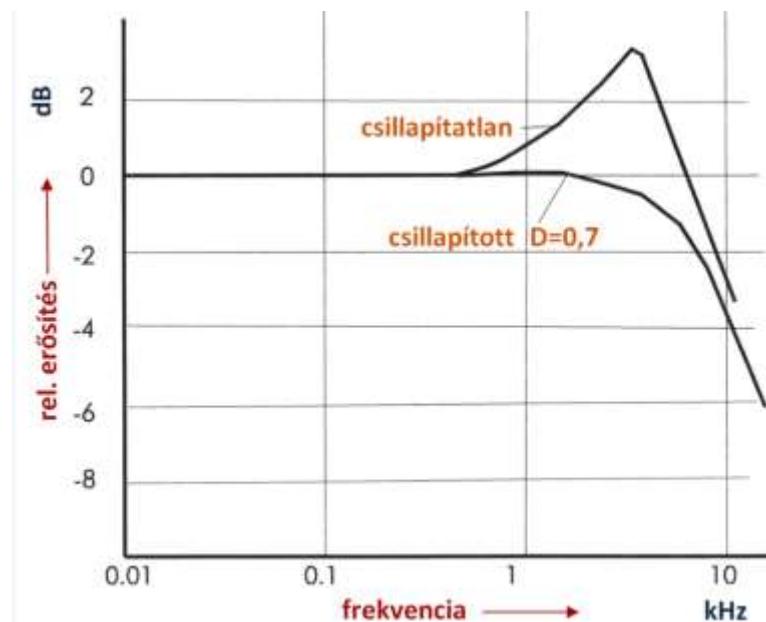
Fontos megérteni, hogy egy szenzor érzékenysége és sajátfrekvenciája szoros összefüggésben van egymással.

$$\omega_0 S = 1 \\ \omega_0^2 S = 1$$

Az egyenlet azt fejezi ki, hogy sajátfrekvencia növelése csak az érzékenység rovására lehetséges, tehát például kétszer nagyobb sajátfrekvencia eléréséhez negyedakkora érzékenység csökkenés tartozik. Másodrendű rendszereknél a rendszer dinamikus működését az energiatárolókon kívül a csillapítási tényező is befolyásolja. Ennek azonban nem az abszolút

értéke, hanem a rendszer többi jellemzőihez viszonyított aránya határozza meg. Ez a csillapítási szám, amelynek nincs dimenziója:

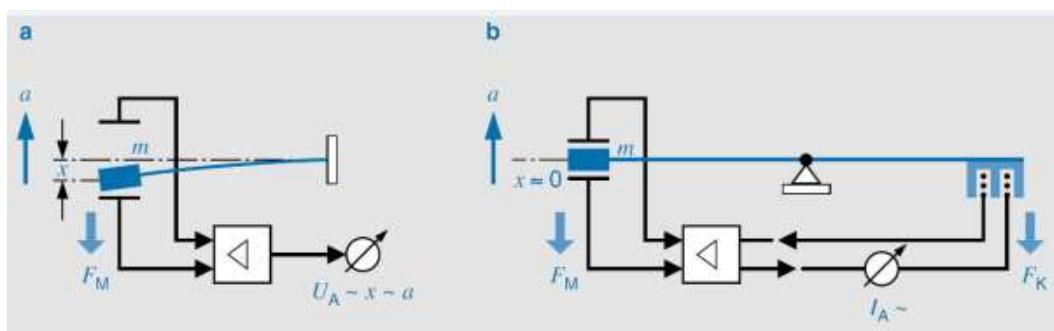
$$D = b^2 c \cdot m \sqrt{D} = b^2 c \cdot m$$



17. ábra Másodrendű rendszerek amplitúdó diagramja a frekvencia függvényében

Az ábrán látható a másodrendű rendszerekre vonatkozó jól ismert Bode-diagram az amplitúdóra és a fázisra vonatkozóan. Látható, hogy változó frekvenciák esetén a $D=0,7$ körüli csillapítási szám adja a legnagyobb amplitúdó sávszélességet, emiatt a szenzorok konstrukciójánál törekedni kell a kívánatos csillapítási szám megtartására.

Mérési elvként vagy a kitéréses, vagy a kompenzációs (erőkompenzációs) elvet választják. Előző esetben az F erő hatására bekövetkező rugalmas alakváltozás nagyságát mérjük, a másodikban szabályzókort használunk, és nem engedünk meg elmozdulást. Ekkor a visszacsatolt erővel kapcsolatos jellemzőt, például egy elektromágnes áramát mérjük, és feltételezzük, hogy ez arányos lesz a mérendő gyorsulással.



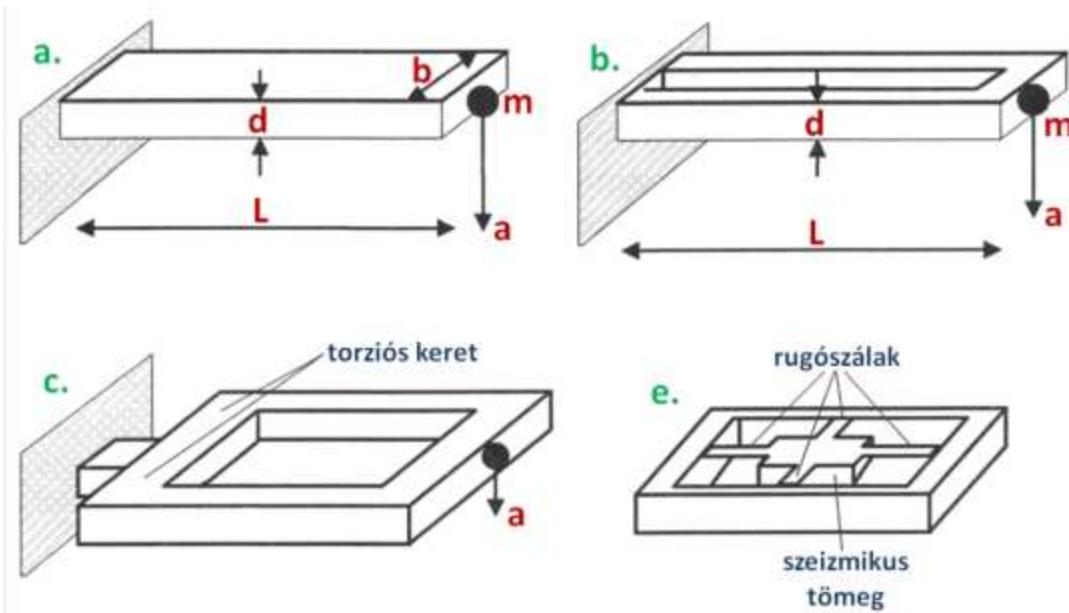
18. ábra A gyorsulásmérőknél alkalmazott mérési elvek: a.) elmozdulás mérés b.) erőmérés

Az ilyen (elmozdulás mérésre, illetve erőmérésre kifejlesztett szenzorok a rezonancia frekvencia (jó közelítéssel a sajátfrekvencia) alatt kielégítő arányosságot mutatnak a mérési érték és a kitérés között. A kitérések villamos jellé alakítására az táblázatban foglalt mérési eljárásokat használják.

a szenzor	elv	műszaki vonatkozások	alkalmazási példák
piezorezisztív	rugó-tömeg rendszerben a gyorsulás által létrejött megnyúlás detektálása ellenállás változással	szilícium-tömbi mikromechanika, vékonyréteg rendszerek	labor alkalmazások, korábban ütközési szenzorok is
piezoelektrikus	töltés szétválasztás kristályban erő hatására (rugó-tömeg rendszer)	kerámia anyagok, PZT, kvarc, PVDF	kopogási szenzorok, korábban ütközési szenzorok is
kapacitív	rugó-tömeg rendszer kiolvasása síkkondenzátoros kapacitásméréssel	szilícium tömbi mikromechanika, felületi mikromechanika	fedett felülettel gépjárműekben
termikus	melegített gáztömeg laterális elmozdulása és a fűtési zóna aszimmetriájának detektálása	tömbi mikromechanika	borulás érzékelés

19. ábra A mikromechanikai gyorsulásmérőknél alkalmazott mérési elvek

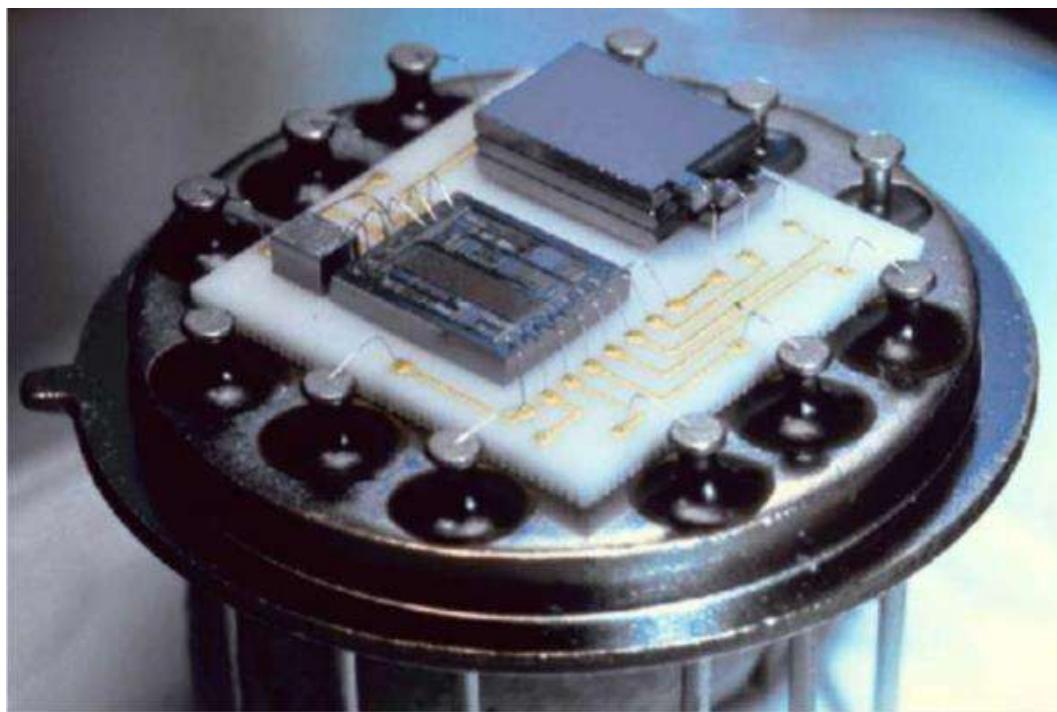
A mikromechanikai gyorsulásmérőknél olyan konstrukciós kialakításokat használnak, amelyek jól illeszkednek a mikromechanikai technológiákhoz. Néhány szokásos megoldást mutat az alábbi ábra. Megjegyezzük, hogy a modellezéskor ezek a kialakítások nem írhatók le koncentrált paraméterekkel, megfelelő közelítést csak osztott paraméteres modellezéssel lehet elérni.



20. ábra Példák mikromechanikai gyorsulásmérők szeizmikus tömegének kialakítására

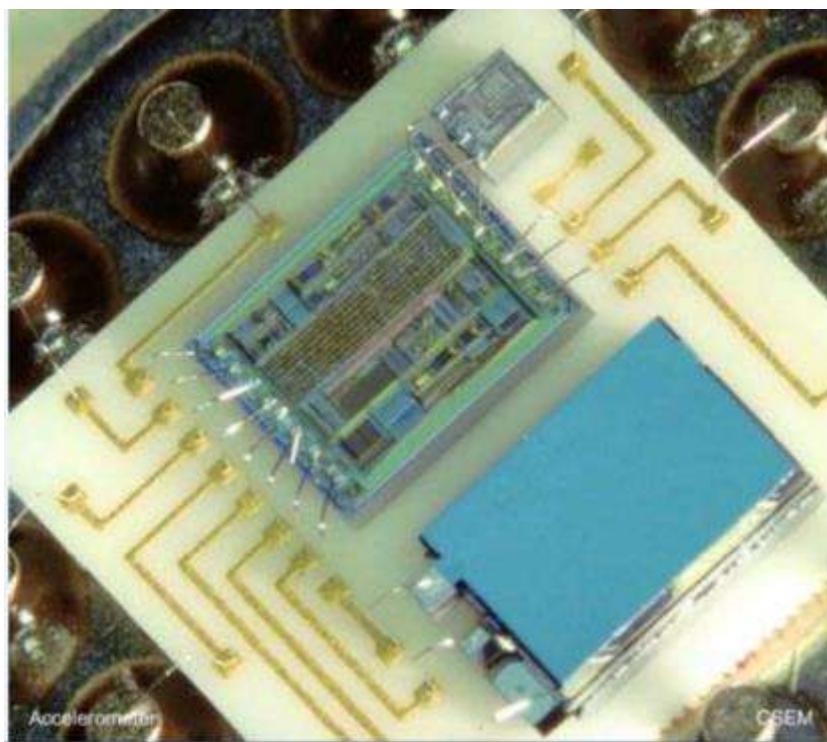
3.3.2.1. Kapacitív gyorsulásmérő

Példaképpen bemutatunk egy kapacitív mérési elven alapuló, tömbi mikromechanikai eljárással készülő gyorsulásmérőt (MS6100), illetve annak gyártástechnológiáját. A gyorsulásmérő két egységből áll, egy mikromechanikai, és egy mikroelektronikai egységből, és minden kettő szilíciumból készül. A hordozó kerámia lapka, a tokozás egy szabványos 12 lábú IC tok. A tokozás előtti állapotot mutatja az ábra, ahol a baloldalon látható a mikroelektronikai rész, jobboldalon a mikromechanikai rész.



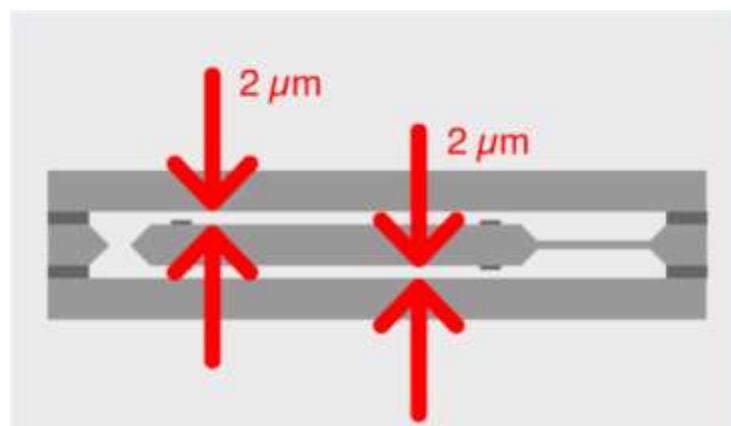
21. ábra Tömbi mikromechanikai technológiákkal kialakított gyorsulásmérő tokozás előtti képe

A gyorsulásmérőket több méréshatárral gyártják (2, 10 és 25 g), a lökésállóságuk 6000 g. Jellemző rájuk a kis áramfelvétel és a nagy precizitás. A következő, ábrán jól látható a kerámia hordozóra szerelt két egység, és a 25 µm átmérőjű aranyhuzalokkal létesített villamos kötések.



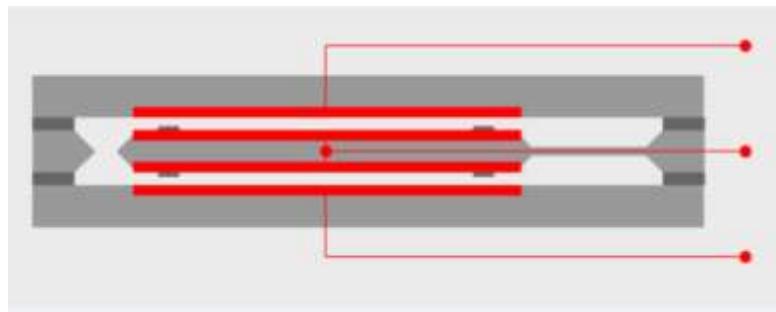
22. ábra A gyorsulásmérő mikroelektronikai és mikromechanikai egysége

A mikromechanikai rész három monokristályos szilícium hordozóból áll. A középsőből kerül kialakításra a szeizmikus tömeg a felfüggesztő rugóval. Erre a középső elemre SiO₂ rétegeket kell felvinni, mert a mérés differenciál-kapacitív elven történik, és az elektródákat egymástól el kell szigetelni. Az így kialakított kondenzátorok fegyverzetei között 2 μm távolság van.



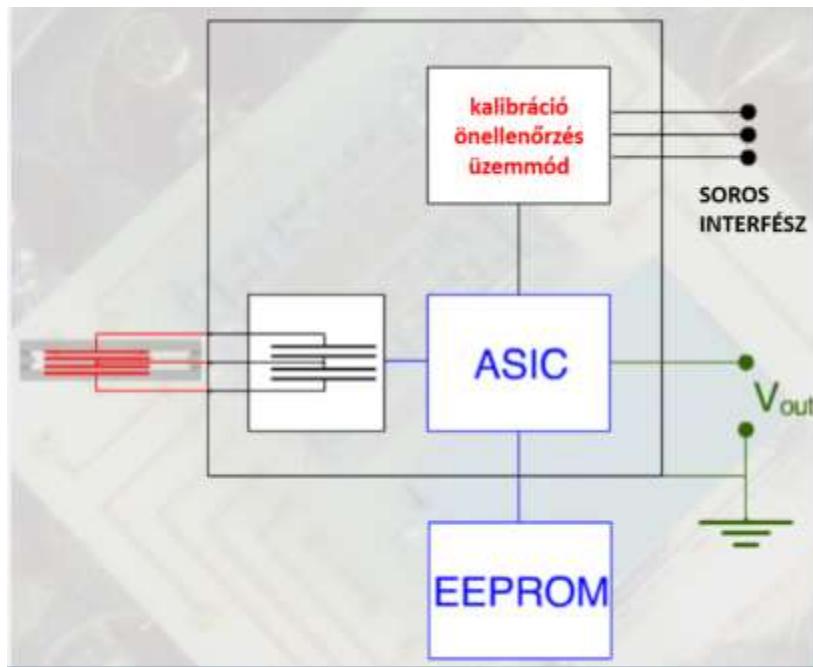
23. ábra A gyorsulásmérő vázlatos keresztmetszete

Az így kialakított kondenzátorok kapacitása kb. 20 pF. A függőleges irányú gyorsulás hatására bekövetkező deformáció (a szeizmikus tömeg elmozdulása) hatására az egyik kapacitás növekedni, a másik csökkenni fog.



24. ábra A gyorsulásmérőben kialakított sikkondenzátorok

A mikromechanikai részben kialakított kondenzátorokat két másik, passzív kondenzátorral Wheatstone-híddá egészítik ki (félhidas mérés) és ennek kimenő jele kerül további feldolgozásra a mikroelektronikai részben.



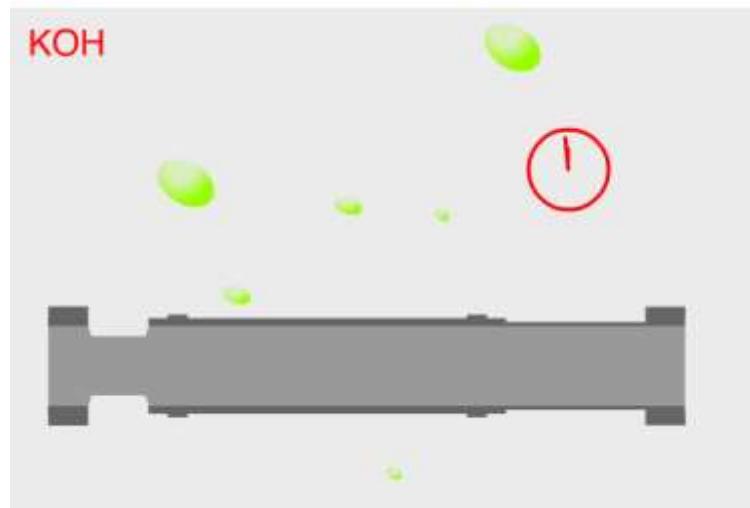
25. ábra A gyorsulásmérő mérő és jelfeldolgozó áramkörei

A szenzor legfontosabb része a középső, 4" átmérőjű és 380 µm vastag szilíciumszeletből kialakított, rugalmasan felfüggesztett szeizmikus tömeg. A technológia főbb lépései a műveletet illusztráló ábrák segítségével mutatjuk be.



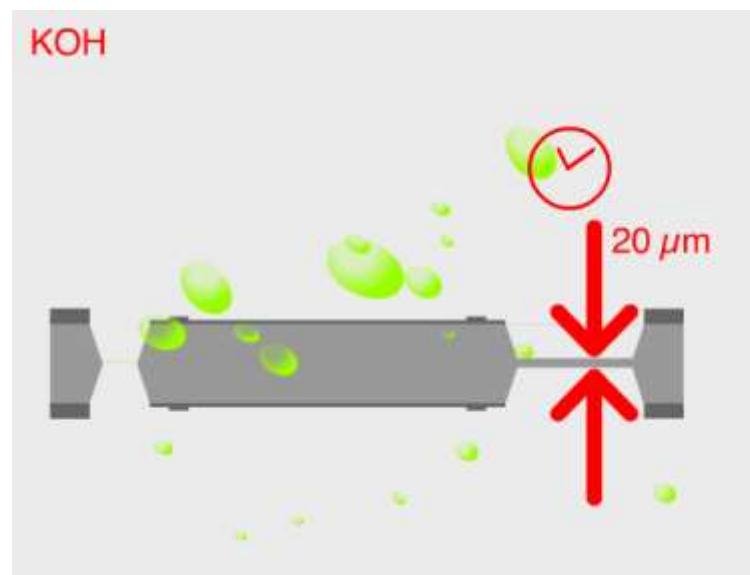
26. ábra A különböző vastagságú oxidrétegek kialakítása

Az első lépés az oxidálás, méghozzá több lépésekben: 2 µm szükséges a túlzottan nagy gyorsulásnál fellépő felütközésnél a rövidzár megakadályozására, de az alsó-felső fegyverzet távolságának biztosítására ennél sokkal vastagabb rétegekre is szükség van. Ezeket a feladatokat fotolitográfiás strukturálással oldják meg.



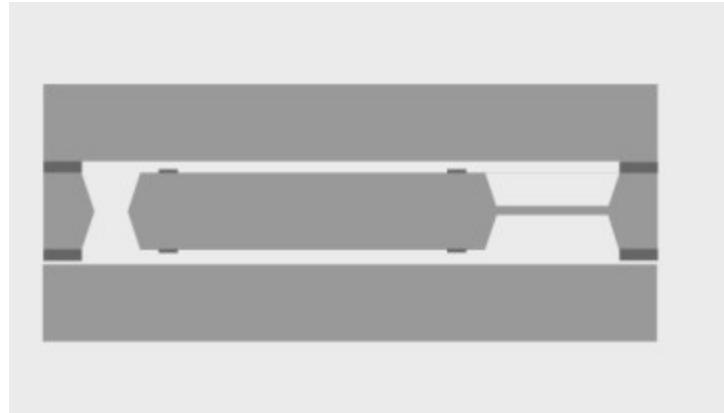
27. ábra A nedves anizotróp marás elindítása

Következő lépésként LPCVD (low pressure chemical vapor deposition) eljárással szilíciumnitrid (Si_3N_4) réteget választanak le. Fotolitográfiával kiválasztják azokat a területeket, ahonnan a szilíciumnitrid réteget el kell távolítani reaktív ionos (reactive ion etching, RIE) marással. Ezután kezdődhet a nedves kémiai anizotróp marás (wet anisotropic etching), káliumhidroxid (KOH) segítségével. Ezt azonban egy bizonyos idő után meg kell szakítani.



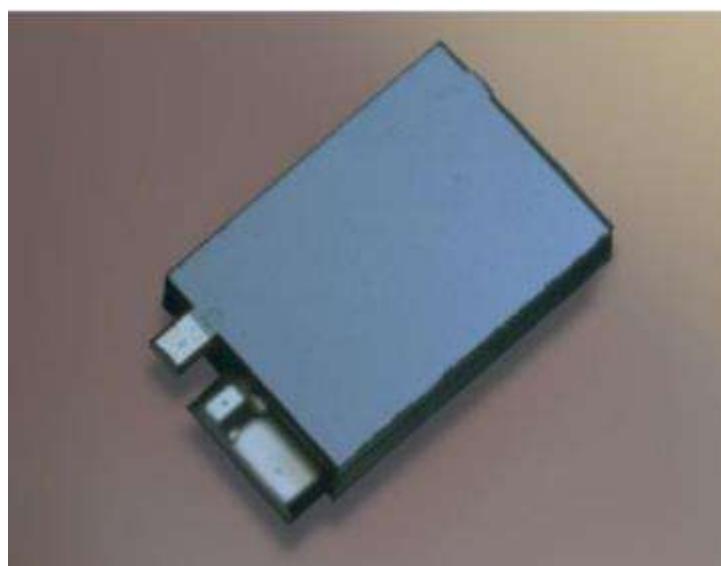
28. ábra A nedves anizotróp marási folyamat befejezése

Ismét strukturálás következik, a rugalmas elem feletti védőréteg eltávolítása, majd a marási folyamat folytatása egészen addig, míg a laprugó vastagsága el nem éri a 20 µm vastagságot, ahogyan az az ábrán is látható.



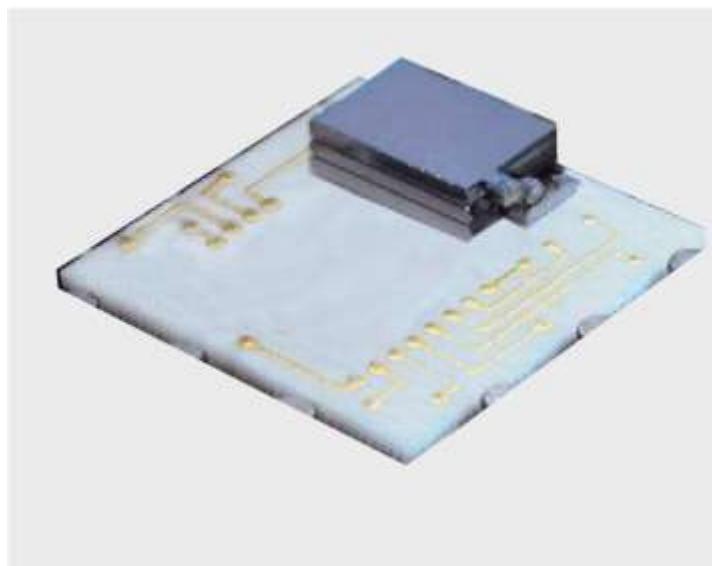
29. ábra A felső és az alsó fegyverzetet alkotó szilíciumszelet hozzákötése a középső szelethez

Ezután anódos kötéssel a középső szilícium szelethez hozzákötik az alsó és felső szilícium szeletet. Ezzel tulajdonképpen a mikromechanikai rész már elkészült.



30. ábra A kész mikromechanikai gyorsulásmérő

Következő műveletként a kivezetéseket alumínium (Al) gőzöléssel vonják be, majd ezután következik a darabolás.



31. ábra A gyorsulásmérő felagasztása a kerámia hordozóra

A gyorsulásmérőt a huzalozást már tartalmazó kerámia lapkára ragasztják (5.51. ábra), majd emellé hozzáerősítik a jelfeldolgozó áramkört, és a kalibrálási konstansokat tartalmazó EEPROM (Erasable Electrically Programmable Read Only Memory) áramköröket is. A kivezetések 25 µm átmérőjű aranyhuzalból készülnek, a tokozás előtti állapotot mutatja a már korábban bemutatott ábra.

A mikromechanikai eljárásokat abból a szempontból két nagy csoportba szokás sorolni, hogy az eszközök előállításánál a laterális, síkbeli méretekhez képest mekkorák a mélyiségi méretek. Ha a mélyiségi méretek relatíve kicsinyek a laterális méretekhez viszonyítva (néhány µm-től néhányszor 10 µm-ig terjednek), felületi mikromechanikáról beszélünk (surface microaching). Amennyiben a mélyiségi méretek nagyobbak, ezek általában más technológiai lépéseket igényelnek, és ilyenkor tömbi mikromechanikáról beszélünk (bulk micromachining). A felületi mikromechanika technológiája általában kevesebb ráfordítást igényel, és így olcsóbb, mint a tömbi mikromechanikai eljárások. A gyorsulásmérőknél mind a felületi, mind a tömbi mikromechanikai technológiákat használják.

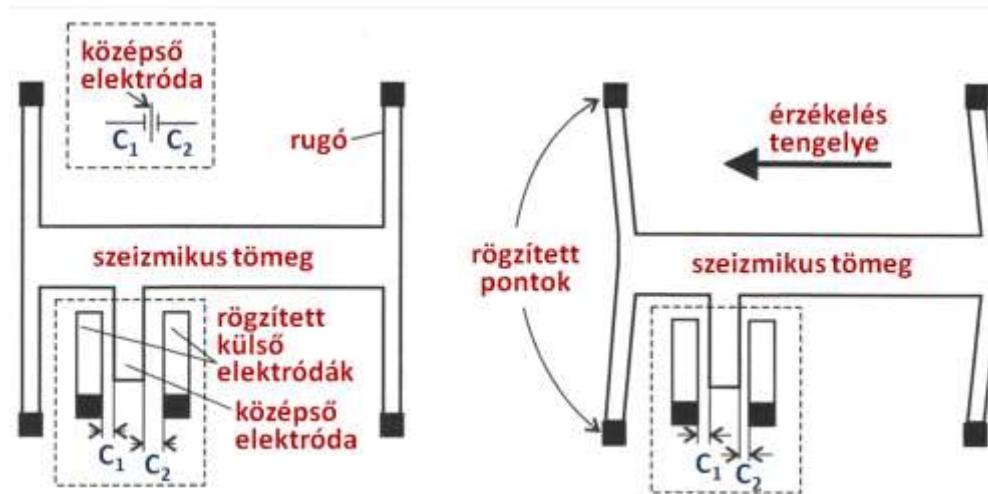
A gépjármű technikában a gyorsulásmérő szenzorok egyik tipikus alkalmazási területe az utas biztonságát szolgáló légzsákok vezérlésére szolgáló szenzorok területe. Ezeket ma felületi mikromechanikai technológiákkal készítik. Egy jellemző kialakítást mutat az ábra.



32. ábra Légsák vezérléshez kifejlesztett gyorsulásmérő

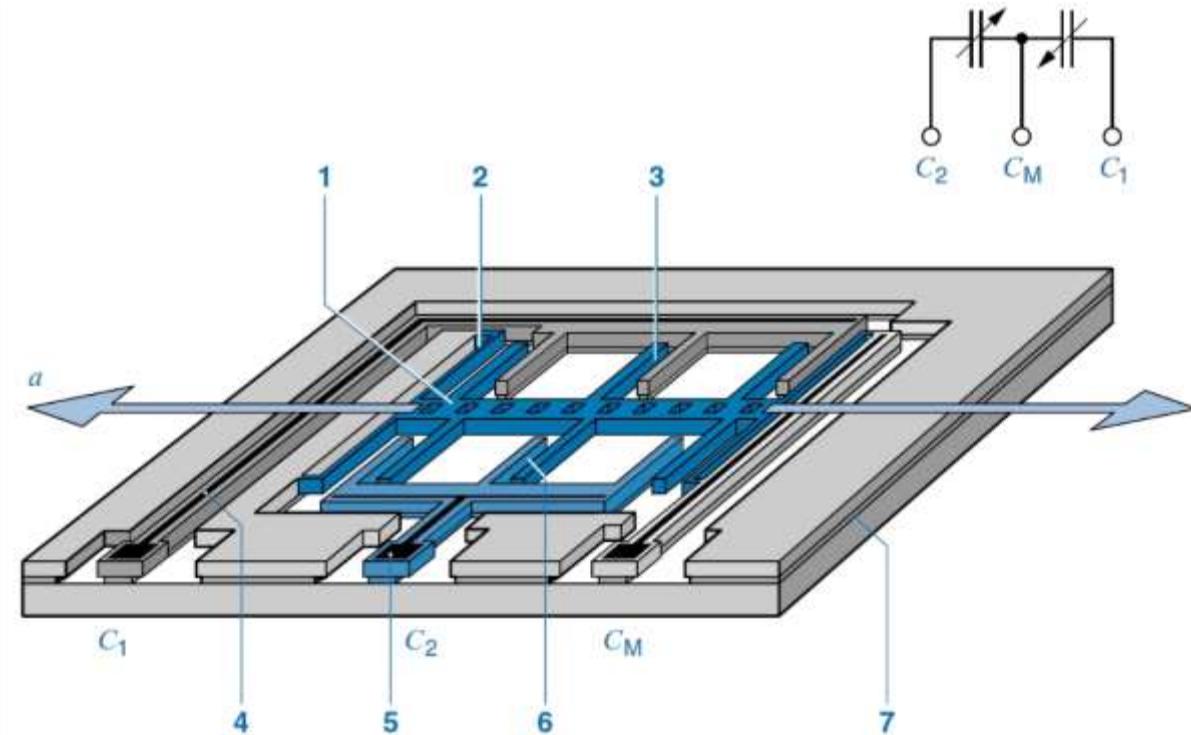
A külső ház (5) robusztus és viszonylag nagy, ebben helyezkedik el a (4) áramköri szerelőlap, amely az 1 gyorsulásmérő modult tartalmazza szabványos SOIC 16 (Small Outline IC) tokozásban. Ezen belül található a felületi mikromechanikával készült (3) gyorsulásmérő szenzor, és a mellé szerelt kiértékelő áramköröket tartalmazó (2) chip. A szenzor természetesen itt is tartalmaz egy rugalmasan felfüggesztett szeizmikus tömeget, amely a gyorsulás (lassulás) hatására elmozdul. Az elmozdulást kapacitív úton mérik, fésűszerűen kialakított fegyverzetekkel, amelyeket a következő ábrákon mutatunk be. A szenzort a gépjármű oldalsó részébe, (oldallégsák) és a lökhárító közelében szerelik fel, ahonnan kábelek segítségével fogják vezérelni az értékhatárt túllépő lassulások esetében a légsákokat, pontosabban a légsákok pirotechnikai töltetét vezérlő elektronikus áramköröket.

Egy ilyen felületi mikromechanikai eljárásokkal készült gyorsulásmérő működési elvét mutatja az ábra. A baloldali ábrán nincs gyorsulás, a szeizmikus tömeg nyugalomban van, a C_1 és C_2 kapacitások egyenlők. Vízszintes gyorsulás esetén a rugalmas felfüggesztés következtében a szeizmikus tömeg elmozdul, C_1 kapacitás kisebb, C_2 kapacitás nagyobb lesz, a kapacitív híd egyensúlya felborul. Ennek megmérése a jelfeldolgozó áramkör feladata lesz.



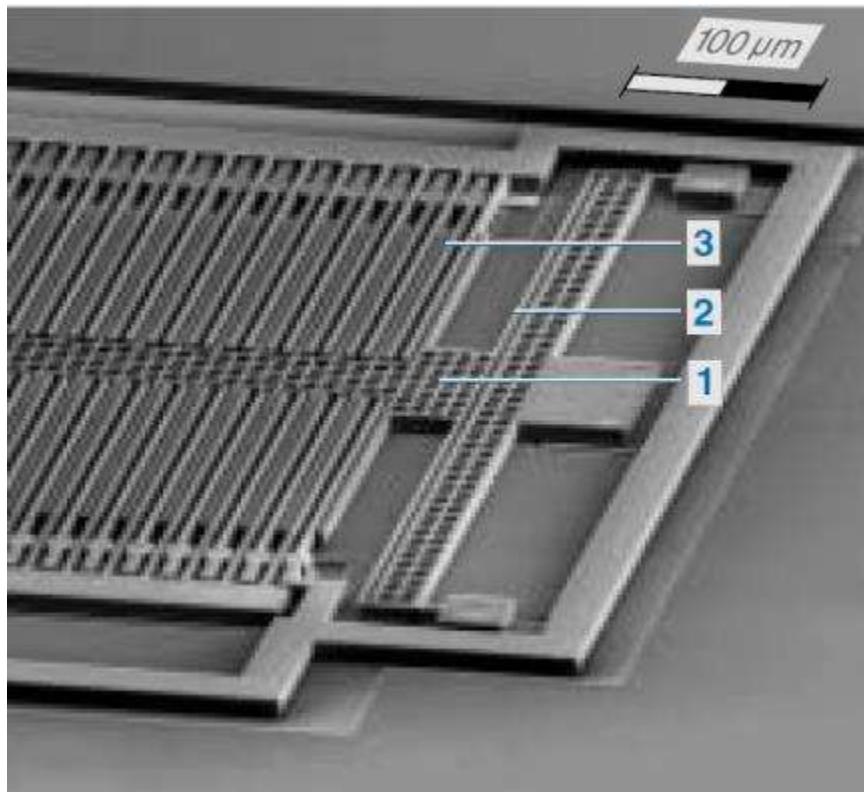
33. ábra A felületi mikromechánikával készült gyorsulásmérő működési elve

A valóságos mikromechanikai struktúra axonometrikus rajzát az ábra mutatja. Az eszköz lényeges eleme az (1) rugóztatott szeizmikus tömeg a mozgó elektródákkal. A felfüggesztés a (2) rugók segítségével történik. A rögzített elektródák (3), és (6) adják a másik két kondenzátor fegyverzetet az Al-vezetőpályák (4) segítségével. A kivezetések (bond-pads) céljára az (5) felületek (pads) állnak rendelkezésre, a (7) SiO₂ réteg a szigetelést szolgálja.



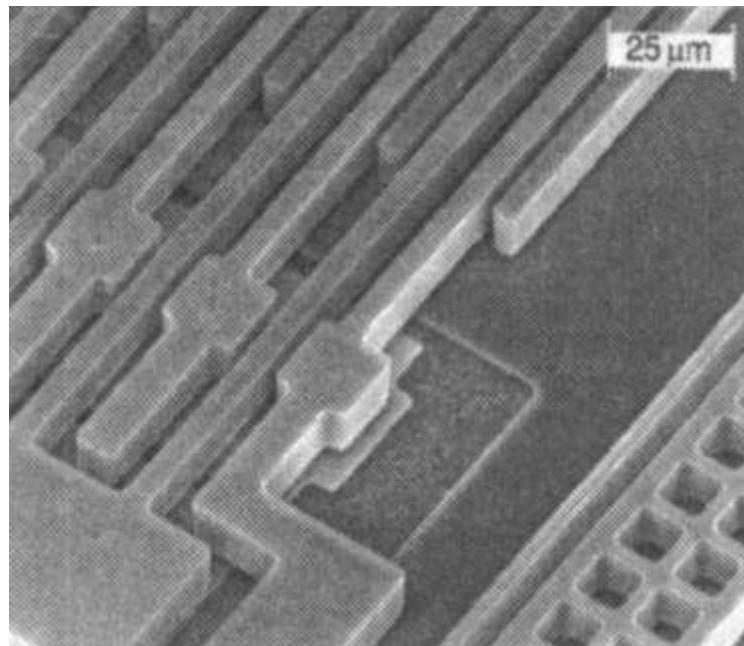
34. ábra Felületi mikromechánikával készült egytengelyes gyorsulásmérő axonometrikus rajza

Az ilyen elven készült gyorsulásmérők mérési tartománya 50-100 g, alsó határfrekvenciájuk akár 0 Hz is lehet (a gravitációs gyorsulást a kapacitív mérési elvnek köszönhetően nyugalmi helyzetben is kimutatják), a felső határfrekvencia a több kHz-es tartományban van. A mikromechanikai szerkezet érhossza 100-500 μm . A gyorsulásmérők kapacitása hozzávetőlegesen 1 pF. Ez igen kis érték, amely miatt a mérő-feldolgozó áramkört a szenzorhoz minél közelebb kell elhelyezni. A gyorsulásmérők gyakran egychipes kivitelűek (one chip design), tehát a kapacitív mérőhidat és jelfeldolgozó áramkört is ugyanazon a Si kristályon alakítják ki, ahol a mikromechanikai szerkezet is van. Az áramkör kompenzációt és öntesztelési lehetőséget is szokott tartalmazni. Elektrosztatikus gerjesztéssel megvizsgálható, hogy a rendszer lengőképes-e, amellyel a szenzor biztonsága és megbízhatósága jelentősen emelhető. Az előbbi mikromechanikai szerkezet elektronmikroszkópos képét az ábra mutatja. Jól látható, hogy a laterális, (síkbeli) méretek sokkal nagyobbak, mint a mélységi méretek, ezért ez egy tipikus felületi mikromechanikai struktúra. Az ábrán (1) a szeizmikus tömeget, (2) a felfüggesztő rugót, és (3) a rögzített elektródákat mutatja.



35. ábra A gyorsulásmérő mikroszkópos képe

Az előbbi ábra kinagyított részletét, a fésűs elektródákat jól láthatóan mutatja az ábra.

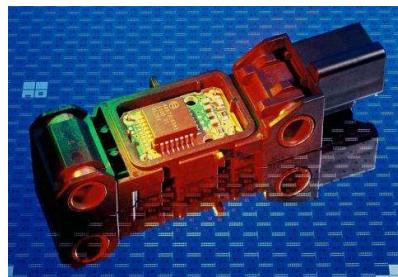


36. ábra A kinagyított fésűs elektródák

Ezek a gyorsulásmérők természetesen csak egy irányban mérnek, kifejezett műszertechnikai feltétel, hogy a keresztirányú érzékenység minél kisebb legyen.

A gyorsulásmérő feldolgozó áramköreit vázlatosan az ábra mutatja. Ezek közül a működéshez legfontosabb az 1 MHz-es oszcillátor, amelynek ellenfázisban érkező jelei az álló elektródákra (fegyverzetekre) kerülnek. A mozgó elektródán megjelenő jelet az impedancia illesztő áramkör után egy szinkron demodulátorra vezetik, ahol a jel amplitúdóján kívül annak az oszcillátorhoz viszonyított fázisát is figyelni kell. Ezután egy erősítőfokozat következik, a kimenet analóg feszültség.

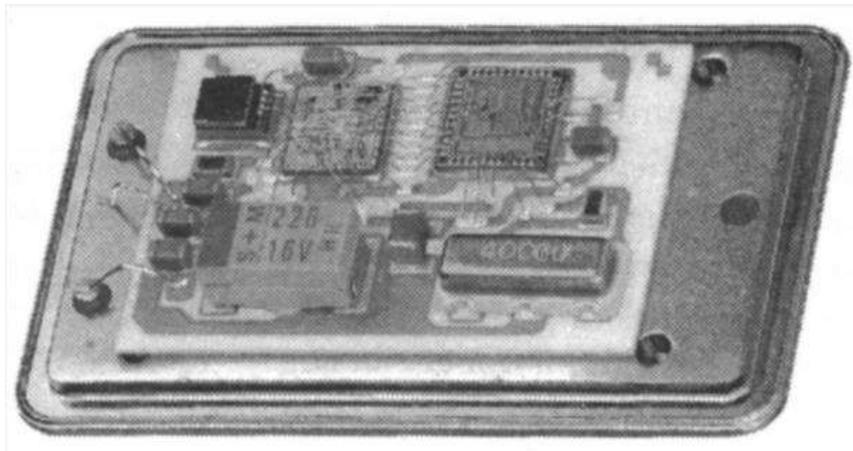
A légzsákok elektronikáját vezérlő gyorsulásmérőket is nagyon robusztus tokozással látják el, érhető okokból, hiszen ezeknek az ütközéskor is helyesen kell működniük. Egy oldallégzsákhoz kifejlesztett mikromechanikai gyorsulásmérő képét mutatja az ábra.



37. ábra Oldal légzsákok vezérlésére kifejlesztett gyorsulásmérő képe

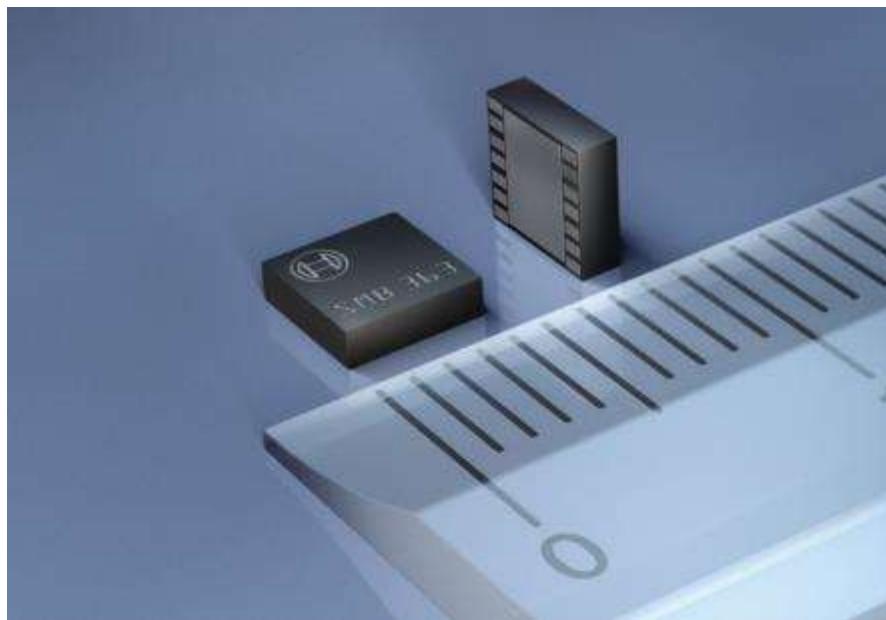
Az ábrán a Bosch korai, PAS 2 (100g) típusú gyorsulás szenzorát mutatja, tokozás előtt. Látható, hogy a fém tokban kerámia hordozón helyezkednek el a mikromechanikai és a

mikroelektronikai chippek, sőt itt még az SMD (Surface Mounting Device) kondenzátorok is, a chippek mikrohuzalokkal csatlakoznak a hordozón kialakított vezetékekhez. A kész modulra egy fém sapkát illesztenek, amelyet a széleken körbe vagy ellenállás, vagy elektronsugaras hegesztéssel zárnak le.



38. ábra A Bosch PAS 2 gyorsulásmérőjének képe tokozás előtt

A jelenlegi (2010-es évek eleje) technológiai szintet a Bosch SMB363 gyorsulásmérője képviseli, amelynek képe az ábrán látható.

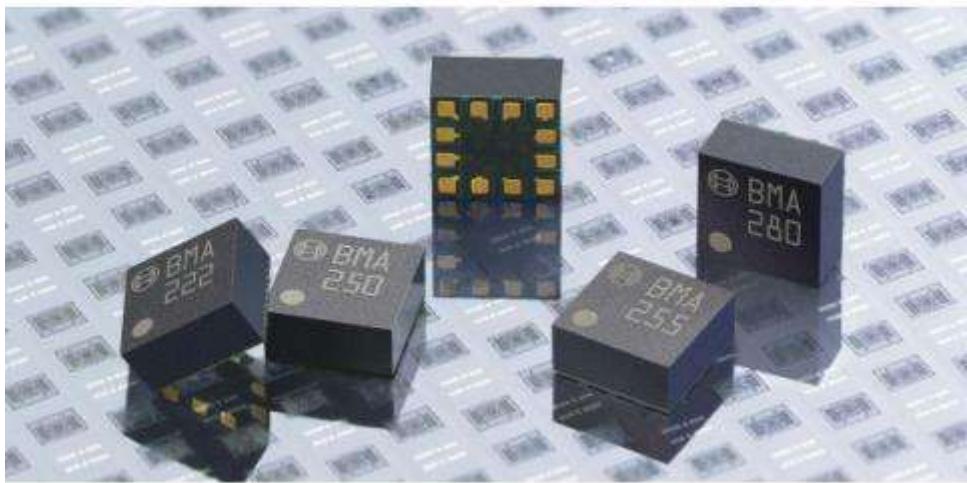


39. ábra A Bosch SMB363 gyorsulásmérőinek képe

Ez a gyorsulásmérő három egymásra merőleges tengely mentén méri a gyorsulást, így lehetőség van a gravitációs gyorsulás abszolút értékének és a szenzorhoz viszonyított irányának meghatározására is. Az eszköz két chipet tartalmaz: egy mikromechanikai szenzoregységet, és egy feldolgozó elektronikus egységet. A mérés elve: differenciális kapacitív mérőhíd. A gyorsulásmérő három független analóg kimenettel rendelkezik. A méréstartomány $\pm 2g$, az

érzékenység minden a három tengelyre $S = V_{DD}/5/1g$, azaz 1 g gyorsulásra a tápfeszültség ötödrésze. A tipikus tápfeszültség $V_{DD} = 2,5$ V. Megjegyezzük, hogy az előzőekben a g nem grammot, hanem a gravitációs gyorsulást jelenti, az mg ennek ezredrészét, a μg a nehézségi gyorsulás milliomodrészét jelenti. Mivel a gyorsulás vektoros mennyisége, fontos műszertechnikai követelmény, hogy az egyes tengelyek mentén mérő gyorsulásmérők a tengelyekre merőleges irányú gyorsulásokra érzéketlenek (vagy legalábbis elhanyagolható mértékben érzékenyek) legyenek. Ez a fogalom a keresztirányú érzékenység, amely ennél a gyorsulásmérőnél $\pm 0,2\%$ a végkitérésre vonatkoztatva. Ha egy eszköznek nagy a keresztirányú érzékenysége, lehetetlen pontosan meghatározni egy ismeretlen térbeli gyorsulás irányát, ami a robottechnikában sokszor nagyon hátrányos lenne. A gyorsulásmérő teljes öntesztelési lehetőséggel rendelkezik, ez a funkció a hagyományos gyorsulásmérőknél nem létezett. Zérus gyorsulásnál a kimeneti feszültség a tápfeszültség fele, amely működés közben a gyorsulás előjelétől függően csökken, vagy nő. Az áramfelvétel rendkívül kicsi, normál üzemben 0,2 mA, készenléti (standby) üzemmódban 1 μA . A készenléti üzemmódból a szenzor 1 ms alatt képes normál üzemmódba átkapcsolni. A gyorsulásmérő sávszélessége kb. 1 kHz, tokozása megfelel a legkorszerűbb felületszerelési technológiának (QFN package, quad flat no lead), a mérete 4x4x1,2 mm. A felhasználó számára az is fontos, hogy a szenzort nem kell utólag kalibrálni, a kimenet gyárilag kalibrálva van. A működési hőmérséklettartomány -40 °C-tól + 65 °C-ig terjed, a zérus gyorsuláshoz tartozó hőmérsékleti drift ± 2 mg/°K. Az eszköz nonlinearitása a végkitérésre vonatkoztatva $\pm 0,5\%$, ami a szenzortechnikában igen jó értéknek számít. A gyártás a szilícium felületi mikrotechnológián alapul, a gyártott darabszám több, mint 100 millió. Összegezve: a bemutatott gyorsulásmérőt nagymértékű integrációs fok jellemzi, párosulva a kiváló műszaki jellemzőkkal, ezekkel is jól példázva a MEMS-ek előnyös (más módon el nem érhető) tulajdonságait.

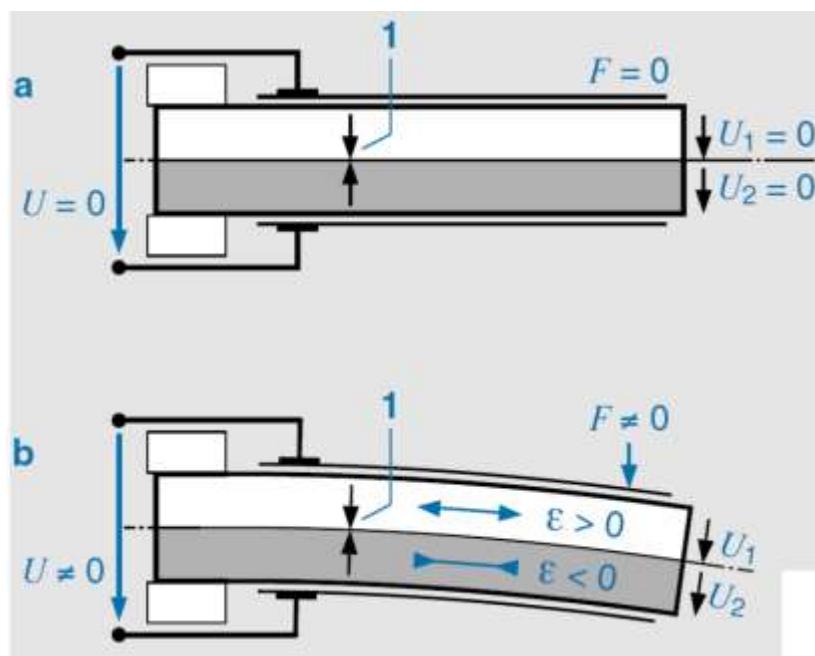
A legújabb fejlesztésű 3 tengelyes gyorsulásmérő családot Bosch BMA 250, 255, 280, 355 típusok, különböző (2 g, 4g, 8g és 16 g) méréshatárokkal, és digitális kimenettel, 10-14 bites feloldással az ábra mutatja. A méretekre jellemző, hogy például a BMA 280 mindössze 2x2x0,95 mm méretű. A gyorsulásmérőket mobil telefonokba és egyéb kereskedelmi elektronikus berendezésekbe ajánlják. A mikromechanikai gyorsulásmérők mérési elve: kapacitív.



40. ábra A Bosch BMA típusú gyorsulásmérőinek képe

3.3.2.2. Piezzo elektromos gyorsulásmérők

Gyorsulásmérő szenzorokat piezoelektronos működési elven is lehet készíteni, sőt, régebben a gyorsulásmérőket szinte kizárolag ezen az elven készítették. A piezoelektronos effektust használó eszközök általános tulajdonsága, hogy mivel működésük a töltés-szétválasztás elvén alapul, a keletkezett töltések előbb-utóbb elfogynak, emiatt alsó határfrekvenciájuk soha nem lehet zérus. A másik fontos tulajdonság, hogy a nagy piezoelektronos effektust mutató anyagoknál a környezeti hatások befolyása (hőmérséklet, légnedvesség) meglehetősen erős. Viszont azoknál az anyagoknál, amelyeknél a piezoelektronos állandó értéke kicsi, a környezeti hatások (például a hőmérséklet) befolyása is kicsi. Ilyen anyag például a monokristályos kvarc, és ezért készül kvarcból a legtöbb piezoelektronos szenzor. Aktuátorokban viszont inkább a pontatlanabb, környezeti változásokra érzékenyebb, de nagy effektust mutató kerámia anyagokat használják fel.

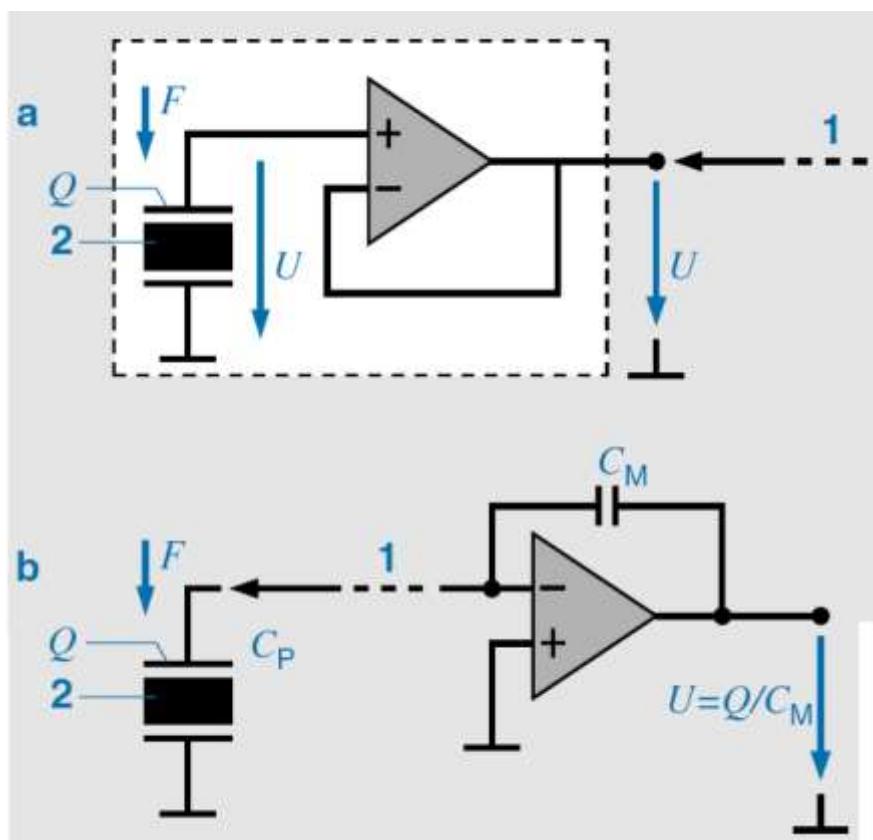


41. ábra A piezoelektronos mérési elv

A mikrotechnikában alkalmazott mérési elvet az ábra mutatja. A szenzor lényege két polaritás helyesen összeragasztott, és egy oldalon befogott piezo elem, amelyet a fellépő gyorsulás hatására fellépő F erő hajlításra kényszerít. A szeizmikus tömeg itt maga a piezo elem. A mechanikai feszültség hatására töltésszétválasztás történik, amelyből villamos feszültséget is (áramot is) könnyű előállítani. A feszültség a töltéstől és a piezo elem saját kapacitásától függ, a jól ismert $q=C \cdot U$ összefüggés alapján. A C kapacitás elsősorban a piezo elem saját kapacitását jelenti, azonban ehhez még hozzá kell számolni a járulékos (feldolgozó áramkör bemeneti kapacitása, kábelkapacitás, stb.) kapacitásokat is. Fontos megjegyezni, hogy a piezo

átalakítók áramkörileg kapacitív forrással, illetve az aktuátorok kapacitív terheléssel modellezhetők, mert a piezo anyagok szigetelők, amelyeknek dielektromos állandójuk van, a töltésekkel pedig fém fegyverzetek gyűjtik össze, amelyeknek felülete van, tehát akarva-akaratlanul a piezo tulajdonság mellett kondenzátort is alkotnak.

A piezoelektronos elven működő szenzorok mérésére két lehetőség kínálkozik. Az egyik a nagyon nagy bemeneti ellenállással rendelkező ún. elektrométer erősítő. Ezek bemeneti ellenállása $10^{16} \Omega$ (100 teraohm) körül van. A nagy bemeneti ellenállás azért szükséges, hogy a nagy bemeneti ellenállás miatt a bemeneti áram minél kisebb legyen, és minél kisebb mértékben fogyassza el azt a töltésmennyiséget, amely a piezo effektus következtében rendelkezésre áll. Más szavakkal kifejezve: minél nagyobb legyen az áramkör időállandója.



42. ábra A piezoelektronos szenzorok jelfeldolgozása. a: elektrométer erősítő, b: töltéserősítő

A másik módszer a piezo effektus során keletkezett töltések mérésére a töltéserősítő, amelynek vázlatát a b ábra mutatja. Ennek a mérési módszernek óriási előnye, hogy a műveleti erősítő invertáló bemenetén gyakorlatilag nulla a feszültség, tehát a piezo elem fegyverzetei között nincs feszültség, amely a töltések kiegyenlítésének az irányában hat. Az áramkör helyes működéséhez az kell, hogy a műveleti erősítő bemeneti árama minél kisebb legyen. A kimeneti feszültséget a C_M visszacsatoló ágban lévő kondenzátor kapacitása határozza meg. (Az

elektrométeres erősítő feszültségerősítése egységnyi.) A piezoelektromos átalakítókkal elvileg sztatikus jeleket nem lehet mérni, ezért az alsó határfrekvencia 1...10 Hz között szokott lenni.

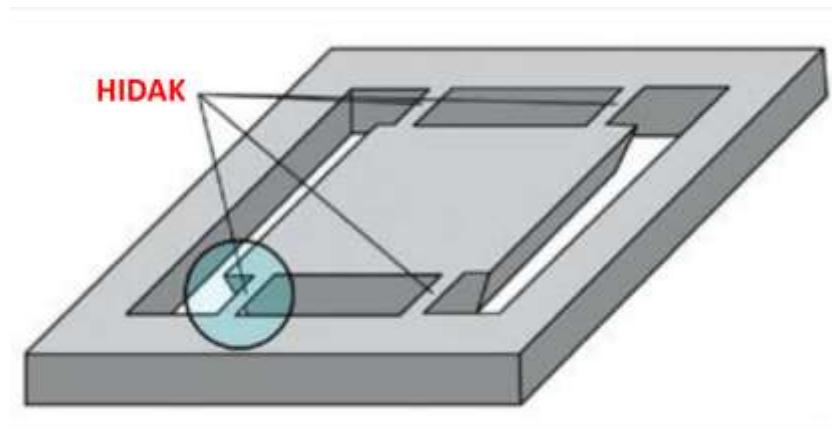


43. ábra A Bosch cég különböző gyorsulásmérői

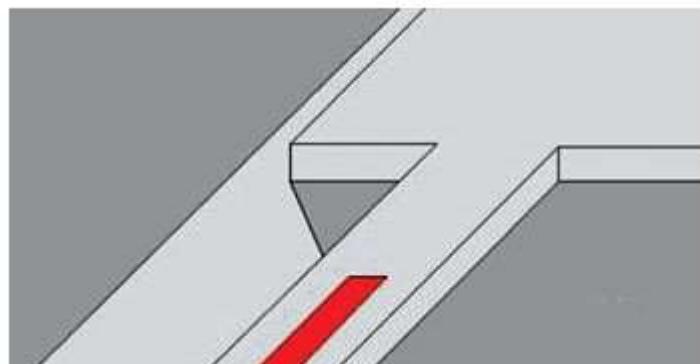
Az ábrán az autóipari gyorsulásmérők tekintetében vezető szerepet játszó Bosch cég különböző generációs gyorsulásmérőit láthatjuk. Az ábra bal oldalán a hibrid szerelésű piezoelektromos gyorsulásmérő látható, amelyet a cég 1984-1997 között gyártott. Középen az 1997-2002 között gyártott, szilícium mikromechanikával készített, kapacitív mérési elvet megvalósító két chip-es gyorsulásmérő látható. Jobb oldalon ennek továbbfejlesztett, 2002-től gyártott változata látható, amely ugyanúgy mikromechanikával készült, és kapacitív mérési elven működik.

3.3.2.3. Gyorsulásmérők nyúlászmérő bélyegek alkalmazásával

Mikromechanikai gyorsulásmérők nyúlászmérő bélyegek alkalmazásával is készíthetők. Ilyenkor a szeizmikus tömeget legtöbbször négy rugalmas elemmel függessztik fel, és a rugalmas elemekre piezorezisztorokat integrálnak, a mikromechanikai kialakítást az ábra, a nyúlászmérő bélyegek felfüggesztésre integrálását az ábra mutatja.



44. ábra Nyúlászmérő bélyegekkel működő gyorsulásmérő

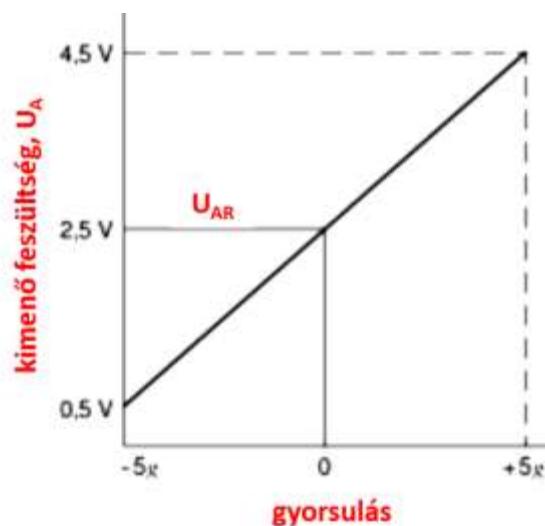


45. ábra A nyúlászmérő bélyegek kialakítása

A bélyegeknél elsősorban a piezorezisztivitást használják fel, tehát ezekben az esetekben a tenzometrikus hatás gyakorlatilag elhanyagolható lesz. A technológia: tömbi mikromechanika, ezen belül is a legjellemzőbb a Si anizotróp marása. A piezorezisztorok ellenállás változásának mérési elve: a jól ismert Wheatstone-híd, a hozzá tartozó teljes jelfeldolgozó elektronikával (vagy egyenfeszültségű, vagy vivőfrekvenciás mérőerősítő, kalibrálás, esetleg A/D átalakítás). A gyorsulásmérők egyik fontos alkalmazási területe az autóiparban a passzív biztonságot szolgáló légszárkok vezérlésére szolgáló gyorsulásmérő. A fejlesztésnek itt is több generációja ismeretes, példaként az ábrán egy tokozott gyorsulásmérő képét mutatjuk be.



46. ábra Oldallégsák vezérlésére kifejlesztett gyorsulásmérő képe

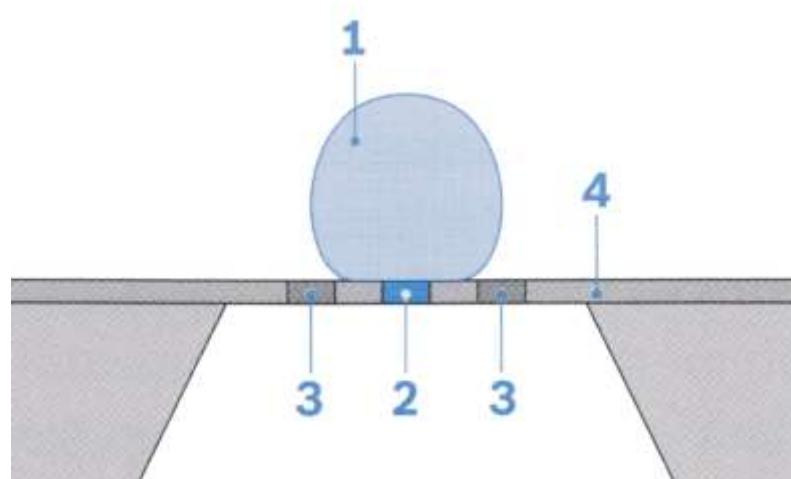


47. ábra Analóg gyorsulásmérő karakterisztikája

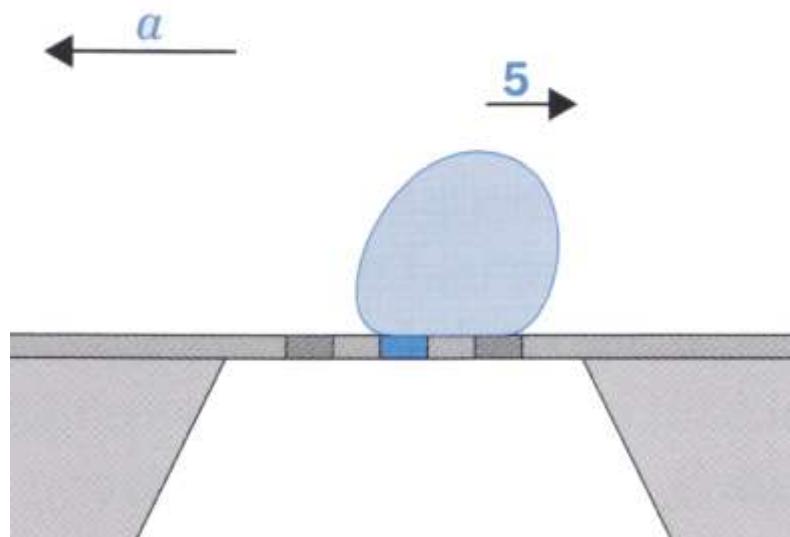
A gyorsulásmérő szenzor névleges mérési tartománya $\pm 5\text{ g}$, tápfeszültsége 5 V, és zérus gyorsulásnál a kimeneti feszültség 2,5 V. Pozitív gyorsulásnál ($+5\text{ g}$) a kimeneti feszültség 4,5 V-ra nő, míg ellenkező irányú gyorsulásnál 0,5 V-ra csökken. Ezt láthatjuk az ábra a pozitív-negatív gyorsulások függvényében.

3.3.2.4. Termodinamikai gyorsulásmérők

A gyorsulásmérők egy újabb generációját képviselik a termodinamikai elven felépített gyorsulásmérők. Az alapvet az ábra mutatja. Ennek lényege, hogy egy szűk térben egy gázbuborékot (1) melegítenek fel, a (2) fűtőtest segítségével. A felmelegített gázbuboréknak lesz egy helytől függő hőmérséklet eloszlása, a hőmérsékletet a szimmetrikusan elhelyezett (3) hőmérséklet érzékelőkkel mérni lehet. Ez az eloszlás abban az esetben szimmetrikus lesz, ha nincs gyorsulás. Ha van laterális gyorsulás, a felmelegített gáz elmozdul, a szimmetria felborul, és a hőmérséklet eloszlása megváltozik. Ez a változás alkalmasan elhelyezett hőmérséklet érzékelőkkel jól detektálható. A gyorsulásmérő mikromechanikai kivitelű, igen kis méretekkel, és igen kis fűtőteljesítményekről van szó.



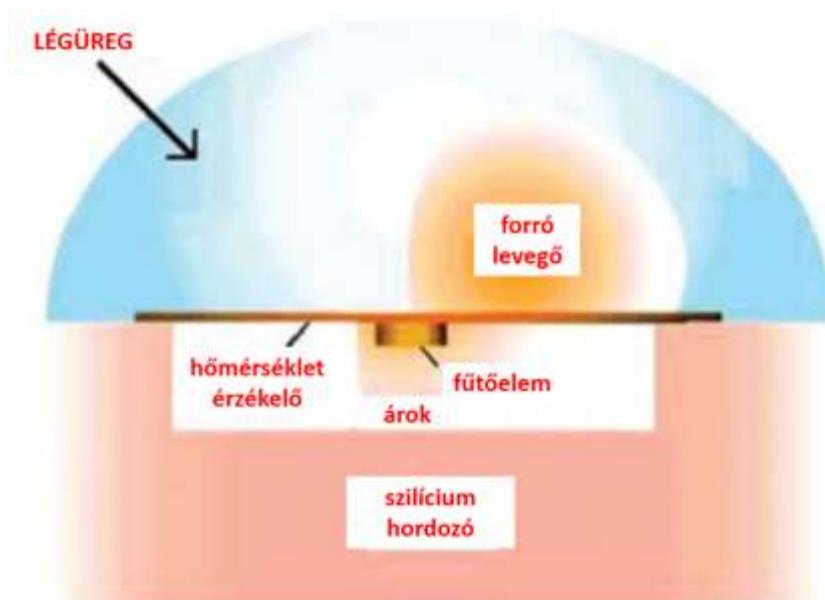
48. ábra Nyugalmi helyzetben a felmelegített rész szimmetrikusan helyezkedik el



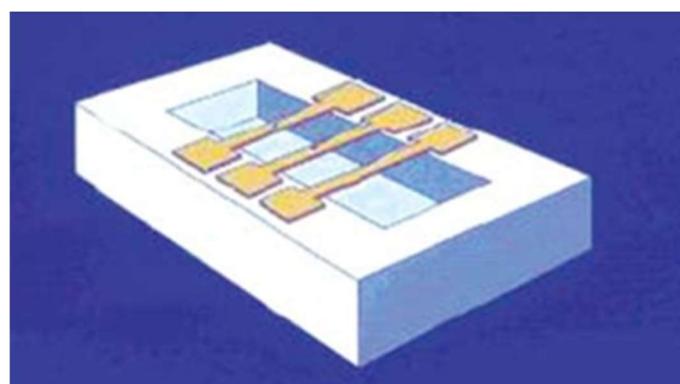
49. ábra Gyorsulás esetén a szimmetria felborul



50. ábra Nyugalmi állapot, amikor a rendszerre nem hat gyorsulás

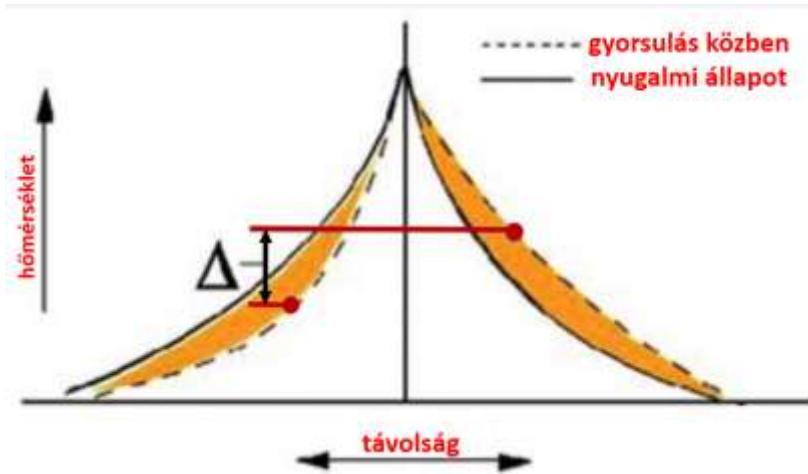


51. ábra A szenzorra gyorsulás hat vízszintesen és bal irányból



52. ábra A fűtőtest és a hőmérők mérő szenzorok mikromechanikai kialakítása

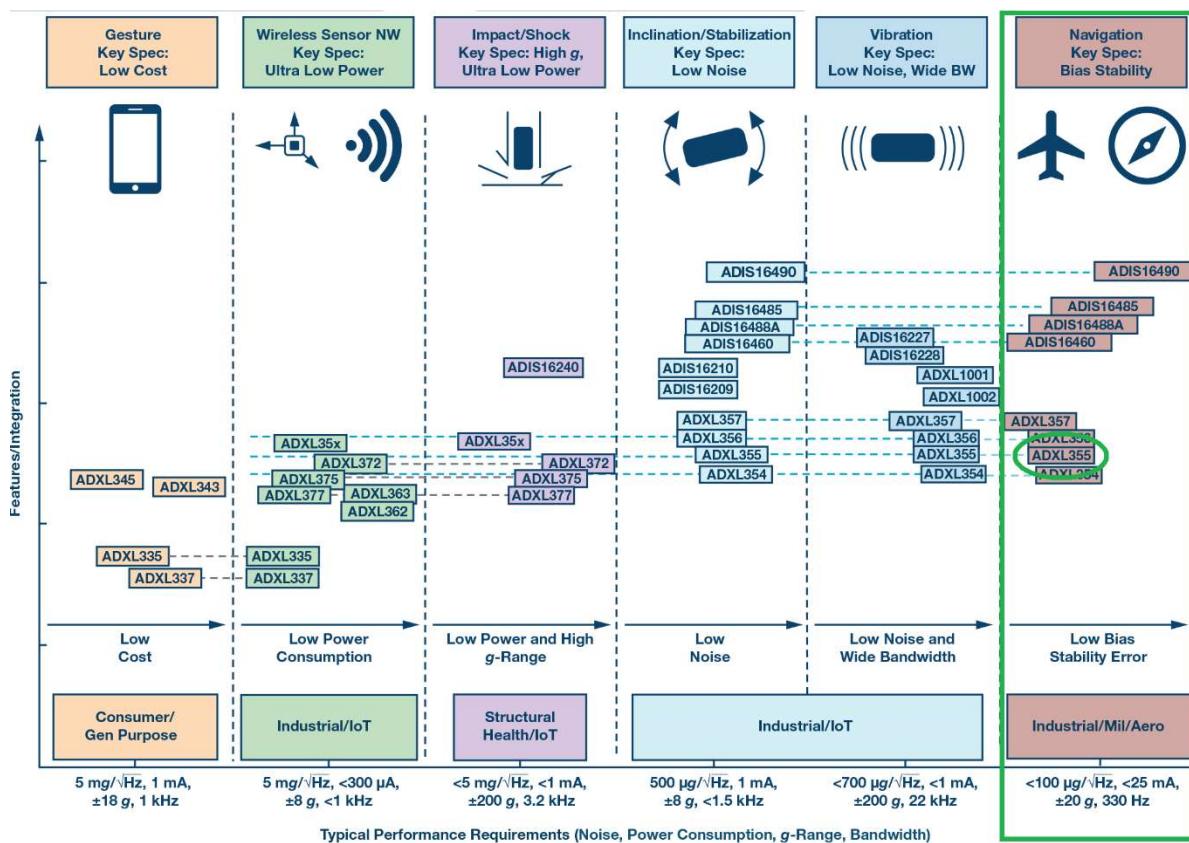
A termikus gyorsulásmérő valóságos kialakítását az ábra mutatja.



53. ábra Gyorsulás esetén a hőmérőkleti eloszlás megváltozik

A fűtőelem mellett kettő, vagy kéttengelyű szenzor esetében négy hőmérséklet érzékelő van. Ha a rendszert gyorsulás éri, az egyensúly felborul (ezt mutatja az 5.74. ábra), ennek mértéke pedig arányos lesz a gyorsulással. Az ilyen elven működő szenzorok érzékenysége az 1 mg-os tartományban van, ütésállósága rendkívüli, hozzávetőlegesen 50 000 g, és határfrekvencia 100 Hz körül van.

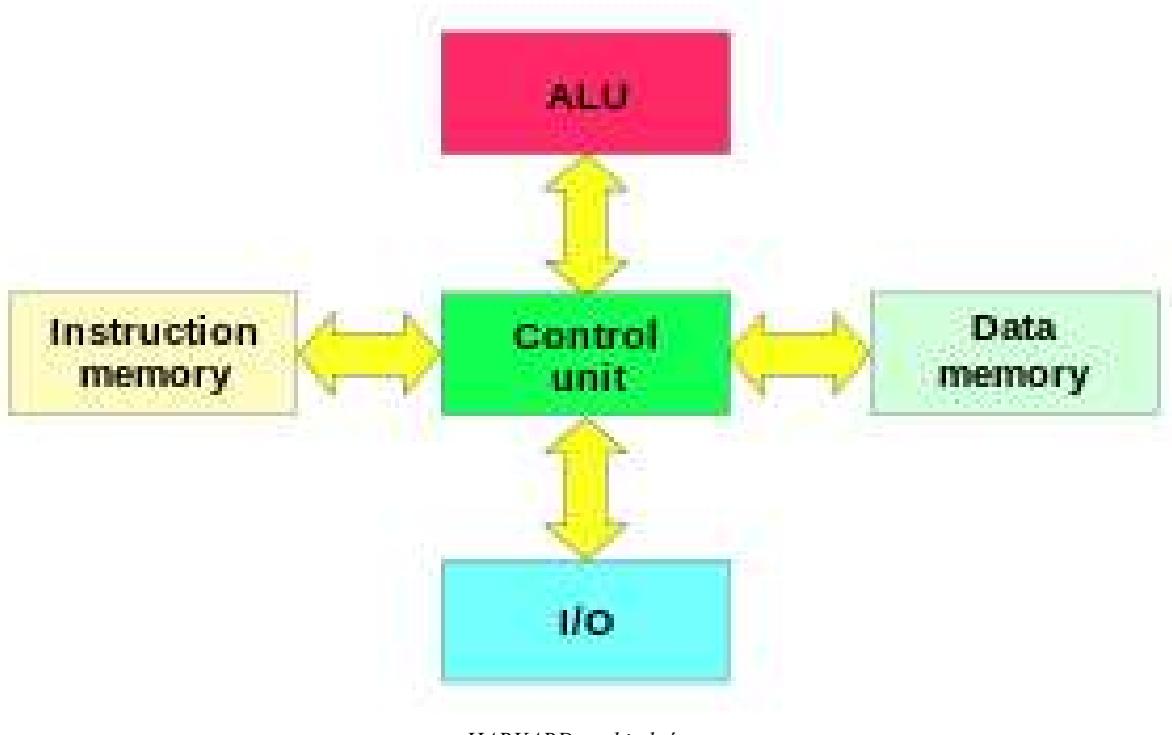
A lehető leg pontosabb alkatrész kiválasztására törekedtünk. Az Analog Devices ADXL3x5 termékcsaládján belül nézelődtünk. Az ábra alapján választottunk a lehetőségek közül. Mivel a MOVIX vezérlőben az egyik legmegalakíthatóbb, leg pontosabb és kifejezetten ipari navigációs célra ajánlott digitális kimenetet adó ADXL355 szenzor található, ezért választottuk fővezérlőnknek. Látható az ábrán, hogy a beépített ADXL355 nagyon alacsony bias és stabilitási hibával rendelkezik.



54. ábra Gyorsulásmérő szenzorok felhasználás szerinti csoportosítása

3.4. Feldolgozó egységek

III. Központi vezérlő egység



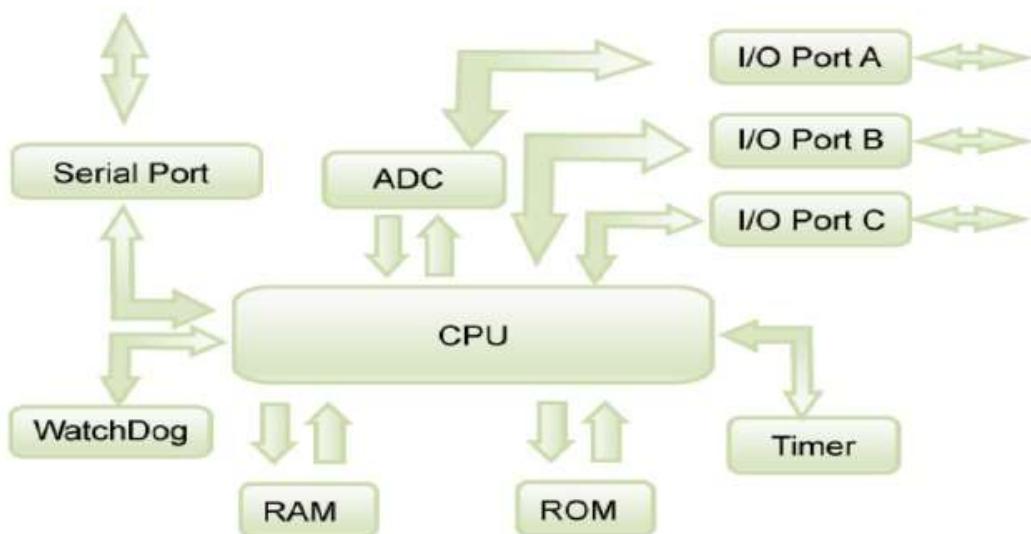
A vezérlő egységek olyan alegységei egy rendszernek, amelyek választott protokollon keresztül küldenek utasításokat, fogadnak adatokat egy vagy több gateway irányából, irányába. A vezérlőegységek alapvetően Harvard vagy Neumann architektúrával rendelkező számítógépek lehetnek. Jellemzően a hatékonyabb Harvard architektúra (fentebbi ábra) terjedt el, melyben a program és az adatmemória két fizikailag külön tárolóban található és ezért két külön buszon érhető el.

A központi vezérlőegység lehet a rendszeren belül egy asztali PC, FPGA alapú vezérlőegység, mikrokontroller, DSP. AZ FPGA-k költségesek, ugyanakkor alacsonyabb fogyasztás mellett a legmagasabb számítási teljesítményt tudják nyújtani párhuzamos architektúrájuk miatt. A DSP szintén nagyobb költségű eszköz, ez azonban az FPGA-val ellentétben nem újra konfigurálható, de jelentős számú számítóegységgel működik. Végül a hagyományos beágyazott rendszerek java részének központi elemét képező mikrovezérlő mellett döntöttünk, ugyanis az megfelelően alacsony fogyasztás mellett, a számunkra szükséges

számítási teljesítményt biztosítani tudja, kezelhetősége is megfelelő számunkra, méretei kicsik, költség és helytakarékos megoldás.

General architecture of a Microcontroller

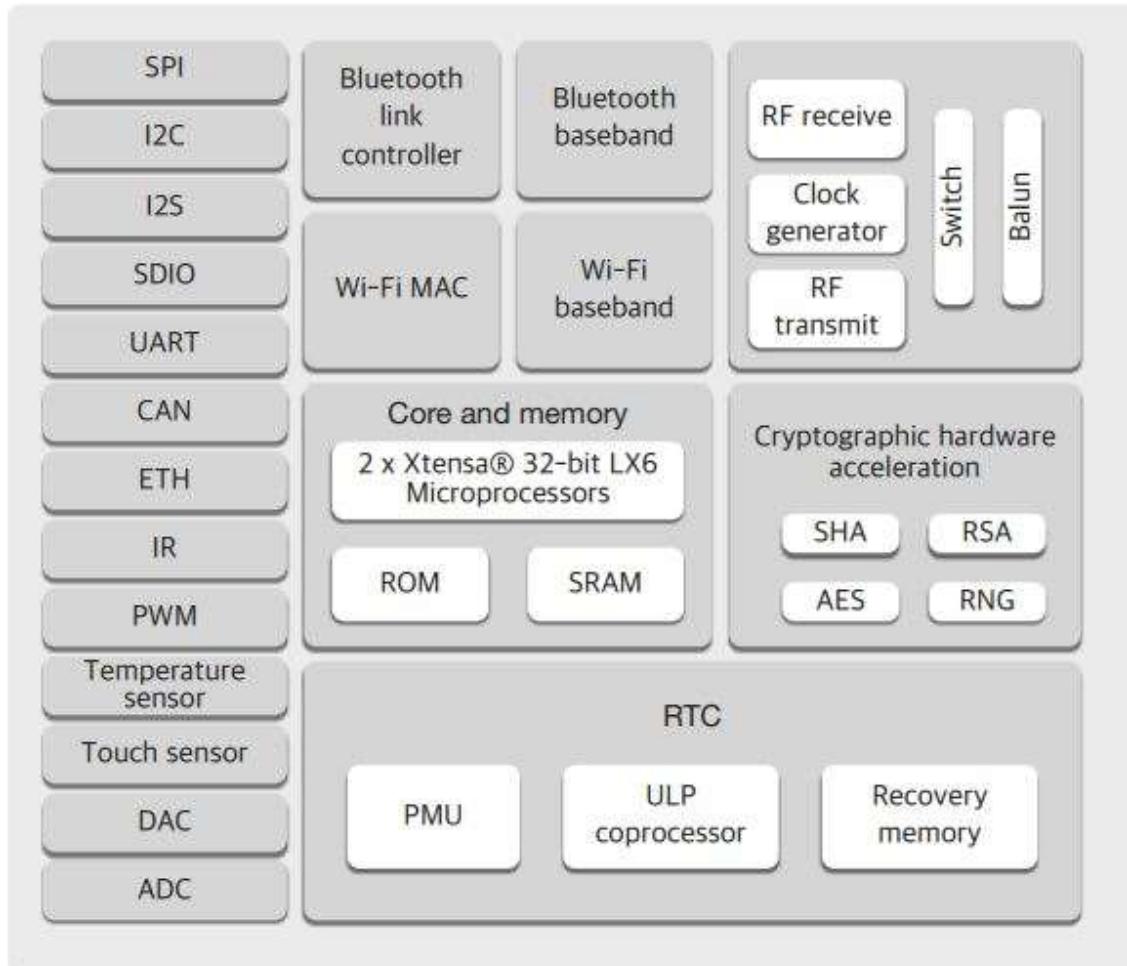
The architecture of a microcontroller depends on the application it is built for. For example, some designs include usage of more than one RAM, ROM and I/O functionality integrated into the package.



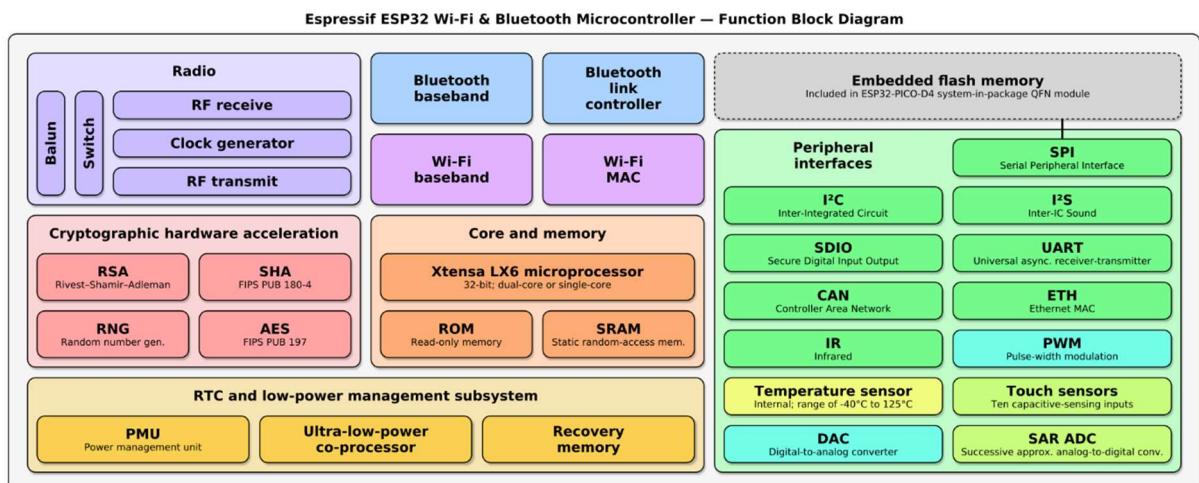
55. ábra Mikrokontrollerek általános felépítése

Központi vezérlő egységnek az AIRobotix Kft. MOVIX™ vezérlőjét választottuk, ugyanis ebben van beépített nagy pontosságú gyorsulásmérő (ADXL355). A vezérlő ezen felül egy ESP32 alapú WiFi/Bluetooth modult is tartalmaz, amellyel az eszköz képes hálózaton kereszttüli vezetéknélküli kommunikációra más gépekkel. Képes azok felé adatokat továbbítani, valamint akár további perifériákat is vezérelni. Ezeken felül a MOVIX™ rendelkezik UART kivezetéssel, így csatlakoztathatók hozzá a RTE-M88 lézeres távolságmérő szenzorok. Egy UART kivezetés található a MOVIX™ eszközön, ugyanakkor az RTE-M88 szenzorok egyedi címmel láthatók el, így ez nem okoz gondot és a kommunikáció busz túlterhelését és áthallást.

A MOVIX egy Espressif ESP32 mikroprocesszor köré épült, amely blokkvázlata az alábbi ábrán látható.



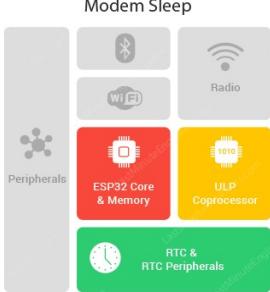
56. ábra ESP32 Blokk diagramm



57. ábra ESP32 funkciók

A MOVIX™ programozása az ESP-IDF nevű IDE programon keresztül történhet soros port segítségével, valamint az AIRobotix Kft. által biztosított PROGIX™ programozóval, amelyet elég a PC-hez kapcsolni és már is programozható a MOVIX™ C, de akár Python nyelven is.

A MOVIX processzora és WiFi modulja -40°C és +125°C között üzemképes, áram felvétele az egyes táblázatban látható.

1. táblázat	
Active Mode  Active: <ul style="list-style-type: none"> WiFi Bluetooth Radio ESP32 Core ULP Coprocessor Peripherals RTC Power Consumption 160~260mA	Modem Sleep  Active: <ul style="list-style-type: none"> ESP32 Core & Memory ULP Coprocessor RTC Inactive: <ul style="list-style-type: none"> WiFi Bluetooth Radio Peripherals Power Consumption 3~20mA
Light Sleep  Active: <ul style="list-style-type: none"> ULP Coprocessor RTC Inactive: <ul style="list-style-type: none"> WiFi Bluetooth Radio Peripherals Paused: <ul style="list-style-type: none"> ESP32 Core Power Consumption 0.8mA	Deep Sleep  Active: <ul style="list-style-type: none"> ESP32 Core & Memory ULP Coprocessor RTC Inactive: <ul style="list-style-type: none"> WiFi Bluetooth Radio Peripherals Power Consumption 10µA
Hibernation  Active: <ul style="list-style-type: none"> RTC Inactive: <ul style="list-style-type: none"> ESP32 Core & Memory ULP Coprocessor WiFi Bluetooth Radio Peripherals Power Consumption 2.5µA	

Az egyedileg fejlesztett MOVIX áramkör egy olyan nyomtatott áramköri lap, amely képes WiFi-n keresztül kommunikálni a külvilággal, így akár egy ipari PC rendszerrel is.

A modul áramkörön elhelyezkedik egy központi vezérlő egység, amely egyben PCB alapú WiFi antennával ellátott wifi modul is. A felhasznált eszköz az Espressif által gyártott ESP32-es mikrovezérlő, amelyben kétmagos Extensa típusú processzor található. A processzor SPI és Uart interfészei vannak felhasználva. Az SPI busz a nyomtatott áramköri lapon belül közti össze a processzort az ADXL355 típusú gyorsulásmérő szenzorral, az UART0 kivezetésen keresztül a processzor programozása történik ESP-IDF környezeten keresztül C vagy Micropython nyelveken. Az UART2 kivezetés az M88-as nagy pontosságú lézer szenzorok csatlakoztatására használhatóak fel, összesen négy szenzor csatlakozására van lehetőség, a három ortogonális tengely irányában és egy plusz párhuzamos irány mely tetszőlegesen kiválasztható annak függvényében, melyik tengely (x, y, z) mentén van szükség hosszabb belátási tartományra.

Az SPI interfész nem kerül kivezetésre, ahogy az 4. ábrán is látszik. Az SPI interfész az ADXL355 típusú szenzorral kommunikál. Ez utóbbi beforrasztva x-y-z irányba forgatva helyezkedik el a NYÁK lapon. Ezt a beállítást kell figyelembe venni a MOVIX mechanikai elhelyezésénél, felhasználásánál. Az 1. ábrán látható a MOVIX áramkör fentebb leírt logikai felépítése és működési blokkvázlata.

A részletesebb áramköri kapcsolási rajz (schematic) az 59-60. ábrákon láthatók. A MOVIX termék és annak hardware információi bizalmas ipari titkok, melyek a AIRobotix Kft tulajdonát képezik. Az AIRobotix Kft csak ezen dokumentáció keretein belül bocsátott rendelkezésre ezen kapcsolási rajzokat és műszaki dokumentumokat, melyek harmadik félnek nem kiadhatók.

A processzor és a szenzorok is egy TVL75733 típusú feszültségszabályozó kimenetéről kapnak tápot, ez 3V3-as szint. A szabályozót egy hagyományos csúszó kapcsolóval lehet tár alá helyezni és így bekapcsolni a MOVIX-ot. A D4 DIP switch furatszerelt kapcsoló a felhúzó ellenállásokat hivatott rákapcsolni az UART vonalakra, ezek szükségesek az M88 lézerszenzorokkal való kommunikációhoz az UART2-es csatornán. Fontos, hogy programozáshoz nincs szükség ezekre az ellenállásokra, így programozó módban (mikor az UART0 használt) off állásba kell kapcsolni ezt a DIP switch kapcsoló mind a négy kapcsolóját. Kivezetésre került továbbá a 26, 27, 32 és 34 IO láb, valamint az A/D konverter és az I2C protokoll lábak is. Ezekben keresztül addcionális szenzorokat és/vagy aktuátorokat lehet

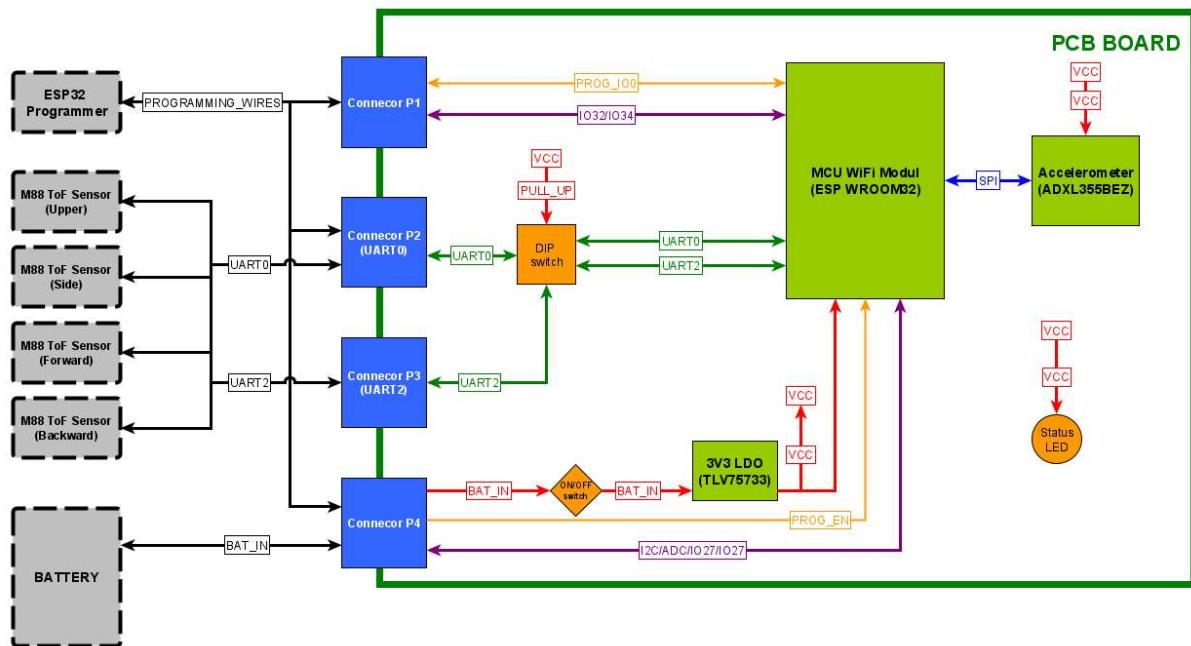
kapcsolni a MOVIX-hoz tetszőleges módon. A beérkező BAT_IN feszültségszintből a feszültségszabályozó 3.3 voltos szintet készít, amely stabil létrejöttét a 'Status LED' felvillanása és folyamatos világítása jelzi.

Az UART-programozót, mellyel a C-ben vagy Pythonban megírt firmware-t rá lehet tölteni a processzorra a P1, P2 és P4 csatlakozókhöz kell kapcsolni, miközben a másik oldala a microUSB csatlakozón keresztül kapcsolódik a PC-hez. A PC-n kiválasztjuk a megfelelő COM port-ot amelyen elérhető a programozó FTDI áramkör és az ESP-IDF környezet segítségével flasheljük a megírt programot.

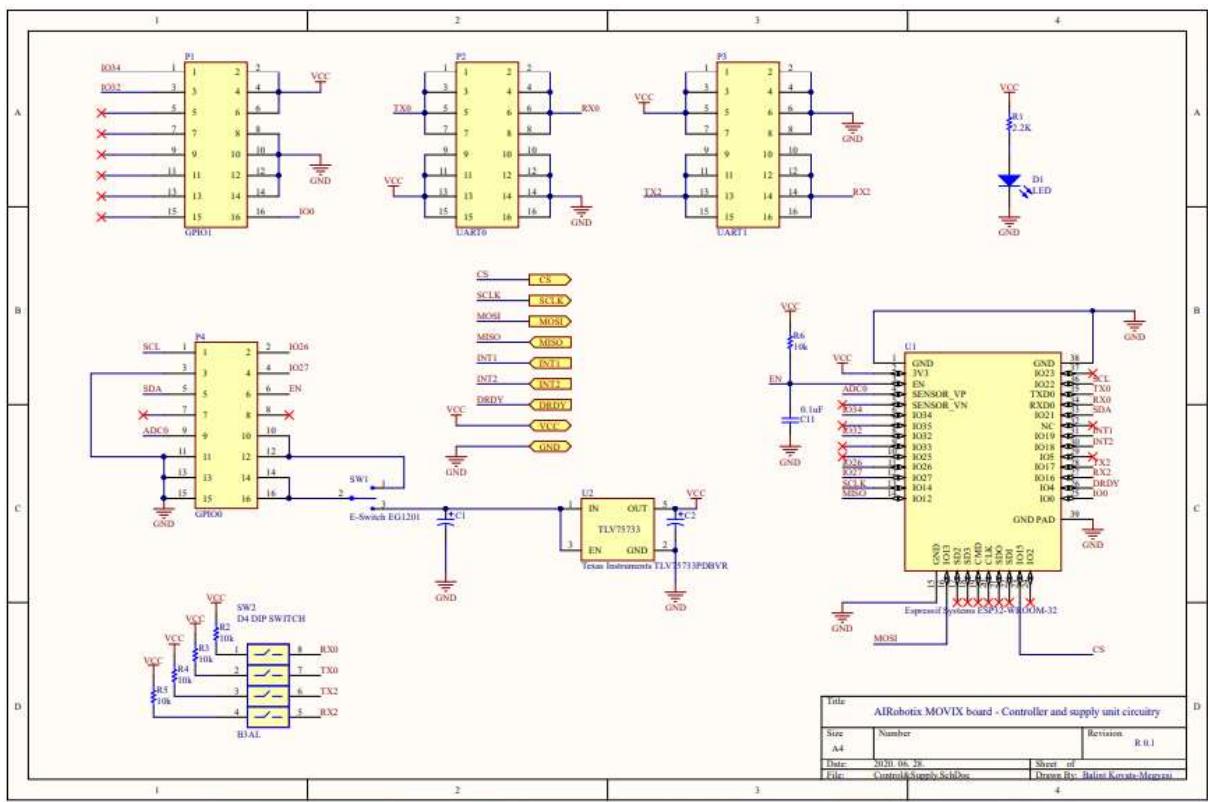
Az M88-as lézer távolságmérők az UART2-es csatornán keresztül kommunikálhatnak a processzorral. Ennek részletes driver programozásáról az 1. Appendix ír részletesen.

A P4-es konnektoron keresztül kapcsolódik a 3.7V-os Li-PO akkumulátor, ehhez az ajánlott modell a PKCELL ICR18650 6600mAh 3.7V (link), amely egy töltéssel hozzávetőleg 11-12 órás folyamatos üzemet képes energiával ellátni: feltételezve azt, hogy napi 10 mérést és WiFi kommunikációt kell végeznie, mely menetek egyenként 30 percesek, akkor becslésünk alapján az akkumulátor egy töltéssel 2 teljes üzemnapot képes végig dolgozni, mielőtt újra tölteni kellene.

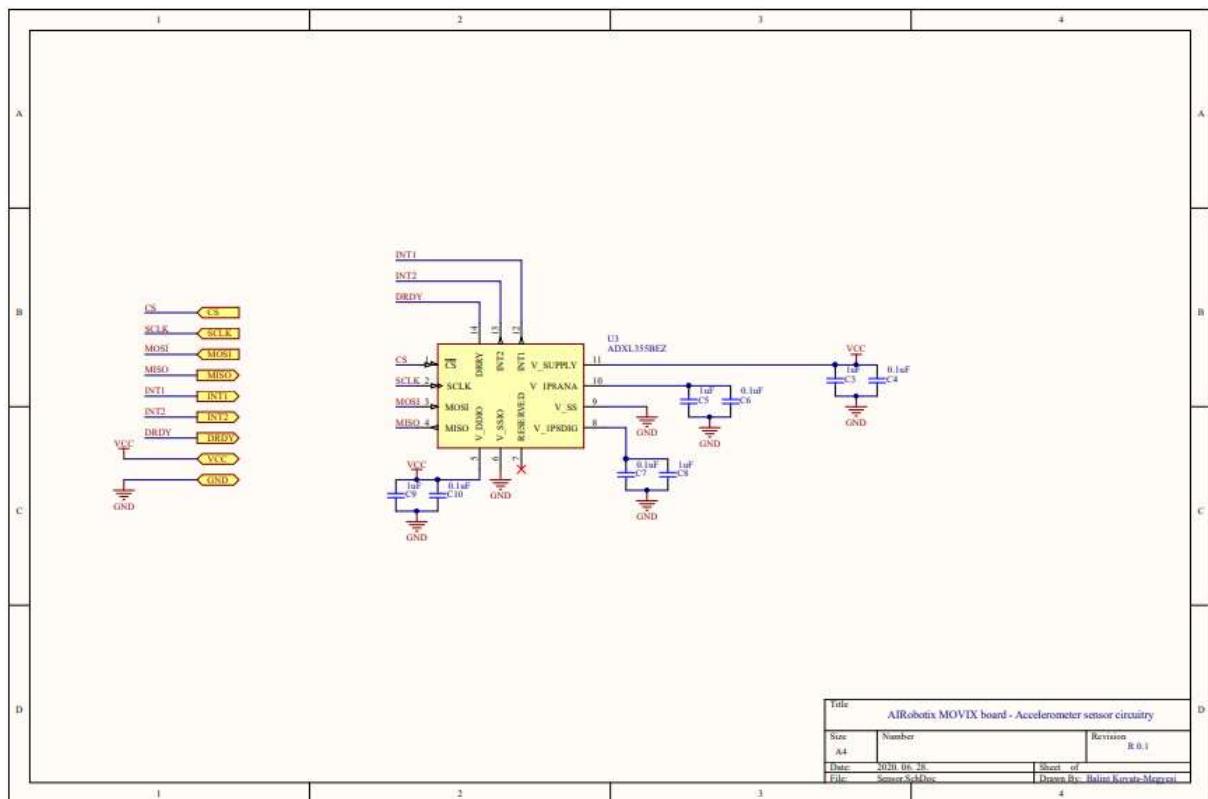
Az ADXL355 típusú gyorsulásmérőt SPI driver dokumentációja a 2. Appendixben található. Az 5. ábrán pedig a mechanikai tulajdonságok láthatóak. A NYÁK 10cm x 15cm felületű és négy M3-as csavar lyuk található a négy sarkában, ezekkel tetszőlegesen tervezhető köré 3D nyomtatott vagy CNC-zett műszerdoboz, melyre a nyilaknak megfelelően kell elhelyezni, hogy a gyorsulásmérő pontos adatokat legyen képes mérni.



58. ábra MOVIX hardware blokkdiagram

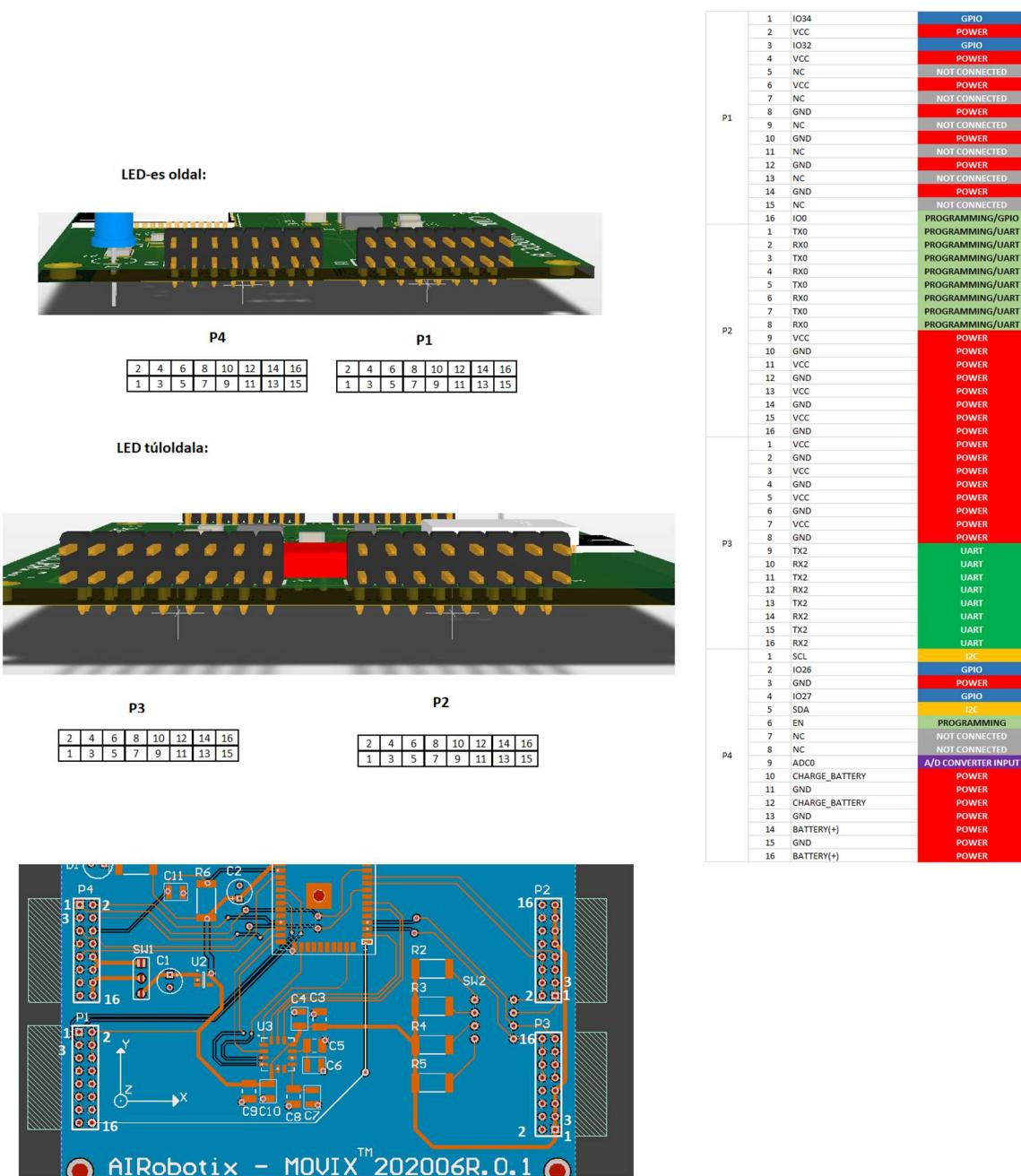


59. ábra MOVIX kapcsolási rajz vezérlő blokk

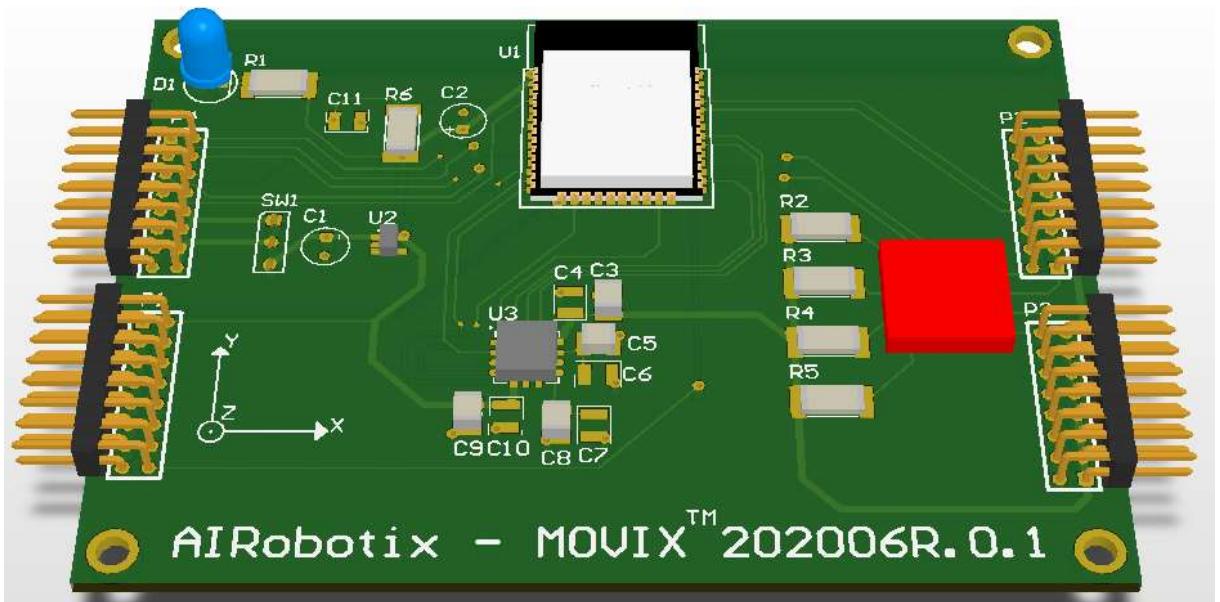


60. ábra MOVIX kapcsolási rajz (szenzor blokk)

J



61. ábra MOVIX lábkiosztás



62. ábra MOVIX nyák

6.2 Control flow char

All communication commands are issued by master board, laser rangefinder module play slave role to answer master's request. The Ask & Answer flow though USART is shown as Figure 6-1.

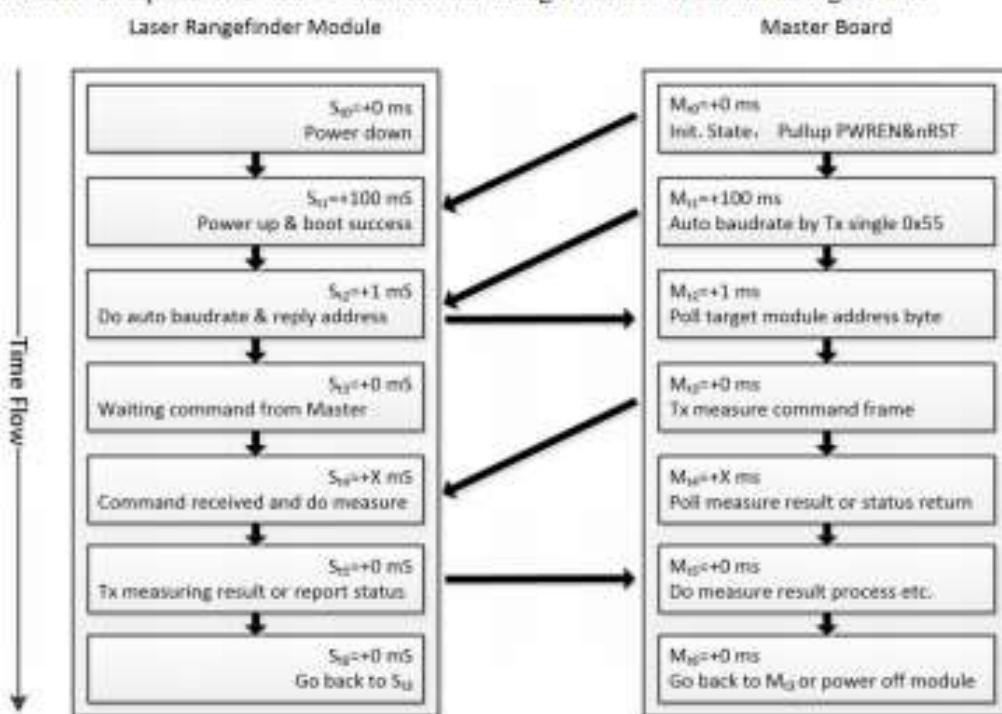


Figure 6-1 Control Flow

In initial state, Slave module (laser rangefinder) is in power down mode before Master pull up the PWREN pin. After PWREN goes high, and if nRST pin. Used please also remember to de-assert the nRST ping by pull it up. Slave will take about 100 milliseconds to do self-boot, and then entering auto baud rate detect stage.

Master transfer 1byte fixed data 0x55 to slave for auto baud rate, if success, slave will reply 1-byte data to master, which present the slave itself address. In one master and multi-slave communication situation, the self-address reply from the slaves may cause USART bus conflict, keep in mind this byte should be ignored.

Communication between master and slave has been established after the successful auto baud rate. Now master can send command frame to slave.

6.3 Command Frame

Command frame may consist 6 parts as Table 6-1 shows.

Table 6-1 command structure

Bytes	0	1		2	3	4	5	6	7	8
Bits	[7:0]	[7]	[6:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]
Name	Head	R/W.	Address	Register		Payload count		Payload		Checksum
Data	0xAA	0	0x51	0x00	0x20	0x00	0x01	0x00	0x00	0x72

Table 6-1 shows the 1-shot measure request command from master to slave. In this command frame:

63. ábra Függelék 1.

SERIAL COMMUNICATIONS

The 4-wire serial interface communicates in either the SPI or I^C protocol. It effectively autodetects the format being used, requiring no configuration control to select the format.

SPI PROTOCOL

Wire the ADXL355 for SPI communication as shown in the connection diagram in Figure 63. The SPI protocol timing is shown in Figure 64 to Figure 67. The timing scheme follows the clock polarity (CPOL) = 0 and clock phase (CPHA) = 0. The SPI clock speed ranges from 100 kHz to 10 MHz.

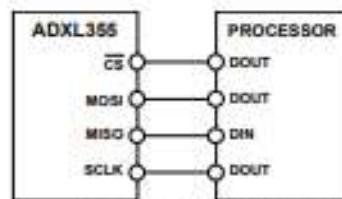


Figure 63. 4-Wire SPI Connection

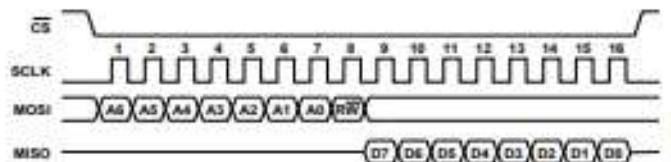


Figure 64. SPI Timing Diagram—Single-Byte Read

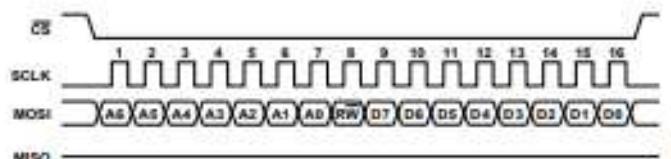


Figure 65. SPI Timing Diagram—Single-Byte Write

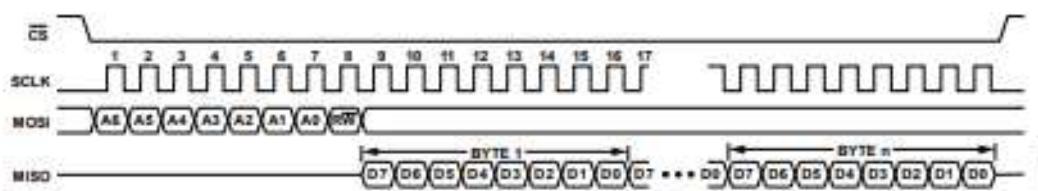


Figure 66. SPI Timing Diagram—Multibyte Read

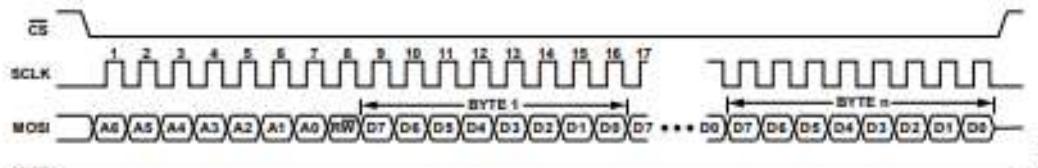


Figure 67. SPI Timing Diagram—Multibyte Write

64. ábra Függelék 2.

3.5. Egyéb hardverelemek

3.5.1. Wifi ESP32

WIFI modul: ESP-WROOM-32

Processor: ESP32-D0WDQ6

Beépített flash memória: 32Mbit

Antenna: Alaplapi PCB antenna

Periféria interfészek UART/GPIO/ADC/DAC/SDIO/PWM/I2C/I2S

WiFi protokol: IEEE 802.11 b/g/n

Bluetooth: Bluetooth 4.2

Frekvencia tartomány: 2.4G ~ 2.5G (2400M ~ 2483.5M)

WIFI módok:: Station / SoftAP / SoftAP+Station

Tápellátás: 5V

Méret: 48.26mm x 25.4mm

3.5.2. Hűtés

Ipari használat során a szenzorokat és az azokat vezérlő elektronikát különböző külső környezeti hatások érik. Ilyen hatás a hőhatás, ami külső és belső forrásból egyaránt származhat.

A darupályák geometriából adódóan a szenzorok a csarnok légterének felső negyedében helyezkednek el. A felfele törekvő meleg levegő miatt ezen a területen a csarnok padlójának környezetéhez képest 5-6 fokkal is melegebb lehet. Erre a plusz hőteherre épül rá az elektronika saját hőtermelése, ami a zárt doboz miatt jelentős többletterhelést jelenthet. A prototípus hűtésének több elvárásnak kell megfelelni:

- a rendszer maradjon pormentes
- ne legyen sérülékeny
- ne keltsen olyan rezgéseket, amik nem szűrhetők egyértelműen, ezáltal befolyásolják a mérést

A megoldás lehet olyan passzív hűtő felszerelése, ami zárt rendszerben, hőelvezető csövekkel juttatja a belső többlethőt a külvilág felé, ahol egy radiátoron képes azt leadni.

3.5.3. Tápellátás - PKCELL ICR18650 6600mAh 3.7V

A szenzorok üzemszerű használata közben a tápellátást az előre kiépített elektromos hálózaton keresztül egy 240V-3.7V transzformátor biztosítja. A prototípus elkészítéskor azonban figyelembe kell venni, hogy a tesztelési fázisban nincs lehetőség a hálózat kiépítésére, ezért az eszköznek hálózatfüggetlen móddal is kell rendelkeznie. A megoldásunk egy nagy kapacitású akkumulátor elhelyezése a rendszerben, ami biztosítja az áramellátását a szenzoroknak a próbamérések során.

A kiválasztott Litium Ion akkumulátorcsomag három darab párosított 2200mAh-ás akkumulátor cellából épül fel, összesen 6600mAh kapacitást biztosítva. A cellák védelmi áramkörrel rendelkeznek, párhuzamosan kapcsoltak. Az áramkör véd a túlfeszültség, alacsony feszültség és a túlárammal szemben. minden cella 2C töltést képes csúcsterhelésen leadni, maximális terhelés mellett több mint 13A képes.

Méret: 69mm x 54 mm x 18 mm

Tömeg: 155g

Maximális töltés: 3A

Maximális leadás: 6A (folyamatosan)



12

65. ábra PKCELL ICR18650 6600mAh

3.5.4. Prototípus hardverdoboz elkészítése

A prototípus tervezésekor figyelembe kell venni az ipari körülményeket, a szenzorok megfelelő pozíciója mellett a védeelmük is fontos. A prototípus vázát adó szerkezetnek ellenállónak kell lenni az ipari körülmények által létrejövő hőmérsékletváltozásnak, rezgésnek, kisebb mechanikai behatásnak. Az egyedi tervezés és kis szériás gyártás miatt a 3D nyomtatást választottuk az eszköz elkészítésére.

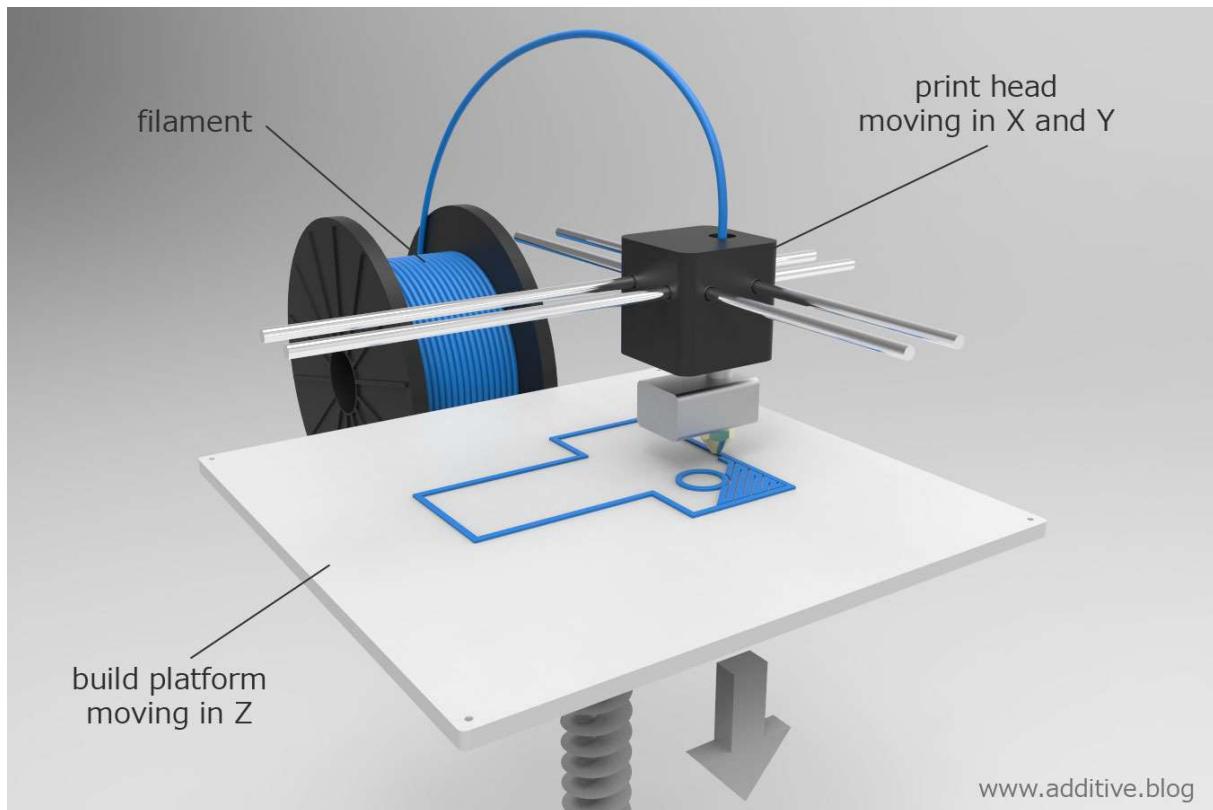
A 3D nyomtatás egy olyan additív eljárás, ami az anyag fokozatos hozzáadásával építi fel a számítógépen tervezett 3D modell valós mását, rétegről-rétegre. A 3D nyomtatók segítségével rövid idő alatt gyártható a szükséges alkatrész vagy tárgy, a kis darabszám miatt költséghatékonyabb a CNC megmunkálásnál vagy fröccsszerszám készítésénél, ráadásul az előbbi két eljáráshoz képest irodai körülmények között is használható. A feladat végrehajtásához különböző nyomtatási eljárásokat használhatunk:

- FFF (Fused Filament Fabrication) FDM (Fused Deposition Modelling)
- SLA (Stereolithography)
- SLS (Selective Laser Sintering)

Az eljárások között a felhasznált alapanyag, az alapanyag rétegenkénti felvitele, a nyomtatás sebessége és végeredmény felbontása is eltér.

FFF vagy FDM nyomtatási eljárás:

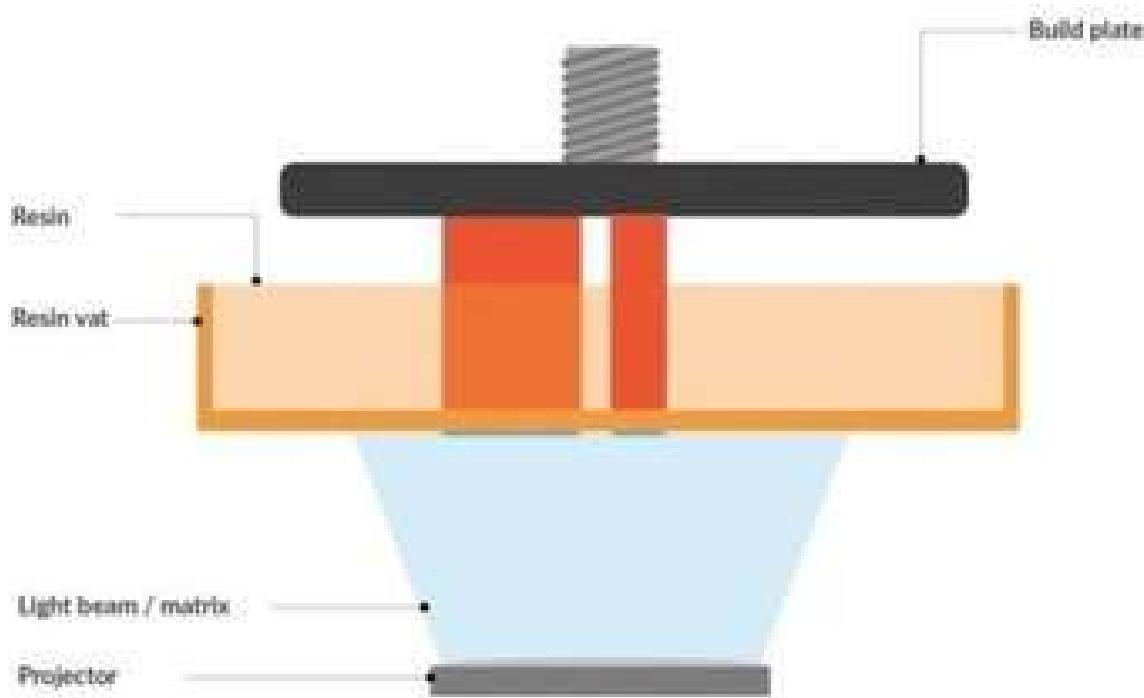
A nyomtatási alapanyag (filament) feltekercselt szál ($d=1.75\text{mm}$, $d=2.85\text{mm}$) kerül a nyomatóba. A nyomatófej egy három tengely mentén mozgó olvasztó (hotend) és adagolófejből (extruder) áll. A négy léptetőmotor és a végálláskapcsolók biztosítják, hogy a nyomatófej pontos, precíz mozgás közben a megfelelő mennyiségű olvasztott filamentet extrudálja a tárgyasztalra. A tárgy rétegről-rétegre épül fel. A rétegek vastagsága határozza meg az elkészült tárgy Z tengely menti felbontását, befolyásolja a tárgy teherbíró képességét is. A filament anyagától függően $160\text{-}250\text{ }^{\circ}\text{C}$ között történik az extrudálás.



66. ábra FFF nyomtatás sematikus ábra

SLA nyomtatási eljárás:

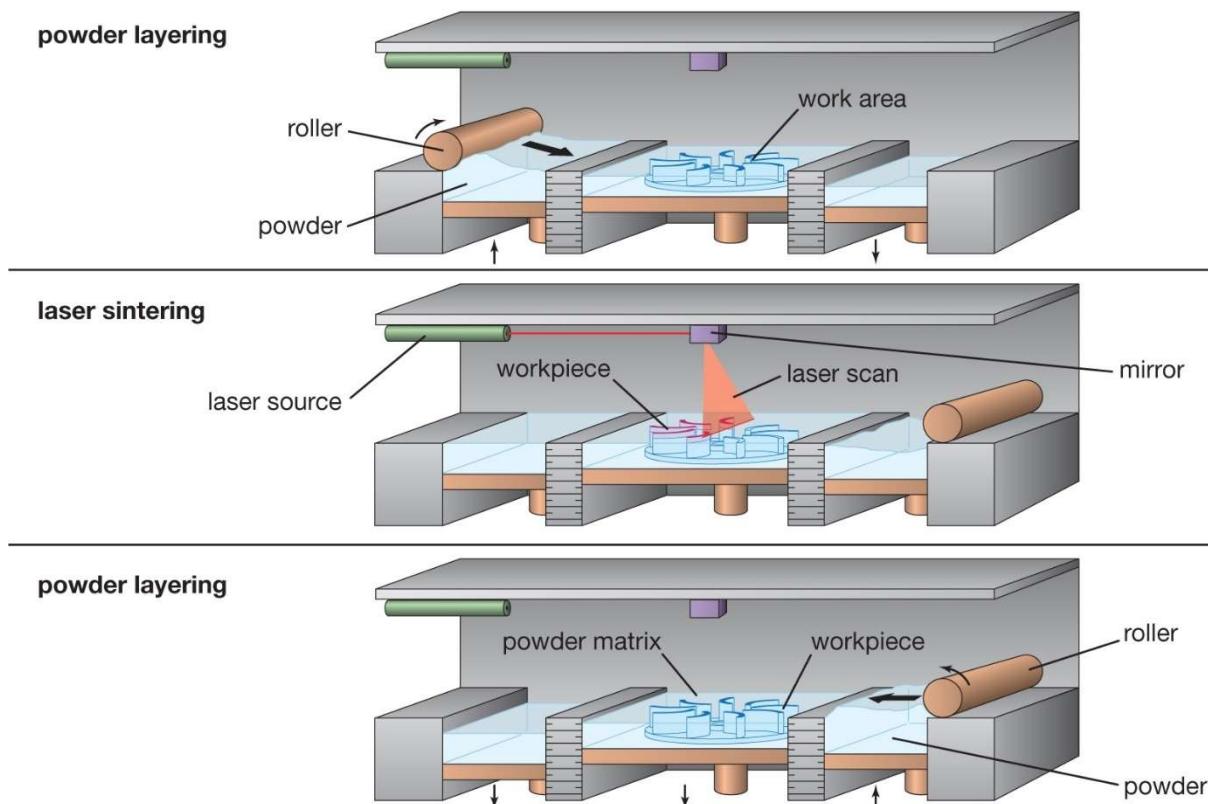
Az SLA nyomtatási eljárás során az alapanyag egy fotoreaktív műgyanta. A műgyanta egy átlátszó aljú tartályban helyezkedik el. A nyomatóban a tárgyasztal Z tengely mentén mozog és a Z irányú felbontástól függően ~ 50 mikron magasságkülönbséggel a folyadékkel teli tartályba merül. A tartály alatt található lézerfény vagy LED fényforrás hatására az 50 mikronnyi műgyanta megszilárdul. A levilágítási idő elteltével a tárgyasztal felemelkedik, az első réteggel együtt. A nyomtatás során rétegről-rétegre haladva épül fel a modell.



67. ábra SLA eljárás sematikus ábra

SLS nyomtatási eljárás:

Az SLS egy poragyfúziós technológia, amely egy hőforrás, pontosabban egy nagy teljesítményű lézer használatával olvasztja össze az alapanyagot egy háromdimenziós tárggyá. A nyomtató vékony rétegben műanyagport terít az építőplatformra egy semleges nitrogéngáz-kamrában, majd a 70 W-os CO₂-lézer egybeolvastja a port. A platform ezután lejjebb süllyed, és egy következőréteg por terül, majd a lézer újra működésbe lép. Ez a folyamat ismétlődik, amíg a tárgy el nem készül. A nyomtatott munkadarabot körbevevő por támaszanyagként szolgál, így kifejezetten bonyolult, akár összeszerelt, vagy mozgó alkatrészekkel is rendelkező nyomatok is készíthetők. A modelleket a nyomtatótartályból való eltávolítását követően nagy nyomású levegővel lefújják, így távolítva el a felesleges, fel nem használt port. Az utómunkálatok tekintetében az SLS kevésbé időigényes, mint sok más 3D nyomtatási módszer.



© 2012 Encyclopædia Britannica, Inc.

68. ábra SLS nyomtatás folyamatábra

A három bemutatott nyomtatási eljárást megismerve a prototípus gyártáshoz a megfelelő eljárások kiválasztása a következő szempontok alapján zajlott:

Felbontás/ méretpontosság: A prototípus háznak XYZ irányban elegendő a 0.4 mm felbontás. Ez a méret az általános FFF nyomtató fűvökamérete 1.75mm filament esetén. A kívánt méretpontosság 0.5 mm. A prototípus egyéb kiegészítő alkatrészeinek 0.1mm felbontás a javasolt. Ezt a felbontás a három tengely mentén SLA és SLS nyomtatóval lehet elérni.

Tartósság: A prototípusnak ellen kell állni kisebb fizikai behatásoknak. A nyomtatásnak rugalmasnak, tömörnek kell lennie, a Z irányú felbontást úgy beállítva, hogy az a rétegek delaminációját a minimálisra csökkentsük. A további paraméterek az alapanyagtól és nem az eljárástól függnek.



69. ábra A három nyomtatási eljárás felbontásbeli különbségei

Alapanyagok:

ABS: Magas (220°C +) hőmérsékleten nyomtatható, kihűléskor vetemedésre hajlamos alapanyag. Nagy szilárdságú, jó mechanikai tulajdonságokkal bír, a rétegek közötti tapadása megfelelő. Kellemetlen szaga miatt elszívás nélkül nem használható.

PET/PETG: Közepesen magas (200°C +) hőmérsékleten nyomtatható, kihűléskor sem vetemedő anyag. Teljesen újrahasznosítható, élelmiszer-biztonsági szempontoknak is eleget tesz, nyomtatása során nem keletkezik kellemetlen szag. Mechanikai tulajdonságai az ABS-hez hasonlóak.

PLA (általános): Alacsony hőmérsékleten (160°C +) nyomtatható, jó alaktartással rendelkező alapanyag. Minimális zsugorodás és torzulás jellemzi. Mechanikai hatásoknak közepese ellenálló, rideg anyag. Nem toxikus, biológiaileg lebomló, kellemetlen szag nélkül nyomtatható.

PLA Carbon: Alacsony hőmérsékleten (160°C +) nyomtatható, jó alaktartással rendelkező alapanyag. Minimális zsugorodás és torzulás jellemzi. Mechanikai hatásoknak ellenáll, a benne

lévő apró szénszálak miatt rugalmas és erős. Nem toxikus, biológiaileg lebomló, kellemetlen szag nélkül nyomtatható.

SC-801 műgyanta: Részletgazdag felbontást tesz lehetővé az apró elemek nyomtatásakor. Mechanikai hatásoknak ellenáll, utókezelés után teljesen víztiszta marad. Levlágítási ideje 8 másodperc. Toxikus, gőze kellemetlen szagú.

Rock-Black Stiff műgyanta: Ipari célra fejlesztett, erős, mechanikai hatásoknak jól ellenáll. Részletgazdag felbontást tesz lehetővé kis alkatrészek esetén is. Levlágítási ideje 15 másodperc. Toxikus, gőze kellemetlen szagú.

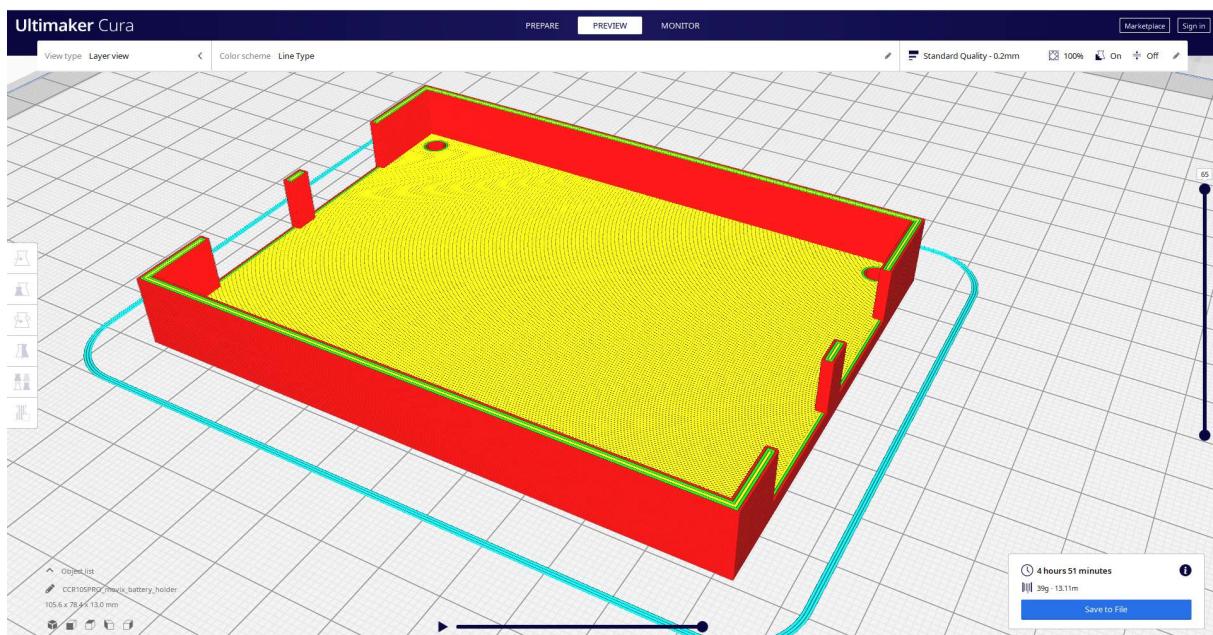
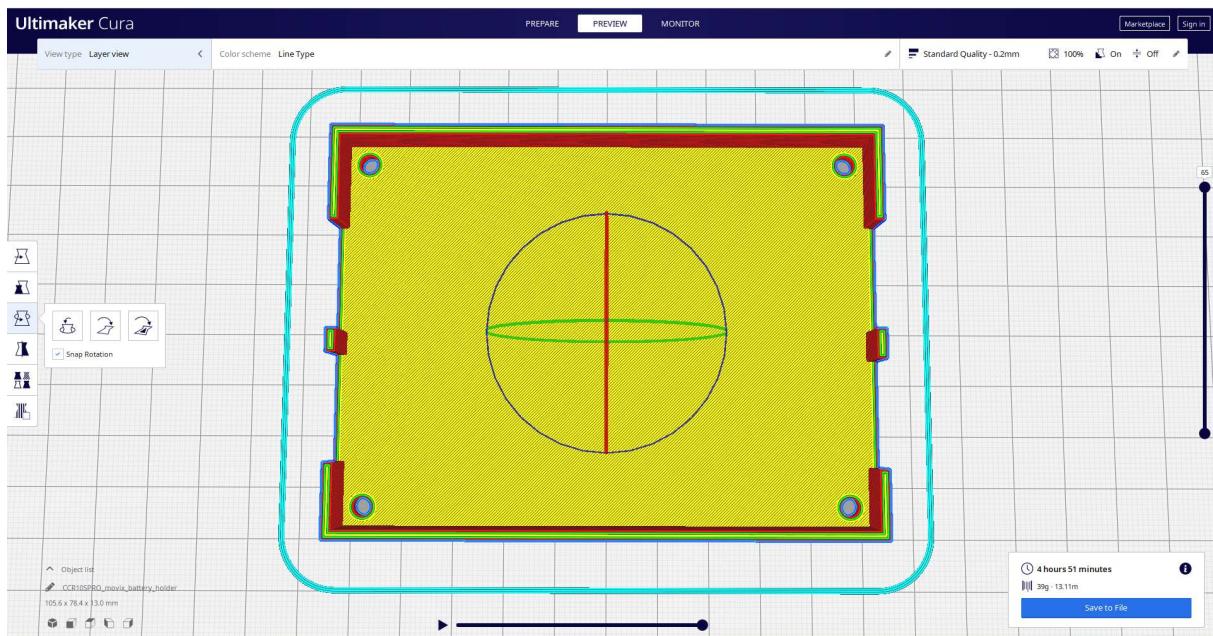
Nylon műgyanta: Ipari célra fejlesztett, rugalmas, mechanikai hatásoknak jól ellenáll. Részletgazdag felbontást tesz lehetővé kis alkatrészek esetén is. Levlágítási ideje 12 másodperc. Toxikus, gőze kellemetlen szagú.

A prototípus eszközök gyártásakor két nyomtatási eljárást használtunk: Az egyik az FFF eljárás, a másik az SLA (DLP) eljárás volt. Az FFF nyomtatával az alaplapot valamint a LiDAR szenzorokat tartó házat készítettük, az SLA eljárással a lencsevédőt, rezgéselnyelő alátéteket és kiegészítő elemeket nyomtattuk.

Az FFF nyomtatás során több anyagot is megvizsgáltunk (ABS, PLA, PETG), végül a PLA és PLA carbon alapanyag mellett döntöttünk. Mind a felbontás, mind a tartóssági igényeket is kielégítette, amellett környezetbarát, újrahasznosítható és biológiaileg lebomló.

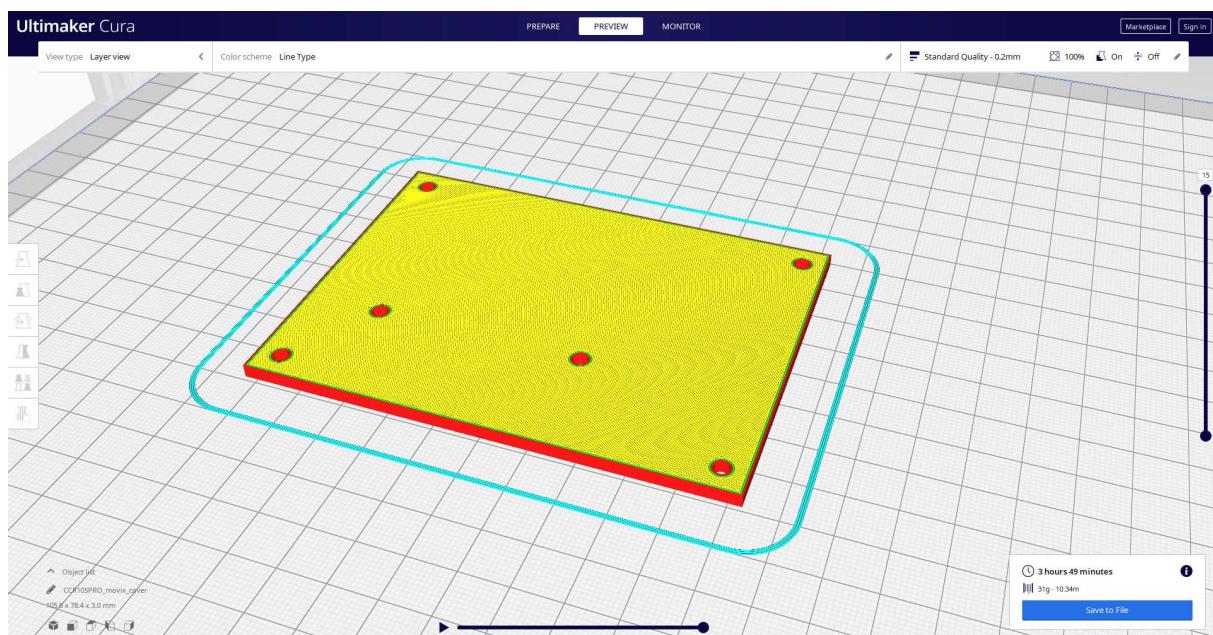
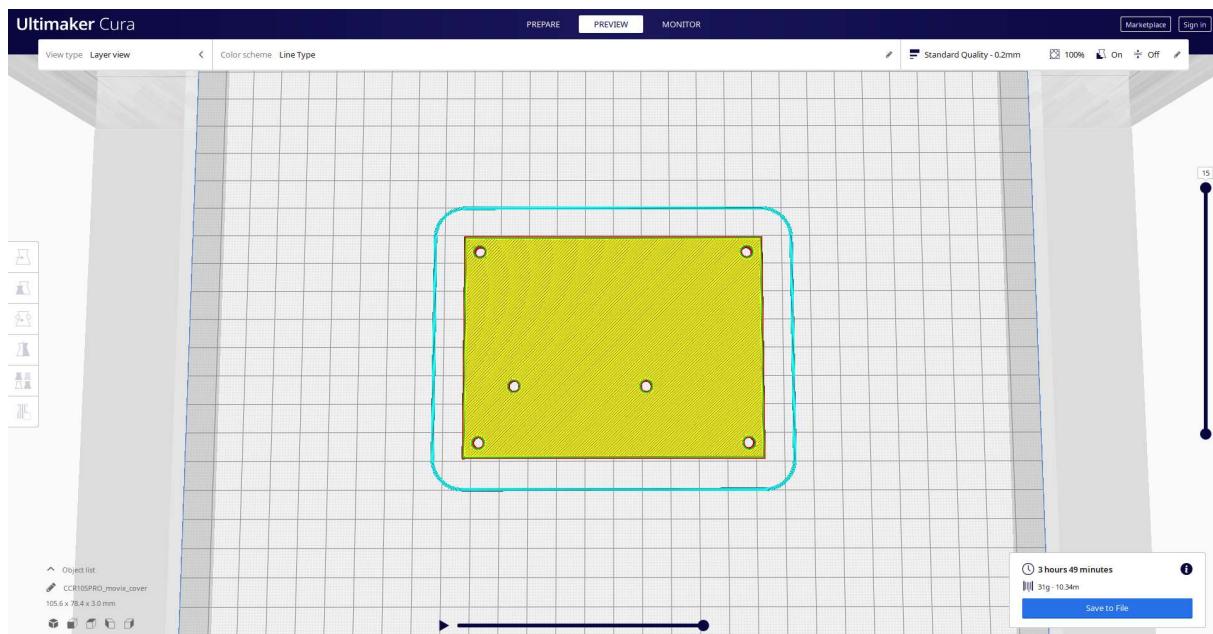
A nyomtatási beállítások:

Akkumulátor tartó:



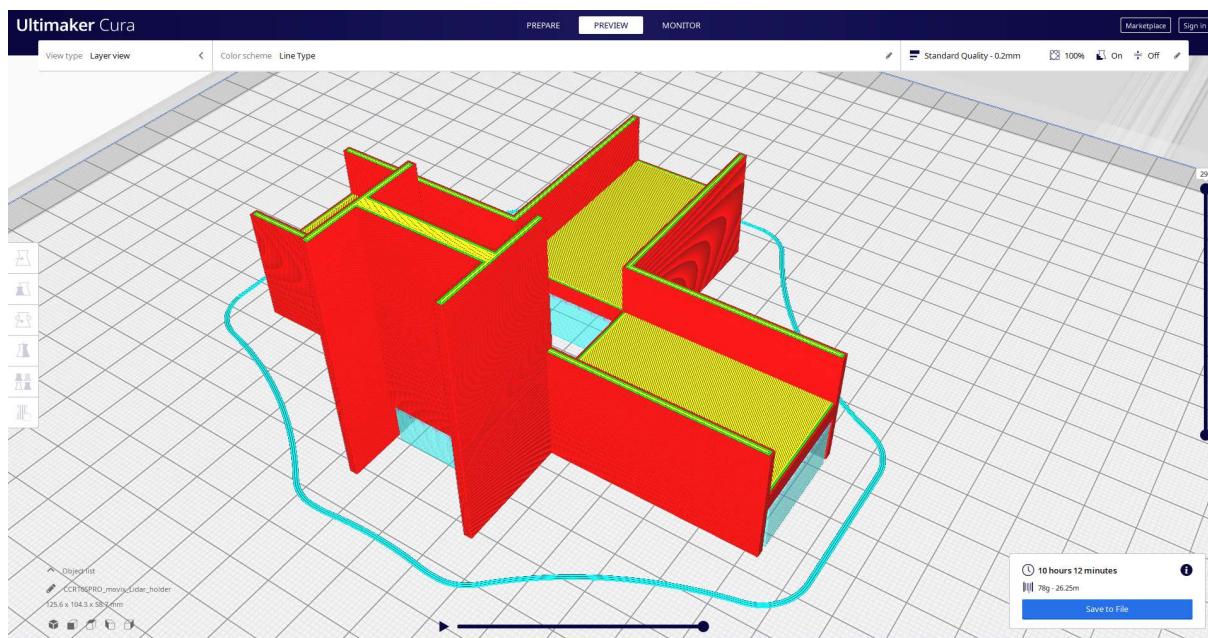
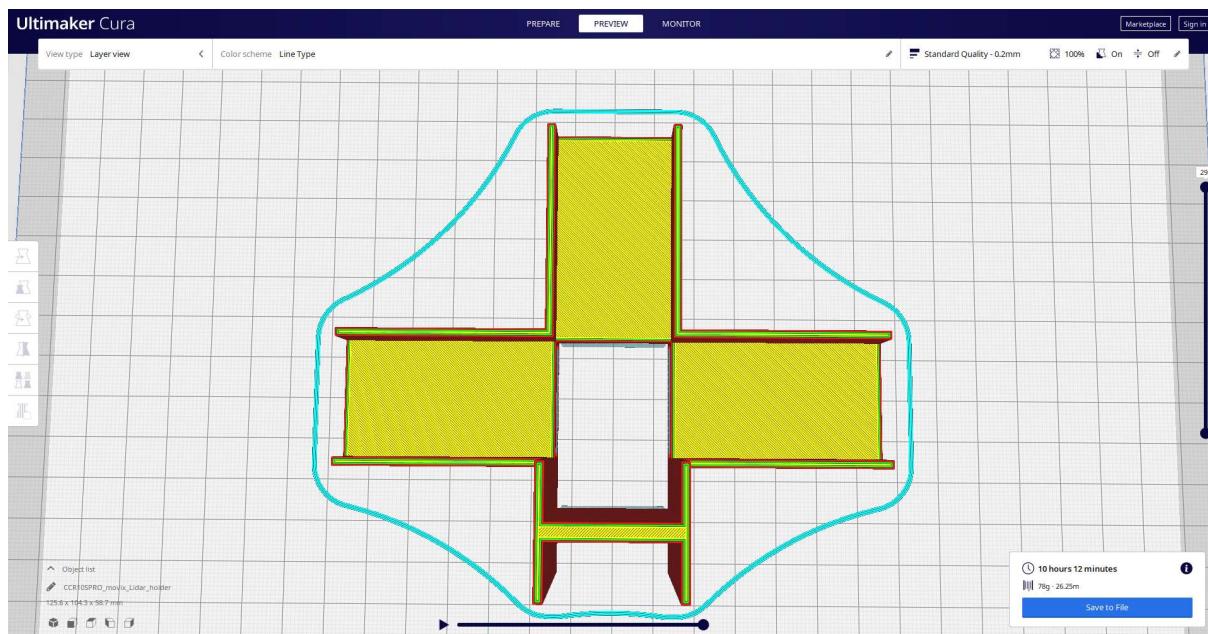
XY felbontás	0.4 mm	Anyaghőmérséklet	200 °C
Z felbontás	0.2 mm	Felhasznált anyag	PLA
Falvastagság	0.8 mm	Anyagáram	105%
Kitöltés	100%	Felhasznált anyag	40g
Sebesség	50mm/s	Felhasznált anyag	13.49m
Support	Nem	Nyomtatási idő	4:58:00
Tapadóhíd	Skirt	Befoglaló méret	105.6 x 78.6 x 14.0

Alaplap tartó fedél:

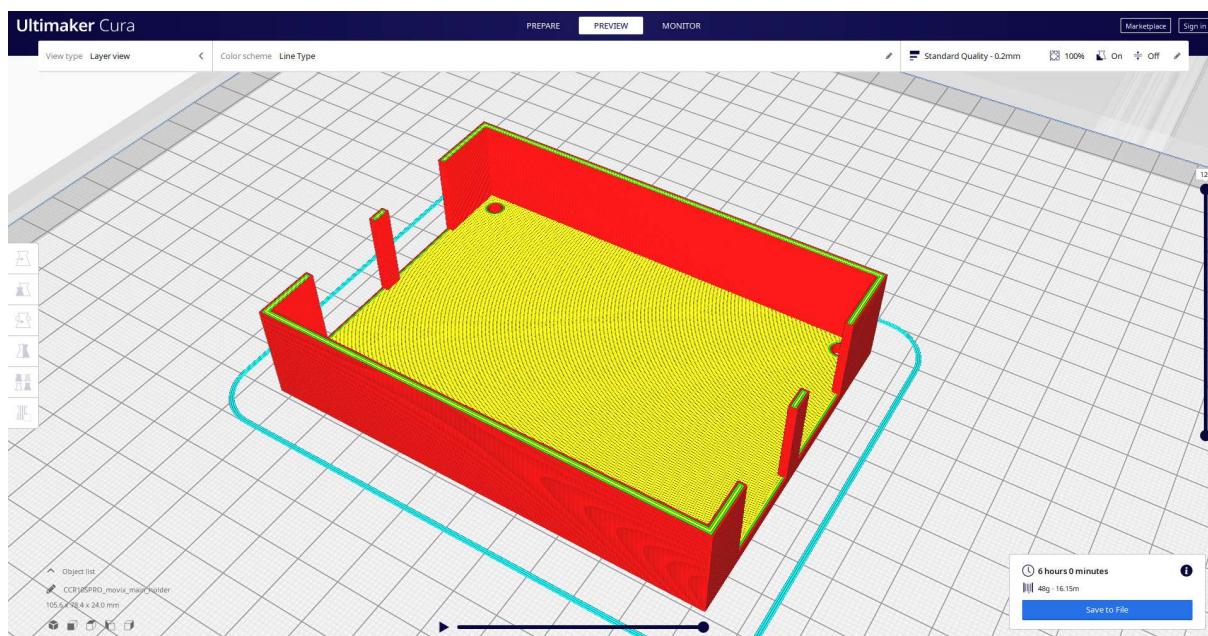
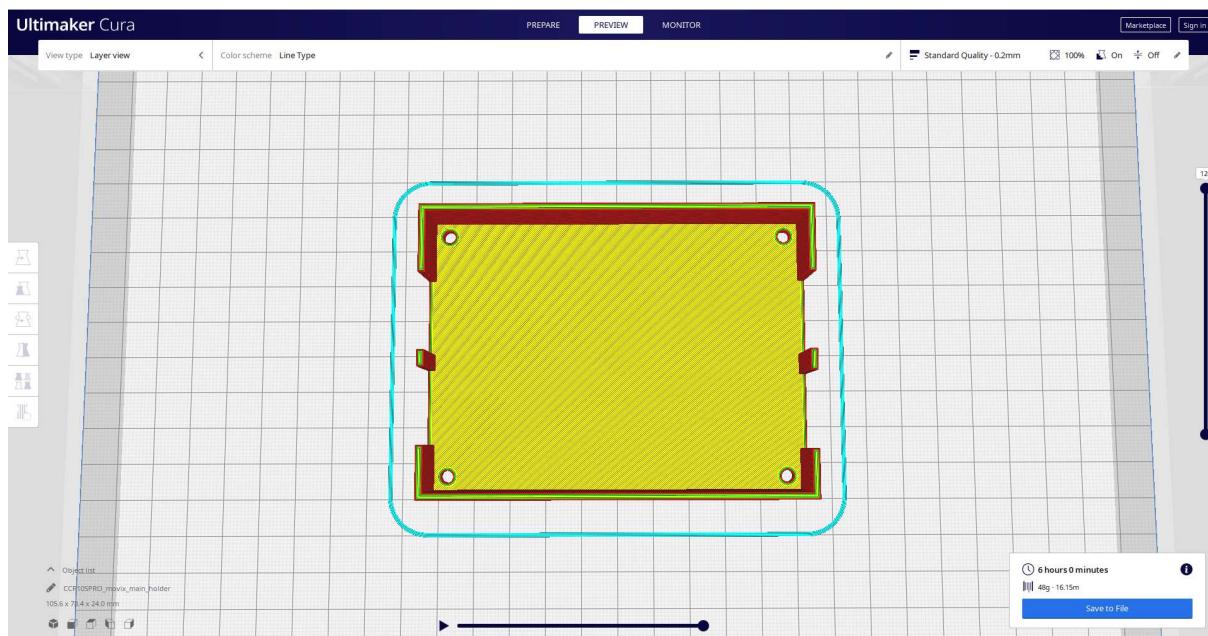


XY felbontás	0.4 mm	Anyaghőmérséklet	200 °C
Z felbontás	0.2 mm	Felhasznált anyag	PLA
Alvastagság	0.8 mm	Anyáram	105%
Kitöltés	100%	Felhasznált anyag	31g
Sebesség	50mm/s	Felhasznált anyag	10.31m
Support	Nem	Nyomtatási idő	3:49:00
Tapadóhíd	Skirt	Befoglaló méret	105.6 x 78.4 x 3.0

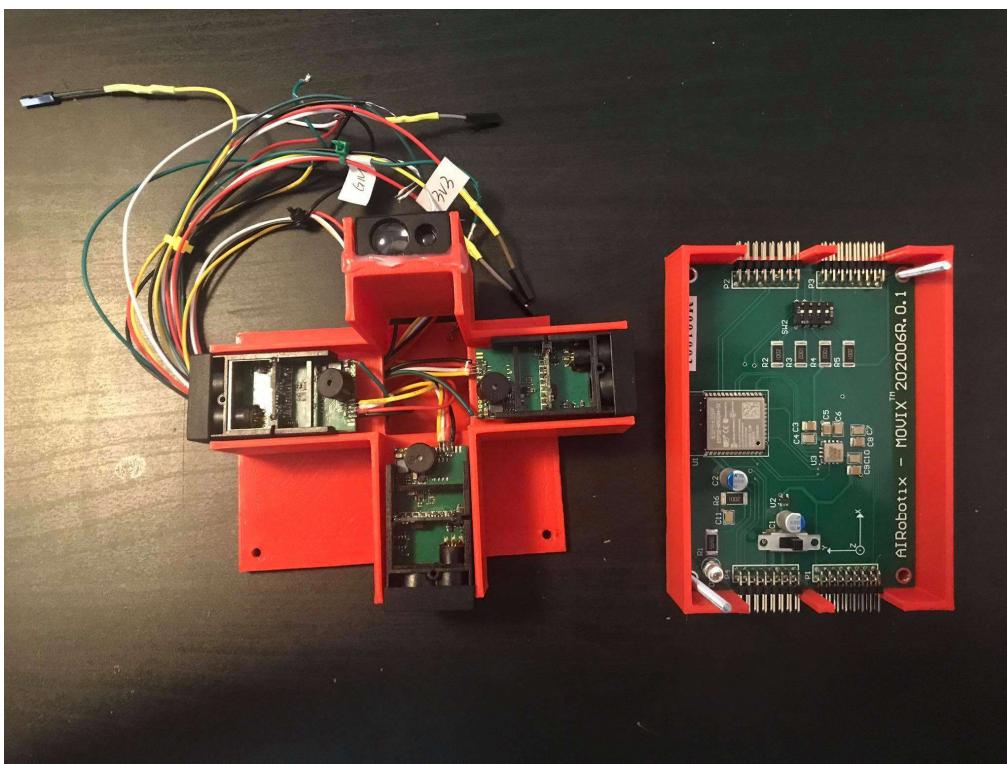
Lidar tartó:



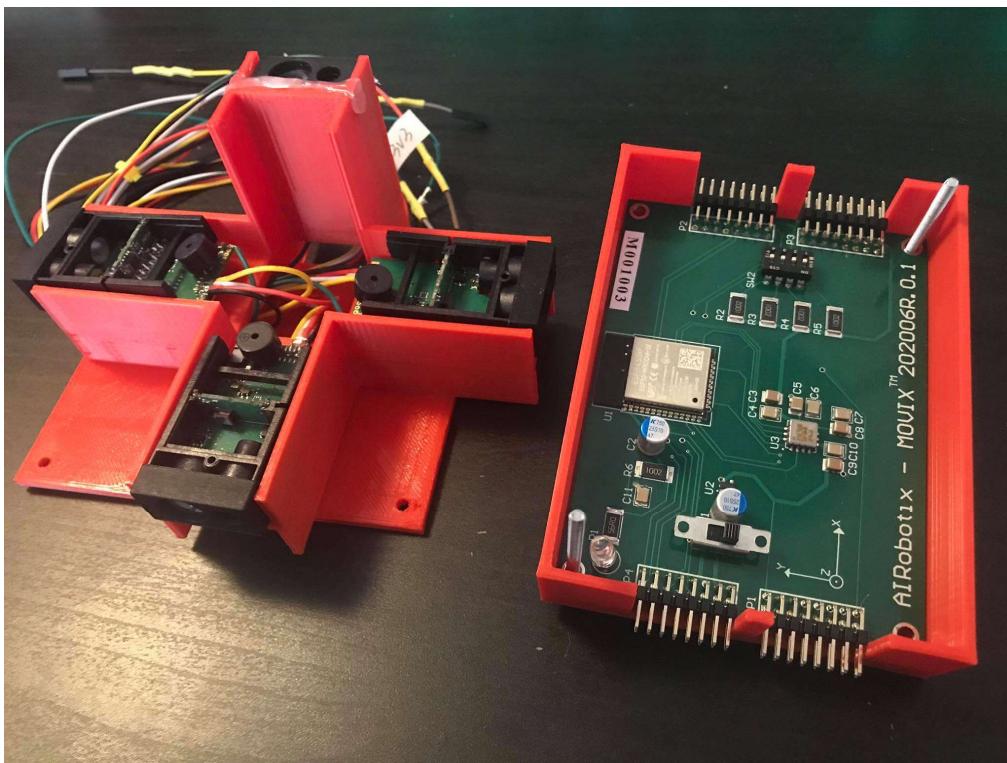
XY felbontás	0.4 mm	Anyaghőmérséklet	200 °C
Z felbontás	0.2 mm	Felhasznált anyag	PLA
Falvastagság	0.8 mm	Anyagáram	105%
Kitöltés	100%	Felhasznált anyag	78g
Sebesség	50mm/s	Felhasznált anyag	26.25m
Support	Igen	Nyomtatási idő	10:12:00
Tapadóhíd	Skirt	Befoglaló méret	125.6 x 104.3 x 58.7



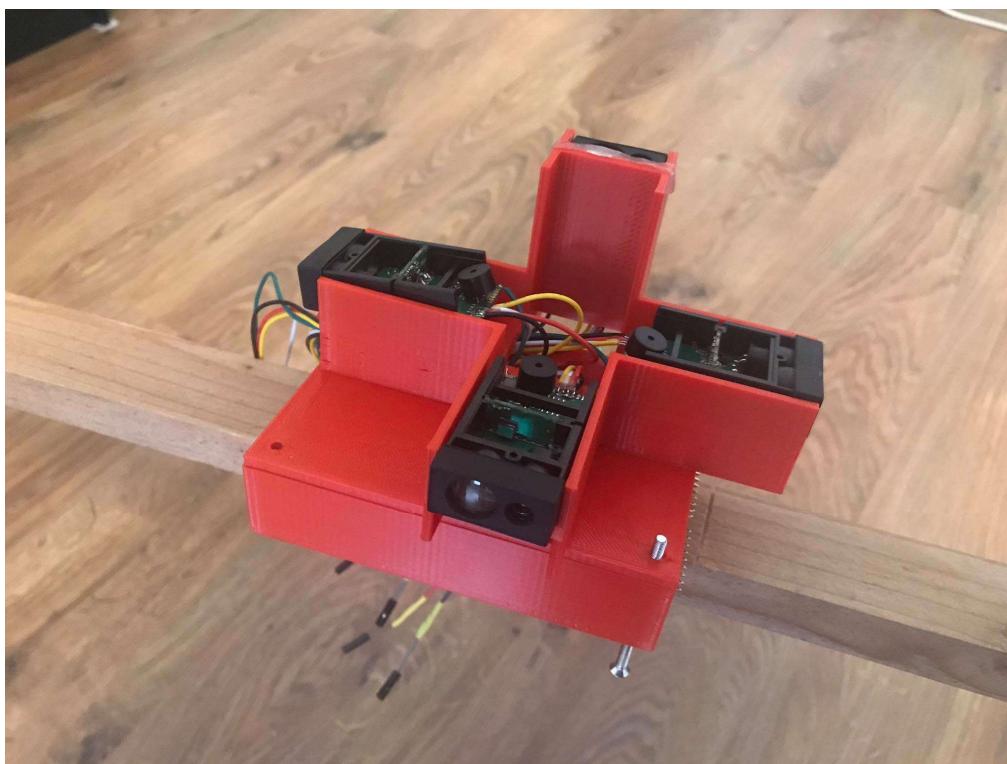
XY felbontás	0.4 mm	Anyaghőmérséklet	200 °C
Z felbontás	0.2 mm	Felhasznált anyag	PLA
Falvastagság	0.8 mm	Anyagáram	105%
Kitöltés	100%	Felhasznált anyag	48g
Sebesség	50mm/s	Felhasznált anyag	16.15m
Support	Nem	Nyomtatási idő	6:00:00
Tapadóhíd	Skirt	Befoglaló méret	105.6 x 78.4 x 24.0



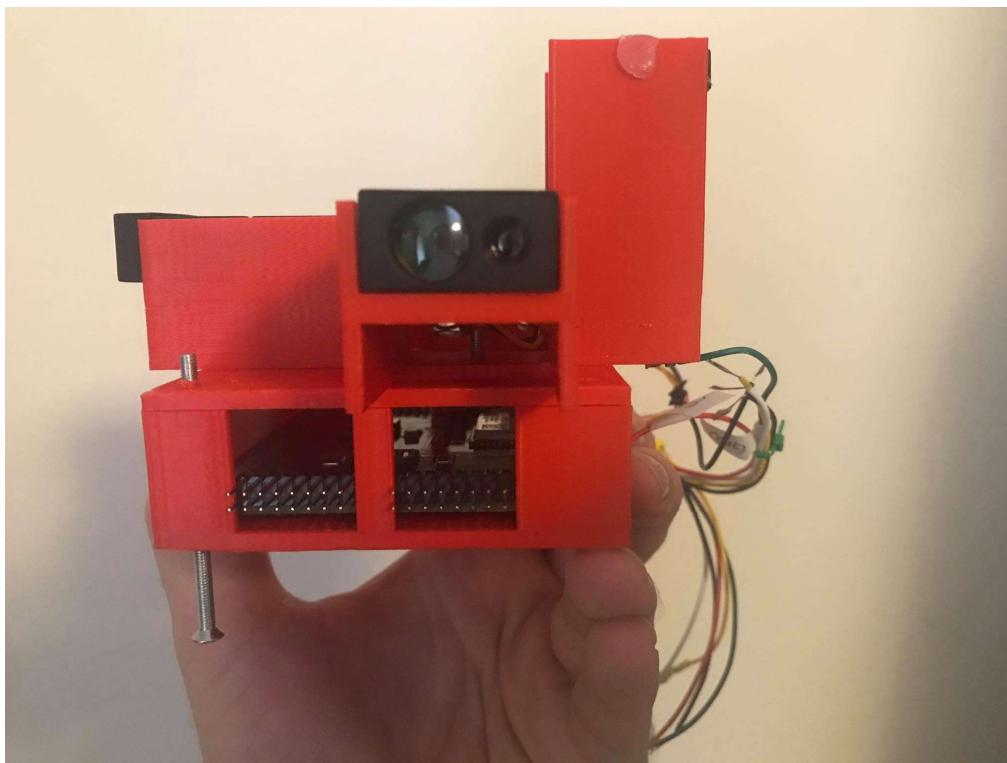
70. ábra Nyomtatott elemek és a szenzorok



71. ábra Nyomtatott elemek és a szenzorok

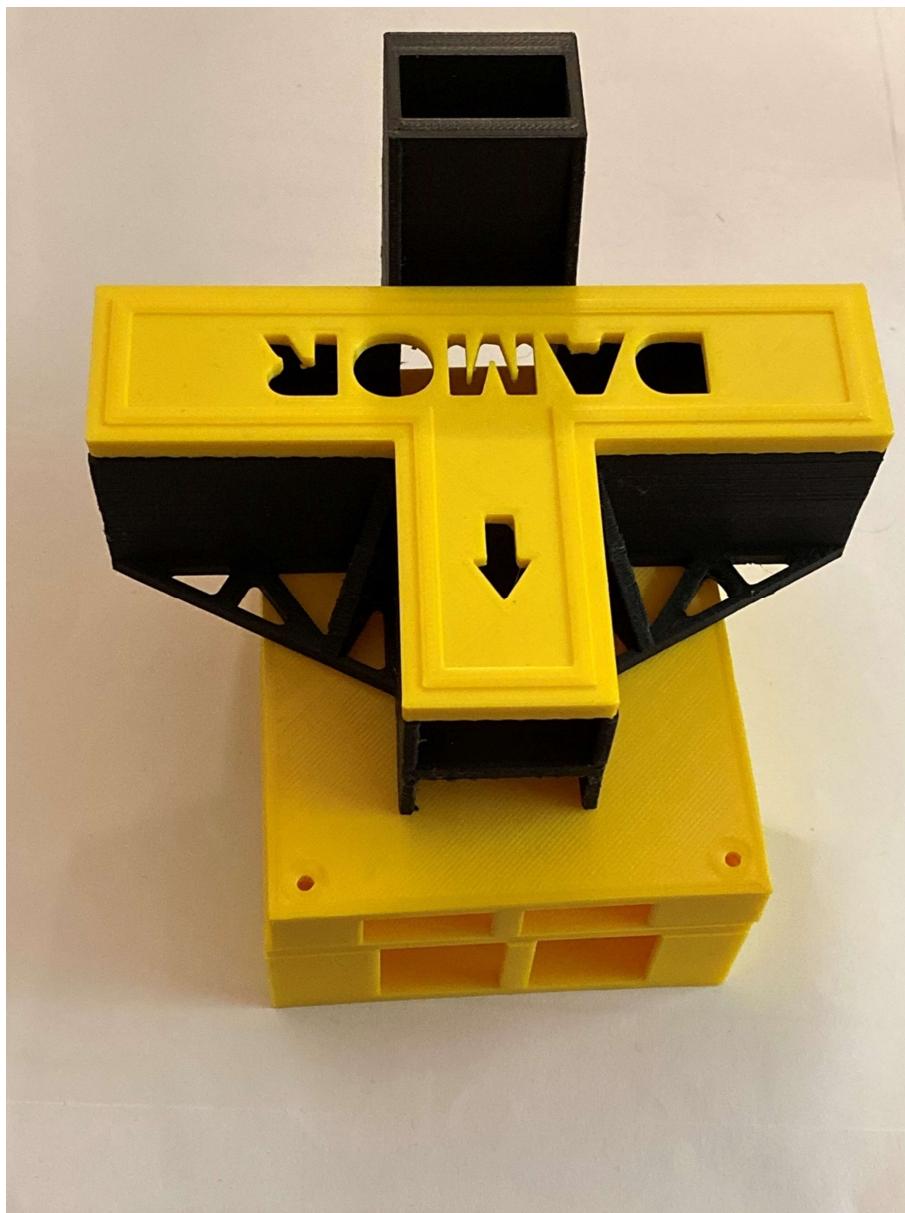


72. ábra Összeépített változat

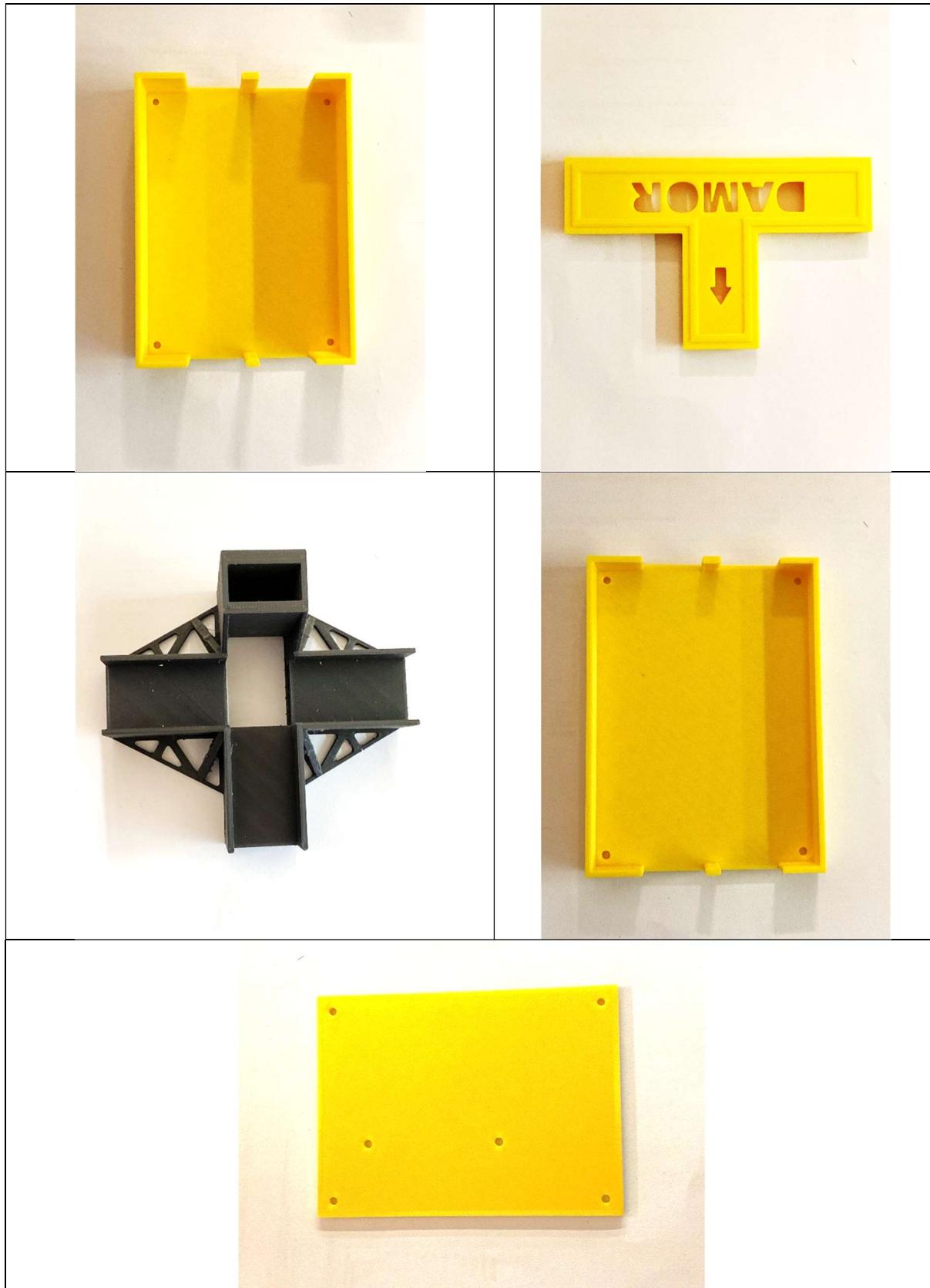


73. ábra 72. ábra Összeépített változat - oldalnézet

A megfelelő anyaghasználat kiválasztása után a prototípusban használt elemek áttervezése és 3D nyomtatása is megtörtént. Az áttervezés során erősítő bordát kapott a test elem, a nagyobb stabilitás érdekében. Az eredetileg nyitott koncepció félig zárt rendszerré alakult, védve ezzel a szenzorokat és az azokat kiszolgáló elektronikát is.



74. ábra Összeillesztett 3D nyomtatott elemek



75. ábra A prototípus vázának elemei. A fekete elem magas karbon tartalmú, rezgésnek ellenálló anyag

3.5.5. Hálózati, vezérlő és adattároló eszközök

3.5.5.1. Vezeték nélküli hálózat kiépítése

Feladata a vezeték nélküli hálózat kiépítése a prototípus elemei között. A routernek nagy hatótávolságúnak, többfrekvenciásnak és többantennás kivitelnek kell lenni a stabil kapcsolat fenntartása érdekében.

Synology RT2600ac Wireless Router

Támogatott szabványok	802.11a (5GHz), 802.11b (2.4GHz), 802.11g (2.4GHz), 802.11n, 802.11ac
Sávok	2,4 GHz , 5 GHz
Átviteli sebesség	2 530 Mb/s
Lan port	1 Gbit
Típus	Dual-band
Titkosítás	WEP 64bit, WPA, WPA2, WPA3
LAN	4 db
WAN	2 db
USB 2.0 és 3.2	1 db- 1db
Külső antennák	4 db
Méret (mm)	169 x 280 x 77
Tömeg (g)	700



3.5.5.2. Hálózati tároló és merevlemezek

A hálózati tároló a folyamatos adatfolyam biztonsági mentésére szolgáló egység. A tárolónak az adatvesztés elkerülése végett RAID vezérlővel kell rendelkeznie. A hálózati tárolóra kerül a nyers adatok mellett a riasztási adatok is. A NAS saját operációs rendszerén belül létrehozott macro segítségével a riasztási mappa folyamatos monitorozás alatt áll. A NAS-minimum 2 HDD befogadására képes, két Gigabites LAN porttal rendelkezik. A NAS-ban használt merevlemezek kifejezetten hálózati tárolóhoz gyártott, hosszú élettartamú, az átlagosnál több írás és olvasási ciklust elviselő HDD-k. A tárkapacitásuk az adatfolyam méretéből adódóan minimum 4 TB/ lemez.

SYNOLOGY DiskStation DS418

Tárhely típusa	HDD
Fiókok száma	4
HDD mérete	3,5"
Maximális kapacitás	40 TB
RAID támogatás	RAID 0, RAID 1 (tükörözés), RAID 5, RAID 6, RAID 10 (1+0), Speciális, JBOD
Rendszermemória	2 GB RAM DDR4
Processzor	Realtek RTD1296, 1,4 GHz 64Bit
Méret (mm)	223 x 199 x 166
Tömeg (g)	2280



WD Red 6TB 3.5" 5400rpm 256MB SATA 3

Tárhely típusa	HDD
Formátum	3,5"
Tárhely	6 TB
Olvasási sebesség	180 MB/s
Írási sebesség	180 MB/s
Hibahatár	1 000 000 óra
Memória cache	256 MB
Interfész	SATA 3
Méret (mm)	101 x 26 x 147



3.5.5.3. Feldolgozó és vezérlő mini PC

Az adatok kiértékelésére és feldolgozására szolgáló mini PC hardveres elvárásai a következők:

A gyors számításhoz és több szálon futó feldolgozáshoz szükséges i7 minimum 8. generációs processzor (mobil), 16 GB gyors elérésű, DDR4 RAM, M.2 csatlakozású SSD.

Intel NUC7i7DNH2E Intel Core i7-8650U, HyperX 16GB DDR4 2666MHz,
SAMSUNG 1TB 970 EVO Plus M.2 PCIe M.2

Processzor	i7, 8. gen.
Órajel	1,9 GHz
Boostolt órajel	4,2 GHz
Grafikus kártya	Intel UHD Graphics 620
Maximum memória méret	32 GB, DDR4-2400 SO-DIMM
USB 2, 3	2 + 2
Sata/M2	1 + 1
Méret (mm)	115 x 111 x 52
<hr/>	
Memória mérete	8 GB
Darabszám	2 db
Sebesség	2666MHz
Típus	DDR4
<hr/>	
Háttértár mérete	1 TB
Csatlakozás	M.2



3.5.5.4. Vezérlő tablet

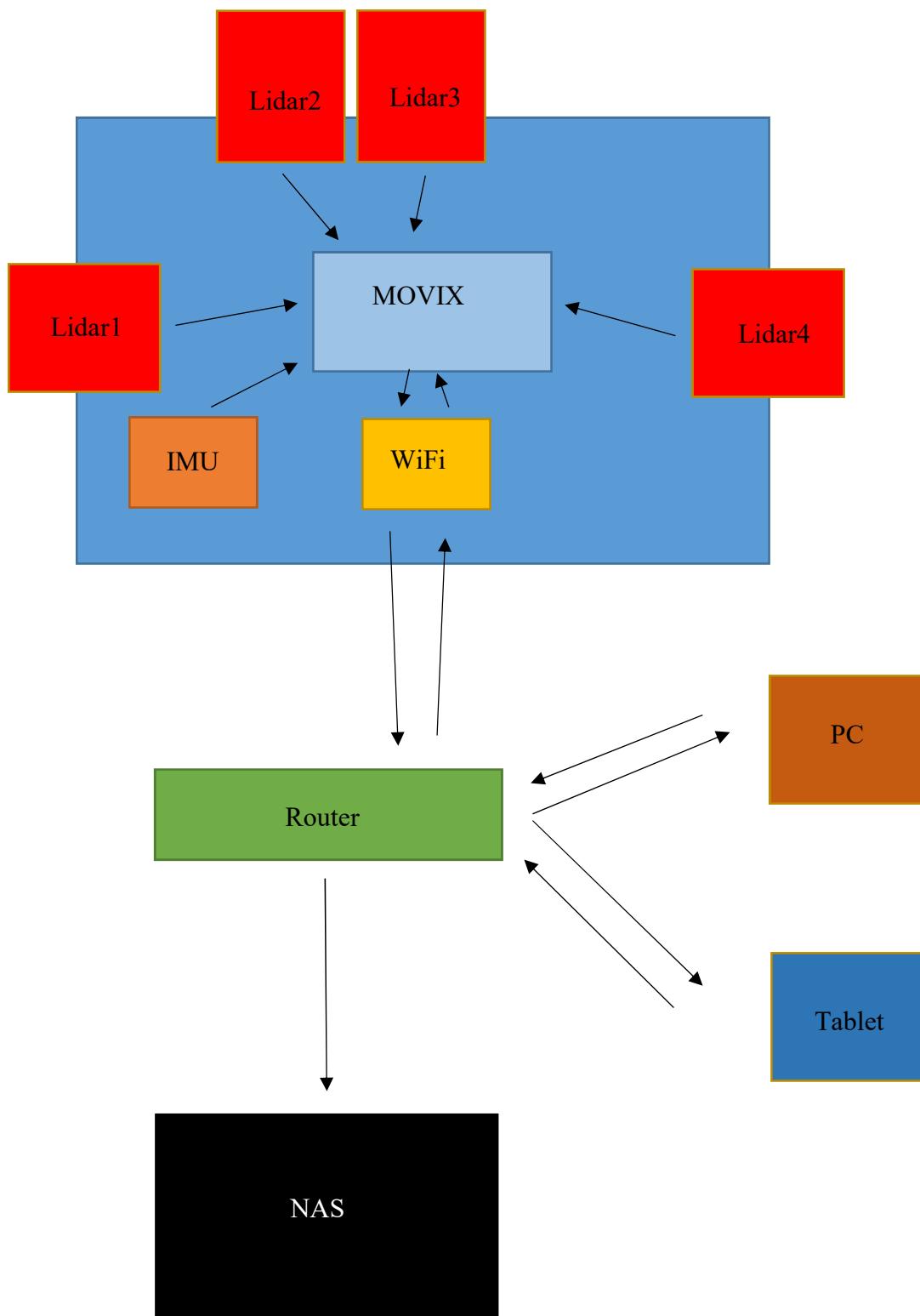
A vezérlő tablet feladata a nem helyhez kötött vezérlés megvalósítása. A tablet segítségével a komparáló mérések elvégzése lehetséges a telepítést követően. A tabletnek minimum 12" kijelzővel kell rendelkeznie, a szoftver stabil futtatásához és az adatok tárolásához az asztali géphez hasonló processzor, memória valamint háttértár méret szükséges.

MICROSOFT Surface Pro X 13" 256GB

Kijelző mérete	13"
Felbontása	2880 x 1920 pixel
Memória	16 GB
Háttértár	256 GB
WiFi protokoll	WiFi 5/ ac
Méret (mm)	287 x 208 x 7,3
Tömeg	770 g



3.6. Hardver elemek összeépítése



4. Prototípus - Szoftver

4.1. Szenzorfúziót megvalósító szoftver

4.1.1. Tervezés

1. Feladatok:

1.1 kommunikáció a felhasználóval (adatok küldése: WiFi, stb...): A felhasználó a PC-n elindítja a DAMOR.exe programot, amelyben a következő lehetőségek állnak rendelkezésre. Ez a program egy interfész, amely socket-ek küldését és fogadását képes végezni. Ezen a programon keresztül történik a 'beállítások' menüben a hálózati konfiguráció is.

1.1.1 **START**: Hálózati setup. Az előző beállításoktól függően elindul az eszköz vagy 'Access point' vagy 'Station' üzemmódba az előző beállításoktól függően. Elindul a szerver, amely az adatkommunikációt biztosítja.

1.1.1.1 Access point: Access point-ot hoz létre megadott/default SSID és Password párossal.

1.1.1.2 Station: Megpróbál WiFi hálózathoz kapcsolódni, ha nem elérhető, mert nem létezik vagy nincs beállítva, akkor visszatérés 1.1.1.2-be.

1.1.1.3 Adat szerver (TCP/IP vagy HTTP) létrehozása: ha ez sikeres **STANDBY** (1.1.1.4), ellenkező esetben 1.1.1 háromszor, ha timeout hibajelzés a MOVIX-on hardware LED jelzővel.

1.1.1.4 **STANDBY**: Várakozás az 1.1.2 pontba való belépérsre, utasításokra.

1.1.2 Hálózati adatkommunikáció/vezérlés:

1.1.2.1 Egyszeri mérés: **SINGLE MEASURE** módba lép: mérési parancs elküldése (részletek: 1.2 alatt), mérési csomag összeállítása. Ezt követően **STANDBY** módba lép (1.1.1.4). Hiba esetén → hibaüzenet, sikeres mérés esetén → eredmény.

1.1.2.2 Folyamatos mérés: **CONTINOUS MEASURE** módba lépteti az eszközt a felhasználó, ekkor data streaming indul el, azaz folyamatos mérések ismétlése és a mért adatok és a hozzá tartozó timestamp kiküldése a felhasználói PC felé.

1.1.2.3 Folyamatos mérés leállítása.

1.1.2.4 Státusz lekérdezés (eszköz állapotának lekérdezése)

1.1.2.5 Hálózati, működési és mérési konfigurációs beállítások

1.1.2.6 **RESET**: az eszköz újraindul és visszalép az 1.1.1-be

1.2 on-board kommunikáció (SPI, UART, egyéb physical user interface...)

1.3 önenellenőrzés (pl.: hőmérsékletmérés, sleep mód beosztása, áramfogyasztás, power management...)

4.2. Felügyeleti szoftver interface

A felügyeleti és megjelenítő szoftver platformfüggetlen, telepítést nem igénylő, böngészőből futtatható. A grafikus felület átlátható, egységes megjelenésű, kis képernyőn, akár mobileszközön is jól olvasható.

4.2.1. Felhasználói felület kód:

```
<head>
  <title>MUI</title>
  <style>
    *
      box-sizing: border-box;
    .log
      background: #f8f8f8;
      list-style-type: none;
    .row
      padding: 5px;
    .row::after
      content: "";
      display: block;
      clear: both;
```

```

.col-1
    width: 100%;
.col-1-2
    width: 50%;
.col-1-3
    width: 33.33%;
.col-2-3 {
    width: 66.67%;
.col-1-4 {
    width: 25%;
.col-3-4 {
    width: 75%;
.col-1-5 {
    width: 20%;
.col-2-5 {
    width: 40%;
.col-3-5 {
    width: 60%;
.col-4-5 {
    width: 80%;
.col-1-6 {
    width: 16.67%;
.col-5-6 {
    width: 83.33%;
.col-1-7 {
    width: 14.28%;
.col-1-8 {
    width: 12.5%;
.col-7-8 {
    width: 87.5%;
[class*="col-"] {
    float: left;
    padding: 5px;
    text-align: center;
input {
    width: 100%;
input[type="number"] {
    border: 1px solid;
#x-sli {
    -webkit-appearance: none;
    width: 80%;
    height: 18px;
    background: #d3d3d3;
#x-sli::-webkit-slider-thumb {
    -webkit-appearance: none;
    width: 18px;
    height: 18px;
    background: #4CAF50;
@media only screen and (max-width: 768px) {
    [class*="col-"] {
        width: 100%;

</style>
</head>

<body>
    <div class="row">
        <div class="col-1"><h1>Movix User Interface</h1></div>

```

```

</div>
<div class="row">
  <div class="col-1-4">
    <div class="row">
      <div class="col-1-4"><input type="button" value="Connect" id="conn-b" /></div>
      <div class="col-1-4"><input type="button" value="Start" id="meas-b" /></div>
      <div class="col-1-2"><input type="text" value="disconnected" id="meas-s" readonly /></div>
    </div>
    <div class="row">
      <div class="col-1-4"><label>Recorder:</label></div>
      <div class="col-1-2"><input type="url" value="http://" id="rec-s" /></div>
    </div>
    <div class="row">
      <div class="col-1-4"><label>Alert:</label></div>
      <div class="col-1-4"><input type="checkbox" value="http://" id="rec-ale" /></div>
      <div class="col-1-4"><input type="button" value="Add" id="rec-add" /></div>
    </div>
    <div class="row">
      <div class="col-1-4"><label>Normal:</label></div>
      <div class="col-1-4"><input type="checkbox" value="http://" id="rec-nor" /></div>
      <div class="col-1-4"><input type="button" value="Remove" id="rec-del" /></div>
    </div>
  </div>
<div class="col-1-2">
  <div class="row">
    <div class="col-1">
      <span id="x-sli-s"><input type="range" id="x-sli" value="0.0" disabled /></span>
    </div>
  </div>
  <div class="row">
    <div class="col-1-7">
      <div class="row"><label>Axis</label></div>
      <div class="row"><label>X:</label></div>
      <div class="row"><label>Y:</label></div>
      <div class="row"><label>Z:</label></div>
    </div>
    <div class="col-1-7">
      <div class="row"><label>Position</label></div>
      <div class="row"><input type="number" id="x-pos" value="0.0" readonly /></div>
      <div class="row"><input type="number" id="y-pos" value="0.0" readonly /></div>
      <div class="row"><input type="number" id="z-pos" value="0.0" readonly /></div>
    </div>
    <div class="col-1-7">
      <div class="row"><label>Velocity</label></div>
      <div class="row"><input type="number" id="x-vel" value="0.0" readonly /></div>
      <div class="row"><input type="number" id="y-vel" value="0.0" readonly /></div>
      <div class="row"><input type="number" id="z-vel" value="0.0" readonly /></div>
    </div>
    <div class="col-1-7">
      <div class="row"><label>Acceleration</label></div>
      <div class="row"><input type="number" id="x-acc" value="0.0" readonly /></div>
      <div class="row"><input type="number" id="y-acc" value="0.0" readonly /></div>
      <div class="row"><input type="number" id="z-acc" value="0.0" readonly /></div>
    </div>
    <div class="col-1-7">
      <div class="row"><label>Threshold</label></div>
      <div class="row"><input type="number" id="x-thr" value="0.0" /></div>
    </div>
  </div>

```

```

<div class="row"><input type="number" id="y-thr" value="0.0" /></div>
<div class="row"><input type="number" id="z-thr" value="0.0" /></div>
</div>
<div class="col-1-7">
    <div class="row"><label>Alert</label></div>
    <div class="row"><input type="checkbox" id="x-ale" /></div>
    <div class="row"><input type="checkbox" id="y-ale" /></div>
    <div class="row"><input type="checkbox" id="z-ale" /></div>
</div>
<div class="col-1-7">
    <div class="row"><label>Fusion</label></div>
    <div class="row"><input type="checkbox" id="x-fus" /></div>
    <div class="row"><input type="checkbox" id="y-fus" /></div>
    <div class="row"><input type="checkbox" id="z-fus" /></div>
</div>
</div>
<div class="col-1-4">
    <ul class="log"></ul>
</div>
</div>

<div class="col-1-3">
    <div class="row"><canvas id="cv-xp"></canvas></div>
    <div class="row"><canvas id="cv-yp"></canvas></div>
    <div class="row"><canvas id="cv-zp"></canvas></div>
</div>
<div class="col-1-3">
    <div class="row"><canvas id="cv-xv"></canvas></div>
    <div class="row"><canvas id="cv-yv"></canvas></div>
    <div class="row"><canvas id="cv-zv"></canvas></div>
</div>
<div class="col-1-3">
    <div class="row"><canvas id="cv-xa"></canvas></div>
    <div class="row"><canvas id="cv-ya"></canvas></div>
    <div class="row"><canvas id="cv-za"></canvas></div>
</div>

<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.4/Chart.min.js"></script>
<script>
    var xslis = document.getElementById("x-sli-s")
    var xsli = document.getElementById("x-sli")
    var xpos = document.getElementById("x-pos")
    xpos.onchange = function () { xsli.value = this.value }
    xslis.onmouseenter = function () { xpos.style.backgroundColor = "#cfc" }
    xslis.onmouseleave = function () { xpos.style.backgroundColor = "#fff" }

    var charts = {
        X: [
            initChart('xp', [0, 255, 0], 'Position on X-axis'),
            initChart('xv', [0, 127, 127], 'Velocity on X-axis'),
            initChart('xa', [0, 0, 255], 'Acceleration on X-axis')
        ],
        Y: [
            initChart('yp', [0, 255, 0], 'Position on Y-axis'),
            initChart('yv', [0, 127, 127], 'Velocity on Y-axis'),
            initChart('ya', [0, 0, 255], 'Acceleration on Y-axis')
        ]
    }

```

```

        ],
        Z: [
            initChart('zp', [0, 255, 0], 'Position on Z-axis'),
            initChart('zv', [0, 127, 127], 'Velocity on Z-axis'),
            initChart('za', [0, 0, 255], 'Acceleration on Z-axis')
        ]
    }

    function log(text) {
        var messages = document.getElementsByClassName('log')[0],
            message = document.createElement('li'),
            content = document.createTextNode(text);
        message.appendChild(content);
        messages.appendChild(message);
    }

    function updateChart(chart, time, val, err) {
        chart.data.datasets[0].data.push({ x: time, y: val });
        chart.data.datasets[1].data.push({ x: time, y: val - err });
        chart.data.datasets[2].data.push({ x: time, y: val + err });
        if (chart.data.datasets[0].data.length > 200) {
            chart.data.datasets[0].data.shift();
            chart.data.datasets[1].data.shift();
            chart.data.datasets[2].data.shift();
        }
        chart.update();
    }

    function addRecord(data) {
        var records = { X: [], Y: [], Z: [] };
        for (var i = 0; i < data.length; i++) {
            var record = data[i].split(',');
            if (record[0] in records && record.length == 8) {
                records[record.shift()] = record;
            }
        }
        console.log(records);

        if (records.X.length > 0) {
            for (var i = 0; i < 3; i++) {
                updateChart(charts.X[i], Number(records.X[0]), Number(records.X[i + 1]),
                    Math.sqrt(records.X[i + 4]));
            }
        }
        if (records.Y.length > 0) {
            for (var i = 0; i < 3; i++) {
                updateChart(charts.Y[i], Number(records.Y[0]), Number(records.Y[i + 1]),
                    Math.sqrt(records.Y[i + 4]));
            }
        }
        if (records.Z.length > 0) {
            for (var i = 0; i < 3; i++) {
                updateChart(charts.Z[i], Number(records.Z[0]), Number(records.Z[i + 1]),
                    Math.sqrt(records.Z[i + 4]));
            }
        }
    }

    function initChart(id, c, t) {
        return new Chart(document.getElementById('cv-' + id).getContext('2d'), {

```

```

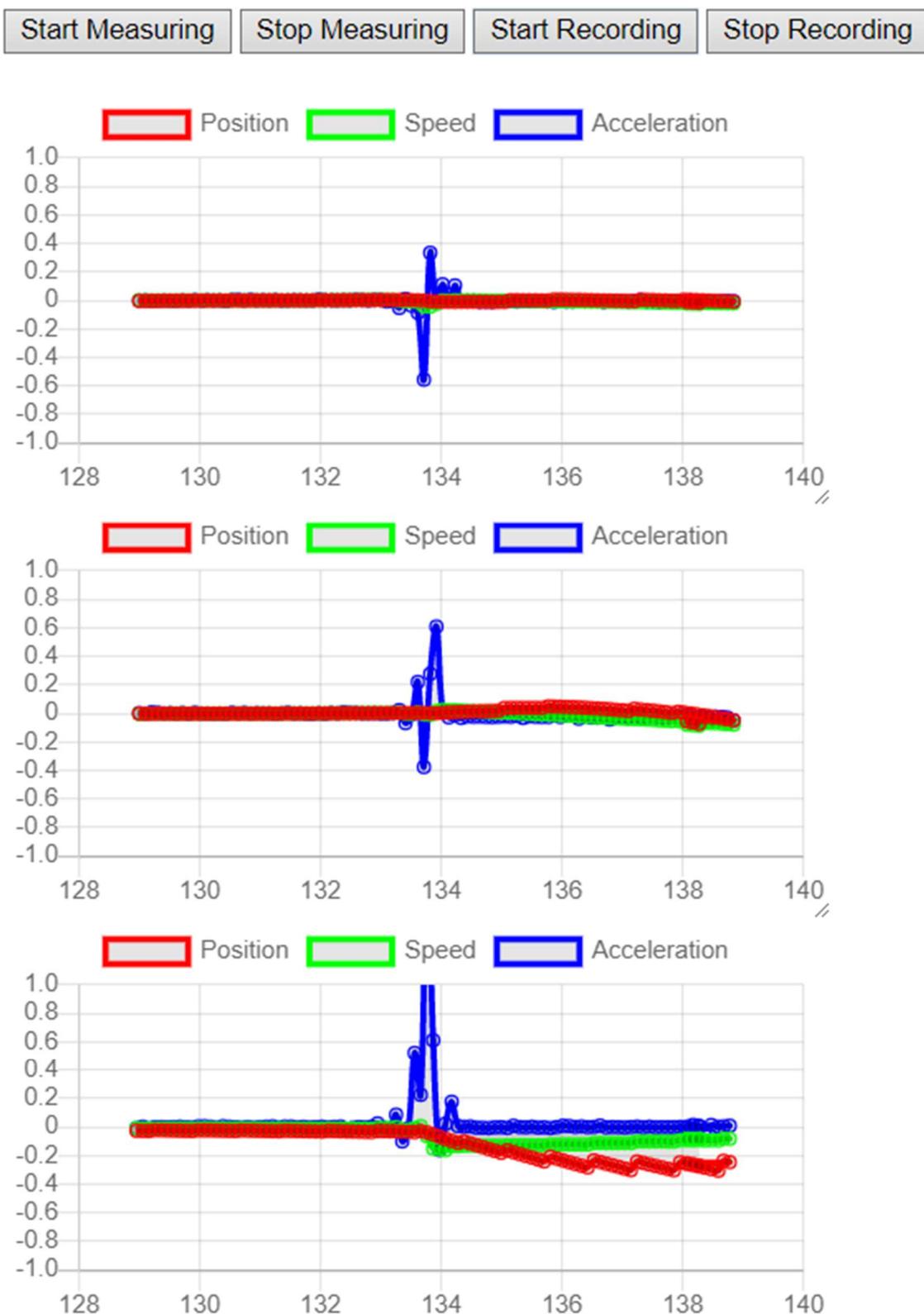
type: 'line',
data: {
    datasets: [
        { borderColor: 'rgb(' + c[0] + ', ' + c[1] + ', ' + c[2] + ', 1.0)', data: [], fill: false },
        { borderColor: 'rgb(' + c[0] + ', ' + c[1] + ', ' + c[2] + ', 0.1)', data: [], fill: 0 },
        { borderColor: 'rgb(' + c[0] + ', ' + c[1] + ', ' + c[2] + ', 0.1)', data: [], fill: 0 }
    ]
},
options: {
    responsive: true,
    maintainAspectRatio: false,
    title: { display: true, text: t },
    legend: { display: false },
    elements: { point: { radius: 0 } },
    animation: { duration: 0 },
    scales: {
        yAxes: [{ type: 'linear', stepSize: 0.1 }],
        xAxes: [{ type: 'linear' }]
    }
}

const socket = new WebSocket("ws://localhost:5678");
socket.onopen = function (e) {
    socket.send("connect");
    log(`[open] Connection established`);
};
socket.onmessage = function (e) {
    var lines = event.data.split('\n');
    if (lines[0] === 'record' && lines.length > 1) {
        lines.shift()
        addRecord(lines);
    }
    log(`[message] ${event.data}`);
};
socket.onclose = function (e) {
    if (event.wasClean) {
        log(`[close] Connection closed, code: ${e.code}, reason: ${e.reason}`);
    }
    else {
        log(`[close] Connection died`);
    }
};
socket.onerror = function (e) {
    log(`[error] ${e.message}`);
};

</script>
</body>

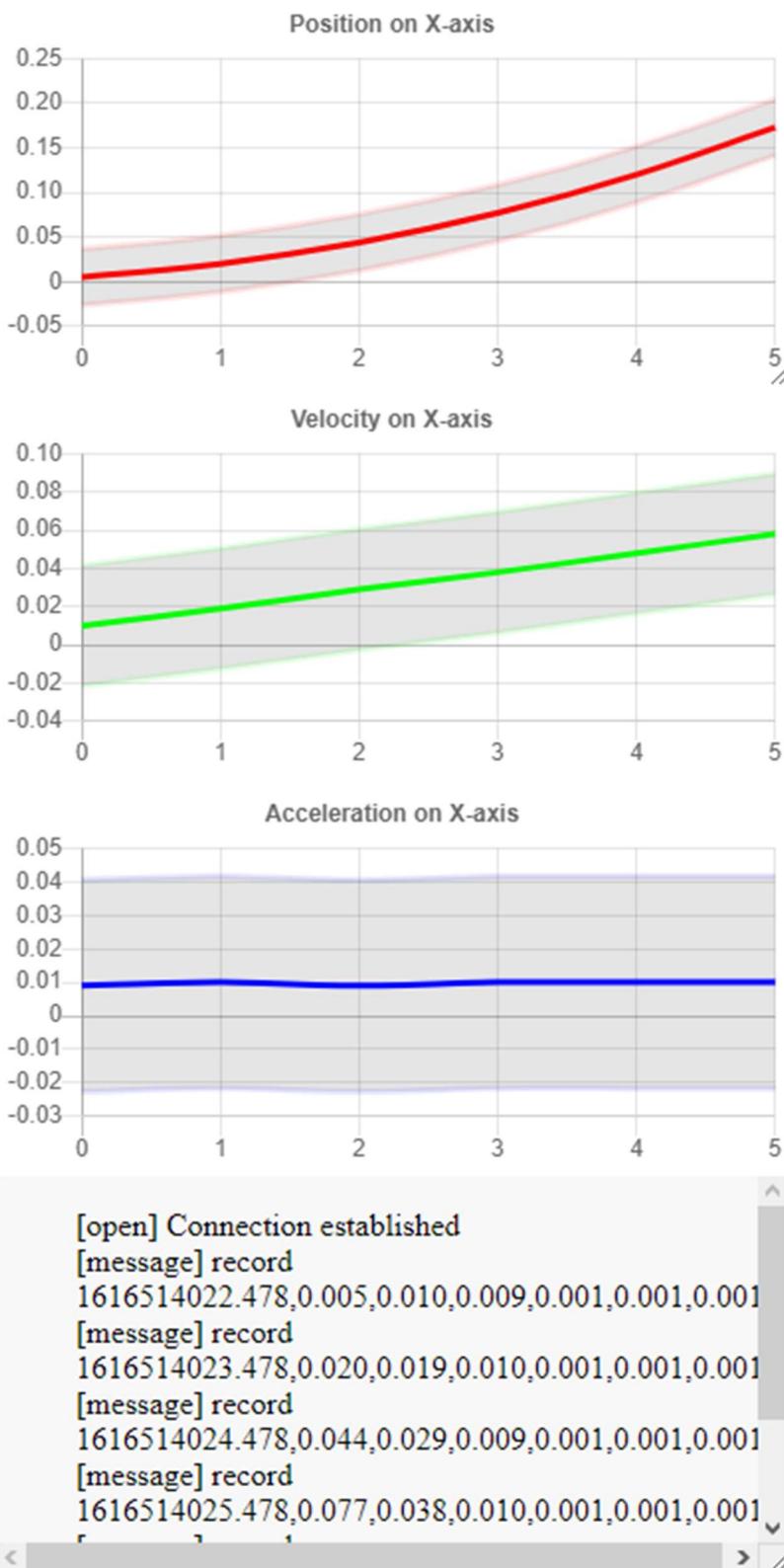
```

4.2.2. Felhasználói felület:

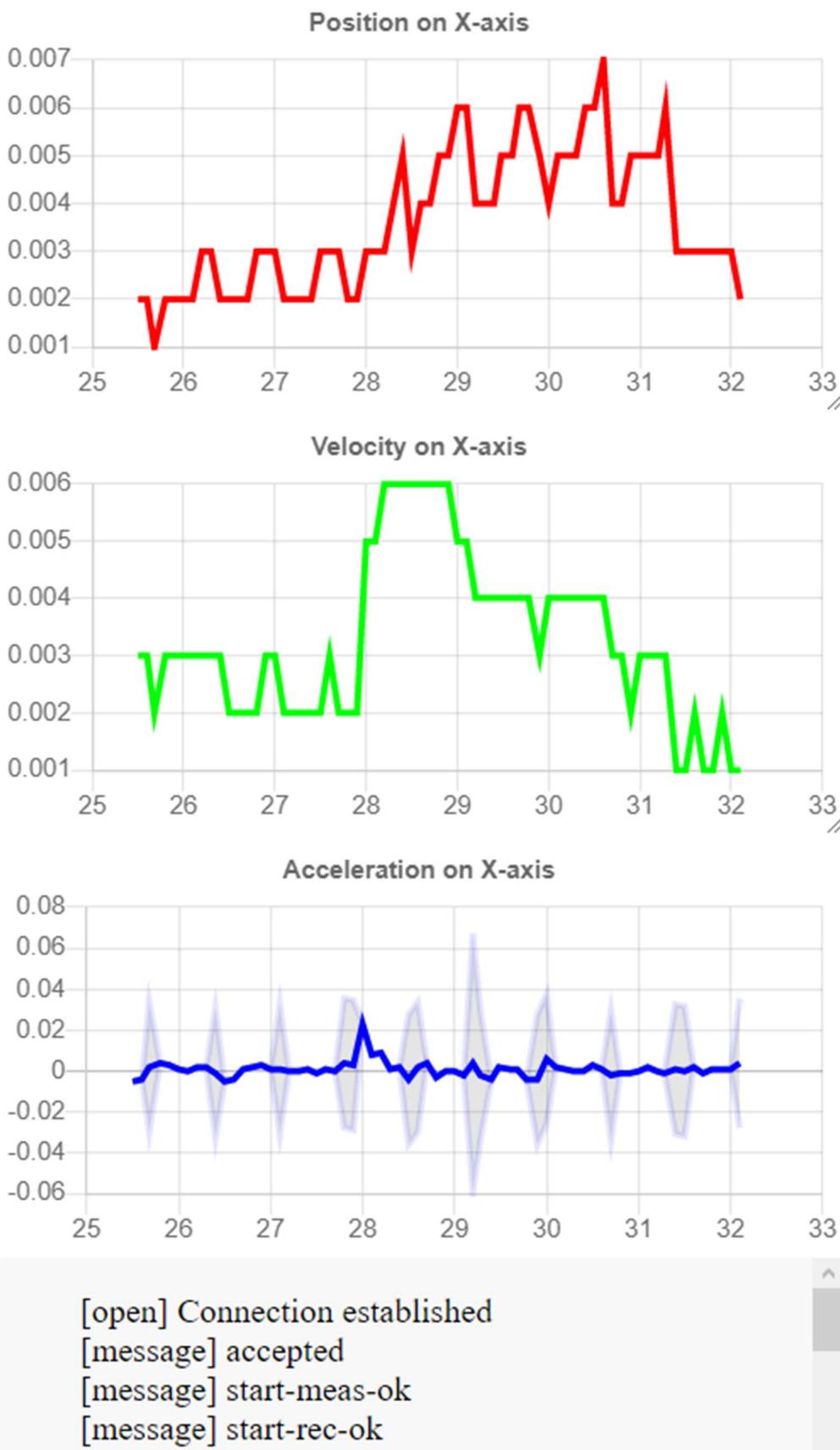


76. ábra verzió 1.0 tesztmérés felület

Start Measuring **Stop Measuring** **Start Recording** **Stop Recording**

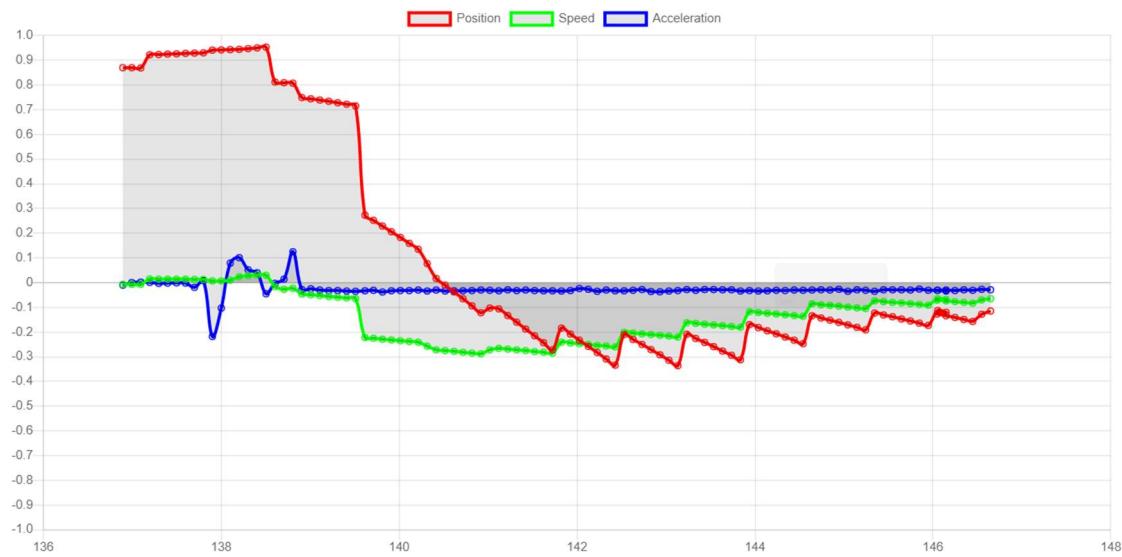


77. ábra verzió 1.0 tesztmérés felület szeparált értékekkel



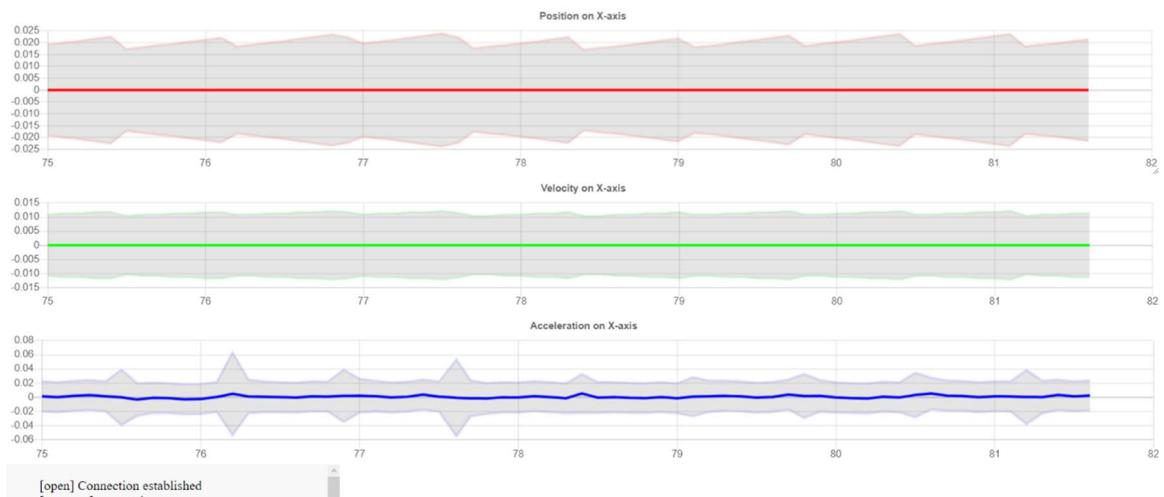
78. ábra verzió 1.0 tesztmérés felület szűrt értékek

[Start Measuring](#) [Stop Measuring](#) [Start Recording](#) [Stop Recording](#)



79. ábra verzió 1.1 tesztmérés felület

[Start Measuring](#) [Stop Measuring](#) [Start Recording](#) [Stop Recording](#)

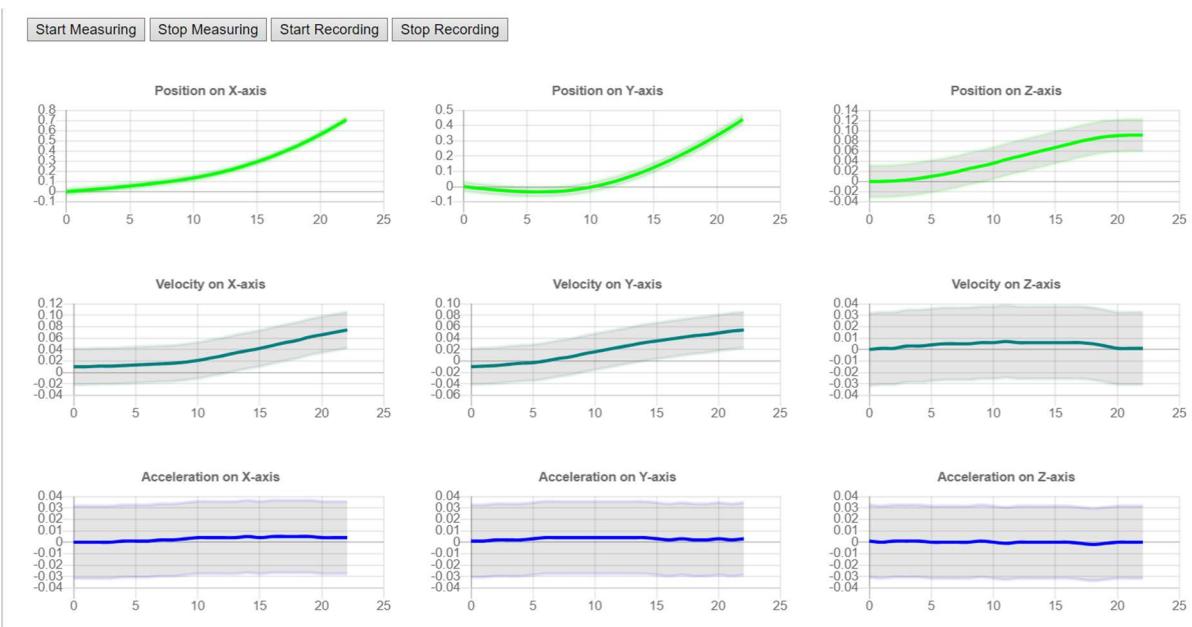


80. ábra verzió 1.1 tesztmérés felület

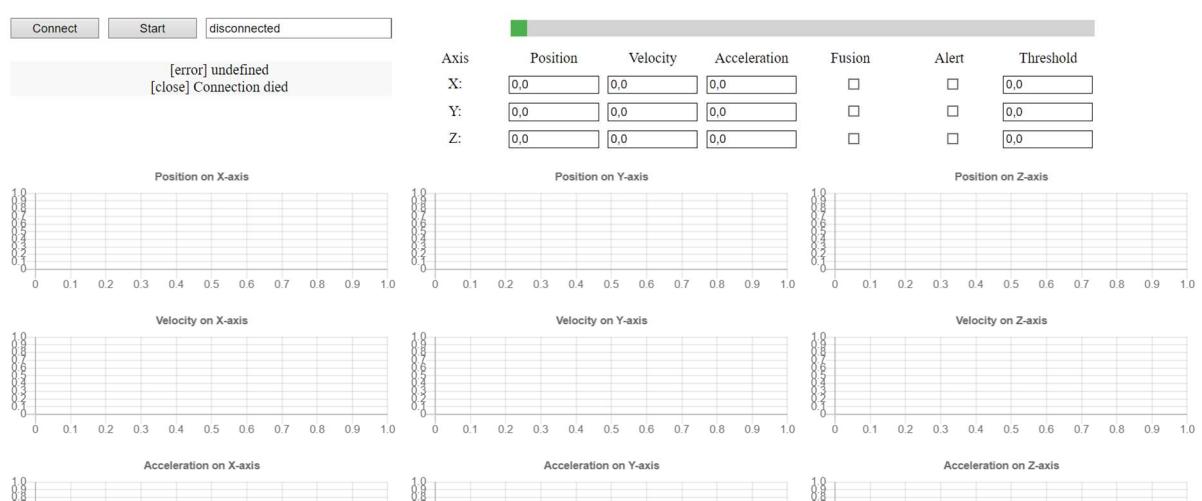


	Pos	Vel	Acc	Fusion	Alert	Threshold
X-axis:	0,0	0,0	0,0	<input type="checkbox"/>	<input type="checkbox"/>	0,0
Y-axis:	0,0	0,0	0,0	<input type="checkbox"/>	<input type="checkbox"/>	0,0
Z-axis:	0,0	0,0	0,0	<input type="checkbox"/>	<input type="checkbox"/>	0,0

81. ábra verzió 2.0 mérés felület darupálya pozíció jelzéssel, alert funkcióval



82. ábra verzió 2.0 tmérés felület a három tengelyen mért adatokkal és a hozzá tartozó limit megjelenítéssel



83. ábra verzió 2.0 tmérés felület, kapcsolódás előtti állapotban, hibaüzenettel

4.3. Szenzorfúziós elemek firmware tervezése

Működés:

1. Movix mérőmodul bekapcsolásakor (startup):
 - a. Ellenőrzi a flash tartós tár állapotát.
 - b. Elindítja a WIFI modult és csatlakozik a beállított hálozathoz.
 - c. Elindítja a hálozati kommunikációért felelős szervert.
 - d. Elindítja a gyorsulásmérő driverét.
 - e. Elindítja a távolságmérő driverét.
 - f. Beállítja a mérési kezdőpontot.
2. Movix mérőmodul bekapcsolása után (run):
 - a. A szerver kezeli a bejövő kéréseket.
 - b. Elindítja a pozíció mérést.
 - c. Leállítja a pozíció mérést.
 - d. Elindítja a pozíció rögzítését.
 - e. Leállítja a pozíció rögzítését.
 - f. Beállítja a működési paramétereket.

rf_hal

Az M88 távolságmérő szenzor hardware absztrakciós rétege. A Movix és a szenzor dokumentációja alapján biztosítja vele a kommunikációt UART-on keresztül. Támogatja több távolságmérő soros bekötösét, ilyenkor az eszközöknek különböző címmel kell rendelkeznie. A könyvtár nem szálbiztos.

Az UART driver elindítása.

Az UART kommunikáció automatikus beállítása.

Az eszköz HW verziójának lekérdezése.

Az eszköz SW verziójának lekérdezése.

Az eszköz sorozatszámának lekérdezése.

Az eszköz feszültségszintjének lekérdezése.

Az eszköz állapotának lekérdezése.

Az eszköz címének beállítása lekérdezése.

A mérési offset beállítása.

A mérési eredmény lekérdezése.

A egyszeri mérés elvégzése.

A teljes mérés indítása.

A folyamatos mérés indítása.

A folyamatos mérés leállítása.

ax_hal

Az ADXL355 gyorsulásmérő szenzor hardware absztrakciós rétege. A Movix és a szenzor dokumentációja alapján biztosítja vele a kommunikációt SPI-on keresztül. A gyorsulásmérő beágyazott szenzor. Megszakítást biztosít az eszköztől érkező vezérlőjelek feldolgozásához. A könyvtár nem szálbiztos.

Az SPI driver elindítása.

Az eszköz alapállapotba történő visszaállítása.

Az eszköz AD verziójának lekérdezése.

Az eszköz MEMS AD verziójának lekérdezése.

Az eszköz azonosítójának lekérdezése.

Az eszköz revizó számának lekérdezése.

Az eszköz állapotának lekérdezése.

Az aktuális hőmérsékleti adat lekérdezése.

Az aktuális gyorsulási adatok lekérdezése.

A FIFO adattároló olvasása.

A FIFO adattároló elemszámának lekérdezése.

A mérési offset lekérdezése és beállítása.

A mérési aktivitás érzékelésének lekérdezése és beállítása

A mérési szűrő lekérdezése és beállítása.

A mérési vezérlőjelek lekérdezés és beállítása.

A mérési tartomány lekérdezése és beállítása.

A mérés elindítása.

A mérés leállítása.

filter

A többszenzoros mérési adatok feldolgozásához, fúziójához biztosít Kálmán-szűrő implementációt. A szűrő az adatokat mátrixokban tárolja, a műveletek ezen végzi el. Az optimalizált mátrix implementációt az ESP-DSP jelfeldolgozásra szolgáló könyvtár biztosítja, hogy nagyfrekenciájú vagy nagyméretű adatrögzításnél se okozzon problémát a feldolgozási idő. A következő állapot megjósolása. (predict) A következő állapot becslése megfigyelés alapján. (estimate)

wifi

A wifi modul megfelelő elindításáért és beállításáért felel. A Movix képes csatlakozni beállított hálózathoz. Ha a csatlakozás sikertelen, akkor saját hálózatot indít el. Így minden esetben elérhező lesz az eszköz wifin keresztül.

server

A hálózati kommunikációért felel. A Movix a startup folyamatának utolsó lépéseként indítja el. A szerver több különböző kapcsolódási végpontot kínál. Az alapértelmezett címre csatlakozva, a kliens egy websocket kapcsolatot létesít, melyet a szerver visszaigazol. Ezt követően a weboldalon végrehajtott műveletek általi kommunikáció a websocketen keresztül történik. A websocket célja, hogy a nagymennyiségű, nagyfrekvenciájú mérési adatok küldésekor elejtével vegyük az overheadnek (amit HTTP jelentene). De lehetőség van aktív és passzív HTTP lekérdezésre is, hogy tetszőleges adatbázisba vagy adatfeldolgozóba kerüljenek a mérési adatok. A websocketen bejövő mérési adatokat a kliens is feldolgozza Chart.js segítségével. A meas-t és wifi-t használja. A kezelőfelület javascriptre épülő html oldal. Ezáltal platformfüggetlen, csak egy böngészőre van szükség. Saját hálozaton csak korlátozottan működik, a kezelőfelület teljes működéséhez internetkapcsolatra van szükség.

meas

A mérést vezérlő project specifikus rész. Rendelkezik egy szűrővel, mely a Movix aktuális állapotát határozza meg és rögzíti. Beállítja a szűrő kezedeti állapotát és működésének dinamikus modelljét. Biztosítja a szenzorok elindítását és beállítását. Mérést végző és rögzítő szálak elindítását és leállítását. A filter-t, ax_hal-t és rf_hal-t használja.

main

A firmware elindítását, a startupot hajtja végre. A meas-t, wifi-t és server-t használja.

4.3.1. Kód

4.3.1.1. Main code:

```
#include "nvs_flash.h"  
  
#include "freertos/FreeRTOS.h"  
  
#include "freertos/task.h"
```

```
#include "wifi.h"
#include "meas.h"
#include "server.h"

#define ABORT(e) if (ESP_OK != e) return

void app_main(void)

    esp_err_t ret = nvs_flash_init();

    if      (ret      ==      ESP_ERR_NVS_NO_FREE_PAGES      ||      ret      ==
ESP_ERR_NVS_NEW_VERSION_FOUND)

        ESP_ERROR_CHECK(nvs_flash_erase());

        ESP_ERROR_CHECK(nvs_flash_init());

    ABORT(WifiInit(WIFI_INIT_MODE_STA));

FilterInit();
AxInit();
RfInit();
ServerStart();
```

4.3.1.2. Server code:

```
#define LOG_LOCAL_LEVEL ESP_LOG_INFO

#include "esp_log.h"

#include "esp_http_server.h"

#include "stdio.h"

#include " cJSON.h"

#include "freertos/FreeRTOS.h"

#include "freertos/task.h"

#include " wifi.h"

#include "meas.h"

#include "server.h"

static const char *TAG = "server";

#define SERVER_REC_WS 1u

#define SERVER_HTML "<!DOCTYPE html>\n"
    "<head>\n"
    "  <title>MUI</title>\n"
    "  <script type=\"module\">\n"
    "    import {InfluxDB, Point} from\n"
    "'https://unpkg.com/@influxdata/influxdb-client/dist/index.browser.mjs';\n"
    "    const url = 'http://localhost:8086';\n"

```

```

    " const token = "
'1cPmbJFojb_1m6GlzFu1iFaRtUZ9WW_r_r8VOGWXzPUgXaQouJKK-
vl80qL82frmX_E7J68BbeEf74cYLtN6xA==';\n  \
    "   const org ='AIRobotix';\n" \
    "   const bucket ='Movix';\n" \
    "   const writeApi = new InfluxDB({url, token}).getWriteApi(org, bucket,
'ms');\n" \
    "   const socket = new WebSocket(`ws:///${url}/ws`);\n" \
    "   socket.onopen = function(e) {\n" \
    "     socket.send(`connect`);\n" \
    "     msg(`[open] Connection established`);\n" \
    "   };\n" \
    "   socket.onmessage = function(event) {\n" \
    "     var lines = event.data.split(`\n`);\n" \
    "     if (lines[0] == `record` ) {\n" \
    "       var d = new Date();\n" \
    "       var t = d.getTime();\n" \
    "       lines.slice(1).forEach(l => l + ` ${t}`);\n" \
    "       writeApi.writeRecords(lines.slice(1));\n" \
    "       writeApi.flush().\n" \
    "         then(() => {console.log('FINISHED')}).catch(e =>
{console.error(e); console.log(`\nERROR`)});\n" \
    "     }\n" \
    "   else {\n" \
    "     msg(`[message] ${event.data}`);\n"

```

```

    "      }\n"
    "      };\n"
    "      socket.onclose = function(event) {\n"
    "          if(event.wasClean) {\n"
    "              msg(`[close] Connection closed, code: ${event.code}, reason:\n${event.reason}`);\n"
    "          }\n"
    "          else {\n"
    "              msg(`[close] Connection died`);\n"
    "          }\n"
    "          writeApi.close();\n"
    "      };\n"
    "      socket.onerror = function(error) {\n"
    "          msg(`[error] ${error.message}`);\n"
    "      };\n"
    "      window.socket = socket;\n"
    "      function msg(text) {\n"
    "          var messages = document.getElementsByTagName('ul')[0],\n"
    "              message = document.createElement('li'),\n"
    "              content = document.createTextNode(text);\n"
    "          message.appendChild(content);\n"
    "          messages.appendChild(message);\n"
    "      }\n"

```

```

    "  </script>\n"
    "</head>\n"
    "<body>\n"
    "  <h1>Movix User Interface</h1>\n"
    "    <button  type=\"button\"  onclick=\"socket.send('start-ax')\">Start
Ax</button>\n"          \
    "    <button  type=\"button\"  onclick=\"socket.send('stop-ax')\">Stop
Ax</button>\n"          \
    "    <button  type=\"button\"  onclick=\"socket.send('start-rf')\">Start
Rf</button>\n"          \
    "    <button  type=\"button\"  onclick=\"socket.send('stop-rf')\">Stop
Rf</button>\n"          \
    "    <button  type=\"button\"  onclick=\"socket.send('start-rec')\">Start
Rec</button>\n"          \
    "    <button  type=\"button\"  onclick=\"socket.send('stop-rec')\">Stop
Rec</button><br>\n"      \
    "  <div style=\"height: 400px; width: 800px; overflow: auto; background:
#f8f8f8;\">\n"
    "    <ul style=\"list-style-type:none;\"></ul>\n"
    "</div>\n"
    "</body>"

```

typedef struct server

```

TaskHandle_t wsTask;

httpd_handle_t handle;

int websocket;

int connected;

```

```
char *html;

} dtServer;

static dtServer *server = NULL;

static const char dirs[] = {'X', 'Y', 'Z'};

static void WsSendAsync(void *arg)

    httpd_ws_frame_t *frame = (httpd_ws_frame_t *)arg;

    esp_err_t ret = httpd_ws_send_frame_async(server->handle, server->websocket,
frame);

    if (ret != ESP_OK)

        ESP_LOGW(TAG, "sending websocket frame failed, %s", esp_err_to_name(ret));

    free(frame->payload);

    free(frame);

static void WsTask(void *arg)

    float state[9u] = {0};

    int64_t last = 0;

    int64_t time = 0;

    int length;

    char record[512];

    ESP_LOGI(TAG, "websocket record start");

    while (pdFALSE == xTaskNotifyWait(0u, 0u, NULL, 0u))

        FilterCurrentState(state, &time);

        if (time > last)

            memset(record, 0, sizeof(record));

            length = sprintf(record, "record\n");
```

```

for (int dir = 0; dir < 3; dir++)

    length      +=      sprintf(&record[length],      "state,axis=%'c\"
pos=%0.3f,vel=%0.3f,acc=%0.3f\n", dirs[dir], state[dir], state[dir + 3], state[dir + 6]);

    last = time;

    ESP_LOGD(TAG, "new record, length: %d\n%s", length, record);

    httpd_ws_frame_t *frame = calloc(1, sizeof(httpd_ws_frame_t));

    frame->type = HTTPD_WS_TYPE_TEXT;

    frame->len = length;

    frame->payload = calloc(length, sizeof(uint8_t));

    memcpy(frame->payload, record, length);

    esp_err_t ret = httpd_queue_work(server->handle, WsSendAsync, (void
*)frame);

    if (ret != ESP_OK)

        ESP_LOGW(TAG, "queing websocket frame failed, %s",
esp_err_to_name(ret));

        vTaskDelay(pdMS_TO_TICKS(1000));

    ESP_LOGI(TAG, "websocket record stop");

    vTaskDelete(NULL);

static esp_err_t RecordStart(uint32_t type)

    esp_err_t err = ESP_ERR_INVALID_STATE;

    switch (type)

        case SERVER_REC_WS:

            if (server->wsTask == NULL)

                err = (pdPASS == xTaskCreate(WsTask, "server-ws-task", 8192, NULL, 2,
&server->wsTask) ? ESP_OK : ESP_ERR_NO_MEM);

```

```
break

default:

ESP_LOGW(TAG, "start failed, invalid arg");

err = ESP_ERR_INVALID_ARG;

break;

return err;

static esp_err_t RecordStop(uint32_t type)

esp_err_t err = ESP_ERR_INVALID_STATE;

switch (type)

case SERVER_REC_WS:

    if (server->wsTask != NULL)

        xTaskNotify(server->wsTask, 0u, eNoAction);

        vTaskDelay(pdMS_TO_TICKS(1000));

        server->wsTask = NULL;

        err = ESP_OK;

    break;

default:

ESP_LOGW(TAG, "stop failed, invalid arg");

err = ESP_ERR_INVALID_ARG;

break;

return err;

static esp_err_t ServerWebsocketHandler(httpd_req_t *req)

httpd_ws_frame_t frame;
```

```

memset(&frame, 0, sizeof(httpd_ws_frame_t));

frame.payload = (uint8_t *)calloc(64, sizeof(uint8_t));

esp_err_t ret = httpd_ws_recv_frame(req, &frame, 64);

if (ESP_OK != ret)

    ESP_LOGE(TAG, "websocket receive failed.");

    return ret;

ESP_LOGD(TAG, "websocket frame received");

ESP_LOGD(TAG, "\tType: %d", frame.type);

ESP_LOGD(TAG, "\tLength: %d", frame.len);

ESP_LOGD(TAG, "\tMessage: %s", frame.payload);

if (frame.type == HTTPD_WS_TYPE_TEXT)

    char *response;

    if (0 == server->connected)

        if (0 == strcmp((char *)frame.payload, "connect"))

            response = "accepted";

            server->websocket = httpd_req_to_sockfd(req);

            server->connected = 1;

    else

        response = "invalid";

    ret = ESP_FAIL;

else

    if (0 == strcmp((char *)frame.payload, "start-rf"))

        response = (ESP_OK == MeasStart(MEAS_SOURCE_RF) ? "start-rf-ok" :

"start-rf-fail");

```

```

else if (0 == strcmp((char *)frame.payload, "stop-rf"))

    response = (ESP_OK == MeasStop(MEAS_SOURCE_RF) ? "stop-rf-ok" :
"stop-rf-fail");

else if (0 == strcmp((char *)frame.payload, "start-ax"))

    response = (ESP_OK == MeasStart(MEAS_SOURCE_AX) ? "start-ax-ok" :
"start-ax-fail");

else if (0 == strcmp((char *)frame.payload, "stop-ax"))

    response = (ESP_OK == MeasStop(MEAS_SOURCE_AX) ? "stop-ax-ok" :
"stop-ax-fail");

else if (0 == strcmp((char *)frame.payload, "start-rec"))

    response = (ESP_OK == RecordStart(SERVER_REC_WS) ? "start-rec-ok" :
"start-rec-fail");

else if (0 == strcmp((char *)(frame.payload), "stop-rec"))

    response = (ESP_OK == RecordStop(SERVER_REC_WS) ? "stop-rec-ok" :
"stop-rec-fail");

else

    response = "invalid";

    ret = ESP_FAIL;

    server->websocket = -1;

    server->connected = -1;

    frame.type = HTTPD_WS_TYPE_TEXT;

    frame.len = strlen(response);

    memset(frame.payload, 0, sizeof(64));

    memcpy(frame.payload, response, frame.len)

    httpd_ws_send_frame(req, &frame);

    free(frame.payload)

```

```
return ESP_OK;

static esp_err_t ServerRequestHandler(httpd_req_t *req)
    return httpd_resp_send(req, server->html, strlen(server->html));

esp_err_t ServerStart()

    static const httpd_uri_t ws = {
        .uri = "/ws",
        .method = HTTP_GET,
        .handler = ServerWebSocketHandler,
        .user_ctx = NULL,
        .is_websocket = true,
    };

    static const httpd_uri_t get = {
        .uri = "/",
        .method = HTTP_GET,
        .handler = ServerRequestHandler,
        .user_ctx = NULL,
    };

    esp_err_t err = ESP_OK;
    if (NULL != server)
        ESP_LOGW(TAG, "already started");
    err = ESP_ERR_INVALID_STATE;
    else
        server = (dtServer *)calloc(1u, sizeof(dtServer));
    if (NULL == server)
```

```
ESP_LOGE(TAG, "not enough memory");

err = ESP_ERR_NO_MEM;

else

    httpd_config_t config = HTTPD_DEFAULT_CONFIG();

    err = httpd_start(&server->handle, &config);

    if (ESP_OK == err)

    {

        ESP_LOGI(TAG, "started");

        httpd_register_uri_handler(server->handle, &ws);

        httpd_register_uri_handler(server->handle, &get);

        server->wsTask = NULL;

        server->websocket = -1;

        server->connected = 0;

        server->html = (char *)calloc(strlen(SERVER_HTML) + 16, sizeof(char));

        int len = sprintf(server->html, SERVER_HTML, GetIPAddress());

        server->html[len] = '\0';

        ESP_LOGD(TAG, "SERVER %s HTML %s (%d/%d)\n%s",

strlen(SERVER_HTML), strlen(server->html), server->html);

        else

            ESP_LOGE(TAG, "start httpd failed");

            free(server);
```

```
return err

void ServerStop()

if (NULL != server)

    httpd_stop(server->handle);

    free(server->html);

    server = NULL;

ESP_LOGI(TAG, "stopped");
```

4.3.1.3. Wifi code:

```
#define LOG_LOCAL_LEVEL ESP_LOG_INFO

#include "esp_log.h"

#include "esp_wifi.h"

#include "string.h"

#include "wifi.h"

#include "freertos/FreeRTOS.h"

#include "freertos/task.h"

#define WIFI_STA_SSID "Hungary38"

#define WIFI_STA_PASS "RoaghanET"

#define WIFI_STA_RETRY 3u

#define WIFI_AP_SSID "Movix"

#define WIFI_AP_PASS "0123456789"

#define WIFI_AP_CONN 3u
```

```
static const char *TAG = "wifi";  
  
static uint8_t retry = 0u;  
  
static char ip[16] = {0};  
  
static esp_netif_t *apNetif = NULL;  
  
static esp_netif_t *staNetif = NULL;  
  
static esp_event_handler_instance_t wifiEventHandler = NULL;  
  
static esp_event_handler_instance_t ipEventHandler = NULL;  
  
  
static void WifiEventHandler(void *arg, esp_event_base_t base, int32_t id, void *event)  
{  
    TaskHandle_t task = (TaskHandle_t)arg;  
  
    switch (id)  
    {  
        case WIFI_EVENT_STA_START:  
            ESP_LOGD(TAG, "connecting to the AP");  
  
            esp_wifi_connect();  
  
            break;  
  
        case WIFI_EVENT_STA_CONNECTED:  
            retry = 0u;  
  
            ESP_LOGI(TAG, "connected");  
  
            xTaskNotify(task, 0u, eNoAction);  
  
            break;  
  
        case WIFI_EVENT_STA_DISCONNECTED:  
            if (retry < WIFI_STA_RETRY)  
            {  
                // Connection failed, attempt to re-connect  
                esp_wifi_connect();  
            }  
            else  
            {  
                // Maximum number of retries reached, exit loop  
                break;  
            }  
    }  
}
```

```

retry++;

ESP_LOGD(TAG, "reconnecting... (%d/%d)", retry, WIFI_STA_RETRY);

esp_wifi_connect();

else

    ESP_LOGD(TAG, "disconnected");

    xTaskNotify(task, 1u, eSetBits);

break;

case WIFI_EVENT_AP_START:

    xTaskNotify(task, 0u, eNoAction);

break;

case WIFI_EVENT_AP_STACONNECTED:

    ESP_LOGD(TAG, "station      " MACSTR      "      join,      AID=%d",
MAC2STR(((wifi_event_ap_staconnected_t *)event)->mac), ((wifi_event_ap_staconnected_t
*)event)->aid);

break;

case WIFI_EVENT_AP_STADISCONNECTED:

    ESP_LOGD(TAG, "station      " MACSTR      "      leave,      AID=%d",
MAC2STR(((wifi_event_ap_stadisconnected_t
*)event)->mac),
((wifi_event_ap_stadisconnected_t *)event)->aid);

break;

default:

    ESP_LOGD(TAG, "wifi event, id: %d", id);

break;
}

static void IpEventHandler(void *arg, esp_event_base_t base, int32_t id, void *event)

```

```

switch (id)

case IP_EVENT_STA_GOT_IP:

    sprintf(ip, IPSTR, IP2STR(&((ip_event_got_ip_t *)event)->ip_info.ip));

    ESP_LOGI(TAG, "ip %s", ip);

    break;

case IP_EVENT_STA_LOST_IP:

    ESP_LOGI(TAG, "lost ip");

    break;

default:

    ESP_LOGD(TAG, "ip event, id: %d", id);

    break;

static void WifiStartAp()

    ESP_LOGI(TAG, "start wifi in access point mode");

    ESP_LOGI(TAG, "ssid: %s, password: %s", WIFI_AP_SSID, WIFI_AP_PASS);

    wifi_config_t apConfig = {

        .ap = {

            .ssid = WIFI_AP_SSID,

            .ssid_len = strlen(WIFI_AP_SSID),

            .password = WIFI_AP_PASS,

            .max_connection = WIFI_AP_CONN,

            .authmode = WIFI_AUTH_WPA_WPA2_PSK,

            esp_wifi_set_mode(WIFI_MODE_AP);

            esp_wifi_set_config(ESP_IF_WIFI_AP, &apConfig);

            esp_wifi_start();

```

```
esp_netif_ip_info_t ipInfo;
esp_netif_get_ip_info(apNetif, &ipInfo);
sprintf(ip, IPSTR, IP2STR(&ipInfo.ip));
ESP_LOGI(TAG, "ip %s", ip);

static void WifiStartSta()
ESP_LOGI(TAG, "start wifi in station mode");
ESP_LOGI(TAG, "ssid: %s, password: %s", WIFI_STA_SSID, WIFI_STA_PASS);
wifi_config_t staConfig = {
    .sta = {
        .ssid = WIFI_STA_SSID,
        .password = WIFI_STA_PASS,
        .threshold = {
            .authmode = WIFI_AUTH_WPA2_PSK,
            .pmf_cfg = {
                .capable = true,
                .required = false,
            }
        }
    }
};

esp_wifi_set_mode(WIFI_MODE_STA);
esp_wifi_set_config(ESP_IF_WIFI_STA, &staConfig);
esp_wifi_start();

static void WifiStop()
ESP_LOGI(TAG, "stop wifi");
esp_wifi_stop();
esp_wifi_restore()
```

```
vTaskDelay(pdMS_TO_TICKS(1000))

const char *GetIPAddress()

    return ip;

esp_err_t WifiInit(uint32_t mode)

    esp_err_t err = ESP_OK;

ESP_ERROR_CHECK(esp_netif_init());

ESP_ERROR_CHECK(esp_event_loop_create_default());

apNetif = esp_netif_create_default_wifi_ap();

staNetif = esp_netif_create_default_wifi_sta();

wifi_init_config_t initConfig = WIFI_INIT_CONFIG_DEFAULT();

ESP_ERROR_CHECK(esp_wifi_init(&initConfig));

ESP_ERROR_CHECK(esp_event_handler_instance_register(WIFI_EVENT,
ESP_EVENT_ANY_ID,      &WifiEventHandler,      xTaskGetCurrentTaskHandle(),
&wifiEventHandler));

ESP_ERROR_CHECK(esp_event_handler_instance_register(IP_EVENT,
ESP_EVENT_ANY_ID,      &IpEventHandler,      xTaskGetCurrentTaskHandle(),
&ipEventHandler);

if (1u & mode)

    WifiStartAp()

    if (pdFALSE == xTaskNotifyWait(0u, 1u, &mode, pdMS_TO_TICKS(30000)))

        WifiStop();
```

```
err = ESP_ERR_TIMEOUT;

else

    WifiStartSta()

    if (pdFALSE == xTaskNotifyWait(0u, 1u, &mode, pdMS_TO_TICKS(30000)))

        WifiStop();

        err = ESP_ERR_TIMEOUT;

        if (1u & mode)

            ESP_LOGW(TAG, "connect to access point failed")

            WifiStop();

            WifiStartAp();

            if (pdFALSE == xTaskNotifyWait(0u, 1u, &mode, pdMS_TO_TICKS(30000)))

                WifiStop();

                err = ESP_ERR_TIMEOUT

            if (ESP_OK == err)

                ESP_LOGI(TAG, "init done");

            else

                ESP_LOGE(TAG, "init fail");

        return err;
```

4.3.1.4. AX code:

```
#define LOG_LOCAL_LEVEL ESP_LOG_INFO

#include "esp_log.h"

#include "esp_timer.h"

#include "string.h"

#include "driver/spi_master.h"

#include "driver/gpio.h"

#include "ax_hal.h"

#define SET_IF_NOT_NULL(x, v) \
    if (NULL != (x))          \
        *(x) = (v);           \


#define AX_SPI_CLK_FREQ 5 * 1000 * 1000

#define AX_SPI_QUEUE_SIZE 2

#define AX_SPI_HOST 1

#define AX_SPI_PIN_MISO 12

#define AX_SPI_PIN_MOSI 13

#define AX_SPI_PIN_CLK 14

#define AX_SPI_PIN_CS 15

#define AX_PIN_INTR1 19

#define AX_PIN_INTR2 18

#define AX_PIN_DRDY 4

#define AX_DEV_REGISTER_DEVID_AD 0x00

#define AX_DEV_REGISTER_DEVID_MST 0x01

#define AX_DEV_REGISTER_PARTID 0x02
```

```
#define AX_DEV_REGISTER_REVID 0x03  
  
#define AX_DEV_REGISTER_STATUS 0x04  
  
#define AX_DEV_REGISTER_FIFO_ENTRIES 0x05  
  
#define AX_DEV_REGISTER_TEMP2 0x06  
  
#define AX_DEV_REGISTER_TEMP1 0x07  
  
#define AX_DEV_REGISTER_XDATA3 0x08  
  
#define AX_DEV_REGISTER_XDATA2 0x09  
  
#define AX_DEV_REGISTER_XDATA1 0x0A  
  
#define AX_DEV_REGISTER_YDATA3 0x0B  
  
#define AX_DEV_REGISTER_YDATA2 0x0C  
  
#define AX_DEV_REGISTER_YDATA1 0x0D  
  
#define AX_DEV_REGISTER_ZDATA3 0x0E  
  
#define AX_DEV_REGISTER_ZDATA2 0x0F  
  
#define AX_DEV_REGISTER_ZDATA1 0x10  
  
#define AX_DEV_REGISTER_FIFO_DATA 0x11  
  
#define AX_DEV_REGISTER_OFFSET_X2 0x1E  
  
#define AX_DEV_REGISTER_OFFSET_X1 0x1F  
  
#define AX_DEV_REGISTER_OFFSET_Y2 0x20  
  
#define AX_DEV_REGISTER_OFFSET_Y1 0x21  
  
#define AX_DEV_REGISTER_OFFSET_Z2 0x22  
  
#define AX_DEV_REGISTER_OFFSET_Z1 0x23  
  
#define AX_DEV_REGISTER_ACT_EN 0x24  
  
#define AX_DEV_REGISTER_ACT_THRESH2 0x25  
  
#define AX_DEV_REGISTER_ACT_THRESH1 0x26
```

```
#define AX_DEV_REGISTER_ACT_COUNT 0x27

#define AX_DEV_REGISTER_FILTER 0x28

#define AX_DEV_REGISTER_FIFO_SAMPLES 0x29

#define AX_DEV_REGISTER_INT_MAP 0x2A

#define AX_DEV_REGISTER_SYNC 0x2B

#define AX_DEV_REGISTER_RANGE 0x2C

#define AX_DEV_REGISTER_POWER_CTL 0x2D

#define AX_DEV_REGISTER_SELF_TEST 0x2E

#define AX_DEV_REGISTER_RESET 0x2F

static const char *AX_TAG = "ax-hal"

typedef struct AxDriver

    spi_host_device_t spiHost;

    spi_device_handle_t spiDev;

} dtAxDriver;

static dtAxDriver *driver = NULL;

static void ISR(void *arg)

static inline int32_t sign_extend(uint32_t x, const int bits)

    int32_t m = 1 << (bits - 1);

    return (x ^ m) - m;

static inline esp_err_t Transmit(uint8_t reg, uint8_t *pload, size_t plen, bool read)

    if (NULL == driver)

        return ESP_ERR_INVALID_STATE;

    spi_transaction_t tr;

    memset(&tr, 0, sizeof(spi_transaction_t));
```

```

tr.addr = reg << 1u | (read ? 1u : 0u);

tr.length = plen * 8;

tr.rxlength = (read ? plen * 8 : 0);

tr.rx_buffer = (read ? (void *)pload : NULL);

tr.tx_buffer = (read ? NULL : (void *)pload);

return spi_device_transmit(driver->spiDev, &tr);

esp_err_t AxHalGetAnalogDevID(uint8_t *id)

    uint8_t buffer = 0u;

    esp_err_t err = Transmit(AX_DEV_REGISTER_DEVID_AD, &buffer, 1u, true);

    ESP_LOGD(AX_TAG, "get analog device id: %02X (%s)", buffer,
esp_err_to_name(err));

    SET_IF_NOT_NULL(id, buffer);

    return err;

esp_err_t AxHalGetAnalogDevMemsID(uint8_t *id)

    uint8_t buffer = 0u;

    esp_err_t err = Transmit(AX_DEV_REGISTER_DEVID_MST, &buffer, 1u, true);

    ESP_LOGD(AX_TAG, "get analog device mems id: %02X (%s)", buffer,
esp_err_to_name(err));

    SET_IF_NOT_NULL(id, buffer);

    return err;

esp_err_t AxHalGetDevID(uint8_t *id)

    uint8_t buffer = 0u;

    esp_err_t err = Transmit(AX_DEV_REGISTER_PARTID, &buffer, 1u, true);

    ESP_LOGD(AX_TAG, "get device id: %02X (%s)", buffer, esp_err_to_name(err));

    SET_IF_NOT_NULL(id, buffer);

```

```

    return err;

esp_err_t AxHalGetRevID(uint8_t *id)

    uint8_t buffer = 0u;

    esp_err_t err = Transmit(AX_DEV_REGISTER_REVID, &buffer, 1u, true);

ESP_LOGD(AX_TAG, "get revision: %02X (%s)", buffer, esp_err_to_name(err));

SET_IF_NOT_NULL(id, buffer);

return err;

esp_err_t AxHalGetStatus(uint8_t *status)

    uint8_t buffer = 0u;

    esp_err_t err = Transmit(AX_DEV_REGISTER_REVID, &buffer, 1u, true);

ESP_LOGD(AX_TAG, "get status: %02X (%s)", buffer, esp_err_to_name(err));

SET_IF_NOT_NULL(status, buffer);

return err;

esp_err_t AxHalGetTemp(uint16_t *temp)

    uint16_t buffer = 0u;

    esp_err_t err = Transmit(AX_DEV_REGISTER_TEMP2, (uint8_t *)&buffer, 2u,
true);

ESP_LOGD(AX_TAG, "get temperature: %04X (%s)", buffer,
esp_err_to_name(err));

SET_IF_NOT_NULL(temp, buffer);

return err;

esp_err_t AxHalGetData(int32_t *x, int32_t *y, int32_t *z)

    uint8_t buffer[9u] = {0u};

    esp_err_t err = Transmit(AX_DEV_REGISTER_XDATA3, buffer, sizeof(buffer),
true);

```

```

int32_t xv = sign_extend((((((0u | buffer[0u]) << 8u) | buffer[1u]) << 8u) | buffer[2u])
>> 4u), 20u);

int32_t yv = sign_extend((((((0u | buffer[3u]) << 8u) | buffer[4u]) << 8u) | buffer[5u])
>> 4u), 20u);

int32_t zv = sign_extend((((((0u | buffer[6u]) << 8u) | buffer[7u]) << 8u) | buffer[8u])
>> 4u), 20u);

ESP_LOG_BUFFER_HEX_LEVEL(AX_TAG, buffer, sizeof(buffer),
ESP_LOG_VERBOSE);

ESP_LOGD(AX_TAG, "get acceleration, x: %08X, y: %08X, z: %08X (%s)", xv, yv,
zv, esp_err_to_name(err));

SET_IF_NOT_NULL(x, xv);

SET_IF_NOT_NULL(y, yv);

SET_IF_NOT_NULL(z, zv);

return err;

esp_err_t AxHalGetFifoCount(uint8_t *count)

    uint8_t buffer = 0u;

    esp_err_t err = Transmit(AX_DEV_REGISTER_FIFO_ENTRIES, &buffer, 1u, true);

    ESP_LOGD(AX_TAG, "get number of fifo entries: %02X (%s)", buffer,
esp_err_to_name(err));

    SET_IF_NOT_NULL(count, buffer);

    return err;

esp_err_t AxHalGetFifoEntry(uint32_t *entry)

    uint32_t buffer = 0u;

    esp_err_t err = Transmit(AX_DEV_REGISTER_FIFO_DATA, (uint8_t *)&buffer,
3u, true)

    ESP_LOGD(AX_TAG, "get fifo entry: %08X (%s)", buffer, esp_err_to_name(err));

```

```

SET_IF_NOT_NULL(entry, buffer);

return err;

esp_err_t AxHalGetOffset(int32_t *x, int32_t *y, int32_t *z)

    uint8_t buffer[6u] = {0u};

    esp_err_t err = Transmit(AX_DEV_REGISTER_OFFSET_X2, buffer, sizeof(buffer),
true);

    int32_t xv = sign_extend((((0u | buffer[0u]) << 8u) | buffer[1u]) << 4u), 20u);

    int32_t yv = sign_extend((((0u | buffer[2u]) << 8u) | buffer[3u]) << 4u), 20u);

    int32_t zv = sign_extend((((0u | buffer[4u]) << 8u) | buffer[5u]) << 4u), 20u);

    ESP_LOG_BUFFER_HEX_LEVEL(AX_TAG,           buffer,           sizeof(buffer),
ESP_LOG_VERBOSE);

    ESP_LOGD(AX_TAG, "get offset, x: %08X, y: %08X, z: %08X (%s)", xv, yv, zv,
esp_err_to_name(err));

    SET_IF_NOT_NULL(x, xv);

    SET_IF_NOT_NULL(y, yv);

    SET_IF_NOT_NULL(z, zv);

    return err;

esp_err_t AxHalSetOffset(int32_t x, int32_t y, int32_t z)

    uint8_t buffer[6u] = {x >> 12u & 0xFFu, x >> 4u & 0xFFu, y >> 12u & 0xFFu, y >>
4u & 0xFFu, z >> 12u & 0xFFu, z >> 4u & 0xFFu};

    esp_err_t err = Transmit(AX_DEV_REGISTER_OFFSET_X2, buffer, sizeof(buffer),
false);

    ESP_LOG_BUFFER_HEX_LEVEL(AX_TAG,           buffer,           sizeof(buffer),
ESP_LOG_VERBOSE);

    ESP_LOGD(AX_TAG, "set offset, x: %08X, y: %08X, z: %08X (%s)", x, y, z,
esp_err_to_name(err));

```

```

    return err;

esp_err_t AxHalGetActivity(uint8_t *axis, uint8_t *counter, uint32_t *threshold)

    uint8_t buffer[4u] = {0u};

    esp_err_t err = Transmit(AX_DEV_REGISTER_ACT_EN, buffer, sizeof(buffer),
true);

    uint8_t av = buffer[0u];

    uint8_t cv = buffer[3u];

    uint32_t tv = (((0u | buffer[1u]) << 8u) | buffer[2u]) << 3u);

    ESP_LOG_BUFFER_HEX_LEVEL(AX_TAG, buffer, sizeof(buffer),
ESP_LOG_VERBOSE);

    ESP_LOGD(AX_TAG, "get activity detection, axis: %02X, counter: %02X, threshold:
%08X (%s)", av, cv, tv, esp_err_to_name(err));

    SET_IF_NOT_NULL(axis, av);

    SET_IF_NOT_NULL(counter, cv);

    SET_IF_NOT_NULL(threshold, tv);

    return err

esp_err_t AxHalSetActivity(uint8_t axis, uint8_t counter, uint32_t threshold)

    uint8_t buffer[4u] = {axis, threshold >> 11u & 0xFFu, threshold >> 3u & 0xFFu,
counter};

    esp_err_t err = Transmit(AX_DEV_REGISTER_ACT_EN, buffer, sizeof(buffer),
false);

    ESP_LOG_BUFFER_HEX_LEVEL(AX_TAG, buffer, sizeof(buffer),
ESP_LOG_VERBOSE);

    ESP_LOGD(AX_TAG, "set activity detection, axis: %02X, counter: %02X, threshold:
%08X (%s)", axis, counter, threshold, esp_err_to_name(err));

    return err;

```

```
esp_err_t AxHalGetFilter(uint8_t *filter)

    uint8_t buffer = 0u;

    esp_err_t err = Transmit(AX_DEV_REGISTER_FILTER, &buffer, 1u, true)

    ESP_LOGD(AX_TAG, "get filter: %02X (%s)", buffer, esp_err_to_name(err));

    SET_IF_NOT_NULL(filter, buffer);

    return err;

esp_err_t AxHalSetFilter(uint8_t filter)

    esp_err_t err = Transmit(AX_DEV_REGISTER_FILTER, &filter, 1u, false);

    ESP_LOGD(AX_TAG, "set filter: %02X (%s)", filter, esp_err_to_name(err));

    return err;

esp_err_t AxHalGetFifoLimit(uint8_t *limit)

    uint8_t buffer = 0u;

    esp_err_t err = Transmit(AX_DEV_REGISTER_FIFO_SAMPLES, &buffer, 1u,
true);

    ESP_LOGD(AX_TAG, "set fifo limit: %02X (%s)", buffer, esp_err_to_name(err));

    SET_IF_NOT_NULL(limit, buffer);

    return err;

esp_err_t AxHalSetFifoLimit(uint8_t limit)

    esp_err_t err = Transmit(AX_DEV_REGISTER_FIFO_SAMPLES, &limit, 1u, false)

    ESP_LOGD(AX_TAG, "set fifo limit: %02X (%s)", limit, esp_err_to_name(err));

    return err;

esp_err_t AxHalGetInterrupt(uint8_t *intr)

    uint8_t buffer = 0u;

    esp_err_t err = Transmit(AX_DEV_REGISTER_INT_MAP, &buffer, 1u, true);
```

```
ESP_LOGD(AX_TAG, "get interrupt map: %02X (%s)", buffer,
esp_err_to_name(err));

SET_IF_NOT_NULL(intr, buffer);

return err;

esp_err_t AxHalSetInterrupt(uint8_t intr)

esp_err_t err = Transmit(AX_DEV_REGISTER_INT_MAP, &intr, 1u, false);

ESP_LOGD(AX_TAG, "set interrupt map: %02X (%s)", intr, esp_err_to_name(err));

return err;

esp_err_t AxHalGetRange(uint8_t *range)

uint8_t buffer = 0u;

esp_err_t err = Transmit(AX_DEV_REGISTER_RANGE, &buffer, 1u, true);

ESP_LOGD(AX_TAG, "set range: %02X (%s)", buffer, esp_err_to_name(err));

SET_IF_NOT_NULL(range, buffer);

return err;

esp_err_t AxHalSetRange(uint8_t range)

esp_err_t err = Transmit(AX_DEV_REGISTER_RANGE, &range, 1u, false);

ESP_LOGD(AX_TAG, "set range: %02X (%s)", range, esp_err_to_name(err));

return err;

esp_err_t AxHalGetControl(uint8_t *control)

uint8_t buffer = 0u;

esp_err_t err = Transmit(AX_DEV_REGISTER_POWER_CTL, &buffer, 1u, true)

ESP_LOGD(AX_TAG, "get control: %02X (%s)", buffer, esp_err_to_name(err));

SET_IF_NOT_NULL(control, buffer);

return err;
```

```

esp_err_t AxHalSetControl(uint8_t control)

    esp_err_t err = Transmit(AX_DEV_REGISTER_POWER_CTL, &control, 1u, false);

    ESP_LOGD(AX_TAG, "set control: %02X (%s)", control, esp_err_to_name(err));

    return err;

esp_err_t AxHalReset()

    uint8_t buffer = 0x52u;

    esp_err_t err = Transmit(AX_DEV_REGISTER_RESET, &buffer, 1u, false);

    ESP_LOGD(AX_TAG, "reset (%s)", esp_err_to_name(err));

    return err;

esp_err_t AxHalInit()

    esp_err_t err = ESP_OK;

    if (NULL != driver)

        ESP_LOGW(AX_TAG, "driver already initialized");

        err = ESP_ERR_INVALID_STATE;

    else

        driver = (dtAxDriver *)calloc(1u, sizeof(dtAxDriver));

        if (NULL == driver)

            ESP_LOGE(AX_TAG, "driver init not enough memory");

            err = ESP_ERR_NO_MEM;

        else

            driver->spiHost = AX_SPI_HOST;

            spi_bus_config_t buscfg = {

                .miso_io_num = AX_SPI_PIN_MISO,

                .mosi_io_num = AX_SPI_PIN_MOSI,

```

```

.sclk_io_num = AX_SPI_PIN_CLK,
.quadwp_io_num = -1,
.quadhd_io_num = -1,
.max_transfer_sz = 0,
.flags = SPICOMMON_BUSFLAG_MASTER,
spi_device_interface_config_t devcfg = {
    .command_bits = 0,
    .address_bits = 8,
    .dummy_bits = 0,
    .clock_speed_hz = AX_SPI_CLK_FREQ,
    .mode = 0,
    .spics_io_num = AX_SPI_PIN_CS,
    .queue_size = AX_SPI_QUEUE_SIZE,
err = spi_bus_initialize(driver->spiHost, &buscfg, 0);
if (ESP_OK == err)
    err = spi_bus_add_device(driver->spiHost, &devcfg, &driver->spiDev);
if (ESP_OK != err)
    ESP_LOGE(AX_TAG, "driver init spi fail");
    spi_bus_free(driver->spiHost);
    free(driver);
    driver = NULL;
    err = ESP_FAIL;
else

```

```
    gpio_install_isr_service(ESP_INTR_FLAG_LEVEL1
ESP_INTR_FLAG_EDGE);

    gpio_reset_pin(AX_PIN_INTR1);

    gpio_set_direction(AX_PIN_INTR1, GPIO_MODE_INPUT);

    gpio_set_pull_mode(AX_PIN_INTR1, GPIO_PULLUP_ONLY);

    gpio_set_intr_type(AX_PIN_INTR1, GPIO_INTR_NEGEDGE);

    gpio_isr_handler_add(AX_PIN_INTR1, ISR, (void *)1u);

    gpio_intr_enable(AX_PIN_INTR1);

    gpio_reset_pin(AX_PIN_INTR2);

    gpio_set_direction(AX_PIN_INTR2, GPIO_MODE_INPUT);

    gpio_set_pull_mode(AX_PIN_INTR2, GPIO_PULLUP_ONLY);

    gpio_set_intr_type(AX_PIN_INTR2, GPIO_INTR_NEGEDGE);

    gpio_isr_handler_add(AX_PIN_INTR2, ISR, (void *)2u);

    gpio_intr_enable(AX_PIN_INTR2);

    gpio_reset_pin(AX_PIN_DRDY);

    gpio_set_direction(AX_PIN_DRDY, GPIO_MODE_INPUT);

    gpio_set_pull_mode(AX_PIN_DRDY, GPIO_PULLDOWN_ONLY);

    gpio_set_intr_type(AX_PIN_DRDY, GPIO_INTR_POSEDGE);

    gpio_isr_handler_add(AX_PIN_DRDY, ISR, (void *)3u);

    gpio_intr_enable(AX_PIN_DRDY);

    ESP_LOGI(AX_TAG, "driver init ok");

return err;
```

4.3.1.5. RF Hal code:

```
#define LOG_LOCAL_LEVEL ESP_LOG_INFO

#include "esp_log.h"

#include "esp_timer.h"

#include "string.h"

#include "driver/uart.h"

#include "rf_hal.h"

#define SET_IF_NOT_NULL(x, v) \
    if (NULL != (x))          \
        *(x) = (v);           \


#define RF_UART_PIN_TX (uint32_t)17u
#define RF_UART_PIN_RX (uint32_t)16u
#define RF_DEV_REGISTER_STATUS (uint16_t)0x0000u
#define RF_DEV_REGISTER_BATTERY (uint16_t)0x0006u
#define RF_DEV_REGISTER_HW_VER (uint16_t)0x000Au
#define RF_DEV_REGISTER_SW_VER (uint16_t)0x000Cu
#define RF_DEV_REGISTER_SERIAL (uint16_t)0x000Eu
#define RF_DEV_REGISTER_ADDRESS (uint16_t)0x0010u
#define RF_DEV_REGISTER_OFFSET (uint16_t)0x0012u
#define RF_DEV_REGISTER_MEASURE (uint16_t)0x0020u
#define RF_DEV_REGISTER_RESULT (uint16_t)0x0022u
#define RF_DEV_REGISTER_LASER (uint16_t)0x01BEu
#define RF_FRAME_HEADER_AA (uint8_t)0xAAu
#define RF_FRAME_HEADER_EE (uint8_t)0xEEu
```

```
#define RF_FRAME_AUTOBAUD (uint8_t)0x55u

#define RF_FRAME_TERMINATE (uint8_t)0x58u

#define RF_BUFFER_MAX_SIZE (size_t)16u

static const char *RF_TAG = "rf-hal"

typedef struct RfTransaction

    uint8_t tx[RF_BUFFER_MAX_SIZE];

    size_t txl;

    int64_t txts;

    uint8_t rx[RF_BUFFER_MAX_SIZE];

    size_t rxl;

    int64_t rxts;

} dtRfTransaction

typedef struct RfDriver

    uart_port_t uartPort;

    uint8_t continous;

} dtRfDriver;

static dtRfDriver *driver = NULL;

static inline uint8_t Checksum(uint8_t *bytes, size_t size)

    uint8_t checksum = 0u;

    for (int i = 0u; i < size; i++)

        checksum += bytes[i];

    return checksum;

static inline void ClearUartRX()

    size_t blen = 0u;
```

```
uart_get_buffered_data_len(driver->uartPort, &blen);

if (0u < blen)

    uart_flush_input(driver->uartPort);

static esp_err_t Transmit(dtRfTransaction *tr, uint32_t timeout)

if (NULL == driver)

    return ESP_ERR_INVALID_STATE;

esp_err_t err = ESP_OK;

int32_t sent = 0;

int32_t received = 0;

ClearUartRX();

if (0u < tr->txl)

    tr->txts = 0u;

sent = uart_write_bytes(driver->uartPort, (void *)tr->tx, tr->txl);

err = (0 == sent ? ESP_ERR_TIMEOUT : ESP_FAIL);

if (tr->txl == (size_t)sent)

    tr->txts = esp_timer_get_time();

err = ESP_OK;

if (0u < tr->rxl)

    memset(tr->rx, 0, RF_BUFFER_MAX_SIZE);

tr->rxl = 0u;

tr->rxts = 0u;

received = uart_read_bytes(driver->uartPort, (void *)&tr->rx[0u], 1u,
pdMS_TO_TICKS(timeout));

err = (0 == received ? ESP_ERR_TIMEOUT : ESP_FAIL);
```

```

    if (0 < received)

        received     +=     uart_read_bytes(driver->uartPort,     (void     *)&tr->rx[1u],
RF_BUFFER_MAX_SIZE - 1u, 0u);

        if (0 < received)

            tr->rxts = esp_timer_get_time();

            tr->txl = (size_t)received;

            err = ESP_OK;

            ESP_LOG_BUFFER_HEX_LEVEL(RF_TAG,  tr->tx,  RF_BUFFER_MAX_SIZE,
ESP_LOG_VERBOSE);

            ESP_LOG_BUFFER_HEX_LEVEL(RF_TAG,  tr->rx,  RF_BUFFER_MAX_SIZE,
ESP_LOG_VERBOSE);

            ESP_LOGV(RF_TAG, "transmit: TX %d/%d/%lld RX %d/%d/%lld (%s)", sent,
RF_BUFFER_MAX_SIZE,  tr->txts,  received,  RF_BUFFER_MAX_SIZE,  tr->rxts,
esp_err_to_name(err));

        return err;

    static esp_err_t SetupFrame(dtRfTransaction *tr, uint8_t addr, uint16_t reg, uint16_t
plen, uint16_t *pload)

        esp_err_t err = ESP_OK;

        memset(tr->tx, 0, sizeof(RF_BUFFER_MAX_SIZE));

        tr->txl = (plen > 0u ? (7u + 2u * plen) : 5u);

        if (RF_BUFFER_MAX_SIZE < tr->txl)

            tr->txl = 0;

            err = ESP_ERR_INVALID_SIZE;

        else

            tr->tx[0u] = RF_FRAME_HEADER_AA;

            tr->tx[1u] = addr;

```

```

tr->tx[2u] = reg >> 8u & 0xFFu;

tr->tx[3u] = reg & 0xFFu;

if (0u < plen)

    tr->tx[4u] = plen >> 8u & 0xFFu;

    tr->tx[5u] = plen & 0xFFu;

    for (uint32_t idx = 0u; idx < plen; idx++)

        tr->tx[6u + 2u * idx] = pload[plen - idx - 1u] >> 8u & 0xFFu;

        tr->tx[7u + 2u * idx] = pload[plen - idx - 1u] & 0xFFu;

    tr->tx[tr->txl - 1u] = Checksum(&tr->tx[1u], tr->txl - 2u);

ESP_LOGV(RF_TAG, "setup frame %d (%s)", tr->txl, esp_err_to_name(err));

return err;

static esp_err_t ParseFrame(dtRfTransaction *tr, uint8_t addr, uint16_t reg, uint16_t
plen, uint16_t *pload)

esp_err_t err = ESP_OK;

if (tr->rxl != (plen > 0u ? (7u + 2u * plen) : 5u))

    err = ESP_ERR_INVALID_SIZE;

else if ((tr->rx[0u] != RF_FRAME_HEADER_AA) && (tr->rx[0u] != RF_FRAME_HEADER_EE))

    err = ESP_ERR_INVALID_VERSION;

else if (tr->rx[tr->rxl - 1u] != Checksum(&tr->rx[1u], tr->rxl - 2u))

    err = ESP_ERR_INVALID_CRC;

else if (((tr->rx[1u] != addr) || (((uint16_t)tr->rx[2u] << 8u | tr->rx[3u]) != reg) ||
((uint16_t)tr->rx[4u] << 8u | tr->rx[5u]) != plen))

    err = ESP_ERR_INVALID_ARG;

else

```

```

for (uint32_t idx = 0u; idx < plen; idx++)

    pload[plen - idx - 1u] = ((uint16_t)tr->rx[6u + 2u * idx] << 8u | tr->rx[7u + 2u *
idx]);

ESP_LOGV(RF_TAG, "parse frame %d (%s)", tr->rxl, esp_err_to_name(err));

return err;

esp_err_t RfHalGetStatus(uint8_t dev, uint16_t *status)

    uint16_t buffer = 0u;

    dtRfTransaction transaction = {0u};

    transaction.rxl = 1u;

    esp_err_t err = SetupFrame(&transaction, (dev | 0x80u),
RF_DEV_REGISTER_STATUS, 0u, NULL);

    if (ESP_OK == err)

        err = Transmit(&transaction, 50u);

        if (ESP_OK == err)

            err = ParseFrame(&transaction, (dev | 0x80u), RF_DEV_REGISTER_STATUS,
1u, &buffer);

            ESP_LOGD(RF_TAG, "device %02X get status: %04X (%s)", dev, buffer,
esp_err_to_name(err));

            SET_IF_NOT_NULL(status, buffer);

    return err;

esp_err_t RfHalGetVoltage(uint8_t dev, uint16_t *voltage)

    uint16_t buffer = 0u;

    dtRfTransaction transaction = {0u};

    transaction.rxl = 1u;

    esp_err_t err = SetupFrame(&transaction, (dev | 0x80u),
RF_DEV_REGISTER_BATTERY, 0u, NULL);

```

```

if(ESP_OK == err)

    err = Transmit(&transaction, 50u);

    if(ESP_OK == err)

        err = ParseFrame(&transaction, (dev | 0x80u),
RF_DEV_REGISTER_BATTERY, 1u, &buffer);

        ESP_LOGD(RF_TAG, "device %02X get voltage: %04X (%s)", dev, buffer,
esp_err_to_name(err));

        SET_IF_NOT_NULL(voltage, buffer);

    return err;

esp_err_t RfHalGetHWVersion(uint8_t dev, uint16_t *hw)

    uint16_t buffer = 0u;

    dtRfTransaction transaction = {0u};

    transaction.rxl = 1u;

    esp_err_t err = SetupFrame(&transaction, (dev | 0x80u),
RF_DEV_REGISTER_HW_VER, 0u, NULL);

    if(ESP_OK == err)

        err = Transmit(&transaction, 50u);

        if(ESP_OK == err)

            err = ParseFrame(&transaction, (dev | 0x80u), RF_DEV_REGISTER_HW_VER,
1u, &buffer);

            ESP_LOGD(RF_TAG, "device %02X get hw version: %04X (%s)", dev, buffer,
esp_err_to_name(err));

            SET_IF_NOT_NULL(hw, buffer);

    return err;

esp_err_t RfHalGetSWVersion(uint8_t dev, uint16_t *sw)

    uint16_t buffer = 0u;

```

```

dtRfTransaction transaction = {0u};

transaction.rxl = 1u;

esp_err_t err = SetupFrame(&transaction, (dev | 0x80u),
RF_DEV_REGISTER_SW_VER, 0u, NULL);

if (ESP_OK == err)

err = Transmit(&transaction, 50u);

if (ESP_OK == err)

err = ParseFrame(&transaction, (dev | 0x80u), RF_DEV_REGISTER_SW_VER,
1u, &buffer)

ESP_LOGD(RF_TAG, "device %02X get sw version: %04X (%s)", dev, buffer,
esp_err_to_name(err));

SET_IF_NOT_NULL(sw, buffer);

return err;

esp_err_t RfHalGetSerialNumber(uint8_t dev, uint32_t *serial)

uint32_t buffer = 0u;

dtRfTransaction transaction = {0u};

transaction.rxl = 1u;

esp_err_t err = SetupFrame(&transaction, (dev | 0x80u),
RF_DEV_REGISTER_SERIAL, 0u, NULL);

if (ESP_OK == err)

err = Transmit(&transaction, 50u);

if (ESP_OK == err)

err = ParseFrame(&transaction, (dev | 0x80u), RF_DEV_REGISTER_SERIAL,
2u, (uint16_t *)&buffer);

```

```

ESP_LOGD(RF_TAG, "device %02X get serial number: %08X (%s)", dev, buffer,
esp_err_to_name(err));

SET_IF_NOT_NULL(serial, buffer);

return err;

esp_err_t RfHalGetMeasureResult(uint8_t dev, uint32_t *distance, uint16_t *quality,
int64_t *timestamp)

    uint16_t buffer[3u] = {0u};

    dtRfTransaction transaction = {0u};

    transaction.rxl = 1u;

    esp_err_t err = ESP_OK;

    if (0u == driver->continous)

        err = SetupFrame(&transaction, (dev | 0x80u), RF_DEV_REGISTER_RESULT,
0u, NULL;

    if (ESP_OK == err)

        err = Transmit(&transaction, 2000u);

    if (ESP_OK == err)

        err = ParseFrame(&transaction, dev, RF_DEV_REGISTER_RESULT, 3u,
buffer);

    ESP_LOGD(RF_TAG, "device %02X get measure result: %04X%04X %04X (%s)",
dev, buffer[2], buffer[1], buffer[0], esp_err_to_name(err));

    SET_IF_NOT_NULL(distance, (uint32_t)buffer[2] << 16u | buffer[1]);

    SET_IF_NOT_NULL(quality, buffer[0]);

    SET_IF_NOT_NULL(timestamp, transaction.rxts);

    return err;

esp_err_t RfHalSingleMeasure(uint8_t dev, uint16_t mode, uint32_t *distance, uint16_t
*quality, int64_t *timestamp)

```

```

        uint16_t buffer[3u] = {0u};

        dtRfTransaction transaction = {0u};

        transaction.rxl = 1u;

        esp_err_t err = SetupFrame(&transaction, dev, RF_DEV_REGISTER_MEASURE,
1u, &mode);

        if (ESP_OK == err)

            err = Transmit(&transaction, 2000u);

            if (ESP_OK == err)

                err = ParseFrame(&transaction, dev, RF_DEV_REGISTER_RESULT, 3u,
buffer);

                ESP_LOGD(RF_TAG, "device %02X single measure result: %04X%04X %04X
(%s)", dev, buffer[2], buffer[1], buffer[0], esp_err_to_name(err));

                SET_IF_NOT_NULL(distance, (uint32_t)buffer[2] << 16u | buffer[1]);

                SET_IF_NOT_NULL(quality, buffer[0]);

                SET_IF_NOT_NULL(timestamp, transaction.rxts);

                return err;

esp_err_t RfHalBroadcastMeasure(uint16_t mode)

        dtRfTransaction transaction = {0u};

        esp_err_t err = SetupFrame(&transaction, 0x7F, RF_DEV_REGISTER_MEASURE,
1u, &mode);

        if (ESP_OK == err)

            err = Transmit(&transaction, 0u);

            ESP_LOGD(RF_TAG, "broadcast measure (%s)", esp_err_to_name(err));

            return err;

```

```

esp_err_t RfHalStartContinousMeasure(uint8_t dev, uint16_t mode)

    dtRfTransaction transaction = {0u};

    esp_err_t err = SetupFrame(&transaction, dev, RF_DEV_REGISTER_MEASURE,
1u, &mode);

    if (ESP_OK == err)

        err = Transmit(&transaction, 0u);

        if (ESP_OK == err)

            driver->continous = 1u;

            ESP_LOGD(RF_TAG, "device %02X start continous measure (%s)", dev,
esp_err_to_name(err));

            return err;

esp_err_t RfHalStopContinousMeasure()

    esp_err_t err = ESP_FAIL;

    if (1u == driver->continous)

        dtRfTransaction transaction = {0u};

        transaction.tx[0u] = RF_FRAME_TERMINATE;

        transaction.txl = 1u;

        transaction.rxl = 1u;

        err = Transmit(&transaction, 1000u);

        if ((ESP_OK == err) || (ESP_ERR_TIMEOUT == err))

            driver->continous = 0u;

            err = ESP_OK;

            ESP_LOGD(RF_TAG, "stop continous measure (%s)", esp_err_to_name(err));

            return err;

esp_err_t RfHalSetDeviceAddress(uint8_t dev, uint16_t address)

```

```

    uint16_t buffer = 0u;

    dtRfTransaction transaction = {0u};

    transaction.rxl = 1u;

    esp_err_t err = SetupFrame(&transaction, dev, RF_DEV_REGISTER_ADDRESS, 1u,
&address);

    if (ESP_OK == err)

        err = Transmit(&transaction, 50u);

        if (ESP_OK == err)

            err = ParseFrame(&transaction, dev, RF_DEV_REGISTER_ADDRESS, 1u,
&buffer);

            if(ESP_OK == err && buffer != address)

                err = ESP_ERR_INVALID_RESPONSE;

                ESP_LOGD(RF_TAG, "device %02X set address: %04X (%s)", dev, buffer,
esp_err_to_name(err));

            return err;

    esp_err_t RfHalSetMeasureOffset(uint8_t dev, int16_t offset)

        uint16_t buffer = 0u

        dtRfTransaction transaction = {0u};

        transaction.rxl = 1u;

        esp_err_t err = SetupFrame(&transaction, dev, RF_DEV_REGISTER_OFFSET, 1u,
(uint16_t *)&offset);

        if (ESP_OK == err)

            err = Transmit(&transaction, 50u);

            if (ESP_OK == err)

                err = ParseFrame(&transaction, dev, RF_DEV_REGISTER_OFFSET, 1u,
&buffer);

```

```

    if(ESP_OK == err && buffer != (uint16_t) offset)
        err = ESP_ERR_INVALID_RESPONSE;

    ESP_LOGD(RF_TAG, "device %02X set offset: %04X (%s)", dev, buffer,
esp_err_to_name(err));

    return err;

esp_err_t RfHalSetLaser(uint8_t dev, uint16_t laser)

    uint16_t buffer = 0u;

    dtRfTransaction transaction = {0u};

    transaction.rxl = 1u;

    esp_err_t err = SetupFrame(&transaction, dev, RF_DEV_REGISTER_LASER, 1u,
(uint16_t *)&laser);

    if (ESP_OK == err)

        err = Transmit(&transaction, 50u);

        if (ESP_OK == err)

            err = ParseFrame(&transaction, dev, RF_DEV_REGISTER_LASER, 1u,
&buffer);

            if(ESP_OK == err && buffer != (uint16_t) laser)

                err = ESP_ERR_INVALID_RESPONSE;

            ESP_LOGD(RF_TAG, "device %02X set laser: %04X (%s)", dev, buffer,
esp_err_to_name(err));

            return err;

esp_err_t RfHalAutobaud()

    dtRfTransaction transaction = {0u};

    transaction.tx[0u] = RF_FRAME_AUTOBAUD;

    transaction.txl = 1u;

```

```
transaction.rxl = 1u;

esp_err_t err = Transmit(&transaction, 1000u);

ESP_LOGD(RF_TAG, "autobaud (%s)", esp_err_to_name(err));

return err;

esp_err_t RfHalInit()

esp_err_t err = ESP_OK;

if (NULL != driver)

    ESP_LOGW(RF_TAG, "driver already initialized");

    err = ESP_ERR_INVALID_STATE;

else

    driver = (dtRfDriver *)calloc(1u, sizeof(dtRfDriver));

    if (NULL == driver)

        ESP_LOGE(RF_TAG, "driver init not enough memory");

        err = ESP_ERR_NO_MEM;

    else

        driver->uartPort = UART_NUM_2;

        driver->continous = 0u;

        uart_config_t uart_config = {

            .baud_rate = 115200,

            .data_bits = UART_DATA_8_BITS,

            .parity = UART_PARITY_DISABLE,

            .stop_bits = UART_STOP_BITS_1,

            .flow_ctrl = UART_HW_FLOWCTRL_DISABLE,

            .source_clk = UART_SCLK_APB,
```

```
err = uart_driver_install(driver->uartPort, 256u, 0u, 0u, NULL, 0);

if (ESP_OK == err)

    err = uart_param_config(driver->uartPort, &uart_config);

    if (ESP_OK == err)

        err = uart_set_pin(driver->uartPort, RF_UART_PIN_TX,
RF_UART_PIN_RX, -1, -1);

        if (ESP_OK != err)

            ESP_LOGE(RF_TAG, "driver init uart fail");

            uart_driver_delete(driver->uartPort);

            free(driver);

            driver = NULL;

            err = ESP_FAIL;

        else

            ESP_LOGI(RF_TAG, "driver init ok");

    return err;
```

4.3.1.6. Filterezés

```
#include "filter.h"

void PredictState(dtFilter *flt)
{
    flt->t += flt->dt;
    flt->x = flt->F * flt->x;
    flt->P = flt->F * flt->P * flt->F.t() + flt->Q;}}

void EstimateState(dtFilter *flt, dtObservation *obs)
{
    dspm::Mat y = obs->z - obs->H * flt->x;
    dspm::Mat S = obs->H * flt->P * obs->H.t() + obs->R;
    dspm::Mat K = flt->P * obs->H.t() * S.inverse();
    dspm::Mat T = dspm::Mat::eye(flt->x.length) - K * obs->H;
    flt->x = flt->x + K * y;
    flt->P = T * flt->P * T.t() + K * obs->R * K.t();}
```

5. Tesztkörnyezet kiválasztása, tesztelés, jegyzőkönyvek

5.1. Tesztkörnyezet

Tesztkörnyezet kiválasztása

A tesztelésre szánt környezet kiválasztása során figyelembe vettük, hogy a felmérődarupálya olyan ipari csarnokban legyen, ahol a referencia mérések és a tesztmérések elvégzésekor nem áll fent folyamatos, a mérést megnehezítő rezgés. A prototípus képes lesz ezeknek a külső rezgéseknek a figyelmen kívül hagyásában, de a referenciául szolgáló mérés nem lehetett terhelve ezekkel a hibákkal.

5.2. Referenciamérés

A referencia mérést egy acélszerkezetek gyártó csarnok darupályáján végeztük el. A csarnok az algyői ipari parkban található. Az acélszerkezetes csarnok egy 5 tonnás és egy 10 tonnás híddaru közösített, laposvas keresztmetszetű sínpályával van szerelve, a sípálya a fő tartóoszlopokra hegesztett konzolokon nyugszik, magassági pozícióját fémlemezes hézagolással állították be.

A felmérés elvégzéséhez több, a dokumentum 2.2 pontjában ismertetett módszer alkalmazható. Ahhoz, hogy a legmegbízhatóbb eredményt kapjuk, az egyik legmodernebb, lézerszkenneres eljárást választottuk.

A lézerszkenner alapú felmérés előnye, hogy a darupálya pontos geometriája mellett az azt tartó acélszerkezet és maga a futódaru is leképződik. Ezáltal a darupályának nemcsak a geometriai torzulását lehet kimutatni, de következtetni lehet a torzulás okára, például egy tartóoszlop alapjának süllyedése, a felszerelt konzol lehajlása stb.

A méréskor felhasznált eszköz egy fázisméréses lézerszkenner volt. A szkenner 360 fokban körbe fordulva rögzíti maga körül a teret. Lehetőség van nagy pontosságú és nagy érzékenységű mérés elvégzésére. A nagy pontosságú mérés 1-50 méter között valósul meg, a 15 méteres sugáron vett távmérési hibája 0,7 mm alatti, 90 százalékos visszaverő képességű anyagoknál 0,12 mm zaj, 10 százalékos visszaverő képességű anyagoknál 0,3mm zaj keletkezik. A nagy érzékenységű mérés 1-110 méter közötti mérési tartományban működik, azonban a nagy pontosságú méréshez képest jelentősebb 0,9 mm távmérési hiba és 0,25-0,7 mm zaj jelentkezik.

A két felmérési mód pontfelvételi sebességében nincs jelentős eltérés. A darupálya referenciaméréséhez a nagy pontosságú felmérési módot választottuk.

A felmérés során az egyes pozíciókat (felmérési álláspontokat) ismert átmérőjű kapcsoló gömbbel illesztettük egymáshoz. Az illesztés során a külön-külön álló pontfelhők közös koordináta-rendszerbe kerültek.

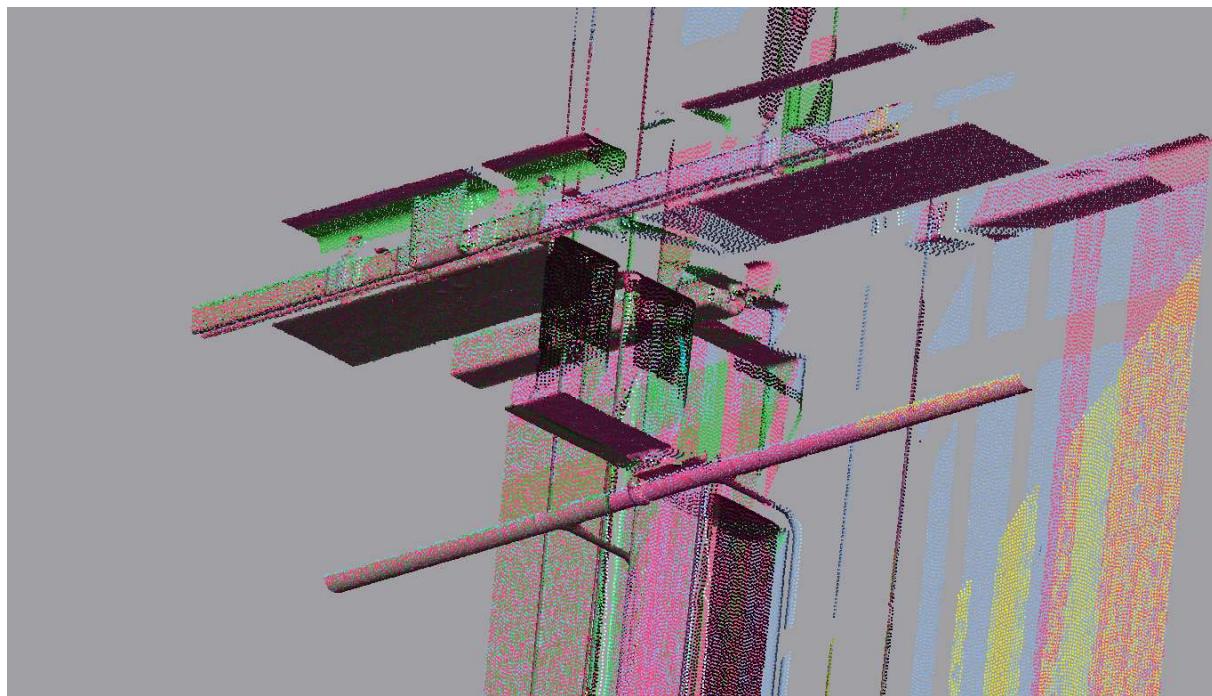
A darupálya felméréshez a műszernek közvetlenül rá kell látni a darupálya felületére. Ez a nehezítő tényező a hagyományos geodéziai felmérés során is fent áll, ugyanakkor a lézerszkenneléskor nincs szükség arra, hogy a felmérést végző személy egy prizmával, előre kiválasztott pontokat bemérve, nagy magasságban végig sétáljon a darupályán



84. ábra Lézerszkenner



85. ábra Csarnok pontfelhőjének keresztmetszete – az egyes színek a különböző állásponlokat jelölik



86. ábra Konzol csomópont

A tesztkörnyezet felmérése után a darupálya jellemző pontjainak és geometriájának meghatározásához volt a cél. A pontfelhő átrajzolása reverse engineering szoftverben történt. Az átrajzoláskor végzett mintavételezés automatizált, síkra illesztett volt. A kiértékelés következtében előállított numerikus jegyzőkönyv a darupálya geometriai jellemzőinek számszerű leírása, ami tartalmazza a pontsor koordinátáit a már irányba forgatott koordinátarendszerben, az egyenestől való Δ eltérési értékeit és a két tengely közötti eltérést. A teszt terület egy a leggyakrabban használt 15 méteres közös szakasz. A szakasz kiválasztásánál figyelembe vettük, hogy két rasztert is érintsen, tehát az alátámasztás és annak csomópontja is vizsgálható.

Távolság (mm), Y tengely mentén	BAL SÍN						JOBB SÍN						Eltérések					
	X	Y	Z	dX	dY	dZ	X	Y	Z	dX	dY	dZ	Z irányú kigyözés	Δz	X irányú kigyözés	Δx		
0	2342	3542	7201	2	0	1	14344	3540	7202	4	-2	2	1	2	-1	2	4	-2
50	2341	3592	7202	1	50	2	14341	3590	7201	1	48	1	2	1	1	1	1	0
100	2342	3642	7202	2	100	2	14344	3640	7200	4	98	0	2	0	2	2	4	-2
150	2342	3692	7201	2	150	1	14343	3690	7202	3	148	2	1	2	-1	2	3	-1
200	2343	3742	7202	3	200	2	14343	3740	7201	3	198	1	2	1	1	3	3	0
250	2344	3792	7200	4	250	0	14346	3790	7199	6	248	-1	0	-1	1	4	6	-2
300	2344	3842	7202	4	300	2	14344	3840	7204	4	298	4	2	4	-2	4	4	0
350	2343	3892	7200	3	350	0	14345	3890	7198	5	348	-2	0	-2	2	3	5	-2
400	2342	3942	7199	2	400	-1	14341	3940	7201	1	398	1	-1	1	-2	2	1	1
450	2341	3992	7202	1	450	2	14342	3990	7202	2	448	2	2	2	0	1	2	-1
500	2341	4042	7198	1	500	-2	14342	4040	7200	2	498	0	-2	0	-2	1	2	-1
550	2341	4092	7198	1	550	-2	14342	4090	7196	2	548	-4	-2	-4	2	1	2	-1
600	2341	4142	7198	1	600	-2	14343	4140	7197	3	598	-3	-2	-3	1	1	3	-2
650	2341	4192	7200	1	650	0	14341	4190	7202	1	648	2	0	2	-2	1	1	0
700	2341	4242	7199	1	700	-1	14340	4240	7198	0	698	-2	-1	-2	1	1	0	1
750	2340	4292	7198	0	750	-2	14339	4290	7198	-1	748	-2	-2	-2	0	0	-1	1
800	2340	4342	7202	0	800	2	14342	4340	7202	2	798	2	2	2	0	0	2	-2
850	2340	4392	7199	0	850	-1	14341	4390	7201	1	848	1	-1	1	-2	0	1	-1
900	2340	4442	7202	0	900	2	14339	4440	7200	-1	898	0	2	0	2	0	-1	1
950	2339	4492	7202	-1	950	2	14339	4490	7204	-1	948	4	2	4	-2	-1	-1	0
1000	2339	4542	7201	-1	1000	1	14340	4540	7200	0	998	0	1	0	1	-1	0	-1
1050	2339	4592	7198	-1	1050	-2	14341	4590	7200	1	1048	0	-2	0	-2	-1	1	-2
1100	2340	4642	7198	0	1100	-2	14339	4640	7199	-1	1098	-1	-2	-1	-1	0	-1	1
1150	2339	4692	7198	-1	1150	-2	14338	4690	7200	-2	1148	0	-2	0	-2	-1	-2	1
1200	2340	4742	7200	0	1200	0	14341	4740	7201	1	1198	1	0	1	-1	0	1	-1
1250	2340	4792	7202	0	1250	2	14339	4790	7204	-1	1248	4	2	4	-2	0	-1	1
1300	2339	4842	7202	-1	1300	2	14338	4840	7202	-2	1298	2	2	2	0	-1	-2	1
1350	2339	4892	7200	-1	1350	0	14339	4890	7199	-1	1348	-1	0	-1	1	-1	-1	0
1400	2338	4942	7203	-2	1400	3	14340	4940	7202	0	1398	2	3	2	1	-2	0	-2
1450	2339	4992	7202	-1	1450	2	14341	4990	7200	1	1448	0	2	0	2	-1	1	-2
1500	2339	5042	7201	-1	1500	1	14339	5040	7200	-1	1498	0	1	0	1	-1	-1	0
1550	2338	5092	7200	-2	1550	0	14338	5090	7198	-2	1548	-2	0	-2	2	-2	-2	0

87. ábra - Felmérésből származó adatok kiértékelése táblázatban

A felmérés teljes adatállománya 300 adatsorból áll, 50 mm mintavételezéssel. A kiértékelés során jelentős elváltozás nem mérhető az adott szakaszon.

Távolság (mm), Y tengely mentén	BAL SÍN						JOBB SÍN						Elterések					
	X	Y	Z	dX	dY	dZ	X	Y	Z	dX	dY	dZ	Z irányú kígyózás	Δz	X irányú kígyózás	Δx		
0	2342	3542	7201	2	0	1	14344	3540	7202	4	-2	2	1	2	-1	2	4	-2
50	2341	3592	7202	1	50	2	14341	3590	7201	1	48	1	2	1	1	1	1	0
100	2342	3642	7202	2	100	2	14344	3640	7200	4	98	0	2	0	2	2	4	-2
150	2342	3692	7201	2	150	1	14343	3690	7202	3	148	2	1	2	-1	2	3	-1
200	2343	3742	7202	3	200	2	14343	3740	7201	3	198	1	2	1	1	3	3	0
250	2344	3792	7200	4	250	0	14346	3790	7199	6	248	-1	0	-1	1	4	6	-2
300	2344	3842	7202	4	300	2	14344	3840	7204	4	298	4	2	4	-2	4	4	0
350	2343	3892	7200	3	350	0	14345	3890	7198	5	348	-2	0	-2	2	3	5	-2
400	2342	3942	7199	2	400	-1	14341	3940	7201	1	398	1	-1	1	-2	2	1	1
450	2341	3992	7202	1	450	2	14342	3990	7202	2	448	2	2	2	0	1	2	-1
500	2341	4042	7198	1	500	-2	14342	4040	7200	2	498	0	-2	0	-2	1	2	-1
550	2341	4092	7198	1	550	-2	14342	4090	7196	2	548	-4	-2	-4	2	1	2	-1
600	2341	4142	7198	1	600	-2	14343	4140	7197	3	598	-3	-3	-3	1	1	3	-2
650	2341	4192	7200	1	650	0	14341	4190	7202	1	648	2	0	2	-2	1	1	0
700	2341	4242	7199	1	700	-1	14340	4240	7198	0	698	-2	-1	-2	1	1	0	1
750	2340	4292	7198	0	750	-2	14339	4290	7198	-1	748	-2	-2	-2	0	0	-1	1
800	2340	4342	7202	0	800	2	14342	4340	7202	2	798	2	2	2	0	0	2	-2
850	2340	4392	7199	0	850	-1	14341	4390	7201	1	848	1	-1	1	-2	0	1	-1
900	2340	4442	7202	0	900	2	14339	4440	7200	-1	898	0	2	0	2	0	-1	1
950	2339	4492	7202	-1	950	2	14339	4490	7204	-1	948	4	2	4	-2	-1	-1	0
1000	2339	4542	7201	-1	1000	1	14340	4540	7200	0	998	0	1	0	1	-1	0	-1
1050	2339	4592	7198	-1	1050	-2	14341	4590	7200	1	1048	0	-2	0	-2	-1	1	-2
1100	2340	4642	7198	0	1100	-2	14339	4640	7199	-1	1098	-1	-2	-1	-1	0	-1	1
1150	2339	4692	7198	-1	1150	-2	14338	4690	7200	-2	1148	0	-2	0	-2	-1	-2	1
1200	2340	4742	7200	0	1200	0	14341	4740	7201	1	1198	1	0	1	-1	0	1	-1
1250	2340	4792	7202	0	1250	2	14339	4790	7204	-1	1248	4	2	4	-2	0	-1	1
1300	2339	4842	7202	-1	1300	2	14338	4840	7202	-2	1298	2	2	2	0	-1	-2	1
1350	2339	4892	7200	-1	1350	0	14339	4890	7199	-1	1348	-1	0	-1	1	-1	-1	0
1400	2338	4942	7203	-2	1400	3	14340	4940	7202	0	1398	2	2	1	-2	0	-2	0
1450	2339	4992	7202	-1	1450	2	14341	4990	7200	1	1448	0	2	0	2	-1	1	-2
1500	2339	5042	7201	-1	1500	1	14339	5040	7200	-1	1498	0	1	0	1	-1	-1	0
1550	2338	5092	7200	-2	1550	0	14338	5090	7198	-2	1548	-2	0	-2	2	-2	-2	0
1600	2337	5142	7202	-3	1600	2	14336	5140	7203	-4	1598	3	2	3	-1	-3	-4	1
1650	2339	5192	7203	-1	1650	3	14340	5190	7201	0	1648	1	3	1	2	-1	0	-1
1700	2337	5242	7200	-3	1700	0	14339	5240	7200	-1	1698	0	0	0	0	-3	-1	-2
1750	2337	5292	7202	-3	1750	2	14336	5290	7204	-4	1748	4	2	4	-2	-3	-4	1
1800	2337	5342	7201	-3	1800	1	14339	5340	7203	-1	1798	3	1	3	-2	-3	-1	-2
1850	2337	5392	7200	-3	1850	0	14338	5390	7199	-2	1848	-1	0	-1	1	-3	-2	-1
1900	2337	5442	7200	-3	1900	0	14339	5440	7200	-1	1898	0	0	0	0	-3	-1	-2
1950	2337	5492	7202	-3	1950	2	14339	5490	7202	-1	1948	2	2	2	0	-3	-1	-2

2000	2338	5542	7203	-2	2000	3	14339	5540	7203	-1	1998	3	3	3	0	-2	-1	-1
2050	2336	5592	7201	-4	2050	1	14337	5590	7200	-3	2048	0	1	0	1	-4	-3	-1
2100	2337	5642	7200	-3	2100	0	14339	5640	7202	-1	2098	2	0	2	-2	-3	-1	-2
2150	2336	5692	7200	-4	2150	0	14335	5690	7202	-5	2148	2	0	2	-2	-4	-5	1
2200	2337	5742	7203	-3	2200	3	14337	5740	7202	-3	2198	2	3	2	1	-3	-3	0
2250	2337	5792	7202	-3	2250	2	14336	5790	7204	-4	2248	4	2	4	-2	-3	-4	1
2300	2337	5842	7203	-3	2300	3	14336	5840	7204	-4	2298	4	3	4	-1	-3	-4	1
2350	2337	5892	7203	-3	2350	3	14338	5890	7201	-2	2348	1	3	1	2	-3	-2	-1
2400	2337	5942	7200	-3	2400	0	14337	5940	7201	-3	2398	1	0	1	-1	-3	-3	0
2450	2337	5992	7202	-3	2450	2	14338	5990	7201	-2	2448	1	2	1	1	-3	-2	-1
2500	2337	6042	7202	-3	2500	2	14336	6040	7204	-4	2498	4	2	4	-2	-3	-4	1
2550	2337	6092	7203	-3	2550	3	14339	6090	7204	-1	2548	4	3	4	-1	-3	-1	-2
2600	2337	6142	7200	-3	2600	0	14339	6140	7201	-1	2598	1	0	1	-1	-3	-1	-2
2650	2338	6192	7200	-2	2650	0	14339	6190	7201	-1	2648	1	0	1	-1	-2	-1	-1
2700	2338	6242	7202	-2	2700	2	14338	6240	7203	-2	2698	3	2	3	-1	-2	-2	0
2750	2338	6292	7203	-2	2750	3	14338	6290	7202	-2	2748	2	3	2	1	-2	-2	0
2800	2338	6342	7202	-2	2800	2	14338	6340	7200	-2	2798	0	2	0	2	-2	-2	0
2850	2338	6392	7201	-2	2850	1	14339	6390	7202	-1	2848	2	1	2	-1	-2	-1	-1
2900	2339	6442	7201	-1	2900	1	14339	6440	7201	-1	2898	1	1	1	0	-1	-1	0
2950	2338	6492	7202	-2	2950	2	14340	6490	7203	0	2948	3	2	3	-1	-2	0	-2
3000	2339	6542	7204	-1	3000	4	14339	6540	7203	-1	2998	3	4	3	1	-1	-1	0
3050	2337	6592	7204	-3	3050	4	14338	6590	7203	-2	3048	3	4	3	1	-3	-2	-1
3100	2339	6642	7202	-1	3100	2	14340	6640	7201	0	3098	1	2	1	1	-1	0	-1
3150	2339	6692	7203	-1	3150	3	14338	6690	7202	-2	3148	2	3	2	1	-1	-2	1
3200	2339	6742	7203	-1	3200	3	14339	6740	7201	-1	3198	1	3	1	2	-1	-1	0
3250	2339	6792	7204	-1	3250	4	14339	6790	7204	-1	3248	4	4	4	0	-1	-1	0
3300	2339	6842	7202	-1	3300	2	14338	6840	7204	-2	3298	4	2	4	-2	-1	-2	1
3350	2340	6892	7201	0	3350	1	14341	6890	7201	1	3348	1	1	1	0	0	1	-1
3400	2342	6942	7201	2	3400	1	14342	6940	7201	2	3398	1	1	1	0	2	2</	

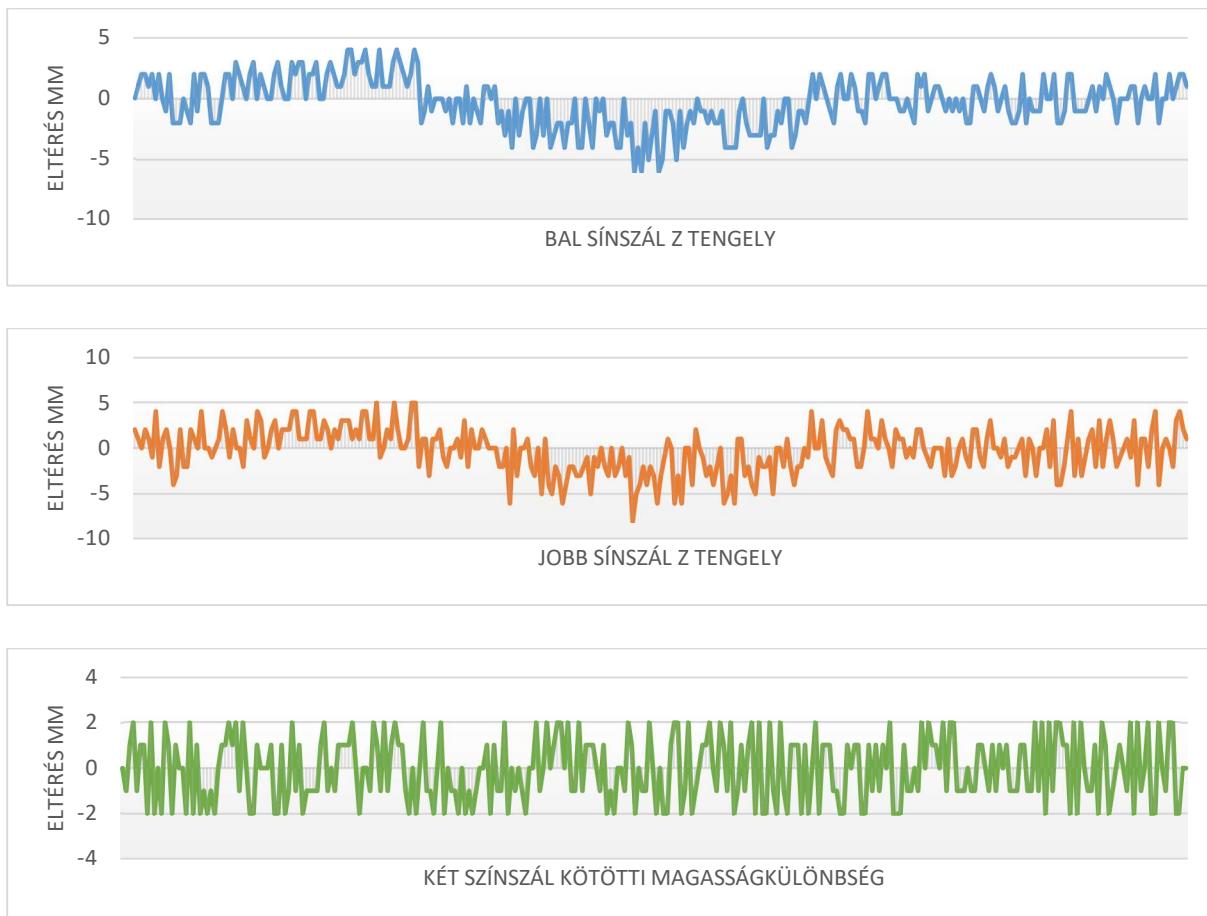
6600	2341	10142	7199	1	6600	-1	14343	10140	7198	3	6598	-2	-1	-2	1	1	3	-2
6650	2341	10192	7200	1	6650	0	14341	10190	7200	1	6648	0	0	0	1	1	1	0
6700	2341	10242	7197	1	6700	-3	14341	10240	7198	1	6698	-2	-2	-3	-1	1	1	0
6750	2342	10292	7198	2	6750	-2	14344	10290	7197	4	6748	-3	-2	-3	1	2	4	-2
6800	2341	10342	7198	1	6800	-2	14343	10340	7200	3	6798	0	-2	0	-2	1	3	-2
6850	2342	10392	7196	2	6850	-4	14341	10390	7197	1	6848	-3	-4	-3	-1	2	1	1
6900	2342	10442	7196	2	6900	-4	14343	10440	7198	3	6898	-2	-4	-2	-2	2	3	-1
6950	2341	10492	7200	1	6950	0	14343	10490	7200	3	6948	0	0	0	0	1	3	-2
7000	2340	10542	7197	0	7000	-3	14341	10540	7197	1	6998	-3	-3	-3	0	0	1	-1
7050	2340	10592	7198	0	7050	-2	14342	10590	7199	2	7048	-1	-2	-1	-1	0	2	-2
7100	2340	10642	7194	0	7100	-6	14342	10640	7192	2	7098	-8	-6	-8	2	0	2	-2
7150	2340	10692	7196	0	7150	-4	14340	10690	7195	0	7148	-5	-4	-5	1	0	0	0
7200	2340	10742	7194	0	7200	-6	14340	10740	7196	0	7198	-4	-6	-4	-2	0	0	0
7250	2341	10792	7198	1	7250	-2	14343	10790	7198	3	7248	-2	-2	-2	0	1	3	-2
7300	2340	10842	7195	0	7300	-5	14341	10840	7196	1	7298	-4	-5	-4	-1	0	1	-1
7350	2340	10892	7197	0	7350	-3	14342	10890	7198	2	7348	-2	-3	-2	-1	0	2	-2
7400	2341	10942	7199	1	7400	-1	14343	10940	7197	3	7398	-3	-1	-3	2	1	3	-2
7450	2340	10992	7194	0	7450	-6	14340	10990	7194	0	7448	-6	-6	-6	0	0	0	0
7500	2341	11042	7195	1	7500	-5	14343	11040	7197	3	7498	-3	-5	-3	-2	1	3	-2
7550	2341	11092	7199	1	7550	-1	14343	11090	7199	3	7548	-1	-1	-1	0	1	3	-2
7600	2341	11142	7199	1	7600	-1	14342	11140	7201	2	7598	1	-1	1	-2	1	2	-1
7650	2341	11192	7198	1	7650	-2	14340	11190	7200	0	7648	0	-2	0	-2	1	0	1
7700	2341	11242	7195	1	7700	-5	14340	11240	7194	0	7698	-6	-5	-6	1	1	0	1
7750	2341	11292	7199	1	7750	-1	14340	11290	7197	0	7748	-3	-1	-3	2	1	0	1
7800	2341	11342	7196	1	7800	-4	14340	11340	7194	0	7798	-6	-4	-6	2	1	0	1
7850	2341	11392	7198	1	7850	-2	14340	11390	7200	0	7848	0	-2	0	-2	1	0	1
7900	2341	11442	7199	1	7900	-1	14340	11440	7200	0	7898	0	-1	0	-1	1	0	1
7950	2341	11492	7198	1	7950	-2	14343	11490	7196	3	7948	-4	-2	-4	2	1	3	-2
8000	2341	11542	7200	1	8000	0	14340	11540	7202	0	7998	2	0	2	-2	1	0	1
8050	2341	11592	7199	1	8050	-1	14345	11590	7200	5	8048	0	-1	0	-1	1	5	-4
8100	2341	11642	7199	1	8100	-1	14345	11640	7199	5	8098	-1	-1	-1	0	1	5	-4
8150	2341	11692	7198	1	8150	-2	14346	11690	7197	6	8148	-3	-2	-3	1	1	6	-5
8200	2340	11742	7199	0	8200	-1	14347	11740	7198	7	8198	-2	-1	-2	1	0	7	-7
8250	2342	11792	7198	2	8250	-2	14348	11790	7196	8	8248	-4	-2	-4	2	2	8	-6
8300	2341	11842	7198	1	8300	-2	14346	11840	7198	6	8298	-2	-2	-2	0	1	6	-5
8350	2341	11892	7199	1	8350	-1	14347	11890	7200	7	8348	0	-1	0	-1	1	7	-6
8400	2341	11942	7196	1	8400	-4	14344	11940	7194	4	8398	-6	-4	-6	2	1	4	-3
8450	2341	11992	7196	1	8450	-4	14344	11990	7195	4	8448	-5	-4	-5	1	1	4	-3
8500	2341	12042	7196	1	8500	-4	14344	12040	7197	4	8498	-3	-4	-3	-1	1	4	-3
8550	2341	12092	7196	1	8550	-4	14344	12090	7194	4	8548	-6	-4	-6	2	1	4	-3
8600	2342	12142	7199	2	8600	-1	14343	12140	7201	3	8598	1	-1	1	-2	2	3	-1
8650	2343	12192	7200	3	8650	0	14343	12190	7201	3	8648	1	0	1	-1	3	3	0
8700	2342	12242	7198	2	8700	-2	14344	12240	7197	4	8698	-3	-2	-3	1	2	4	-2
8750	2343	12292	7197	3	8750	-3	14345	12290	7198	5	8748	-2	-3	-2	-1	3	5	-2
8800	2342	12342	7197	2	8800	-3	14342	12340	7196	2	8798	-4	-3	-4	1	2	2	0
8850	2343	12392	7197	3	8850	-3	14345	12390	7195	5	8848	-5	-3	-5	2	3	5	-2
4300	2346	7842	7200	6	4300	0	14348	7840	7201	8	4298	1	0	1	-1	6	8	-2
4350	2347	7892	7200	7	4350	0	14347	7890	7202	7	4348	2	0	2	-2	7	7	0
4400	2348	7942	7199	8	4400	-1	14350	7940	7199	10	4398	-1	-1	-1	0	8	10	-2
4450	2347	7992	7200	7	4450	0	14349	7990	7198	9	4448	-2	0	-2	2	7	9	-2
4500	2346	8042	7198	6	4500	-2	14345	8040	7200	5	4498	0	-2	0	-2	6	5	1
4550	2345	8092	7200	5	4550	0	14344	8090	7200	4	4548	0	0	0	0	5	4	1
4600	2345	8142	7200	5	4600	0	14346	8140	7201	6	4598	1	0	1	-1	5	6	-1
4650	2345	8192	7198	5	4650	-2	14345	8190	7199	5	4648	-1	-2	-1	-1	5	5	0
4700	2344	8242	7201	4	4700	1	14344	8240	7203	4	4698	3	1	3	-2	4	4	0
4750	2345	8292	7198	5	4750	-2	14346	8290	7198	6	4748	-2	-2	-2	0	5	6	-1
4800	2346	8342	7200	6	4800	0	14348	8340	7202	8	4798	2	0	2	-2	6	8	-2
4850	2345	8392	7199	5	4850	-1	14344	8390	7200	4	4848	0	-1	0	-1	5	4	1
4900	2345	8442	7198	5	4900	-2	14345	8440	7200	5	4898	0	-2	0	-2	5	5	0
4950	2345	8492	7201	5	4950	1	14344	8490	7202	4	4948	2	1	2	-1	5	4	1
5000	2345	8542	7201	5	5000	1	14345	8540	7201	5	4998	1	1	1	0	5	5	0
5050	2344	8592	7200	4	5050	0	14346	8590	7200	6	5048	0	0	0	0	4	6	-2
5100	2345	8642	7201	5	5100	1	14344	8640	7200	4	5098	0	1	0	1	5	4	1
5150	2345	8692	7198	5	5150	-2	14347	8690	7200	7	5148	0	-2	0	-2	5	7	-2
5200	2345	8742	7199	5	5200	-1	14346	8740	7198	6	5198	-2	-1	-2	1	5	6	-1
5250	2345	8792	7197	5	5250	-3	14344	8790	7198	4	5248	-2	-3	-2	-1	5	4	1
5300	2345	8842	7199	5	5300	-1	14345	8840	7200	5	5298	0	-1	0	-1	5	5	0
5350	2343	8892	7196	3	5350	-4	14342	8890	7194	2	5348	-6	-4	-6	2	3	2	1
5400	2343	8942	7200	3	5400	0	14345	8940	7202	5	5398	2	0	2	-2	3	5	-2
5450	2343	8992	7197	3	5450	-3	14345	8990	7197	5	5448	-3	-3	-3	0	3	5	-2
5500	2343	9042	7199	3	5500	-1	14342	9040	7200	2	5498	0	-1	0	-1	3	2	1
5550	2342	9092	7200	2	5550	0	14344	9090	7200	4	5548	0	0	0	0	2	4	-2
5600	2345	9142	7200	5	5600	0	14347	9140	7201	7	5598	1	0	1	-1	5</td		

8900	2342	12442	7197	2	8900	-3	14342	12440	7199	2	8898	-1	-3	-1	-2	2	2	0	
8950	2344	12492	7200	4	8950	0	14344	12490	7198	4	8948	-2	0	-2	2	4	4	0	
9000	2343	12542	7196	3	9000	-4	14345	12540	7198	5	8998	-2	-2	-2	-2	3	5	-2	
9050	2342	12592	7197	2	9050	-3	14344	12590	7199	4	9048	-1	-3	-1	-2	2	4	-2	
9100	2343	12642	7197	3	9100	-3	14344	12640	7195	4	9098	-5	-3	-5	2	3	4	-1	
9150	2342	12692	7199	2	9150	-1	14343	12690	7200	3	9148	0	-1	0	-1	2	3	-1	
9200	2344	12742	7198	4	9200	-2	14345	12740	7200	5	9198	0	-2	0	-2	4	5	-1	
9250	2343	12792	7200	3	9250	0	14343	12790	7198	3	9248	-2	0	-2	2	3	3	0	
9300	2342	12842	7200	2	9300	0	14344	12840	7201	4	9298	-1	0	1	-1	2	4	-2	
9350	2343	12892	7196	3	9350	-4	14344	12890	7198	4	9348	-2	-4	-2	-2	3	4	-1	
9400	2342	12942	7197	2	9400	-3	14343	12940	7196	3	9398	-4	-3	-4	1	2	3	-1	
9450	2343	12992	7199	3	9450	-1	14342	12990	7198	2	9448	-2	-1	-2	1	3	2	1	
9500	2343	13042	7199	3	9500	-1	14342	13040	7198	2	9498	-2	-1	-2	1	3	2	1	
9550	2343	13092	7198	3	9550	-2	14343	13090	7200	3	9548	0	-2	0	-2	3	3	0	
9600	2342	13142	7200	2	9600	0	14344	13140	7199	4	9598	-1	0	-1	1	2	4	-2	
9650	2342	13192	7202	2	9650	2	14343	13190	7204	3	9648	4	2	4	-2	2	3	-1	
9700	2342	13242	7200	2	9700	0	14343	13240	7200	3	9698	0	0	0	0	2	3	-1	
9750	2342	13292	7202	2	9750	2	14341	13290	7200	1	9748	0	2	0	2	2	1	1	
9800	2343	13342	7201	3	9800	1	14343	13340	7203	3	9798	3	1	3	-2	3	3	0	
9850	2343	13392	7200	3	9850	0	14342	13390	7199	2	9848	-1	0	-1	1	3	2	1	
9900	2343	13442	7199	3	9900	-1	14343	13440	7198	3	9898	-2	-1	-2	1	3	3	0	
9950	2343	13492	7198	3	9950	-2	14343	13490	7197	3	9948	-3	-2	-3	1	3	3	0	
10000	2343	13542	7201	3	10000	1	14344	13540	7202	4	9998	2	1	2	-1	3	4	-1	
10050	2343	13592	7202	3	10050	2	14342	13590	7203	2	10048	3	2	3	-1	3	2	1	
10100	2343	13642	7200	3	10100	0	14343	13640	7202	3	10098	2	0	2	-2	3	3	0	
10150	2343	13692	7200	3	10150	0	14345	13690	7202	5	10148	2	0	2	-2	3	5	-2	
10200	2343	13742	7202	3	10200	2	14344	13740	7201	4	10198	1	2	1	1	3	4	-1	
10250	2343	13792	7201	3	10250	1	14344	13790	7201	4	10248	1	1	1	0	3	4	-1	
10300	2343	13842	7199	3	10300	-1	14343	13840	7198	3	10298	-2	-1	-2	1	3	3	0	
10350	2343	13892	7199	3	10350	-1	14343	13890	7198	3	10348	-2	-1	-2	1	3	3	0	
10400	2343	13942	7198	3	10400	-2	14342	13940	7200	2	10398	0	-2	0	-2	3	2	1	
10450	2342	13992	7202	2	10450	2	14342	13990	7204	2	10448	4	2	4	-2	2	2	0	
10500	2344	14042	7202	4	10500	2	14343	14040	7201	3	10498	1	2	1	1	4	3	1	
10550	2344	14092	7200	4	10550	0	14344	14090	7201	4	10548	1	0	1	-1	4	4	0	
10600	2344	14142	7201	4	10600	1	14346	14140	7200	6	10598	0	1	0	1	4	6	-2	
10650	2344	14192	7202	4	10650	2	14343	14190	7203	3	10648	3	2	3	-1	4	3	1	
10700	2344	14242	7202	4	10700	2	14345	14240	7201	5	10698	1	2	1	1	4	5	-1	
10750	2344	14292	7200	4	10750	0	14343	14290	7200	3	10748	0	0	0	0	4	3	1	
10800	2344	14342	7200	4	10800	0	14343	14340	7198	3	10798	-2	-2	0	-2	2	4	3	
10850	2344	14392	7200	4	10850	0	14345	14390	7202	5	10848	2	0	2	-2	4	5	-1	
10900	2345	14442	7199	5	10900	-1	14344	14440	7201	4	10898	1	-1	1	-2	5	4	1	
10950	2346	14492	7199	6	10950	-1	14346	14490	7201	6	10948	1	-1	1	-2	6	6	0	
11000	2345	14542	7200	5	11000	0	14345	14540	7199	5	10998	-1	0	-1	1	5	5	0	
11050	2345	14592	7199	5	11050	-1	14346	14590	7200	6	11048	0	-1	0	-1	5	6	-1	
11100	2345	14642	7198	5	11100	-2	14345	14640	7199	5	11098	-1	-2	-1	-1	5	5	0	
11150	2345	14692	7202	5	11150	2	14347	14690	7202	7	11148	2	2	2	0	0	5	7	-2

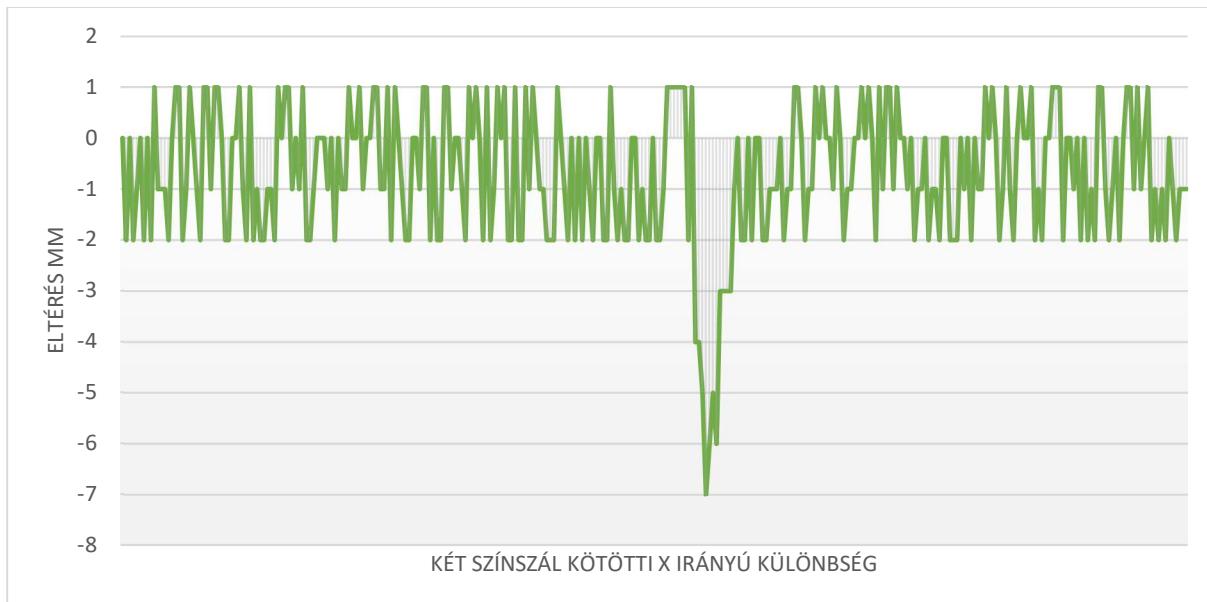
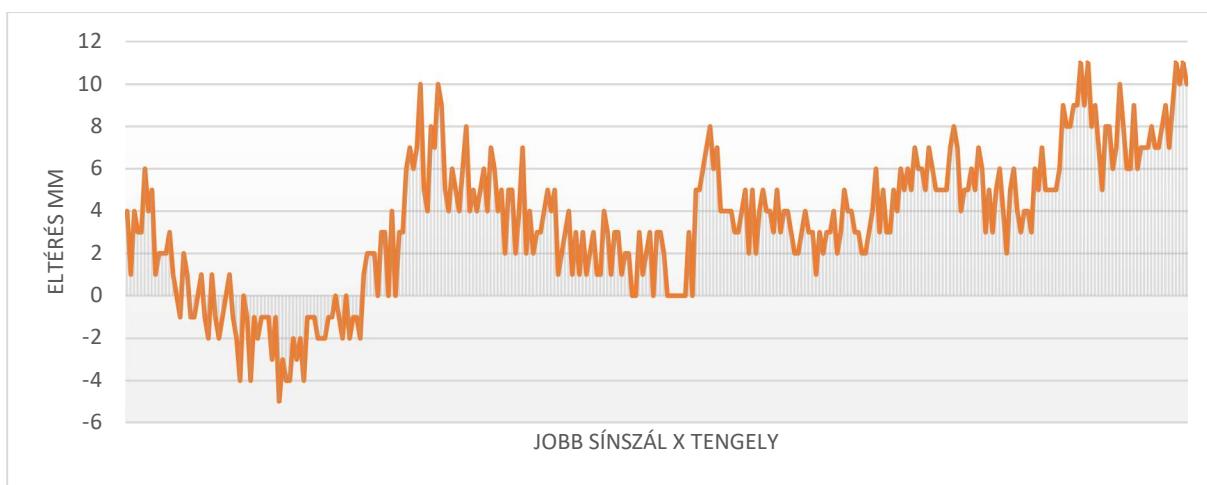
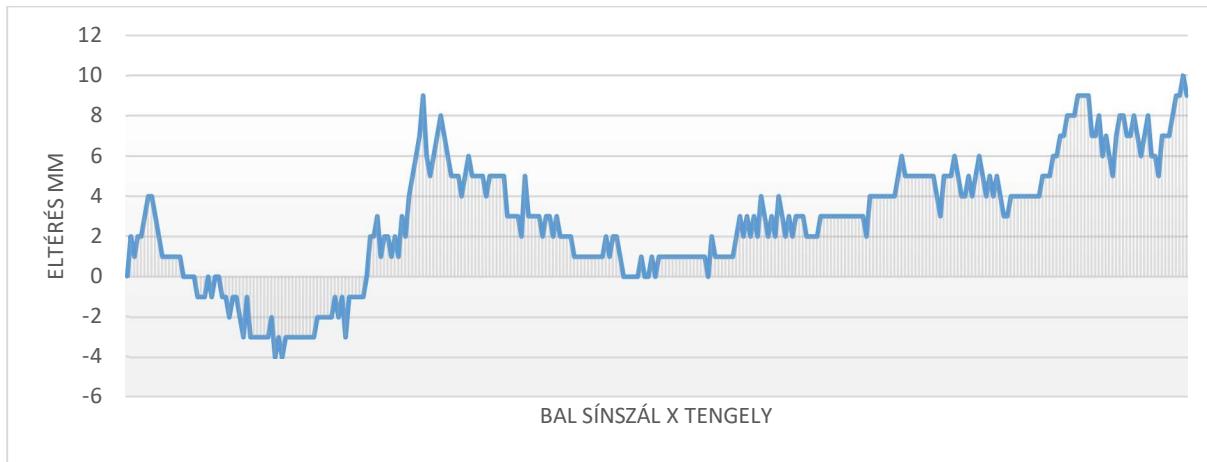
11200	2345	14742	7201	5	11200	1	14346	14740	7202	6	11198	2	1	2	-1	5	6	-1
11250	2345	14792	7202	5	11250	2	14346	14790	7200	6	11248	0	2	0	2	5	6	-1
11300	2345	14842	7199	5	11300	-1	14345	14840	7199	5	11298	-1	-1	-1	0	5	5	0
11350	2345	14892	7200	5	11350	0	14347	14890	7198	7	11348	-2	0	-2	2	5	7	-2
11400	2345	14942	7201	5	11400	1	14346	14940	7200	6	11398	0	1	0	1	5	6	-1
11450	2344	14992	7201	4	11450	1	14345	14990	7200	5	11448	0	1	0	1	4	5	-1
11500	2343	15042	7200	3	11500	0	14345	15040	7200	5	11498	0	0	0	0	3	5	-2
11550	2345	15092	7199	5	11550	-1	14345	15090	7197	5	11548	-3	-1	-3	2	5	5	0
11600	2345	15142	7200	5	11600	0	14345	15140	7201	5	11598	1	0	1	-1	5	5	0
11650	2345	15192	7199	5	11650	-1	14347	15190	7197	7	11648	-3	-1	-3	2	5	7	-2
11700	2346	15242	7200	6	11700	0	14348	15240	7198	8	11698	-2	0	-2	2	6	8	-2
11750	2345	15292	7199	5	11750	-1	14347	15290	7200	7	11748	0	-1	0	-1	5	7	-2
11800	2344	15342	7200	4	11800	0	14344	15340	7201	4	11798	1	0	1	-1	4	4	0
11850	2344	15392	7198	4	11850	-2	14345	15390	7199	5	11848	-1	-2	-1	-1	4	5	-1
11900	2345	15442	7198	5	11900	-2	14345	15440	7198	5	11898	-2	-2	-2	0	5	5	0
11950	2344	15492	7201	4	11950	1	14346	15490	7202	6	11948	2	1	2	-1	4	6	-2
12000	2345	15542	7201	5	12000	1	14345	15540	7202	5	11998	2	1	2	-1	5	5	0
12050	2346	15592	7200	6	12050	0	14347	15590	7199	7	12048	-1	0	-1	1	6	7	-1
12100	2345	15642	7199	5	12100	-1	14346	15640	7198	6	12098	-2	-1	-2	1	5	6	-1
12150	2344	15692	7201	4	12150	1	14343	15690	7201	3	12148	1	1	0	0	4	3	1
12200	2345	15742	7202	5	12200	2	14345	15740	7203	5	12198	3	2	3	-1	5	5	0
12250	2344	15792	7201	4	12250	1	14343	15790	7200	3	12248	0	1	0	1	4	3	1
12300	2345	15842	7199	5	12300	-1	14345	15840	7200	5	12298	0	-1	0	-1	5	5	

13500	2349	17042	7199	9	13500	-1	14351	17040	7197	11	13498	-3	-1	-3	2	9	11	-2
13550	2349	17092	7199	9	13550	-1	14349	17090	7199	9	13548	-1	-1	-1	0	9	9	0
13600	2349	17142	7200	9	13600	0	14351	17140	7201	11	13598	1	0	1	-1	9	11	-2
13650	2347	17192	7201	7	13650	1	14348	17190	7202	8	13648	2	1	2	-1	7	8	-1
13700	2347	17242	7199	7	13700	-1	14349	17240	7198	9	13698	-2	-1	-2	1	7	9	-2
13750	2348	17292	7201	8	13750	1	14347	17290	7203	7	13748	3	1	3	-2	8	7	1
13800	2346	17342	7200	6	13800	0	14345	17340	7198	5	13798	-2	0	-2	2	6	5	1
13850	2347	17392	7202	7	13850	2	14348	17390	7201	8	13848	1	2	1	1	7	8	-1
13900	2346	17442	7201	6	13900	1	14348	17440	7203	8	13898	3	1	3	-2	6	8	-2
13950	2345	17492	7200	5	13950	0	14346	17490	7201	6	13948	1	0	1	-1	5	6	-1
14000	2347	17542	7198	7	14000	-2	14347	17540	7198	7	13998	-2	-2	-2	0	7	7	0
14050	2348	17592	7200	8	14050	0	14350	17590	7199	10	14048	-1	0	-1	1	8	10	-2
14100	2348	17642	7200	8	14100	0	14348	17640	7200	8	14098	0	0	0	0	8	8	0
14150	2347	17692	7200	7	14150	0	14346	17690	7201	6	14148	1	0	1	-1	7	6	1
14200	2347	17742	7201	7	14200	1	14346	17740	7199	6	14198	-1	1	-1	2	7	6	1
14250	2348	17792	7201	8	14250	1	14349	17790	7203	9	14248	3	1	3	-2	8	9	-1
14300	2347	17842	7198	7	14300	-2	14346	17840	7196	6	14298	-4	-2	-4	2	7	6	1
14350	2346	17892	7200	6	14350	0	14347	17890	7201	7	14348	1	0	1	-1	6	7	-1
14400	2347	17942	7201	7	14400	1	14347	17940	7201	7	14398	1	1	1	0	7	7	0
14450	2348	17992	7200	8	14450	0	14347	17990	7198	7	14448	-2	0	-2	2	8	7	1
14500	2346	18042	7200	6	14500	0	14348	18040	7202	8	14498	2	0	2	-2	6	8	-2
14550	2346	18092	7202	6	14550	2	14347	18090	7204	7	14548	4	2	4	-2	6	7	-1
14600	2345	18142	7198	5	14600	-2	14347	18140	7196	7	14598	-4	-2	-4	2	5	7	-2
14650	2347	18192	7200	7	14650	0	14348	18190	7200	8	14648	0	0	0	0	7	8	-1
14700	2347	18242	7200	7	14700	0	14349	18240	7201	9	14698	1	0	1	-1	7	9	-2
14750	2347	18292	7202	7	14750	2	14347	18290	7200	7	14748	0	2	0	2	7	7	0
14800	2348	18342	7200	8	14800	0	14349	18340	7198	9	14798	-2	0	-2	2	8	9	-1
14850	2349	18392	7201	9	14850	1	14351	18390	7203	11	14848	3	1	3	-2	9	11	-2
14900	2349	18442	7202	9	14900	2	14350	18440	7204	10	14898	4	2	4	-2	9	10	-1
14950	2350	18492	7202	10	14950	2	14351	18490	7202	11	14948	2	2	2	0	10	11	-1
15000	2349	18542	7201	9	15000	1	14350	18540	7201	10	14998	1	1	1	0	9	10	-1

A Z irányú eltérések a két sínszál között +2 mm szórást mutatnak. A bal és a jobb sínszál nál a Z irányú változások követik a darupálya geometriai adottságait, az eltérésdiagramot alátámasztja a 8 méterenként elhelyezett konzolokból adódó lokális maximumok, hiszen itt a legmérévebb a szerkezet.



Az X irányú eltérésvizsgálat elvégzéséhez a nyers adatok vizsgálata elengedhetetlen. A két sínszál egymás alá helyezett diagramjából jól megfigyelhető egy a darupályában és valószínűleg az egész szerkezetben jelenlévő torzulás, Y irány mentén előre haladva, a meghatározott tengelytől való X irányú eltérés folyamatosan nő. Ennek oka, a koordinátarendszer felvételének pontatlansága. Ez a hiba azonban nem befolyásolja a mérési adatok kiértékelését, hiszen két szál közötti eltérés mértéke a mértékadó. A különbséggrafikonon jól megfigyelhető egy lokális minimum/maximum érték, -8 mm nagyságú csúccsal. Ez az érték a két sínszál párhuzamosságának eltérését mutatja. A hiba Y irányú koordinátáit összevetve a geometriai felméréssel megfigyelhető, hogy egy konzolos alátámasztás helyén rögzíthető az eltérés.



5.2.1. Teszt1

Tesztadatok:

I (30) boot: ESP-IDF v4.3-dev-771-gc77c4ccf6-dirty 2nd stage bootloader

I (30) boot: compile time 20:00:52

I (31) boot: chip revision: 1

I (35) boot_comm: chip revision: 1, min. bootloader chip revision: 0

I (42) boot.esp32: SPI Speed : 40MHz

I (47) boot.esp32: SPI Mode : DIO

I (51) boot.esp32: SPI Flash Size : 4MB

I (56) boot: Enabling RNG early entropy source...

I (61) boot: Partition Table:

I (65) boot: ## Label	Usage	Type	ST	Offset	Length
-----------------------	-------	------	----	--------	--------

I (72) boot: 0 nvs	WiFi data	01 02	00009000	00006000	
--------------------	-----------	-------	----------	----------	--

I (80) boot: 1 phy_init	RF data	01 01	0000f000	00001000	
-------------------------	---------	-------	----------	----------	--

I (87) boot: 2 factory	factory app	00 00	00010000	00100000	
------------------------	-------------	-------	----------	----------	--

I (95) boot: End of partition table

I (99) boot_comm: chip revision: 1, min. application chip revision: 0

I (106) esp_image: segment 0: paddr=0x00010020 vaddr=0x3f400020 size=0x1d6d0
(120528) map

I (161) esp_image: segment 1: paddr=0x0002d6f8 vaddr=0x3ffb0000 size=0x02920 (10528) load

I (165) esp_image: segment 2: paddr=0x00030020 vaddr=0x400d0020 size=0x881d0
(557520) map

0x400d0020: _stext at ???:?

I (380) esp_image: segment 3: paddr=0x000b81f8 vaddr=0x3ffb2920 size=0x01320 (4896) load

I (382) esp_image: segment 4: paddr=0x000b9520 vaddr=0x40080000 size=0x00404 (1028) load

0x40080000: _WindowOverflow4 at D:/Munka/DAMOR/esp-idf/components/freertos/xtensa/xtensa_vectors.S:1730

I (387) esp_image: segment 5: paddr=0x000b992c vaddr=0x40080404 size=0x18808 (100360) load

I (452) boot: Loaded app from partition at offset 0x10000

I (452) boot: Disabling RNG early entropy source...

I (464) cpu_start: Pro cpu up.

I (464) cpu_start: Starting app cpu, entry point is 0x40081550

0x40081550: call_start_cpu1 at D:/Munka/DAMOR/esp-idf/components/esp_system/port/cpu_start.c:107

I (0) cpu_start: App cpu up.

I (480) cpu_start: Pro cpu start user code

I (480) cpu_start: Application information:

I (480) cpu_start: Project name: damor

I (485) cpu_start: App version: 1

I (490) cpu_start: Compile time: Sep 28 2020 19:59:40

I (496) cpu_start: ELF file SHA256: 01c4d3ed63c1d504...

I (502) cpu_start: ESP-IDF: v4.3-dev-771-gc77c4ccf6-dirty

I (509) heap_init: Initializing. RAM available for dynamic allocation:

I (516) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM

I (522) heap_init: At 3FFB9518 len 00026AE8 (154 KiB): DRAM

I (528) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM

I (534) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM

I (541) heap_init: At 40098C0C len 000073F4 (28 KiB): IRAM

I (548) spi_flash: detected chip: generic

I (552) spi_flash: flash io: dio

I (557) cpu_start: Starting scheduler on PRO CPU.

I (0) cpu_start: Starting scheduler on APP CPU.

I (568) gpio: GPIO[4]| InputEn: 1| OutputEn: 0| OpenDrain: 0| Pullup: 0| Pulldown: 1|

Intr:1

I (578) gpio: GPIO[18]| InputEn: 1| OutputEn: 0| OpenDrain: 0| Pullup: 0| Pulldown: 1|

Intr:1

I (588) gpio: GPIO[19]| InputEn: 1| OutputEn: 0| OpenDrain: 0| Pullup: 0| Pulldown: 1|

Intr:1

I (598) Adxl355: Driver init done.

D (598) Rangefinder: 55

D (5598) Rangefinder: 00

I (598) Rangefinder: autobaud (-2)

D (5598) Rangefinder: aa 81 00 00 81

D (5598) Rangefinder: aa 81 00 00 00 01 00 81 03

I (5598) Rangefinder: status (129)

I (5598) Rangefinder: Driver initialized.

I (5658) wifi:wifi driver task: 3ffc5c10, prio:23, stack:6656, core=0

I (5658) system_api: Base MAC address is not set

I (5658) system_api: read default base MAC address from EFUSE

I (5678) wifi:wifi firmware version: 9c89486

I (5678) wifi:wifi certification version: v7.0

I (5678) wifi:config NVS flash: enabled
I (5678) wifi:config nano formating: disabled
I (5688) wifi:Init data frame dynamic rx buffer num: 32
I (5688) wifi:Init management frame dynamic rx buffer num: 32
I (5698) wifi:Init management short buffer num: 32
I (5698) wifi:Init dynamic tx buffer num: 32
I (5708) wifi:Init static rx buffer size: 1600
I (5708) wifi:Init static rx buffer num: 10
I (5718) wifi:Init dynamic rx buffer num: 32
I (5718) Wifi: Started in station mode.
I (5818) phy: phy_version: 4350, 18c5e94, Jul 27 2020, 21:04:07, 0, 0
I (5828) wifi:mode : sta (fc:f5:c4:28:f9:40)
I (5828) Wifi: Connecting to the access point...
I (5948) wifi:new:<11,0>, old:<1,0>, ap:<255,255>, sta:<11,0>, prof:1
I (5948) wifi:state: init -> auth (b0)
I (5988) wifi:state: auth -> assoc (0)
I (5998) wifi:state: assoc -> run (10)
I (6128) wifi:connected with UPC54DA919, aid = 2, channel 11, BW20, bssid = ac:22:05:80:89:57
I (6128) wifi:security: WPA2-PSK, phy: bgn, rssi: -51
I (6128) wifi:pm start, type: 1

I (6198) wifi:AP's beacon interval = 102400 us, DTIM period = 1
I (7148) esp_netif_handlers: sta ip: 192.168.0.213, mask: 255.255.255.0, gw: 192.168.0.1

I (7148) Wifi: Connected to the access point.

I (7148) Wifi: IP:192.168.0.213

I (7158) Wifi: Starting server on port '80'.

D (12878) Wifi: Websocket got packet.

D (12888) Wifi: Type: 2

D (12888) Wifi: Length: 116

D (12888) Wifi: Message:
eyJ0eXB1IjogInJlcXVlc3QiLCAib3BlcmF0aW9uIjogImNvbmZpZ3VyZSIzICJhcndzIjogeyJyYW5nZSI6IDMsICJvZHiiOiA0LCAiaHBmIjogMn19

I (12898) Application: Task notified about websocket request. (0x00, 0x01)

I (12898) Application: Adxl355 sensor configured.

D (12988) Wifi: Websocket got packet.

D (12988) Wifi: Type: 2

D (12988) Wifi: Length: 112

D (12988) Wifi: Message:
eyJ0eXB1IjogInJlcXVlc3QiLCAib3BlcmF0aW9uIjogImNhbGlicmF0ZSIzICJhcndzIjogeyJpdGVyYXRpb24iOiAxNiwgInNpemUiOiAzMn19

I (12998) Application: Task notified about websocket request. (0x00, 0x01)

I (13268) Adxl355: Sensor calibration read #1 retrieved 96 fifo entries.

I (13388) Adxl355: Sensor calibration read #2 retrieved 96 fifo entries.

I (13518) Adxl355: Sensor calibration read #3 retrieved 96 fifo entries.

I (13648) Adxl355: Sensor calibration read #4 retrieved 96 fifo entries.

I (13778) Adxl355: Sensor calibration read #5 retrieved 96 fifo entries.

I (13908) Adxl355: Sensor calibration read #6 retrieved 96 fifo entries.

I (14038) Adxl355: Sensor calibration read #7 retrieved 96 fifo entries.

I (14168) Adxl355: Sensor calibration read #8 retrieved 96 fifo entries.

I (14288) Adxl355: Sensor calibration read #9 retrieved 96 fifo entries.

I (14418) Adxl355: Sensor calibration read #10 retrieved 96 fifo entries.

I (14548) Adxl355: Sensor calibration read #11 retrieved 96 fifo entries.

I (14678) Adxl355: Sensor calibration read #12 retrieved 96 fifo entries.

I (14808) Adxl355: Sensor calibration read #13 retrieved 96 fifo entries.

I (14938) Adxl355: Sensor calibration read #14 retrieved 96 fifo entries.

I (15058) Adxl355: Sensor calibration read #15 retrieved 96 fifo entries.

I (15188) Adxl355: Sensor calibration read #16 retrieved 96 fifo entries.

I (15188) Application: Adxl355 sensor calibrated.

D (15258) Wifi: Websocket got packet.

D (15258) Wifi: Type: 2

D (15258) Wifi: Length: 120

D (15258) Wifi: Message:

eyJ0eXBIIjogInJlcXVlc3QiLCaib3BlcmF0aW9uIjogImNvbmZpZ3VyZSIIsICJhcmdzIjogeyJyYW5nZSI6IDMsICJvZHliOiAxMCwgImhwZiI6IDF9fQ==

I (15268) Application: Task notified about websocket request. (0x00, 0x01)

I (15278) Application: Adxl355 sensor configured.

D (15328) Wifi: Websocket got packet.

D (15328) Wifi: Type: 2

D (15328) Wifi: Length: 100

D (15328) Wifi: Message:

eyJ0eXBIIjogInJlcXVlc3QiLCaib3BlcmF0aW9uIjogIm1lYXN1cmUiLCaIYXJncyI6IHsibW9kZSI6IDYsICJkZXViOiAwfX0=

I (15338) Application: Task notified about websocket request. (0x00, 0x01)

D (15348) Rangefinder: Driver thread notified. (6)

I (15348) Application: Adxl355 sensor start measuring.

I (15358) Rangefinder: Measure. (0) (6)

D (15368) Rangefinder: aa 01 00 20 00 01 00 06 28

D (15868) Rangefinder: aa 01 00 22 00 03 00 00 03 28 00 50 a1

I (15868) Application: Rangefinder sensor measurement notification.

D (16268) Rangefinder: aa 01 00 22 00 03 00 00 03 29 00 53 a5

I (16268) Application: Rangefinder sensor measurement notification.

D (16678) Rangefinder: aa 01 00 22 00 03 00 00 03 28 00 4c 9d

I (16678) Application: Rangefinder sensor measurement notification.

D (17078) Rangefinder: aa 01 00 22 00 03 00 00 03 29 00 48 9a

5.2.2. Teszt2

D:\Munka\damor>"C:/Program Files/Python38/python.exe"
d:/Munka/DAMOR/esp-projects/damor/wsclient.py

{'type': 'response', 'operation': 'configure', 'status': 'accept', 'args': {}}

{'type': 'response', 'operation': 'configure', 'status': 'done', 'args': {}}

{'type': 'response', 'operation': 'calibrate', 'status': 'accept', 'args': {}}

{'type': 'response', 'operation': 'calibrate', 'status': 'done', 'args': {}}

{'type': 'response', 'operation': 'configure', 'status': 'accept', 'args': {}}

{'type': 'response', 'operation': 'configure', 'status': 'done', 'args': {}}

{'type': 'response', 'operation': 'measure', 'status': 'accept', 'args': {}}

{'type': 'response', 'operation': 'measure', 'status': 'done', 'args': {}}

acceleration X: 0.00 m/s^2, acceleration Y: 0.00 m/s^2, acceleration Z: 0.00 m/s^2,

temperature: 223.85 C, timestamp: 14.80 sec

acceleration X: 0.09 m/s², acceleration Y: -0.04 m/s², acceleration Z: 4.69 m/s²,
temperature: 22.08 C, timestamp: 15.06 sec

distance: 80.80 cm, quality: 80.00, timestamp: 15.32 sec

acceleration X: 0.17 m/s², acceleration Y: -0.09 m/s², acceleration Z: 9.36 m/s²,
temperature: 21.09 C, timestamp: 15.31 sec

distance: 80.90 cm, quality: 83.00, timestamp: 15.72 sec

acceleration X: 0.17 m/s², acceleration Y: -0.09 m/s², acceleration Z: 9.22 m/s²,
temperature: 22.19 C, timestamp: 15.57 sec

acceleration X: 0.16 m/s², acceleration Y: -0.08 m/s², acceleration Z: 9.07 m/s²,
temperature: 21.75 C, timestamp: 15.83 sec

distance: 80.80 cm, quality: 76.00, timestamp: 16.12 sec

acceleration X: 0.16 m/s², acceleration Y: -0.08 m/s², acceleration Z: 8.93 m/s²,
temperature: 21.86 C, timestamp: 16.08 sec

distance: 80.90 cm, quality: 72.00, timestamp: 16.53 sec

acceleration X: 0.16 m/s², acceleration Y: -0.08 m/s², acceleration Z: 8.79 m/s²,
temperature: 22.08 C, timestamp: 16.34 sec

acceleration X: 0.16 m/s², acceleration Y: -0.08 m/s², acceleration Z: 8.65 m/s²,
temperature: 21.64 C, timestamp: 16.60 sec

distance: 80.90 cm, quality: 86.00, timestamp: 16.93 sec

acceleration X: 0.15 m/s², acceleration Y: -0.08 m/s², acceleration Z: 8.52 m/s²,
temperature: 22.30 C, timestamp: 16.85 sec

distance: 80.90 cm, quality: 78.00, timestamp: 17.33 sec

acceleration X: 0.15 m/s², acceleration Y: -0.08 m/s², acceleration Z: 8.39 m/s²,
temperature: 22.30 C, timestamp: 17.11 sec

acceleration X: 0.15 m/s², acceleration Y: -0.08 m/s², acceleration Z: 8.25 m/s²,
temperature: 21.86 C, timestamp: 17.37 sec

distance: 80.80 cm, quality: 85.00, timestamp: 17.73 sec

acceleration X: 0.15 m/s², acceleration Y: -0.08 m/s², acceleration Z: 8.13 m/s², temperature: 21.75 C, timestamp: 17.62 sec

distance: 80.80 cm, quality: 65.00, timestamp: 18.13 sec

acceleration X: 0.14 m/s², acceleration Y: -0.07 m/s², acceleration Z: 8.00 m/s², temperature: 21.20 C, timestamp: 17.88 sec

acceleration X: 0.14 m/s², acceleration Y: -0.07 m/s², acceleration Z: 7.87 m/s², temperature: 22.08 C, timestamp: 18.14 sec

distance: 80.80 cm, quality: 64.00, timestamp: 18.53 sec

acceleration X: 0.14 m/s², acceleration Y: -0.07 m/s², acceleration Z: 7.75 m/s², temperature: 22.30 C, timestamp: 18.39 sec

acceleration X: 0.14 m/s², acceleration Y: -0.07 m/s², acceleration Z: 7.63 m/s², temperature: 22.30 C, timestamp: 18.65 sec

distance: 80.80 cm, quality: 70.00, timestamp: 18.93 sec

acceleration X: 0.13 m/s², acceleration Y: -0.07 m/s², acceleration Z: 7.51 m/s², temperature: 21.42 C, timestamp: 18.91 sec

distance: 80.80 cm, quality: 86.00, timestamp: 19.33 sec

acceleration X: 0.13 m/s², acceleration Y: -0.07 m/s², acceleration Z: 7.39 m/s², temperature: 22.19 C, timestamp: 19.16 sec

acceleration X: 0.13 m/s², acceleration Y: -0.07 m/s², acceleration Z: 7.28 m/s², temperature: 21.75 C, timestamp: 19.42 sec

distance: 80.90 cm, quality: 77.00, timestamp: 19.74 sec

acceleration X: 0.13 m/s², acceleration Y: -0.07 m/s², acceleration Z: 7.16 m/s², temperature: 22.52 C, timestamp: 19.68 sec

distance: 80.90 cm, quality: 67.00, timestamp: 20.14 sec

acceleration X: 0.13 m/s², acceleration Y: -0.06 m/s², acceleration Z: 7.05 m/s², temperature: 21.20 C, timestamp: 19.93 sec

acceleration X: 0.12 m/s², acceleration Y: -0.06 m/s², acceleration Z: 6.94 m/s²,
temperature: 22.08 C, timestamp: 20.19 sec

distance: 80.90 cm, quality: 80.00, timestamp: 20.54 sec

acceleration X: 0.12 m/s², acceleration Y: -0.06 m/s², acceleration Z: 6.83 m/s²,
temperature: 21.75 C, timestamp: 20.45 sec

distance: 80.90 cm, quality: 72.00, timestamp: 20.94 sec

acceleration X: 0.12 m/s², acceleration Y: -0.06 m/s², acceleration Z: 6.73 m/s²,
temperature: 21.86 C, timestamp: 20.70 sec

acceleration X: 0.12 m/s², acceleration Y: -0.06 m/s², acceleration Z: 6.62 m/s²,
temperature: 21.42 C, timestamp: 20.96 sec

distance: 80.90 cm, quality: 69.00, timestamp: 21.34 sec

acceleration X: 0.12 m/s², acceleration Y: -0.06 m/s², acceleration Z: 6.52 m/s²,
temperature: 22.08 C, timestamp: 21.22 sec

acceleration X: 0.12 m/s², acceleration Y: -0.06 m/s², acceleration Z: 6.42 m/s²,
temperature: 21.20 C, timestamp: 21.47 sec

distance: 80.90 cm, quality: 71.00, timestamp: 21.74 sec

acceleration X: 0.11 m/s², acceleration Y: -0.06 m/s², acceleration Z: 6.32 m/s²,
temperature: 22.41 C, timestamp: 21.73 sec

distance: 80.80 cm, quality: 76.00, timestamp: 22.14 sec

acceleration X: 0.11 m/s², acceleration Y: -0.06 m/s², acceleration Z: 6.22 m/s²,
temperature: 21.31 C, timestamp: 21.99 sec

acceleration X: 0.11 m/s², acceleration Y: -0.06 m/s², acceleration Z: 6.12 m/s²,
temperature: 22.19 C, timestamp: 22.25 sec

distance: 80.80 cm, quality: 69.00, timestamp: 22.54 sec

acceleration X: 0.11 m/s², acceleration Y: -0.06 m/s², acceleration Z: 6.02 m/s²,
temperature: 21.42 C, timestamp: 22.50 sec

distance: 80.80 cm, quality: 74.00, timestamp: 22.95 sec

acceleration X: 0.11 m/s², acceleration Y: -0.05 m/s², acceleration Z: 5.93 m/s², temperature: 21.75 C, timestamp: 22.76 sec

acceleration X: 0.10 m/s², acceleration Y: -0.05 m/s², acceleration Z: 5.84 m/s², temperature: 22.30 C, timestamp: 23.02 sec

distance: 80.80 cm, quality: 75.00, timestamp: 23.35 sec

acceleration X: 0.10 m/s², acceleration Y: -0.05 m/s², acceleration Z: 5.75 m/s², temperature: 22.08 C, timestamp: 23.27 sec

distance: 80.90 cm, quality: 76.00, timestamp: 23.75 sec

acceleration X: 0.10 m/s², acceleration Y: -0.05 m/s², acceleration Z: 5.66 m/s², temperature: 22.41 C, timestamp: 23.53 sec

acceleration X: 0.10 m/s², acceleration Y: -0.05 m/s², acceleration Z: 5.57 m/s², temperature: 22.08 C, timestamp: 23.79 sec

distance: 80.80 cm, quality: 71.00, timestamp: 24.15 sec

acceleration X: 0.10 m/s², acceleration Y: -0.05 m/s², acceleration Z: 5.48 m/s², temperature: 22.30 C, timestamp: 24.04 sec

distance: 80.80 cm, quality: 84.00, timestamp: 24.55 sec

acceleration X: 0.10 m/s², acceleration Y: -0.05 m/s², acceleration Z: 5.40 m/s², temperature: 21.97 C, timestamp: 24.30 sec

acceleration X: 0.10 m/s², acceleration Y: -0.05 m/s², acceleration Z: 5.31 m/s², temperature: 22.30 C, timestamp: 24.56 sec

distance: 80.80 cm, quality: 71.00, timestamp: 24.95 sec

acceleration X: 0.09 m/s², acceleration Y: -0.05 m/s², acceleration Z: 5.23 m/s², temperature: 21.64 C, timestamp: 24.81 sec

acceleration X: 0.09 m/s², acceleration Y: -0.05 m/s², acceleration Z: 5.15 m/s², temperature: 22.19 C, timestamp: 25.07 sec

distance: 80.80 cm, quality: 70.00, timestamp: 25.35 sec

acceleration X: 0.09 m/s², acceleration Y: -0.04 m/s², acceleration Z: 5.07 m/s²,
temperature: 21.31 C, timestamp: 25.33 sec

distance: 80.80 cm, quality: 55.00, timestamp: 25.76 sec

acceleration X: 0.09 m/s², acceleration Y: -0.04 m/s², acceleration Z: 4.99 m/s²,
temperature: 22.30 C, timestamp: 25.58 sec

acceleration X: 0.09 m/s², acceleration Y: -0.04 m/s², acceleration Z: 4.91 m/s²,
temperature: 21.42 C, timestamp: 25.84 sec

distance: 80.80 cm, quality: 75.00, timestamp: 26.16 sec

acceleration X: 0.09 m/s², acceleration Y: -0.04 m/s², acceleration Z: 4.83 m/s²,
temperature: 21.20 C, timestamp: 26.10 sec

distance: 80.90 cm, quality: 75.00, timestamp: 26.56 sec

acceleration X: 0.09 m/s², acceleration Y: -0.09 m/s², acceleration Z: 4.76 m/s²,
temperature: 22.08 C, timestamp: 26.35 sec

acceleration X: 0.08 m/s², acceleration Y: -0.09 m/s², acceleration Z: 4.68 m/s²,
temperature: 21.20 C, timestamp: 26.61 sec

distance: 81.00 cm, quality: 72.00, timestamp: 26.96 sec

acceleration X: 0.08 m/s², acceleration Y: -0.17 m/s², acceleration Z: 4.61 m/s²,
temperature: 21.09 C, timestamp: 26.87 sec

distance: 81.00 cm, quality: 59.00, timestamp: 27.36 sec

acceleration X: 0.08 m/s², acceleration Y: -0.24 m/s², acceleration Z: 4.53 m/s²,
temperature: 22.19 C, timestamp: 27.12 sec

acceleration X: 0.08 m/s², acceleration Y: -0.25 m/s², acceleration Z: 4.47 m/s²,
temperature: 22.19 C, timestamp: 27.38 sec

distance: 80.90 cm, quality: 61.00, timestamp: 27.76 sec

acceleration X: 0.11 m/s², acceleration Y: -0.35 m/s², acceleration Z: 4.38 m/s²,
temperature: 21.64 C, timestamp: 27.64 sec

acceleration X: 0.11 m/s², acceleration Y: -0.37 m/s², acceleration Z: 4.32 m/s²,
temperature: 22.08 C, timestamp: 27.89 sec

distance: 80.90 cm, quality: 57.00, timestamp: 28.16 sec

acceleration X: 0.11 m/s², acceleration Y: -0.37 m/s², acceleration Z: 4.25 m/s²,
temperature: 21.09 C, timestamp: 28.15 sec

distance: 81.00 cm, quality: 72.00, timestamp: 28.57 sec

acceleration X: 0.11 m/s², acceleration Y: -0.36 m/s², acceleration Z: 4.18 m/s²,
temperature: 21.20 C, timestamp: 28.41 sec

acceleration X: 0.10 m/s², acceleration Y: -0.35 m/s², acceleration Z: 4.12 m/s²,
temperature: 22.19 C, timestamp: 28.66 sec

distance: 81.00 cm, quality: 60.00, timestamp: 28.97 sec

acceleration X: 0.09 m/s², acceleration Y: -0.30 m/s², acceleration Z: 4.06 m/s²,
temperature: 21.20 C, timestamp: 28.92 sec

distance: 81.10 cm, quality: 56.00, timestamp: 29.37 sec

acceleration X: 0.09 m/s², acceleration Y: -0.30 m/s², acceleration Z: 3.99 m/s²,
temperature: 21.64 C, timestamp: 29.18 sec

acceleration X: 0.11 m/s², acceleration Y: -0.33 m/s², acceleration Z: 3.93 m/s²,
temperature: 21.20 C, timestamp: 29.43 sec

distance: 81.00 cm, quality: 84.00, timestamp: 29.77 sec

acceleration X: 0.10 m/s², acceleration Y: -0.33 m/s², acceleration Z: 3.86 m/s²,
temperature: 21.86 C, timestamp: 29.69 sec

distance: 81.10 cm, quality: 79.00, timestamp: 30.17 sec

6. Prototípus - Telepítési és használati útmutató

Az elkészült prototípus darura történő telepítésének és kalibrációs folyamatának lépései:

1. Wifi router helyzetének kimérése a csarnokban
2. Wifi router hálózatra kötése, jelerősség mérés elvégzése a darupálya távolabbi pontjain
3. Wifi router pozíció váltás, ha a mérési eredmények nem elégségesek
4. DAMOR szenzor előzetes elhelyezése a híddaru sínszál feletti elemén úgy, hogy a két végfalra szabad rálátást biztosítson az Y irányú Lidar szenzoroknak.
5. DAMOR szenzor előzetes pozícionálása úgy, hogy az X irányú Lidar szenzor a szemközti oldalon lévő tengely felett elhelyezett, jó fényvisszaverő képességű lapra akadálytalanul rálásson.
6. DAMOR szenzor ráhelyezése az OBEN két irányú makrósínre, a sín ideiglenes rögzítése a darutestre.
7. DAMOR szenzor tápfeszültség biztosítása akkumulátorról, MOVIX rendszer elindítása
8. Közvetlen WiFi kapcsolódás megteremtése a DAMOR szenzor és a tablet között (2.4G)
9. A tableten böngészőből futtatva a folyamatos adatfolyam elindítása, tesztmérés indítása
10. Az Y irányú mérések megfigyelése, párhuzamosság beállítása a makrósínnel. Null érték meghatározása a végpontra történő mozgatással
11. Az X irányú érték finomhangolása a makrósínnel, egész cm értékre hozása, Null érték meghatározása
12. Tengelyek feljelölése markerrel a darutestre
13. OBEN rendszer eltávolítása, a DAMOR szenzor fix rögzítése kétoldalú, gumírozott vagy szivacsbetétes ragasztószalag segítségével.
14. Rögzítés ellenőrzése
15. Null állapotok ellenőrzése
16. Tesztmérés a két végpont között
17. Tesztmérésből származó adatok PC-n történő elemzése, meglévő anomáliák figyelmen kívül hagyásához TARE állapot létrehozása
18. Rendszer kapcsolódása a helyi WiFi hálózathoz, PC-hez, NAS-hoz.

19. Határértékek finomhangolása
20. Készenléti állapot bekapcsolása