



**Budapesti Műszaki és Gazdaságtudományi Egyetem**  
Villamosmérnöki és Informatikai Kar  
Automatizálási és Alkalmazott Informatikai Tanszék

Sásdi András

# **KÉT TENGYELŰ FORGÓASZTAL TERVEZÉSE GYORSULÁSMÉRŐ SZENZOROK KARAKTERIZÁLÁSÁHOZ**

KONZULENS

Hajnal Dániel

Dr. Stumpf Péter Pál

BUDAPEST, 2025

# Tartalomjegyzék

<b>Összefoglaló .....</b>	<b>I</b>
<b>Abstract.....</b>	<b>II</b>
<b>1 Bevezetés .....</b>	<b>1</b>
1.1 A munka indoka és célja .....	1
1.2 Specifikáció .....	1
1.3 Feladatok áttekintése.....	2
1.4 Projekt arculata .....	2
1.5 Szakirodalmi áttekintés, elméleti összefoglaló és tervezési irányelvek.....	3
1.5.1 MEMS szenzorok .....	3
1.5.2 Biztonságtechnikai lépések .....	4
1.5.3 Kéttengelyű forgatómechanizmusok .....	5
<b>2 Modulok .....</b>	<b>8</b>
2.1 Mikrovezérlő.....	9
2.2 Motorok .....	11
2.2.1 Motorvezérlő IC.....	12
2.2.2 Léptetőmotorok sebességprofil vezérlése .....	13
2.2.3 Trigonometrikus sebességprofil.....	14
2.2.4 Numerikus módszerek .....	17
2.2.5 Módszer implementálása beágyazott eszközre .....	21
2.2.6 Enkóderek .....	21
2.2.7 Driver .....	23
2.3 Kezelőfelület.....	24
2.3.1 Állapotgép.....	25
2.3.2 Számítógépes vezérlés .....	27
2.4 Kijelző.....	27
2.4.1 Nyomtatott áramkör .....	28
2.4.2 Driver .....	31
2.4.3 Grafikus kezelői felület.....	32
2.5 Gomb panel.....	33
2.6 Numerikus billentyűzet.....	36
2.7 Referenciaszenzor .....	41

2.7.1 Driver .....	42
2.8 Alaplap .....	44
2.8.1 Táp áramkör .....	44
2.8.2 Vészkapcsoló .....	46
2.8.3 Külső trigger jel .....	50
2.8.4 SPI lehallgató .....	51
2.8.5 EEPROM .....	51
2.8.6 EEPROM driver .....	52
2.8.7 Mikrovezérlő és további modularitási/bővítési megfontolások .....	53
<b>3 Összefoglalás/eredmények értékelése .....</b>	<b>56</b>
3.1 Eredmények .....	56
3.2 Javaslatok/Következtetések/Tanulságok .....	57
<b>4 Irodalomjegyzék .....</b>	<b>58</b>
<b>Függelék .....</b>	<b>61</b>

# HALLGATÓI NYILATKOZAT

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

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

Kelt: Budapest, 2025. 12. 18.

.....  
Sásdi András

# Összefoglaló

A diplomaunka MEMS gyorsulásmérő szenzorok karakterizálására alkalmas két tengely körüli forgatást megvalósító eszköz tervezését és kivitelezését mutatja be. Mint azt az 1.5.1 fejezetben is említettem, ezek a szenzorok működésükből adódóan rendelkeznek ofszettel, amit bizonyos szenzorok képesek kompenzálni (hasonlóan ahhoz, ahogy egy konyhai mérleg bekapcsolás után tárazza önmagát). Egyebek mellett ennek a viselkedésnek a karakterizálásához is szükséges lehet mérés közben változtatni a szenzor orientációját. A jelenlegi eljárás szerint ezt a forgatást a mérést végző karakterizáló mérnök manuálisan hajtja végre. Ennek a jelenlegi protokollnak több hátránya is létezik, mint például a manuális forgatáshoz szükséges idő, a méréshez való jelenlét szükségessége, a nem automatizálhatóság, a rossz ismételhetőség és a forgatás paramétereinek gyenge reprodukálhatósága és befolyásolhatósága. A diplomamunka során megvalósítandó eszköz ezeket a problémákat oldja meg, így csökkentve a mérést végző mérnökön a terhelést és a méréshez szükséges időt.

A diplomaunka első fele a szükséges irodalomkutatásról szól, míg a második fele az eszköz konkrét megvalósításáról. A mű során bemutatom a döntési láncaimat, beszámolok a kihívásokról és eredményekről. Az eszköz megvalósítása során drivereket írtam, áramköröket terveztem, mechanikai kivitelezéseket hasonlítottam össze és biztonságtechnikai megfontolásokat gyűjtöttem össze, a teljes tervezési folyamat végig követhető.

# Abstract

The goal of the thesis is to design and implement a controlled two-axis rotary table that is adequate for the characterization of MEMS accelerometers. As discussed in chapter 1.5.1 due to their physical properties and working principles, MEMS sensors have an offset that some of these sensors can automatically cancel out (think of a digital kitchen scale that tares itself upon startup). Among others, the characterization of the above-mentioned behaviour is one of the instances where it is necessary to change the orientation of the characterised sensor during a measurement. The current protocol is manual rotation of the sensor, which requires the presence of the engineer performing the measurement. This current method has many downsides, such as the time it takes to manually rotate the sensor, the impossibility of doing these measurements remotely or the poor reproducibility of the exact timing and speed of such rotations. The device created in this thesis aims to solve these issues thereby easing the workload of the characterization engineers and shortening the runtime of these measurements.

The thesis shows the thought process behind the design and implementation of certain modules and features. The first part shows the literature review, while the second part focuses on the designs and implementation of the modules. To realize the project I implemented drivers, designed circuits and compared different mechanical approaches and taken the relevant safety guidelines into account.

# Előszó

*A következő dokumentum egy teljes mérőeszköz tervezésének folyamatát mutatja be. Ennek megvalósításához szükségem volt tanulmányaim során elsajátított tudásomhoz. Úgy érzem sikeresen mutatja be jelenlegi képességeimet és azt, hogy miben kell még fejlődnöm.*

\*\*\*

Ezúton szeretnék köszönetet nyilvánítani mindazoknak, akik segítségemre voltak ennek a diplomamunkának megvalósításában.

Első sorban szeretném kifejezni hálámat Hajnal Dániel témavezetőmnek, aki rengeteget segített a feladat megoldásában, tanácsokat adott, megoldásokat javasolt és segített a jó irányból meglátni a kihívásokat és ahol tudott ott segített.

Köszönöm egyetemi tanárainknak a tudásért, ami nélkül nem lettem volna képes erre a feladatra. Külön köszönetet szeretnék mondani Stumpf Péter egyetemi docensnek, akihez bármikor fordulhattam szakmai kérdéseimmel.

Végezetül szeretnék köszönetet mondani családomnak, különösen anyai nagypapámnak, aki mindig is példa volt számomra örökös tudásvágyával és szerteágazó műszaki tudásával.

*Budapest, 2025*

*Sásdi András*

# 1 Bevezetés

## 1.1 A munka indoka és célja

A feladat mikroelektromechanikai (MEMS) gyorsulásszenzor karakterizálásához felhasználható vezérelhető kéttengelyes forgóasztal tervezése és megvalósítása. A MEMS szenzorok, ahogy azt a 1.5.1 fejezetben kifejttem működésükből és fizikai kialakításukból adódóan ofszettel rendelkeznek, melyet ezen eszközök egyes típusai automatikusan lenulláznak (akárcsak egyes digitális konyhai mérlegek melyek bekapcsolás után tárázzák önmagukat). A szenzorok karakterizálása során többek közt ezen viselkedések kiméréséhez szükséges a szenzor orientációjának változtatása. Jelen eljárás szerint ilyen mérések során a mérést végző mérnök jelenléte szükséges, hogy a mért szenzort manuálisan a megfelelő orientációba forgassa (természetesen a gyártásban lévő szenzorok esetén ez nem manuálisan történik). Ez nem csak meglassítja a mérési folyamatot, de ellehetetleníti a mérés távolról végzését, valamint az ismételhetséget is korlátozza. A diplomaterv kereteiben létrehozott eszköz célja ezen problémák megoldása, ezáltal a tesztelést végző karakterizáló mérnök munkájának könnyítése és a mérési idők csökkentése.

## 1.2 Specifikáció

A gyorsulásmérő szenzorok karakterizálását segítő eszközt a következő elvárások teljesítésével kell megvalósítani:

- Két egymásra merőleges tengely mentén legyen képes forgatni egy mért szenzort, tengelyenként egy teljes elfordulásra legyen képes!
- A (legalább egy) mért szenzor és a szenzort kezelő külső mérőeszköz (SPI-Simulyzer) között a forgatás során biztosítva legyen a folytonos kommunikációs kapcsolat!
- Háromtengelyes gyorsulásmérő szenzor bármely tengelyét lehessen pozícionálni a gravitációs térhez képest!
- A mért szenzor pozíciója egy referenciaszenzorral legyen visszamérve!
- A mért szenzort a kiválasztott pozícióban mechanikus zaj nélkül tartsa!



- Az eszköz mindkét tengelyen legalább  $0.5^\circ$  beállási pontossággal rendelkezzen!
- Külső számítógép nélkül vezérelhető legyen az eszköz!
- Az mérőeszköz számítógépről is vezérelhető legyen!
- Feleljen meg alapvető biztonságtechnikai megfontolásoknak!

### 1.3 Feladatok áttekintése

Minden feladat első lépése annak alfeladatokra bontása. A diplomamunka első feléve során a lehetséges mechanikai megvalósításokat és a biztonságtechnikai elvárásokat tártam fel, tehát a legelső alfeladata az irodalomkutatás. Ezután körvonalazódott, hogy milyen modulok szükségesek az eszköz megvalósításához. A modulok első megvalósítása és funkcionális tesztje deszkamodell formájában valósult meg. Miután a modulok külön-külön működtek, beillesztettem ezeket az eszköz funkcionális törzsét adó állapotgépbe, itt a modulok együttműködését is ellenőrizhettem. A következő lépés a deszkamodell alapján megtervezni a kapcsolásokat és ezek alapján megalkotni a nyomtatott áramkörök tervét. Az eszköz mechanikájának konkrét megvalósítása nem képezte a diplomamunka tárgyát, de ez a közeljövőben megtörténik.

### 1.4 Projekt arculata

Annak érdekében, hogy az eszközről könnyen és egyértelműen lehessen beszélni el kellett neveznem. Célom volt olyan nevet vagy betűszót találni, ami egyértelmű viszont könnyedebb hangulatú. Ezeknek megfelelően a projekt neve Kinematic Test Environment lett, aminek a rövidítése K.I.T.T.E.N., ami angolul „kiscica”. Ennek megfelelően macskás grafikákat használtam az áramkörökön. Az ilyen és ehhez hasonló elnevezéseknek és praktikáknak köszönhetően emberközelibb munkaközeg hozható létre.

## 1.5 Szakirodalmi áttekintés, elméleti összefoglaló és tervezési irányelvek

### 1.5.1 MEMS szenzorok



1-1. ábra BMA508 [1]

A mikroelektromechanikai (azaz MEMS) szenzorok valamilyen mikroelektromechanika segítségével alakítják mérendő elektromos jellé. MEMS gyorsulásérzékelők esetén ez gyakran egy fésűs mechanizmus, melynek kapacitásváltozása arányos a gyorsulással. A MEMS szenzorok a mikromechanikát és a jelfeldolgozó áramkört is tartalmazzák egy tokozáson belül. A mért jelet továbbítás előtt feldolgozza (erősíti, digitalizálja, szűri, esetleg kompenzálja stb.). Az érzékelt fizikai mennyiség többek közt lehet akár nyomás, gyorsulás, hőmérséklet, akár egy bizonyos gáz jelenléte. Elterjedésük okai közt áll kis méretük (mint a képen látható Bosch BMA 580 gyorsulásmérő), alacsony energiafogyasztásuk, magas pontosságuk és olcsó gyárthatóságuk. A MEMS gyorsulásérzékelők népszerűsége egyes területeken vetekszik az egykor domináns piezoelektromos gyorsulásérzékelőkkel, valamint a MEMS szenzorok képesek statikus gyorsulás detektálására is [2]. Felhasználási területük közt szerepelnek az okos eszközök és a gépjárművek is. Megbízhatóságukat alátámasztja, hogy ez utóbbiban aktív és passzív biztonsági rendszerek részeként is felhasználásra kerülnek, például légzsákrendszerek szenzorjaiként.

A MEMS gyorsulásérzékelő szenzorok felépítésükből és működési elvükből adódóan belső mechanikai feszültség következtében (mely akár hőmérsékletváltozás, illetve beforrasztáskor is kialakulhat) DC ofszettel terhelt gyorsulás jelet adnak. Ezt nevezik 0g ofszetnek, melynek kikompenzálása a magasabb minőségű szenzorok esetén már a szenzor belső jelfeldolgozásánál megtörténik, így a kiküldött digitális jel ezt már nem tartalmazza. Ez a folyamat az offset cancellation (ofszet törlés), mely nemcsak a szenzor bekapcsolásakor, de az esetleges hőmérsékletváltozások miatt a szenzor működése közben is üzemel.

A fejezethez felhasznált források: [3] [4] [5].

### **1.5.2 Biztonságtechnikai lépések**

A mérőberendezés használatánál felmerülhet a személyi sérülés kockázata, melyet megszüntetni vagy csökkenteni kell. A személyi sérülés forrása a forgó elemek melyek lehetséges következménye az ütés, nyírás [6].

Ezeknek a kockázatoknak a legegyszerűbb minimalizálása egy védőburkolat beépítése. A védőburkolatot tervezésekor a megfontolandó szempontokat a következőkben részletezem. A burkolat feladata a veszélyforrás emberrel való érintkezését megakadályozni. Általános szabályai, hogy legyen szilárd, ne idézzon elő további veszélyeket, ne legyen egyszerűen megkerülhető vagy kiiktatható, tartson a veszélyes tértől kellő távolságot, a munkafolyamat megfigyelését csak a lehető legszükségesebb módon korlátozza [7]. Továbbá fontos elvárás, hogy a védőburkolat ne akadályozza a munkát a szükségesnél nagyobb mértékben, ezáltal ne ösztönözze a felhasználót annak kiiktatására. Emiatt fontos, hogy csak annyit burkoljon, amely a munkavégzés biztonsága szempontjából feltétlenül szükséges.

A megfigyelhetőséget figyelembe véve egy víztiszta polikarbonátból kialakított védőburkolat kerül megtervezésre. A munkavégzés minimális gátlása érdekében védőburkolat csak a forgó részeket takarhatja, nem akadályozhatja a kezelőfelületek és csatlakozók hozzáférhetőségét, valamint szerszám nélkül legyen nyitható.

A felhasználó számára sérülésveszélyes gépi funkciók csak lezárt védőburkolat esetén működhetnek. Amennyiben működés közben kinyílik a védőburkolat, a veszélyes folyamatnak le kell állnia, a burkolat visszacsukása esetén nem folytatódhatnak automatikusan, azaz egy reteszelő berendezéssel lesz ellátva a védőburkolat.

További védőberendezésnek kell biztosítania az eszköz felborulása esetén a motorok leállítását. Ennek kialakításakor a működési elv is kettőzve lesz, hiszen a felborulás érzékelésére alkalmas lesz a mérőasztalon található referencia gyorsulásérzékelő is. Ezen felül beépítésre kell kerülnie egy vészleállító gombnak is.

A biztonsági berendezések vizsgálhatóságát is figyelembe kell venni a tervezéskor. A berendezés által kiadott jelet, valamint a megszakítók állapotát is vizuális visszajelzéssel kell ellátni, például egy piros és egy zöld LED-del.

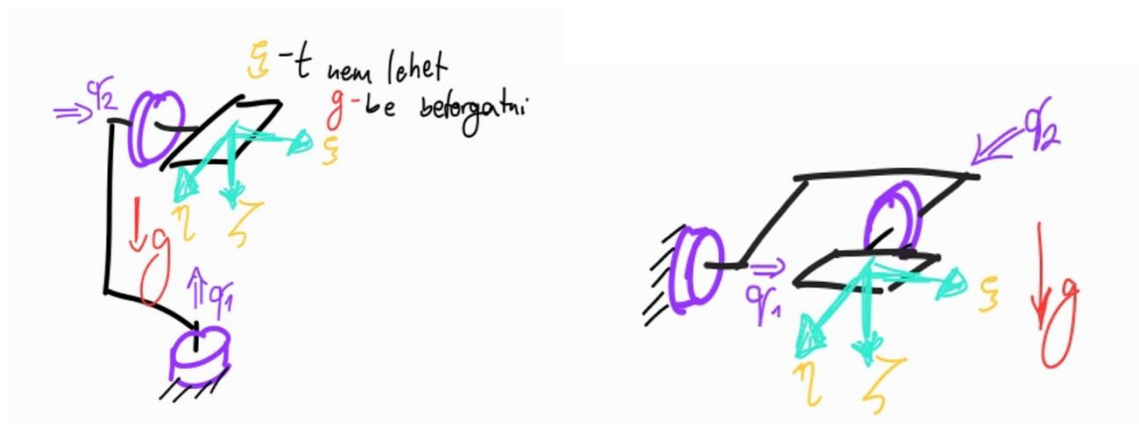
### 1.5.3 Kéttengelyű forgatómechanizmusok

#### 1.5.3.1 Tengelyek elhelyezése

A kéttengelyű forgatómechanizmus alapvetően két egymásra merőleges forgatási tengelyből áll. Amennyiben a két tengely egymással párhuzamos lenne a gimbal lock jelenség lépne fel, ezért a forgatás csak a közös tengely mentén lehetne lehetséges.

A mechanizmusnak a háromtengelyű gyorsulásmérő szenzor bármely tengelyét a gravitációs térrel párhuzamos pozícióba kell tudja forgatni. Ezt minél jobb ismételtelhetőséggel, mechanikai stabilitással kell tudnia végrehajtani, a motorok tengelyének terhelését csapágyazás felhasználásával minimalizálni kell. A mechanizmusnak lehetőleg egyszerűnek kell lennie, a meghibásodási lehetőségeket lehetőleg minimalizálni kell.

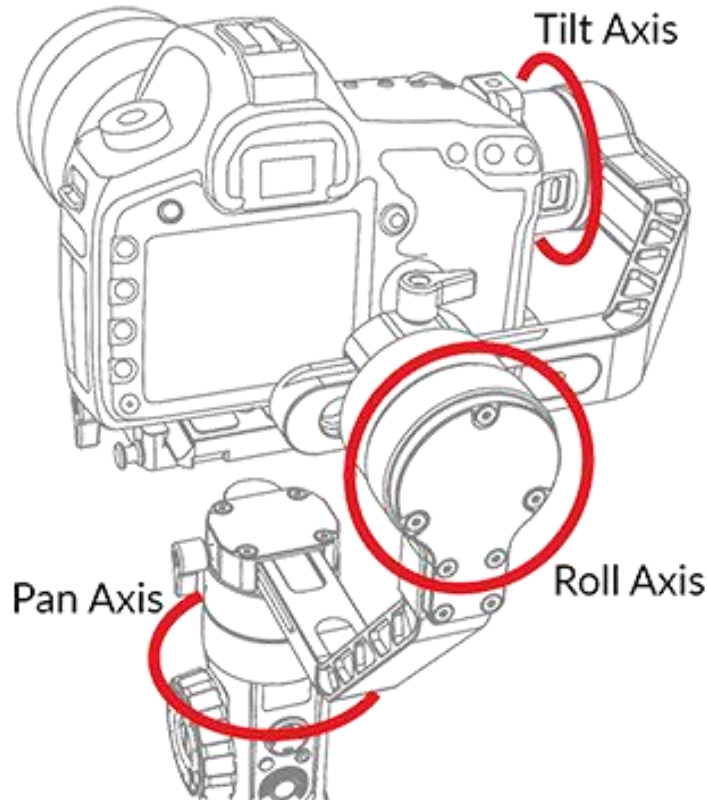
Az első elvárás teljesítésének szükséges feltétele, hogy nem lehet olyan tengely, ami a gravitációs térrel állandóan párhuzamos, onnan ki nem fordítható. Tehát a legegyszerűbb mechanikai megvalósításhoz célszerű a rögzített tengelynek ( $q_1$ ) a gravitációs térre merőlegesnek kell lennie, ahogy ez az 1-2. ábra is mutatja.



1-2. ábra Nem megfelelő tengely elrendezés (balra) és megfelelő tengely elrendezés (jobbra)

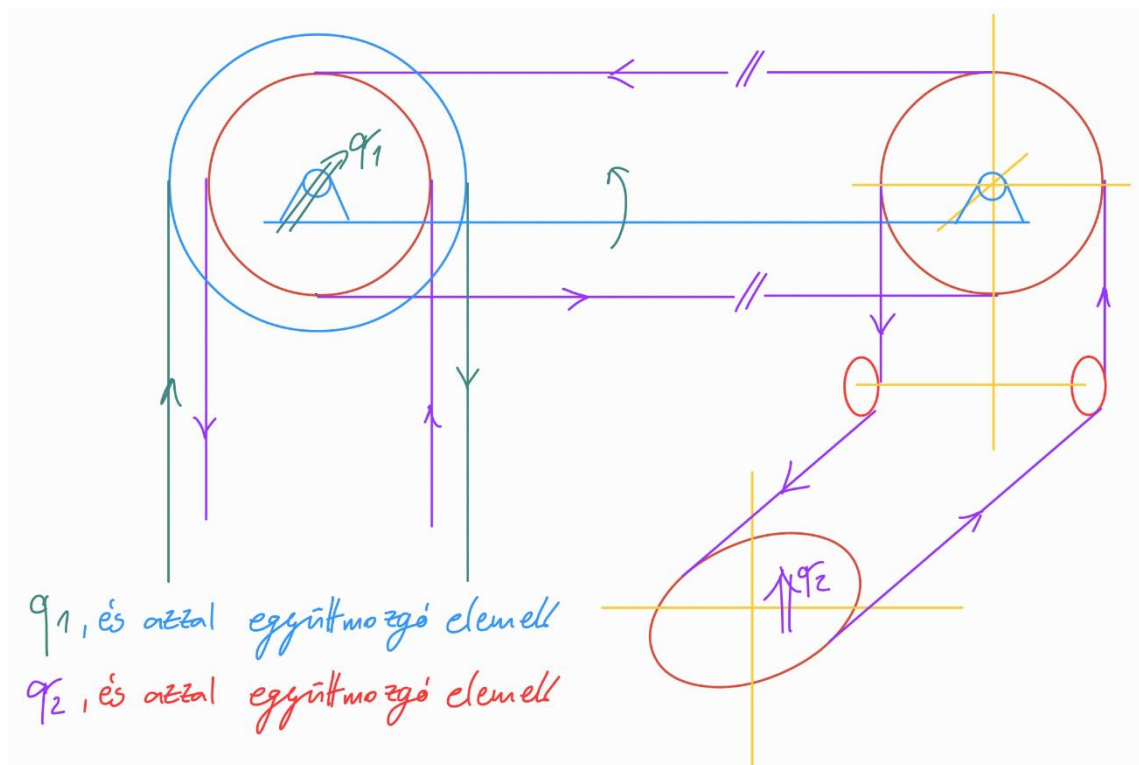
### 1.5.3.2 Tengelyek meghajtása

A tengelyek meghajtásának alapvetően két lehetséges megközelítése van. Az első megközelítésben a tengelyeket közvetlenül hajtják a motorok. A második megoldásnál a meghajtást valamilyen nyomaték továbbító rendszeren segítségével vezetjük a tengelyekhez.



1-3. ábra Első megoldás 3 tengelyes kamerastabilizátor esetén [8]

Az első megoldás a mechanikailag legegyszerűbb. Ebben az esetben egy, a gravitációs tengellyel merőleges tengelyű, rögzített motor forgat egy második motort. A mozgatott motor, a rögzített mozgató motor tengelyével merőleges tengelyű, hasonlóan az 1-3. ábra által mutatott kamerastabilizátorhoz. Ennél a megoldásnál az egyik fő kihívás, hogy a mérőasztalhoz menő jelvezetékeken kívül a forgatott motor tápvezetékeit és az enkóder jelvezetékeit is továbbítani kell. További kihívás, hogy a forgató motornak így nem csak az eszközt kell forgatnia, hanem a forgatott motor tömegét is. Végezetül számolni kell azzal, hogy ennél az elrendezésnél a motorok tengelye nem csak forgató, hanem hajlító nyomatéknak is ki van téve, de ez megfelelő csapágyazással kiküszöbölhető.

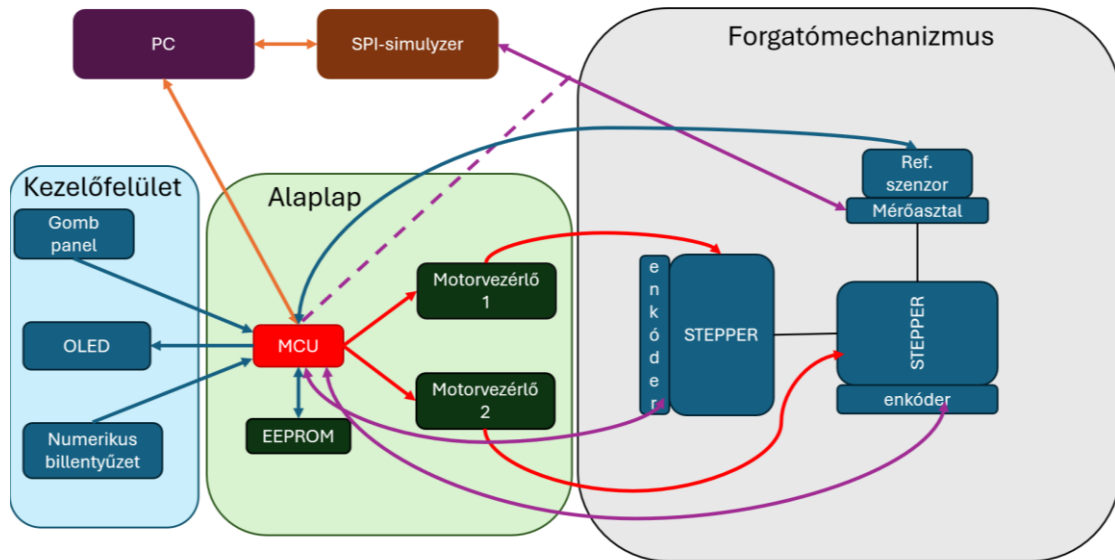


1-4. ábra Kézi vázlat a második megoldásra

A második kivitelezés esetén a motorok rögzítettek, így nem szükséges a második motornak sem mozgatója, sem annak jeleinek és tápvezetékeinek átvitele. Ebben az esetben egy megoldási lehetőség bonyolult hajtósíjjas mechanizmus használata (1-4. ábra). Vagy fogaskerekek használatával egy fordítottan hajtott differenciálműhöz hasonló megoldás, viszont ebben az esetben a legnagyobb kihívás a rendszer kotyogásmentesítése.

A tervezés során az egyszerűbb mechanika érdekében az első megoldással megyek tovább.

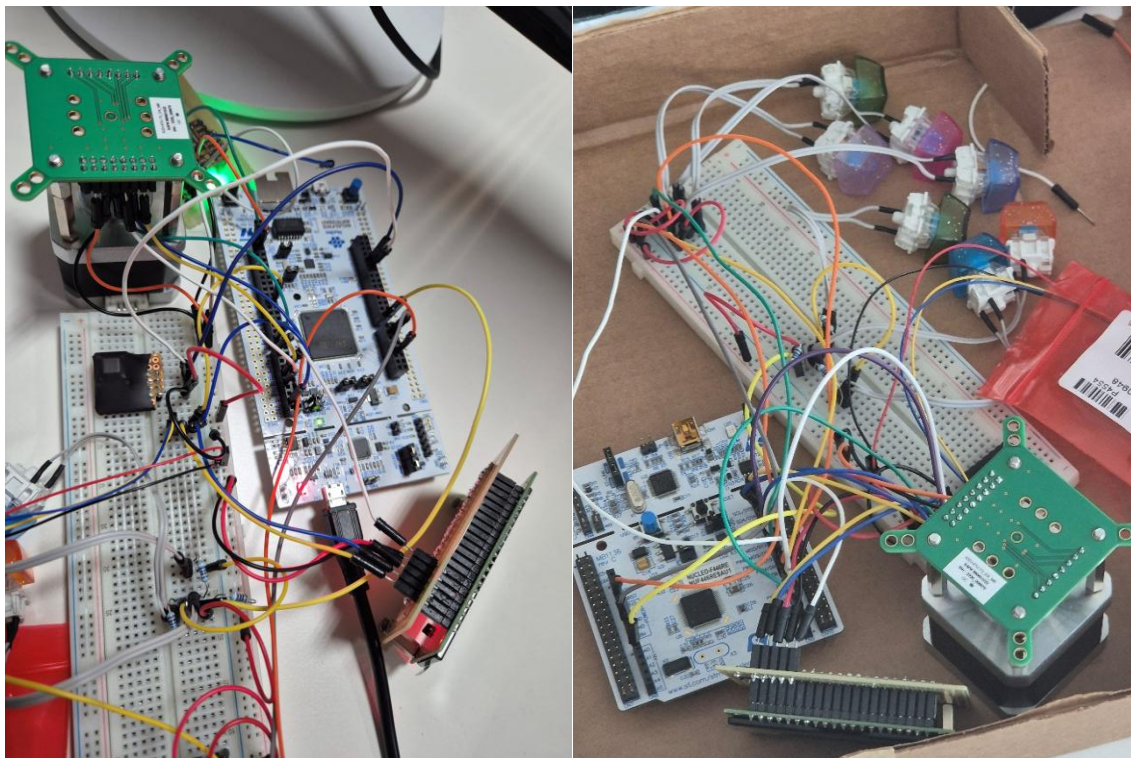
## 2 Modulok



2-1. ábra Az eszköz blokkdiagramja

Az eszköz blokkdiagramját a 2-1. ábra mutatja. Narancssárgával a soros kommunikációt, késsel az I2C-t, lilával az SPI-t, pirossal pedig a motorvezérlőknek küldött lépés impulzusokat és az azok által kiküldött modulált tápfeszültséget jelöltem. Az eszköz központi eleme a mikrovezérlő (MCU). Ez köti össze a rendszer elemeit és írja le a rendszer viselkedését. A felhasználó a kezelőfelület segítségével vagy távoli vezérlés esetén számítógépről küldött parancsokkal tud kapcsolatba lépni az eszközzel. A felhasználó által küldött beállításokat az EEPROM-ba mentve a használat kényelmesebb. A mikrovezérlő a kapott beállítások alapján parancsra a motorvezérlő IC-ken keresztül lépteti a motorokat abba a szöghelyzetbe, ami megfelel a beállításoknak. Ehhez természetesen szükséges a motorok tengelypozíciójának ismerete. Ezt az információt az enkóderek szolgáltatják a mikrovezérlőnek. A motorok által pozicionált mérőasztalon a mért szenzor mellett egy referenciaszenzor is megtalálható, ami az asztal orientációjáról küld információt az MCU-nak. A mérőasztallal, csak a mért szenzorral kommunikáló SPI-simulyzer van kapcsolatban. Ezt az SPI-csatornát a mikrovezérlő lehallgatja, így a pozicionálás akár adott SPI üzenetek észlelésére is ütemezhető.





**2-2. ábra Deszkamodellek különböző fejlesztési fázisokban**

A megvalósításhoz szükséges egységek fejlesztését úgynevezett deszkamoddellen végeztem, ami a gyors prototípusgyártás egyik elterjedt módja (2-2. ábra). Ebben a fázisban az elektronikai alkatrészek próbapanelen vannak, nincs nyomtatott áramkör, próbanyákon vannak a csatlakozók kialakítva. Így a kiválasztott eszközök (pl. OLED modul) megfelelőségét kis előkészülettel tudtam ellenőrizni és megkezdhettem a modulokhoz szükséges kezelő szoftver megvalósítását. Az STM32CubeIDE fejlesztői környezetet használtam a beágyazott szoftver fejlesztésére. Ez a szoftver a gyártócég által szolgáltatott környezet, a választott STM mikrokontrollerre érdemes ebben a fejlesztői környezetben szoftvert fejleszteni a hasznos beépített eszközök miatt.

Miután megbizonyosodtam a választott alkatrészek megfelelőségéről és összeállt a deszkamodell, megterveztem a modulokhoz szükséges nyomtatott áramköröket. Az áramkörök tervezéséhez a KiCAD szoftvert használtam, ami egy ingyenes, jól dokumentált és elterjedt szoftver.

## **2.1 Mikrovezérlő**

Az eszköz központi egysége, amin az ezt működtető szoftver fut, az maga a mikrovezérlő. A projektjeimhez (szakdolgozatom, önálló projekt stb.) általában STM32

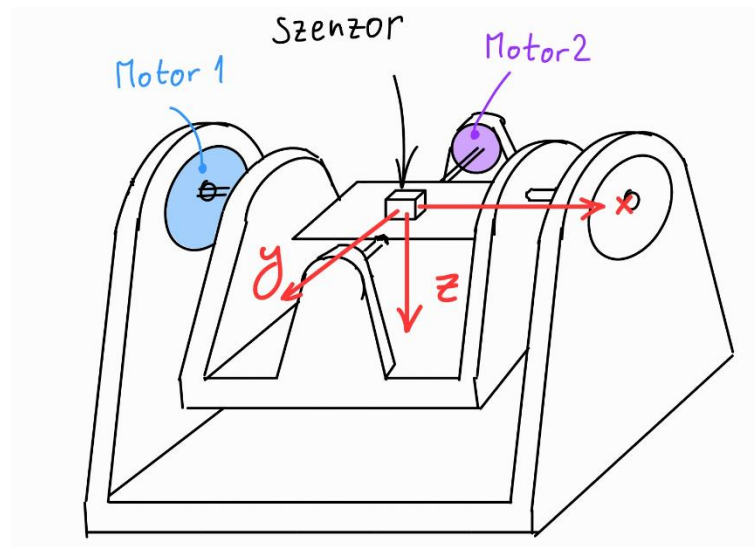


mikrovezérlőt használok, valamint a cégnél használt belső fejlesztésű eszköz is STM32 alapú. Ezekhez a mikrovezérlőkhöz a korábban említett CubeIDE fejlesztői környezet tartozik. Ebben a környezetben a fejlesztést segítik a perifériák úgynevezett hardveres absztrakciós szint, azaz HAL kezelő osztályainak megléte. Így a modulok drivereinek megírásakor a figyelmemet tudtam az adott modul kezelésére fordítani és nem a kommunikációs protokollal kellett töltenem a fejlesztési időt. További előnye ennek a platformnak, hogy a mikrovezérlők megvásárolhatóak „NUCLEO” fejlesztői panelre felszerelve, ahol a működéshez szükséges minden áramköri elem és kapcsolat ki van alakítva, valamint programozó és debugger (hibakereső) modullal vannak felszerelve.

A következő feladat az STM32 mikrokontroller családból kiválasztani a tervezési szempontoknak legjobban megfelelőt. A szempontok minél több GPIO (későbbi bővíthetőség miatt), legalább 3 SPI-csatorna (2 csatorna a lehallgatáshoz és egy az SPI perifériák kezelésére), legalább 4 DMA (2 a motorok vezérléséhez, 2 az SPI lehallgatáshoz), legalább 4 időzítő (2 a motorok vezérléséhez, 2 az AD konvertereknek), Ethernet és USB kommunikáció. Ezen megfontolások alapján a választásom az STM32F767ZIT6-re esett. Ennek főbb tulajdonságai [9]:

- Arm® 32-bit Cortex®-M7 CPU
- Akár 2Mbyte flash memória
- Akár 512 Kbyte ram: keresőtáblák és kijelző puffer miatt szükséges
- Általános felhasználású 16 csatornás DMA: mivel marad nem felhasznált, DMA, a későbbiekben akár a képernyő frissítés és egyéb (akár újabb) perifériák kezelése is megoldható DMA-val.
- Akár 18 időzítő
- Akár 168 IO port
- Akár 4 I2C
- Akár 4 USART
- Akár 6 SPI
- USB2.0
- Ethernet

## 2.2 Motorok



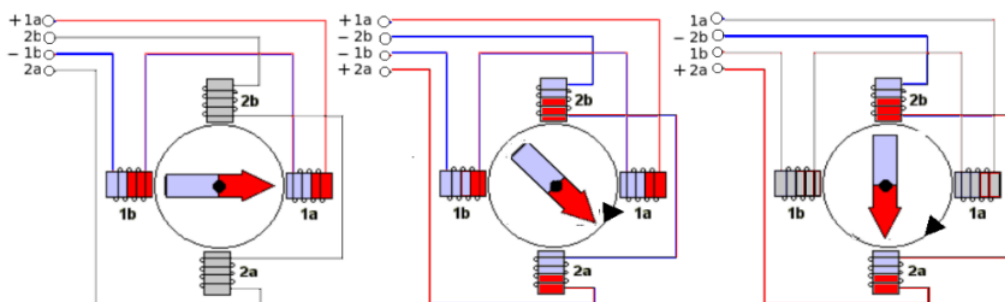
2-3. ábra A motorok mechanikában való elhelyezésének kézi vázlata

Ahhoz, hogy a későbbiekben tárgyaltak érthetőbbek és elképzelhetőek legyenek a 2-3. ábra mutatja a motorok elrendezését a mechanizmus vázlatában. A színnel jelölt elemek szimbolizálják a léptetőmotorokat.

Abból, hogy a mérőeszköz gyorsulásmérő szenzorokat pozicionál következik az elvárás, hogy a mérőeszköz miután felvette a kívánt pozíciót, azt tartsa. Tehát nulla szögsebességnél is legyen nyomatéka a motoroknak, valamint oszcilláció lehetősége ne állhason fenn, hiszen az mechanikai zajt vinne a rendszerbe, amit a mért szenzor mérne, ezzel torzítaná a mérési eredményeket. Ezen elvárások alapján léptetőmotorokat választottam aktuátoroknak. A léptetőmotor tekercseinek gerjesztésével a forgórész léptethető, vagy az adott pozícióban tartható. A tekercsek gerjesztését TMC2209 IC-t használó SilentStepStick meghajtóval végeztem. A meghajtónak forgási irányt, mikrolépés elbontást és lépés impulzust adva lépteti a motort lépés impulzusonként. Mivel a lépés impulzusok hatására a motor adott szöggel fordul el, a motorok nyílt hatásláncú pozíció irányítását választottam (ez elterjedt megoldás léptetőmotorok esetén). A mozgás végén felvett pozíciót mágneses abszolút enkóderekkel mérem vissza a mozgítás pontosságának ellenőrzése miatt ezen kívül a mérés kiértékelésénél hasznosak lehetnek a mért szögállások. Tehát a szöghelyzet mérése nem hat vissza a szabályzásra, a rendszer a visszamérés ellenére nyílt hatásláncú.

## 2.2.1 Motorvezérlő IC

A léptetőmotorok vezérléséhez, mint azt feljebb is említettem, a TMC2209 IC-t használtam, mely a szükséges passzív áramköri elemekkel ellátva is kapható, beépítve a SilentStepStick elnevezésű panelbe. A TMC2209 egy elterjedt léptetőmotor meghajtó, amit gyakran alkalmaznak például 3D nyomtatók motorjainak vezérlésére is. Választásom azért erre esett, mivel elterjedtsége miatt könnyen beszerezhető, sok információ található róla, valamint csendesebb és energiahatékonyabb léptetést ígér. A meghajtó IC képes UART-csatornán kommunikálni, akár ezen keresztül is megadható a léptetés sebessége, de ezt a képességét nem használom ki, mivel ez nem képes gyorsító és lassító profilokra. Emiatt a későbbiekben tárgyalt sebességprofilvezérlést alkalmaztam. Az IC-nek további nagyon hasznos funkciója, hogy képes nyomaték ugrást detektálni, ezáltal végálláskapcsoló nélkül is lehet alaphelyzetet találni a motorokkal, valamint képes kihagyott lépések detektálására. Az optimális motorvezérlés érdekében a meghajtó kalibrálható. Ezeket a funkcióit egyelőre még nem használtam ki, de az alaplap tervezésekor úgy hoztam létre a kapcsolást, hogy az UART-csatorna használható legyen.



2-4. ábra Kétfázisú bipoláris léptetőmotor elvi ábrája [10]

Ez a motorvezérlő kétfázisú bipoláris léptetőmotorok (azaz fázisonként egy tekercs) vezérlésére alkalmas 2.8A maximális árammal. A pontos motorválasztásra akkor kerülhet sor, amikor a mechanikai terv már elkészült és ismert a szükséges nyomaték. Mivel a motor mérete nem befolyásolja az alapvető működési és vezérlési elvet, ezért a motor pontos típusának ismeretének hiánya nem jelent akadályt a fejlesztés során.

A motorvezérlő mikrolépés felbontást konfiguráló lábain a kiválasztható felbontások az 1/8, 1/16, 1/32 és 1/64. Ez 1.8° felbontású léptetőmotorok már 1/8 mikrolépés beállítás esetén is esetén  $\alpha_{min} = 0,225^\circ$ -os felbontást jelent, tehát a specifikációknál jelzett beállási pontosságot tudja szolgáltatni.

A fejezethez felhasznált forrás: [11].

### 2.2.2 Léptetőmotorok sebességprofil vezérlése

Feltételezve, hogy nem történik lépéstévesztés, a léptetőmotorok által a mozgás végén felvett szöghelyzetet a léptető impulzusok száma határozza meg (2.1) egyenlet, míg a mozgás közbeni szögsebességet az impulzusok közötti idő határozza meg (2.2) egyenlet.

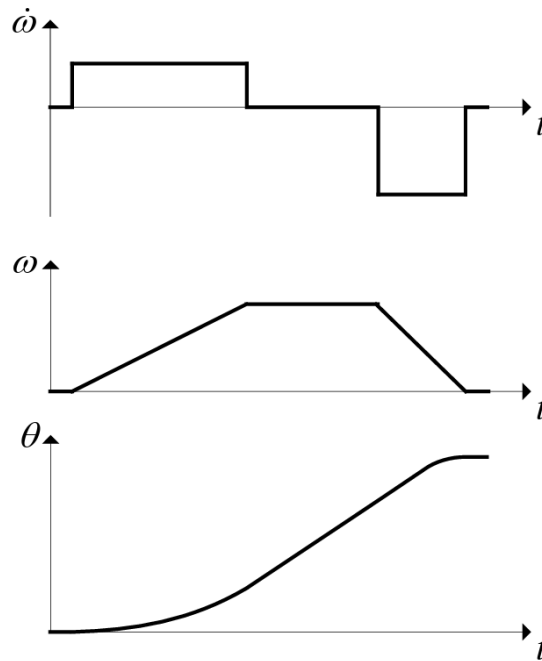
$$\Theta = \alpha \cdot n \quad (2.1)$$

$$\omega = \frac{\alpha}{\tau} \quad (2.2)$$

Az egyenletekben a  $\alpha$  az egy lépésimpulzus hatására tett elfordulást,  $n$  az impulzusok számát,  $\Theta$  a mozgás során tett teljes elfordulást,  $\tau$  az impulzusok között eltelt időt,  $\omega$  pedig a szögsebességet jelöli.

Belátható viszont, hogy nem vezérelhető a motor oly módon, hogy álló helyzetében azonnal egy kívánt szögsebességhez tartozó  $\tau$  időintervallumonként lépés impulzust adunk a vezérlőnek. A sebességprofil, amit így előírnánk a motornak nem lenne teljesíthető, hiszen az ugrásszerű szögsebességváltozáshoz elméletileg végtelenül nagy szöggyorsulás, ezáltal nyomaték tartozna, amit természetesen nem tud a motor szolgáltatni. Ez nem csak magas áramfelvételhez, hanem esetleges lépésvesztéshez is vezethet, valamint az éles gyorsulásból adódóan a motor „rángatna”. Ezen nem kívánatos hatások elkerülése érdekében a léptetőmotorokat megfelelő sebességprofillal kell vezérelni.

Gyakran használt megoldás a sebességprofilt lineáris szakaszokból összeilleszteni, ahogy azt a 2-5. ábra 1-1. ábra is mutatja.



2-5. ábra Szöggyorsulás ( $\dot{\omega}$ ), szögsebesség ( $\omega$ ), szögpozíció ( $\theta$ ) [12]

Meg kell jegyezni, hogy ez a sebességprofil sem deriválható folytonosan, ugrások vannak a gyorsulásban, így a gyorsító nyomatékban is ugrások vannak. Ennek kiküszöbölése érdekében tetszőleges számban deriválható sebességprofil választottam. Kézenfekvő megoldás volt a trigonometrikus függvényekhez nyúlni.

Másik problémám a [12] forrásban bemutatott megoldással, hogy a következő lépésig eltelt időt a motor forgása közben lépésről-lépésre számították. Annak érdekében, hogy a motorok forgatása akár párhuzamosan is történhessen, célom volt, hogy mozgás során ne terheljék a processzort. Ezt a problémát úgy oldottam meg, hogy az időzítést előre számolom, mozgás során pedig DMA segítségével másolom ezeket az értékeket az időzítő ARR regiszterébe.

### 2.2.3 Trigonometrikus sebességprofil

A választott pozícióbeállási görbét az (2.3) függvény adja meg, ahol  $\theta$  a végpozíció,  $T$  pedig a mozgás ideje. A kívánt mozgási profilokat az 2-6. ábra mutatja.

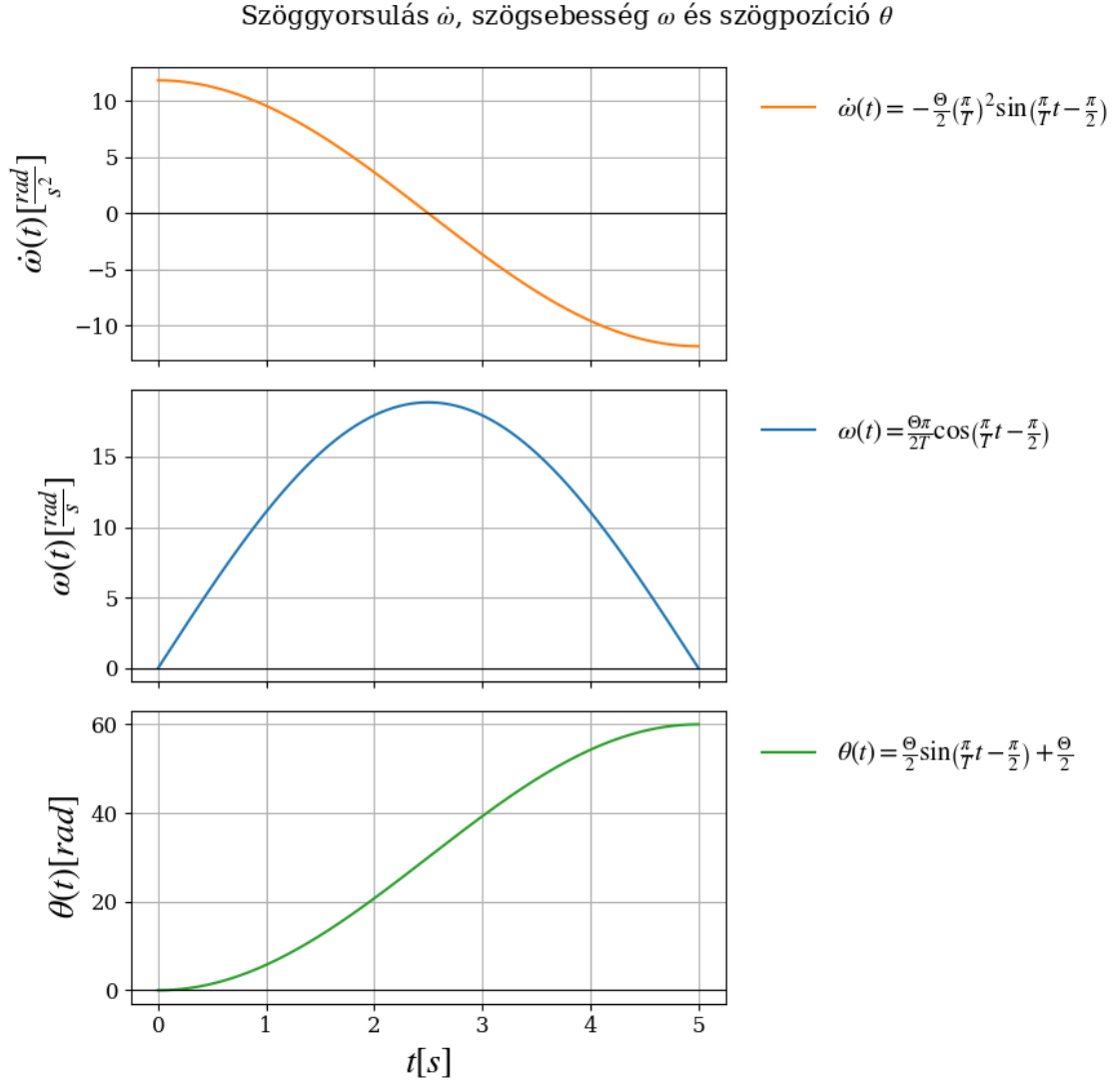
$$\theta(t) = \frac{\theta}{2} \cdot \sin\left(\frac{\pi}{T} \cdot t - \frac{\pi}{2}\right) + \frac{\theta}{2} \quad (2.3)$$

Ennek deriváltja pedig az (2.4) függvény:

$$\omega(t) = \frac{\pi \cdot \Theta}{2T} \cdot \cos\left(\frac{\pi}{T} \cdot t - \frac{\pi}{2}\right) \quad (2.4)$$

Aminek deriváltja a szöggyorsulást leíró függvény:

$$\dot{\omega}(t) = -\frac{\Theta}{2} \cdot \left(\frac{\pi}{T}\right)^2 \cdot \sin\left(\frac{\pi}{T} \cdot t - \frac{\pi}{2}\right) \quad (2.5)$$



2-6. ábra Trigonometrikus mozgásprofil,  $T = 5$ ,  $\Theta = 60^\circ$

Mivel a motor forgási sebessége a lépésimpulzusok között eltelt időtől függ, a cél a lépések közötti  $t_k$  idők meghatározása.

Első megközelítésben a sebességprofilot számoltam először. Ezt kiértékeltem  $N \left(= \frac{\Theta}{\alpha}\right)$  db helyen, majd az így kapott  $\omega_i$  sebességekhez tartozó  $t_i$  időzítéseket a  $t_i = \frac{\omega_i}{\alpha}$

összefüggéssel számoltam. Ennek a megoldásnak a problémája, hogy  $\omega(t = 0)$  helyen 0 értéket vesz fel, ott kiértékelve  $\omega_i = 0$ -t kapunk, amiből  $t_i = \infty$  adódik és a motor sosem fog elindulni. Erre megoldást adhat az  $\omega(t)$  függvény  $\omega_{min}$  minimális szögsebességgel való eltolása, így viszont a  $\theta(t)$  függvényt is korrigálni kell az ebből adódó hibával.

A megoldás a  $\theta(t)$  függvény inverzének, a  $t(\theta)$  függvénynek használata.

$$\theta(t) = \frac{\Theta}{2} \cdot \sin\left(\frac{\pi}{T} \cdot t - \frac{\pi}{2}\right) + \frac{\Theta}{2} \quad (2.6)$$

Kifejezve a  $t(\theta)$  függvényt:

$$t(\theta) = \left( \arcsin\left(\left(\theta - \frac{\Theta}{2}\right) \cdot \frac{2}{\Theta}\right) + \frac{\pi}{2} \right) \cdot \frac{T}{\pi} \quad (2.7)$$

Áttérve diszkrét időre a  $\theta_k = k \cdot \alpha$  helyettesítéssel:

$$t_k = \left( \arcsin\left(\left(k \cdot \alpha - \frac{\Theta}{2}\right) \cdot \frac{2}{\Theta}\right) + \frac{\pi}{2} \right) \cdot \frac{T}{\pi}, \quad (2.8)$$

$$\left\{ k \mid 1 \leq k \leq \frac{\Theta}{\alpha}, k \in \mathbb{Z} \right\}$$

A függvény argumentuma egyszerűsíthető:

$$t_k = \left( -\arcsin\left(1 - 2 \cdot \frac{k \cdot \alpha}{\Theta}\right) + \frac{\pi}{2} \right) \cdot \frac{T}{\pi}, \quad (2.9)$$

Az így kapott összefüggéssel egyből a  $k$ . lépésben alkalmazandó  $t_k$  időzítést kapom, amivel a kívánt sebességprofil érhető el.

Amennyiben nem a megtételhez szükséges idő, hanem a maximális elért szögsebességet szeretnénk paraméterként felhasználni, először meg kell találni az összefüggést  $T$  és  $\omega_{max}$  között. A szögsebesség maximuma könnyen belátható, hogy  $t = \frac{T}{2}$  pontban található. Az (2.4) függvénybe behelyettesítve ebben a pontban:

$$\omega\left(t = \frac{T}{2}\right) = \frac{\pi \cdot \Theta}{2T} \cdot \cos\left(\frac{\pi}{T} \cdot \frac{T}{2} - \frac{\pi}{2}\right) \quad (2.10)$$

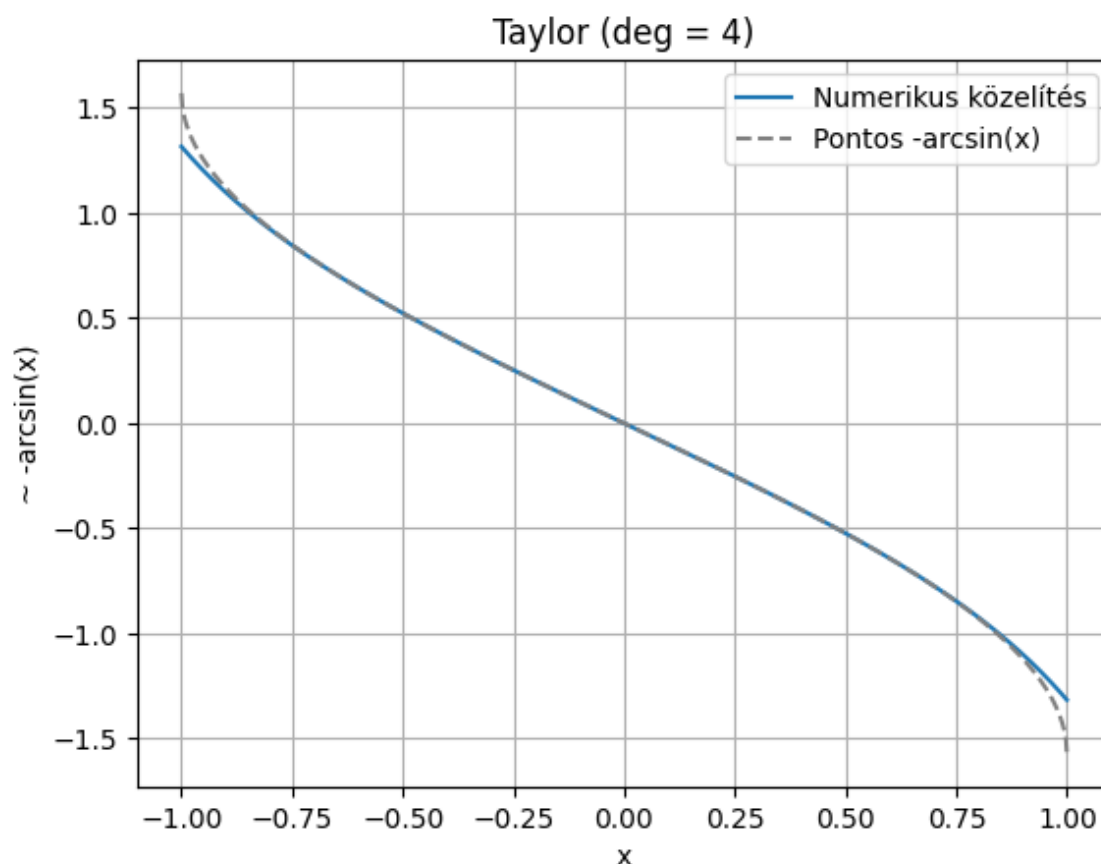
majd kiértékelve, az (2.11) összefüggést kapjuk. Ebből akár visszaszámolható a szükséges  $T$  a kívánt  $\omega_{max}$  alapján.

$$\omega_{max} = \frac{\pi \cdot \Theta}{2T} \quad (2.11)$$

## 2.2.4 Numerikus módszerek

A következő felmerülő probléma, az arcsin függvény számítása. Erre több megoldás is lehetséges: Taylor sor használata, aszimptotikus közelítés  $\pm 1$ -körül [13] vagy keresőtábla (másnéven LUT) lineáris interpolációval. A numerikus módszereket először Python környezetben valósítottam meg, majd összehasonlítottam az eredményeket és ez alapján döntöttem, hogy melyiket implementáljam az eszközön.

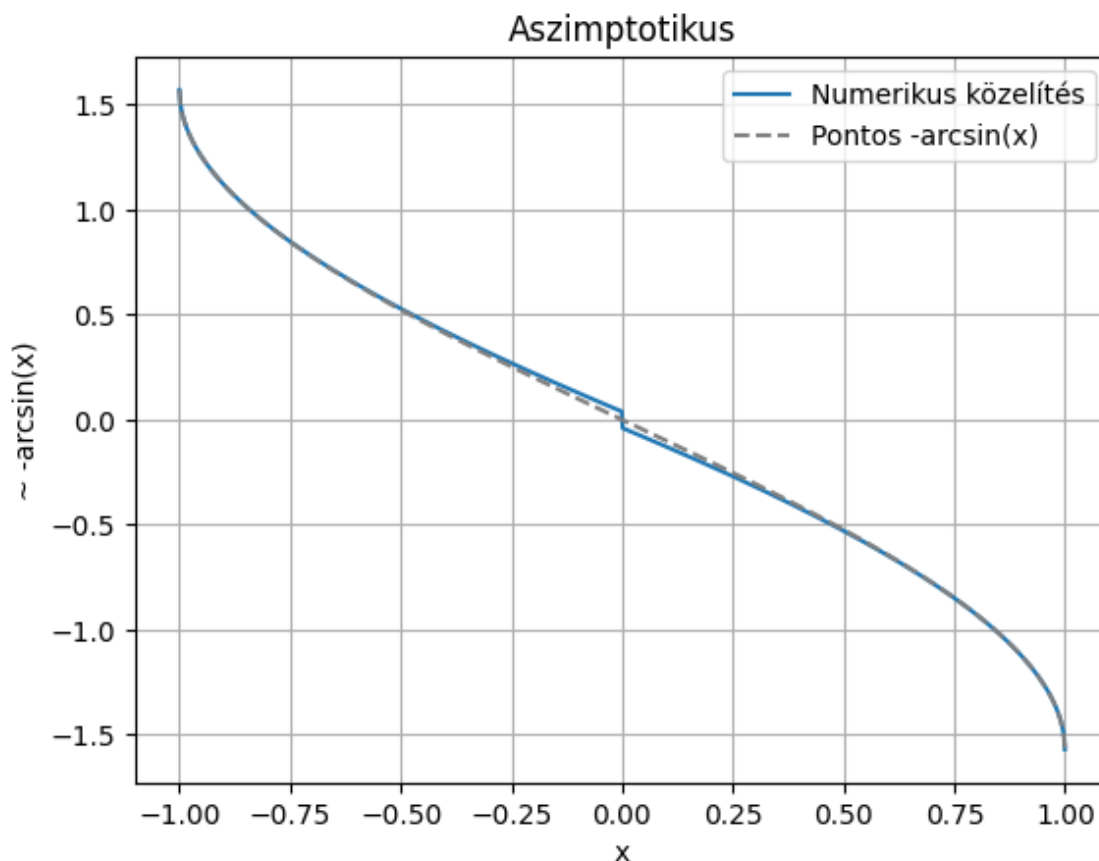
Az összehasonlításnál negyedfokú Taylor sort használtam. Ezzel a  $-\arcsin$  függvény közelítését a 2-7. ábra mutatja. Mint látható, nullától távolodva egyre jobban torzul a függvény közelítése.



2-7. ábra  $-\arcsin$  közelítése negyedfokú Taylor sorral

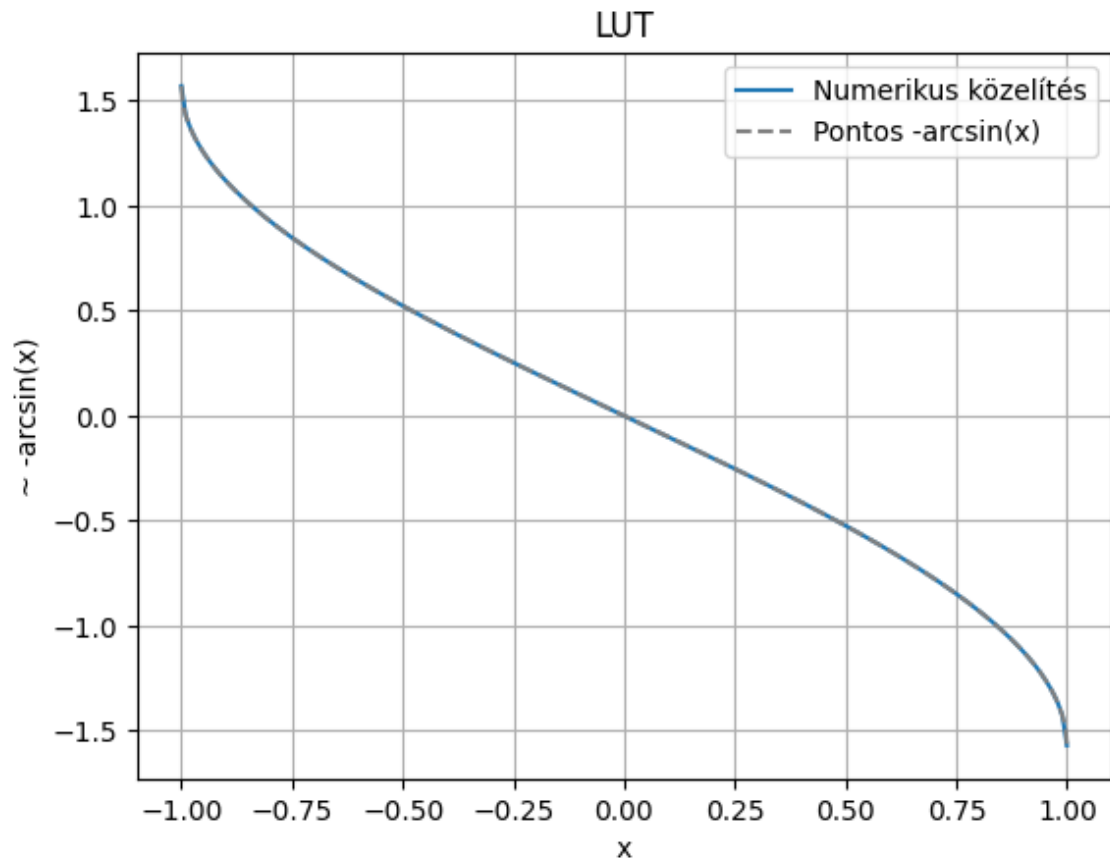


A következő vizsgált megoldás az aszimptotikus közelítés. A Taylor sor 1 és -1 körüli torzítását kívántam ellensúlyozni azzal, hogy az aszimptotikus közelítést -1 (és a szimmetria által ezáltal 1) köré választottam. Az így kapott  $-\arcsin$  közelítést a 2-8. ábra mutatja. Látható, hogy itt törés van nullánál. Magasabb fokú kifejtésnél ez az effektus csökken, de a nem folytonosságból adódó hiba sosem tűnik el. [13]



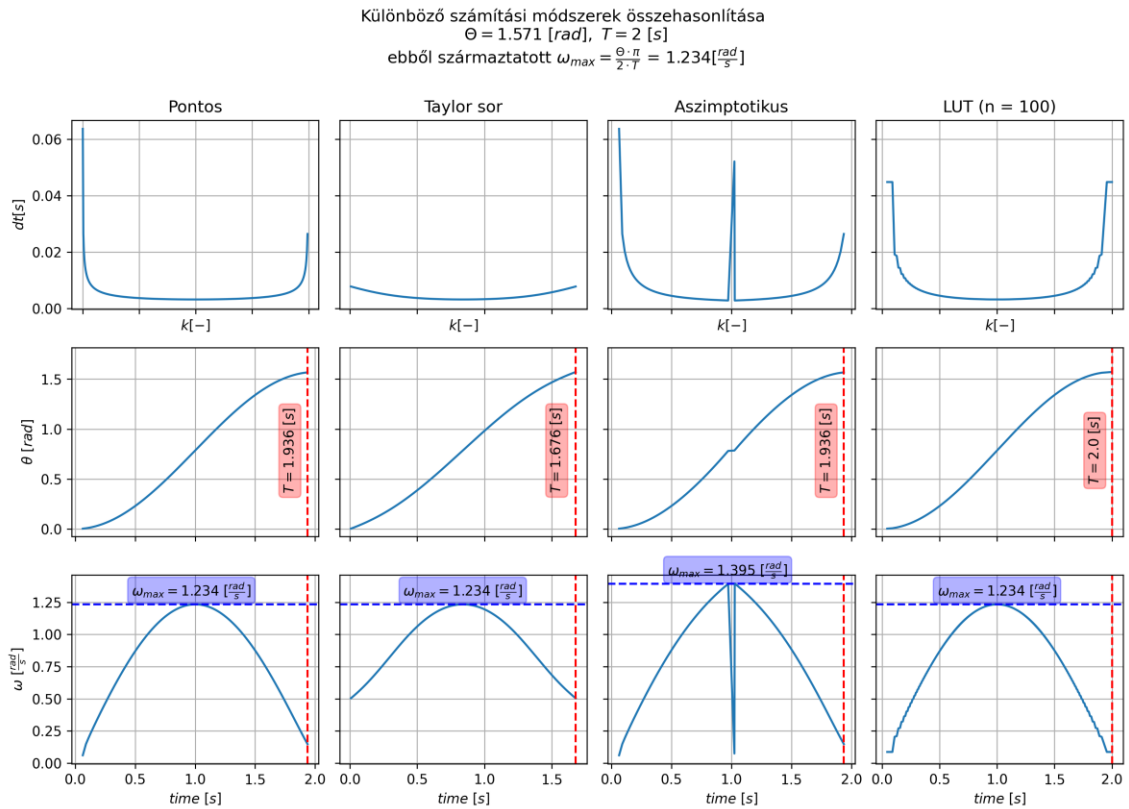
**2-8. ábra Aszimptotikus közelítés**

A harmadik módszer keresőtáblát használni, azaz adott pontokban előre meghatározni a függvény értékét, és ezen pontok között becsülni a függvény értékét. Az összehasonlítás során 100 elemű táblát használtam. A tábla csak a negatív  $x$  értékekhez tartozó  $y$  értékeket tartalmazza, hiszen a függvény szimmetrikus. Így valójában 200 elemes tábla felbontását adja. Az így kapott  $-\arcsin$  közelítést a 2-9. ábra mutatja.



2-9. ábra  $-\arcsin$  közelítése keresőtáblával (a pontos és a közelített fedik egymást)

A módszereknek az adott feladatra való alkalmasságát úgy ellenőriztem, hogy a különböző módszerekkel kiszámítottam egy adott mozgásprofilhoz tartozó  $t_k$  értékeket, illetve ez alapján számítottam ebből adódó szögsebességet és szögelfordulást is. Annak érdekében, hogy legyen összehasonlítási alap, a Numpy csomag beépített arcsin függvényével is kiszámoltam egy „pontos” sorozatot. Az eredményeket a 2-10. ábra foglalja össze.



**2-10. ábra közelítési módszerek**

A megadott  $T$  beállási időt a Taylor soros megoldás közelítette a legpontosabban, illetve a Taylor soros számítás esetén a kapott szögsebesség profil is sokkal nagyobb kezdeti ugrással indul, mint a többi módszer. Az aszimptotikus közelítés esetén a nulla körüli törés által okozott torzításon kívül a sebességprofil formája messze van a kívánt profiltól, ezért a nulla körüli kiugrás numerikus kiszűrésével sem foglalkoztam, hiszen ez a módszer több szempontból is rossz megközelítés. A 100 elemes keresőtáblás megoldás a tesztelt módszerek közül a legjobban követte a pontos számítás eredményeit. Nem tekinthetünk el attól a sem, hogy a keresőtáblás megközelítés jóval kisebb számítási igényt igényel, viszont a keresőtábla keresése időigényes, valamint a többi megközelítéssel szemben a keresőtábla letárolása miatt memória igényesebb ez a módszer. Ezek a hátrányok elfogadhatóak, hiszen a választott mikrovezérlő tárhelye bőven elegendő a 100db, akár 32bit-es értékpárok letárolásához, valamint az aszinkron kivitelezés miatt (először  $t_k$  táblázat számítása majd parancsra a mozgás megkezdése) nem probléma, ha a mozgásprofil számítása időigényesebb.

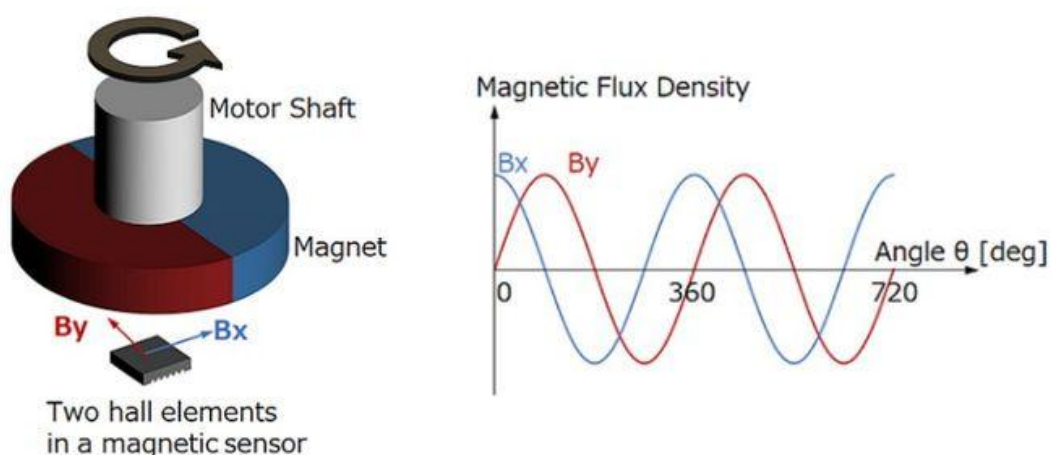
### 2.2.5 Módszer implementálása beágyazott eszközre

Kis számokkal kellene számolni, így könnyen bekövetkezhetne, hogy számolás közben nullává válik egy eredmény a számábrázolás miatt. Azért, hogy ezt elkerüljem, felskálázott keresőtáblát használok és az értékeit 32 biten tárolom. A számítások leírásánál többek között az osztások és szorzások sorrendjére is külön figyelnem kellett, hogy ne történjenek nullázódások kis részeredmények következtében.

A megoldásom validációjához vizuális jelzéssel láttam el a motor tengelyét a mozgás jobb szemléltetése érdekében. Ezután olyan beállási szöveget választottam, amik összege egy teljes körbefordulást eredményez. Mivel a motor többszöri körül fordulásra is a kiinduló szögállásba érkezett, valamint ezt a kívánt idő alatt tette, megfelelőnek találtam a megoldást.

### 2.2.6 Enkóderek

Mivel a motorokat nyílt hatáslánccal irányítom (tehát nincs visszacsatolás) elengedhetetlen, hogy a mozgás végén visszamérjem a motorok pontos pozícióját. A pozíció visszamérése továbbá azért is fontos, mert a következő pozícióba való beálláshoz szükséges a jelenlegi pozíció ismerete. A pozícióméréssel szemben támasztott elvárásom, hogy bekapcsoláskor is abszolút szöghelyzetet szolgáltatson, szöghelbontása legalább annyira legyen finom, mint a legkisebb használt motorlépés, könnyű legyen beszerezni és könnyen beszerezhető legyen.



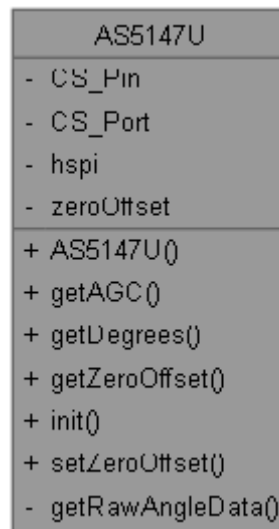
2-11. ábra Mágneses enkóderek működési vázlata [14]

Ezen megfontolások alapján mágneses alapú enkódert választottam. Ezek az eszközök a Hall-effektuson alapulnak és a tengelyre rögzített mágnes terének abszolút

A választott szenzor képes az optikai enkódereknél megszokott impulzus jelek kiadására, de SPI kommunikáción keresztül kiolvasható belőle a mágneses tér pozíciója, így a tengely abszolút szögpozíciója bármikor lekérdezhető a szenzortól. A szenzor továbbá képes diagnosztikai adatok szolgáltatására, így a megfelelő beépítés ellenőrizhető, érzéketlen a külső mágneses terek zavaró hatására.

22

## 2.2.7 Driver



2-13. ábra AS5147U kezelőosztályának UML diagramja

Mivel nem találtam szabadon felhasználható drivert erre az enkóderre, ezért azt nekem kellett megvalósítani az adatlapja alapján.

Az eszközök driverének megírásához a C++ nyelvet választottam. A drivereket objektum orientált megközelítéssel írtam a könnyű használat, modularitás és bővíthetőség érdekében. Külön header fájlba kigyűjtöttem az eszköz regisztereit, majd létrehoztam az eszközt kezelő osztályt. Jelenleg csak a modul működéséhez szükséges, legalapvetőbb funkciókat valósítottam meg.

A kommunikációhoz létrehoztam az üzeneteket küldő- és fogadó metódusokat a HAL könyvtár SPI függvényeinek segítségével. Az eszköz többféle kommunikációs módokat is támogat: 16 bites üzenetek CRC nélkül, 24 bites üzenetek CRC-vel, 32 bites üzenetek CRC-vel és 8-bites PAD-dal több eszköz sorba kötése és további kommunikációs védelemre. A magasabb biztonság és jobb hibakezelés érdekében a 24 bites kommunikációt választottam. Ennek a kezelésére külön metódusokat valósítottam meg, melyek kezelik a CRC számítást és a megfelelő üzenet összeállítását. Ezekhez fejlesztői csomagban adott HAL SPI metódusokat használják fel.

Az inicializáló metódust a konstruktortól külön hoztam létre, így annak logikai visszatérési értéke jelezheti az inicializálás sikerességét. Az inicializálás során az eszköz szűrőjének K értékének minimumát az adatlapban [16] megadottak szerint a kis szögsebességeknél használatosan 0-ra állítom, mivel a pozíció kiolvasás álló helyzetben történik a jelenlegi felhasználás szerint. Ezen túl kiolvasom az eszköz hibaregiszterét is,

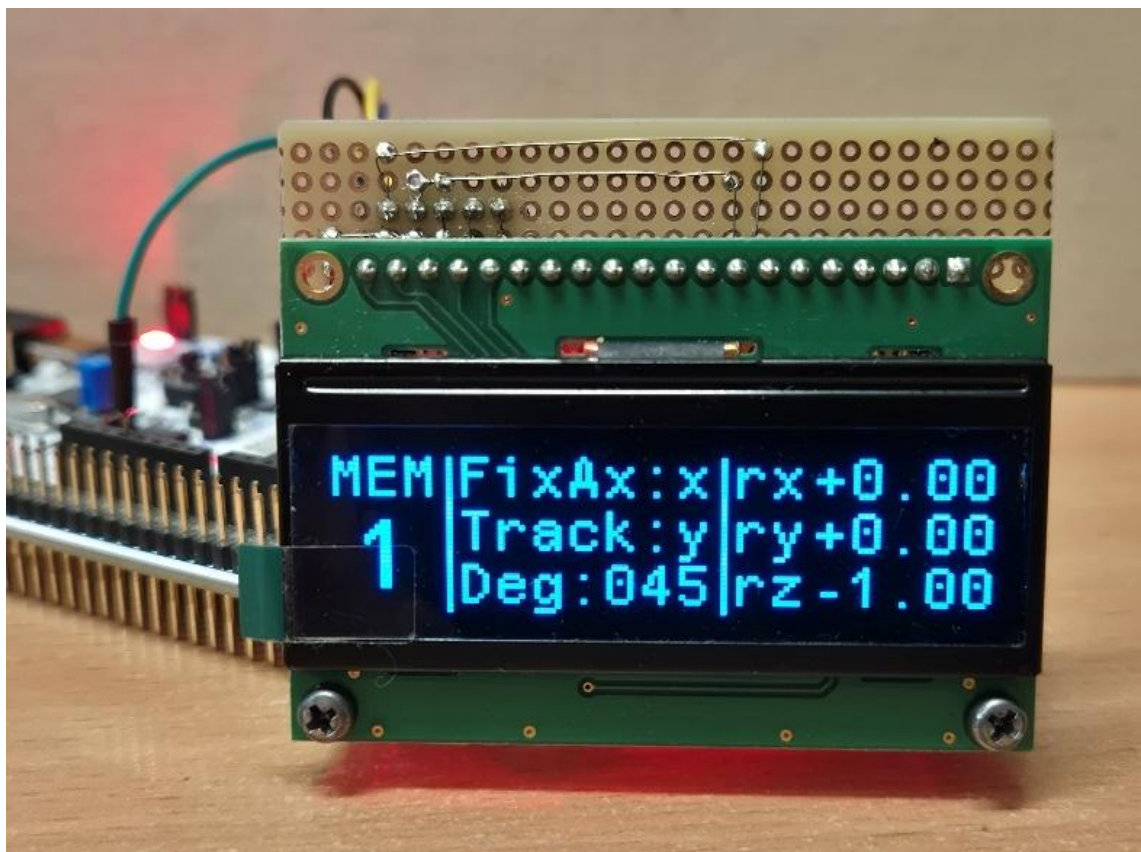
így például nem megfelelő mágnes távolság esetén az inicializálás hibára fut. Egyelőre a pontos hibadiagnózis implementálatlan, ez a jövőbeli fejlesztések során pótlendő a modul jobb kezelhetősége érdekében.

További megvalósított metódusok a nyers pozíció adatoknak kiolvasása, valamint egy ezt a metódust felhasználó szögpozíciót olvasó függvény, ami a nyers adatokból pozíciót számol fokokban.

Az osztály rendelkezik továbbá nullofszet rögzítésére változóval, ez egyelőre csak beállítható és kiolvasható, a későbbiekben ennek felhasználása implementálható a többi metódusba. Jelenleg a felhasználó ezt saját számításaira használhatja.

A mágnes megfelelő távolságának beállítása érdekében diagnosztikai célokból megvalósítottam az AGC (Automatic Gain Control) értéket kiolvasó metódust is. Ez az érték a mágneses tér erősségével arányos.

## 2.3 Kezelőfelület



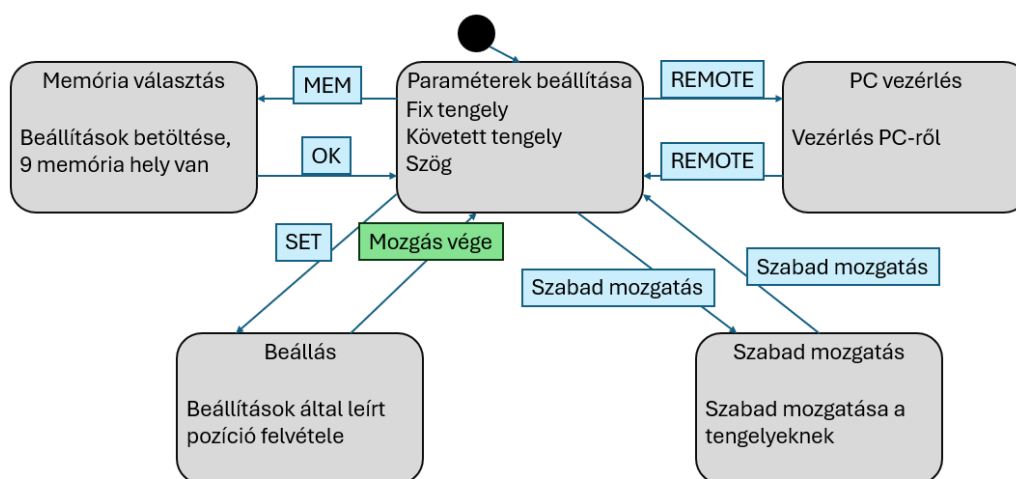
2-14. ábra OLED modulon megjelenített kezelőfelület (OLED próbapanelen, fejlesztés alatt)

Mivel cél, hogy az eszköz számítógép nélkül is intuitívan használható legyen, meg kellett terveznem a kezelőfelületét. Célom az volt, hogy a laboratóriumi tápegységek kezelőfelületével hasonlóan egyszerű és egyértelmű interfészt alkossak. Itt is (akárcsak a labortápoknál is) a felhasználó kiválaszt 2 paramétert, majd a gép gombnyomásra végrehajtja a paraméterek szerinti állapot felvételét.

Az eszköz feladata az, hogy a 3 tengelyű gyorsulásmérő szenzor bármely adott tengelyét képes legyen a gravitációs tengelyhez képest adott szögbe állítani. Ennek egy nagyon intuitív megoldása, ha a kezelő kiválaszthat egy fix, gravitációs térre merőleges tengelyt, ami körül forgatni fog a gép. Ez után kiválasztható a maradék két tengely közül, hogy melyiknek a szögét szeretnék megadni, majd megadható a kívánt szög. Ezzel a módszerrel egyértelműen tudni fogja az eszköz, hogy melyik forgató tengelyt hova forgassa és nem lesz olyan konfiguráció, ahonnan a beállítás nem egyértelmű. Ezen kívül a kezelőnek is egy intuitív eszközt ad a kívánt szög megadására.

A használatot nagyban egyszerűsíti, ha a felhasználó elmenthet néhány állapotot. Ennek érdekében az eszközt kiegészítettem egy EEPROM IC-vel. A felhasználó amikor beállít egy adott pozíciót, az az éppen kiválasztott „memória helyre” kerül mentésre. A memória választó gombbal a felhasználó beléphet a memóriaválasztás állapotba, ahol számlapon megadhatja a 9 közül melyik memóriát szeretné betölteni és akár módosítani.

### 2.3.1 Állapotgép



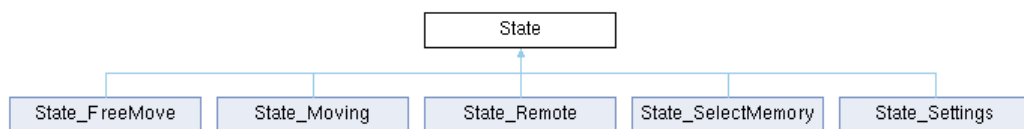
2-15. ábra állapotgép



Az eszköz vezérlését egy egyszerű állapotgéppel valósítottam meg (2-15. ábra). A felhasználó a gép bekapcsolásakor a *Paraméterek beállítása* állapotba lép (2-14. ábra). Ezen az állapoton belül egy belső állapotgép fut. A felhasználó kiválaszthatja a követett és a rögzített tengelyt, valamint a kívánt szöveget. A *Memória választás* menüben korábbi beállításokat hívhat vissza. A *Beállítás* állapotban nem lehet további parancsokat adni az eszköznek, amíg az a beállást végzi. A *Szabad mozgás* állapotban a felhasználó a numerikus billentyűzetet használhatja „nyilakként”, hogy a két mozgatott tengelyt a mozgástartományon belül tetszőleges pozícióba forgassa.

### 2.3.1.1 Állapotgép objektum orientált megoldása

Az állapotgépet a komplexitása miatt nem a klasszikus switch-case megoldással valósítottam meg, hanem objektumokkal. A switch-case megoldást csak a beállítások állapoton belüli alállapotgép kezelésére használtam.



2-16. ábra Állapot osztály öröklődési fája

Az állapotokat egy közös őosztályból származtatom le (2-16. ábra Állapot osztály öröklődési fája). Minden állapotnak van belépő (onEnter), kilépő (onExit) és futó (run) függvénye. A leszármazott, specifikus állapotok felülírhatják a be- és kilépő függvényt, de a futást leíró függvényt mindenképpen felül kell írniuk. Ezzel a struktúrával nagyon jól kezelhetők az egyes állapotok és azok tranziens viselkedései. Ennek a megközelítésnek a hátránya a nagyobb memóriaigény, de ez a választott mikrovezérlő esetén nem okoz problémát.

Az állapotgép maga is egy külön osztály. Az állapotgép konstruktorának meg kell adni a már lepdányosított állapotok pointereit. Ez mind statikusan történik, ezért nem áll fenn a memóriaszivárgás veszélye. Az állapotgépnek a konstruktorán kívül további kettő, egy futás (run) és egy állapotváltás (change state) függvénye van. Az osztály változói: minden típusú állapot pointer típusú változók, a jelenlegi állapotot számon tartó jelenlegi (current) állapotra mutató őosztály pointer típusú változó, valamint a felhasznált erőforrásokat kezelő objektumok is pointerekként vannak megadva (az OLED, a referencia szenzor, az EEPROM).

Az állapotgép jelenleg még nem teljesen befejezett, egyelőre a beállítások és a memória előhívás állapotok funkcionalitása van implementálva.

### **2.3.2 Számítógépes vezérlés**

A számítógépes vezérléshez a nyílt forráskódú „Shellminator” programot használtam. Ez a program lényegében egy terminál, mely mikrovezérlők távoli elérésére lett fejlesztve. Futtatása nem igényel beágyazott operációs rendszert. Mivel a kommunikáció egyszerű soros kommunikáció formájában zajlik, a számítógépre sem szükséges semmilyen specifikus szoftver telepítése.

Ez a keretrendszer lehetővé teszi a számítógépről érkező parancsok egyszerű kezelését. Az eszköz terminálból való kezelését segíti, visszaküldi a leütött betűket, lehetővé teszi az visszakeresést, valamint további terminál alkalmazásokban megszokott kiegészítő funkciót implementál.

A fejezethez felhasznált forrás: [17].

## **2.4 Kijelző**

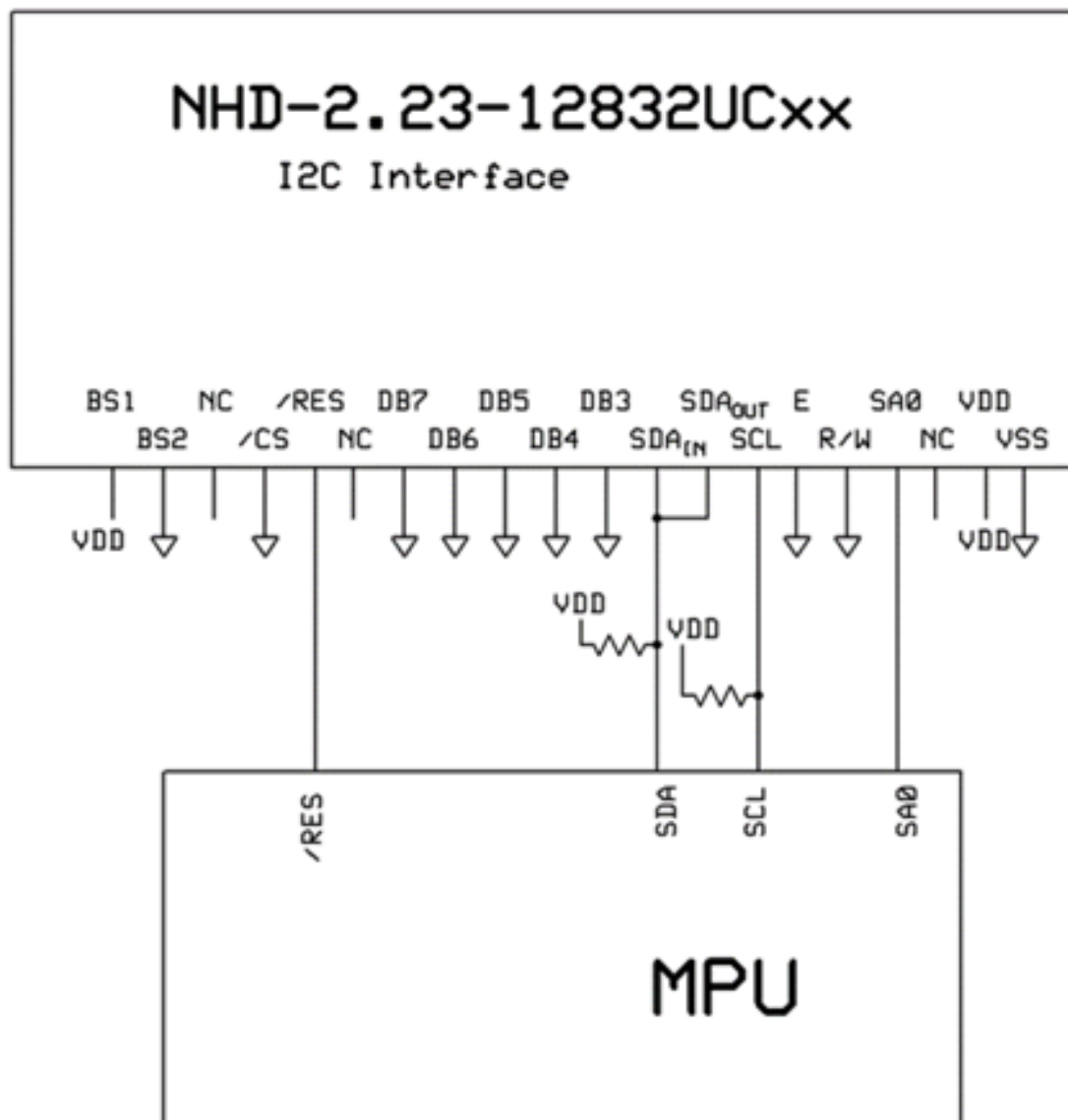
A kijelző kiválasztásánál a következő szempontokat vettem figyelembe: grafikus kijelző (azaz képpontonként címezhető, nem szegmenses), kontraszt, megfelelő méret, I2C vagy SPI támogatása, jó dokumentáció elérhetősége, professzionális megjelenés.

A fenti elvárásaimat tökéletesen teljesítette egy, a Newhaven Display International által gyártott 128x32 felbontású kék OLED panel (NHD-2.23-12832UCB3). Ez az OLED panel az SSD1305 típusú vezérlő IC-vel van szerelve. Ez a vezérlő IC támogat párhuzamos és soros kommunikációt is, és maga az OLED panel is úgy van megvalósítva, hogy ezeket a kommunikációs protolloknak a megvalósítása lehetséges legyen. Én az I2C kommunikációt választottam. A választásom indoka a kevesebb vezeték és egyszerű kommunikáció.

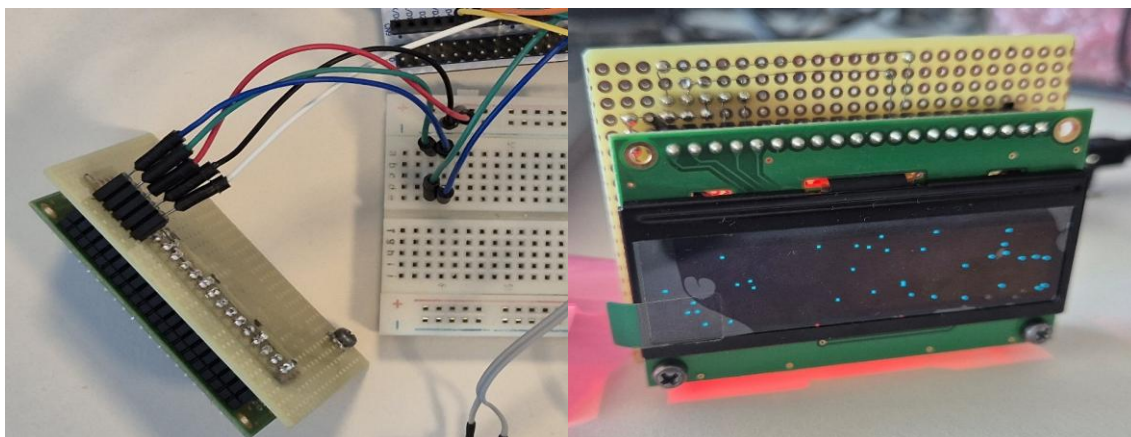
Felhasznált források: [18] [19].

### 2.4.1 Nyomtatott áramkör

Az OLED panel egy nyomtatott áramkörre szerelve kapható, amin 20 csatlakozási hely van kialakítva. Ezeket a csatlakozási pontokat a kívánt kommunikációs protokollnak és egyéb paramétereknek megfelelően kell földre vagy logikailag magas feszültségre „húzni”, illetve a megfelelő csatlakozási pontokon keresztül kell a mikrovezérlővel összekötni, hogy megvalósuljon a kommunikáció. A megfelelő kapcsolást a panel adatlapja részletezni (2-17. ábra).



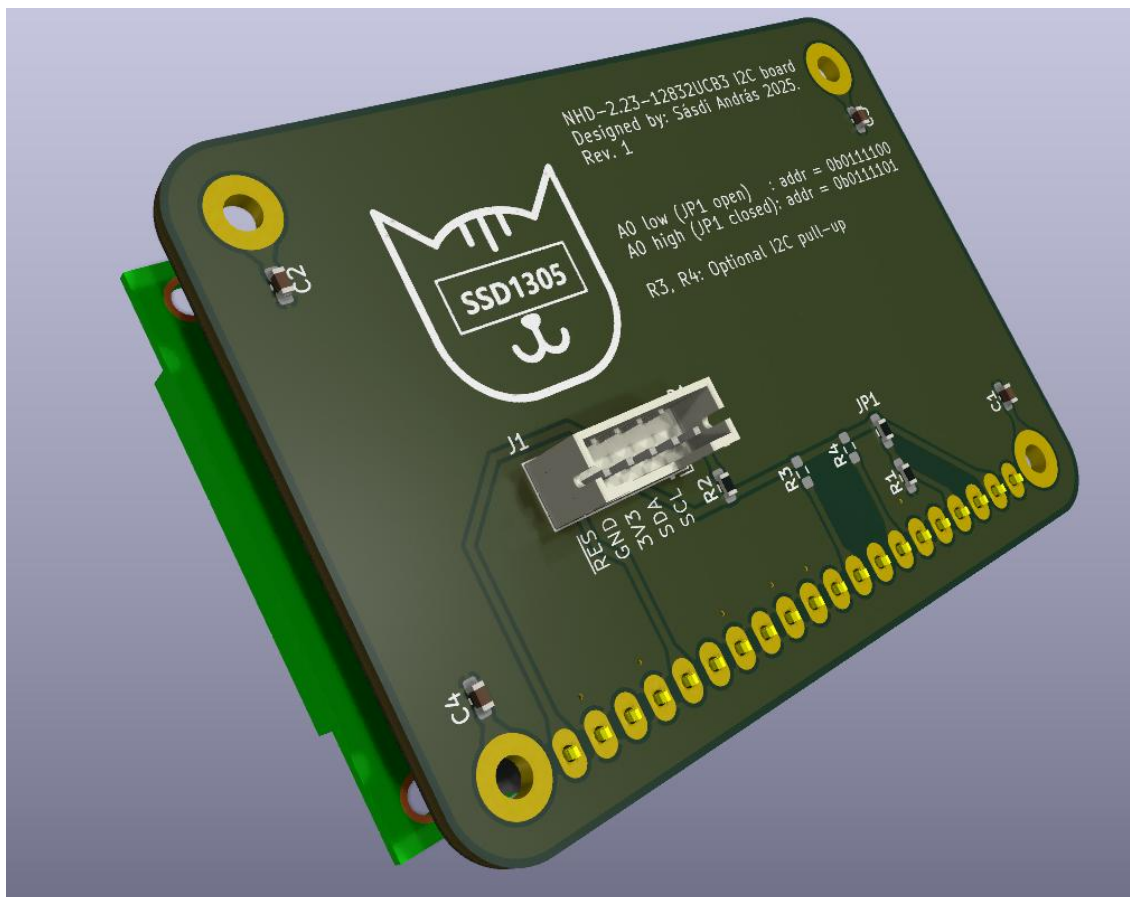
2-17. ábra OLED panel adatlapja I2C konfigurációhoz [19]



**2-18. ábra OLED modul próbapanelen**

Amíg a panel funkcionalitásait teszteltem és a drivert írtam hozzá, addig ideiglenesen egy próbapanelen hoztam létre ezt a kapcsolást és azon helyeztem el az I2C felhúzó ellenállásait is. Így a mikrovezérlőhöz már csak 5 vezetékot kellett vezetnem.

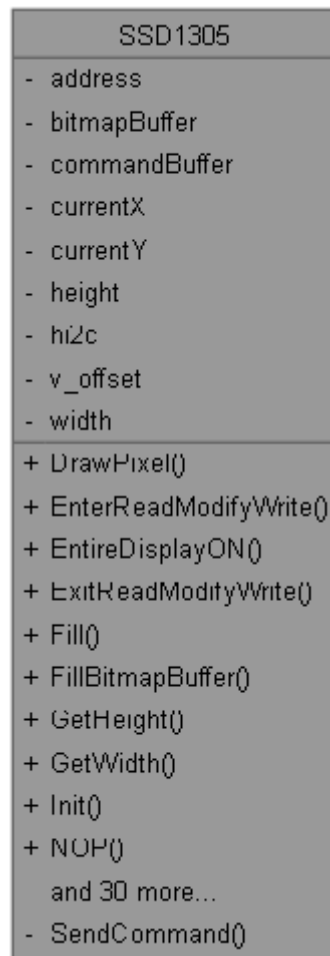
A végső megvalósításhoz ez alapján nyomtatott áramkört terveztem. Az áramkörön az A0 kiválasztására 0R ellenállás beforrasztásával van lehetőség, valamint bele terveztem egy tápfeszültség visszajelző LEDet is (ez a próbapanelen is helyet kapott).



**2-19. ábra OLED modul nyomtatott áramkörön**

A képen is látható R3 és R4 ellenállások az I2C felhúzó ellenállásainak a helyei. Mivel az I2C busz felhúzó ellenállásai az alaplapon, a mikrovezérlőhöz minél közelebb lesznek beépítve, ezért ezeket nem szabad jelen esetben beültetni. Helyük azért van kialakítva, hogy ettől a projekttől függetlenül is, akár önállóan felhasználható legyen az áramkör.

## 2.4.2 Driver



2-20. ábra SSD1305 kezelőosztályának UML diagramja

Sajnos ehhez a meghajtó IC-hez nem találtam szabadon felhasználható I2C kommunikációt használó drivert, ezért ezt nekem kellett megvalósítanom az adatlapja alapján. Az I2C kommunikációhoz, mint korábban az SPI hoz is, a beépített HAL könyvtárat használtam.

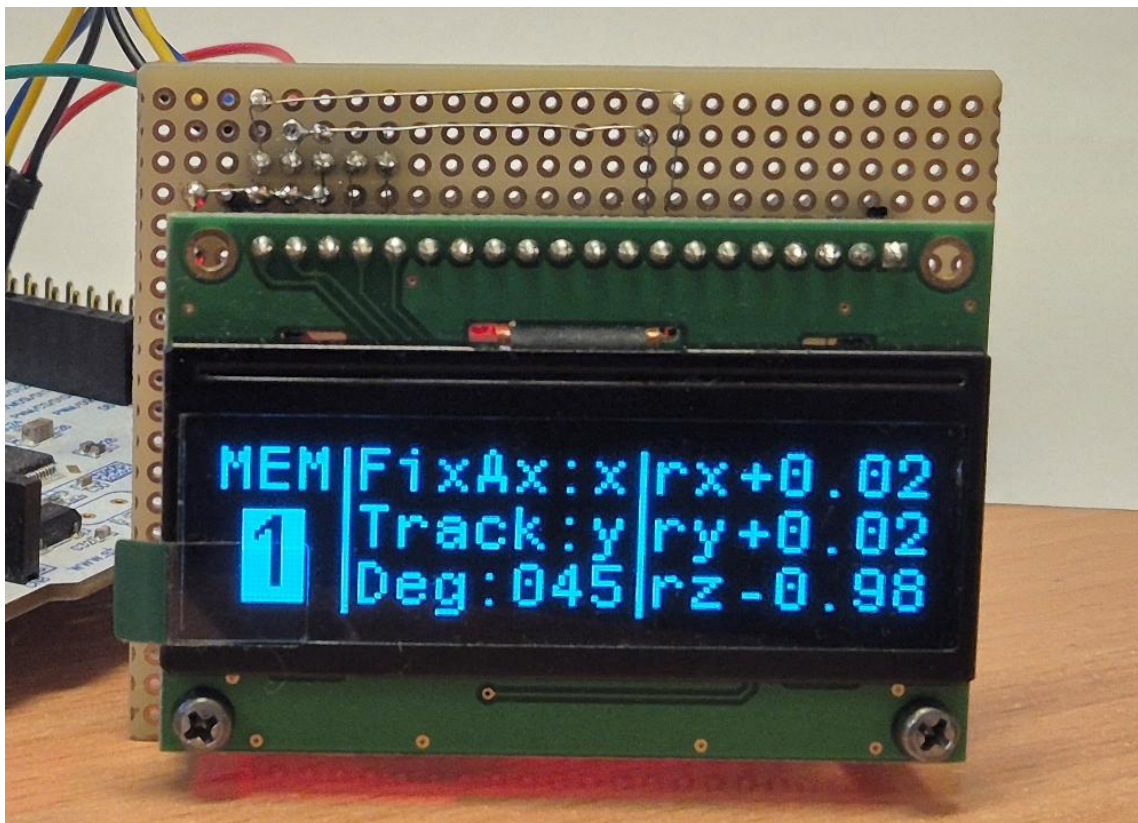
Ezt a drivert is C++ nyelven valósítottam meg, objektum orientált módon. A driver megírásához a következő linken elérhető MIT licenszes driverből merítettem ötleteket néhány megoldásra, a betűk bitmapjeit viszont teljes egészében innen másoltam, hiszen megrajzolásuk időigényes lett volna, ami valós hozzáadott érték nélkül hosszabbította volna a tervezési folyamatot. (<https://github.com/4ilo/ssd1306-stm32HAL>). A driver megírása ennek ellenére nagyon sok időt vett igénybe, főleg a sorok és oszlopok kalibrálása miatt. Ehhez nagy segítségre volt egyszerű bitmapek használata (például kalibráló keresztek).

A vezérlő IC nagyon sok funkcióval rendelkezik, igyekeztem ezeknek többségét implementálni. Az IC működését parancsok küldésével lehet vezérelni. A drivert is ennek a kezelési megközelítés szerint alakítottam ki. Az OLED kezelő objektum metódusai a parancsok megvalósításai, illetve a kirajzolás logikája.

Az inicializáló függvény itt is a konstruktortól külön áll, annak érdekében, hogy a hibakezelés könnyebben megoldható legyen. Az inicializálás során adhatjuk meg a kijelző orientációját, illetve inicializálás közben állítom be például a címzési sorrendet és a kalibrációt is.

A kijelző osztály rendelkezik egy bitmap puffer tömbbel. A felhasználó ebbe töltheti be a kiíratni való bitmapet, majd ezután egy külön metódust meghívva a vezérlő IC ez alapján a tábla alapján tölti ki a kijelzőt. Jelenleg ez nem DMA használatával van megoldva (ami a processzor terhelését csökkentené képfrissítéskor), ez egy jövőbeli potenciális fejlesztési lehetőség, ahogyan a jobb hibakezelés is.

### 2.4.3 Grafikus kezelői felület



2-21. ábra Grafikus kezelői felület, memóriaválasztás állapotban



A grafikus kezelői környezete függvényeit külön header és cpp fájlba írtam a rendezett kód érdekében. A 2-21. ábra mutatja a grafikus felületet. Alapvetően egy statikus ábra, amin csak a számok változnak, illetve az éppen kiválasztott és beállított értéket kiemeléssel jelzi.

Jelenleg a grafikus kezelői felületet kirajzoló függvények az OLED kijelzőt kezelő objektum pointerét kapják meg. Fejlesztési lehetőség létrehozni egy absztrakt kijelző osztályt és ennek a függvényeivel megvalósítani a grafikus kijelzést. Így a későbbiekben esetleges kijelző csere esetén könnyen beilleszthető a változtatás a kódbázisba, amennyiben a grafikus függvények egy általános kijelző osztály pointerét kaphatják meg.

## 2.5 Gomb panel

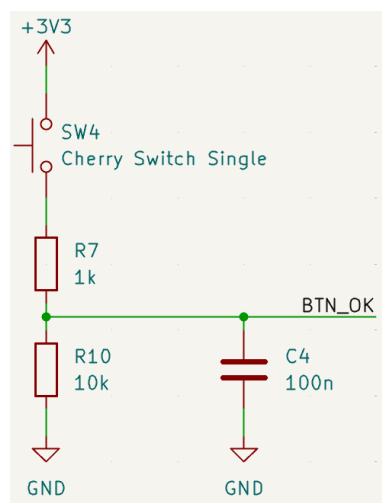
A felhasználó az eszközt (ha nem számítógépen keresztül kommunikál) gombok segítségével tudja kezelni. Annak érdekében, hogy a gombokat az alaplaptól függetlenül el tudjam helyezni az eszközön, a gomboknak külön panelt hoztam létre.

A nyomógombok típusának kiválasztásában fontos szempont volt a taktilis visszajelzés. A gumimembrános gomboknál a felhasználók sokszor nem érzik biztosra, hogy lenyomták-e a gombot, van, hogy túl erősen nyomják és összességében „olcsó” hatást adnak az eszköznek. Ezzel szemben a mechanikus gombok, amelyeknek van érezhető átbillenési pontja visszajelzést adnak a felhasználónak a gomb lenyomásáról. Ismert kapcsoló gyártó cég a Cherry, aminek a kék típusú kapcsolóját választottam. Ezek a gombok egyértelmű aktivációs ponttal és érezhető kattanással és hanggal rendelkeznek. A lenyomásról a felhasználó biztos visszajelzést kap.

A mechanikus kapcsolók (nyomógombok, relék stb.) tulajdonsága az úgynevezett „pergés”. Ez a jelenség abból adódik, hogy a mechanikailag bistabil rendszert az egyik állapotából a másikba billentve a rugalmas érintkező tagok visszapattanhatnak egymásról, így több kapcsolódás-szakadás periódus alakulhat ki az állapot állandósulása előtt [20]. Ez a jelenség ahhoz vezet, hogy a felhasználó egyszeri gombnyomását a mikrovezérlő sok egymás utáni lenyomásként érzékeli. Ennek a nem kívánatos jelenségnek több elterjedt és bevált kezelése is létezik. A gomb panel megépítése előtt, az állapotgép tesztelésekor szoftveres megoldást használtam. A hardveres megoldások közt elterjedt az aluláteresztő RC tag, illetve ennek ötvözése egy hiszterézises kapcsolóval (Schmitt

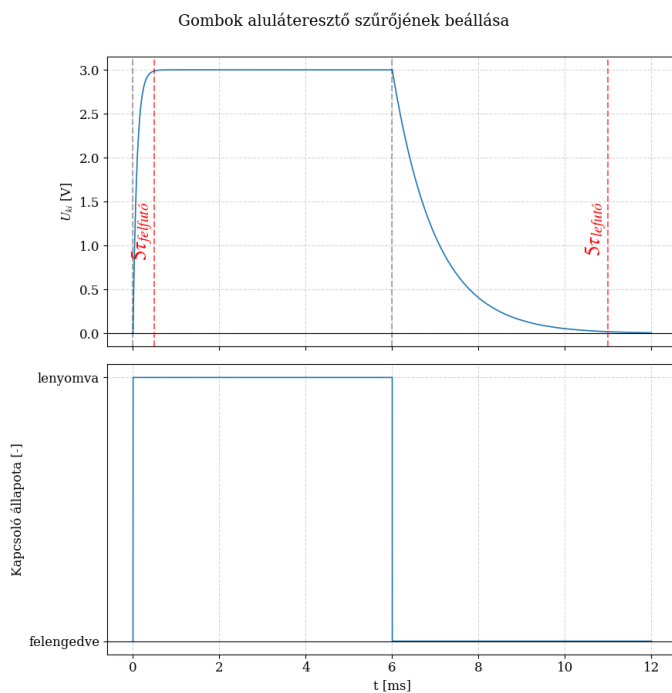


triggerrel). Én schmitt trigger nélküli, egyszerű RC tagos megoldást választottam (2-22. ábra), ami lényegében egy aluláteresztő szűrő. Schmitt triggerre azért nincs külön szükség, mivel a mikrovezérlő digitális bemenetei rendelkeznek hiszterézissel. Első iterációban ezt 0.5ms-os feltöltési és 5ms kisütési idővel terveztem. Ennek megfelelőségét az áramkör megérkezésekor tesztelhető, majd ez alapján szükség esetén módosíthatóak az ellenállás és kondenzátor értékek.



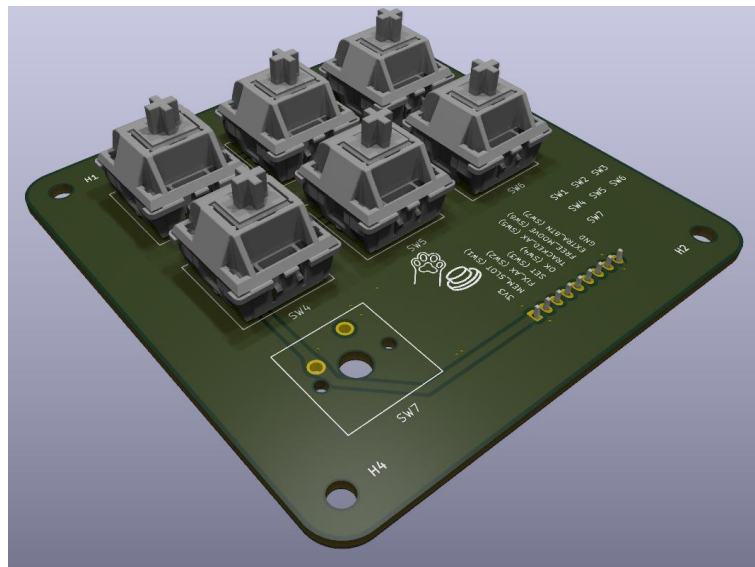
2-22. ábra Nyomógomb pergésmentesítése

A szűrő választ a gomb lenyomására és felengedésére (pergést nem szimulálva) a 2-23. ábra mutatja



2-23. ábra RC kör beállása

A nyomtatott áramkörön a bővíthetőség érdekében a szükségesnek ítélt gombokon kívül kialakítottam még egy, további gombnak a helyét.



**2-24. ábra Nyomógombok nyomtatott áramkörön**

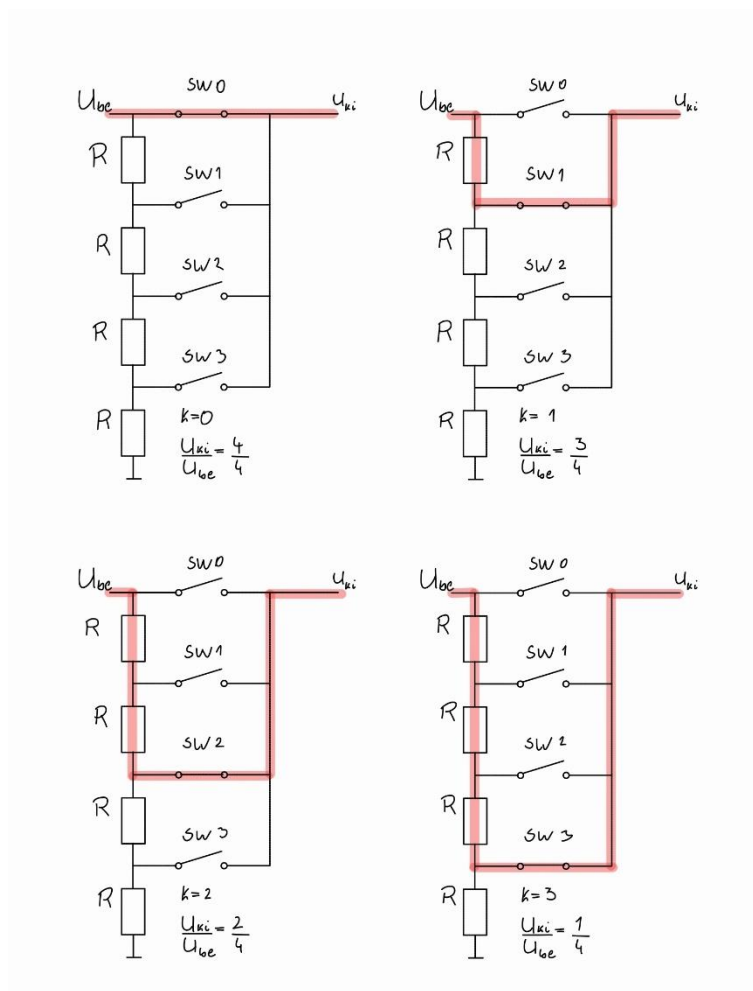
## 2.6 Numerikus billentyűzet

Az eszköz másik fontos kezelőszerve a numerikus billentyűzet. Ezzel tudja a felhasználó bevinni a kívánt szög értéket, valamint szabad mozgítás állapotban a mozgítás is ezzel valósítható meg. A számlap gombjaival szemben is a korábban támasztott elvárásaim voltak, ezért itt is Cherry Blue nyomógombokat használtam.

Mivel az eszköznek sok perifériát kell kezelnie és szempont a bővíthetőség, minimalizálni szerettem volna a számlap kezeléséhez felhasznált GPIO-k számát. A lehetséges megoldások közül egy analóg megközelítést választottam. A kimeneti feszültség a lenyomott gomb indexével arányos. Ezt a megoldást a vezetékes fülhallgatók gombjainak kivitelezéséből merítettem [21]. Lényegében tehát egy feszültségosztó ellenállásainak arányát állítom diszkrét értékek között a gombok lenyomásával. A kimeneti feszültség szintet a mikrovezérlő egyik beépített analóg-digitális átalakítója (ADC) segítségével visszamérve állapítható meg a lenyomott gomb. A szükséges ADC felbontás kiválasztásához a kimeneti feszültség szinteket kell végig venni. A kimeneti feszültség a  $k$ . gomb lenyomásakor  $n$  db gomb esetén:

$$U_{ki} = U_{be} \cdot \frac{n - k}{n} \quad (2.12)$$

Ezt 4 kapcsolóval szemlélteti a 2-25. ábra. A numerikus billentyűzetnek 10 kapcsolója van, tehát két gomb lenyomás között a legkisebb feszültségkülönbség a bemeneti feszültségnek egy tizede. 3V-os referencia esetén ez 0,3V. A mikrovezérlő 12 bites 3,3V maximális feszültségű ADC felbontása  $\cong 0,008 \frac{V}{LSB}$ , ez nagyságrendekkel kisebb a legkisebb feszültségkülönbségnél, tehát biztosan megfelelő. Meg kell jegyezni továbbá, hogy a valós legkisebb feszültségkülönbség ennél valamivel rosszabb a később tárgyalt aluláteresztő szűrő ellenállása miatt, viszont azt elég nagyra választva ez a hatás elhanyagolható.



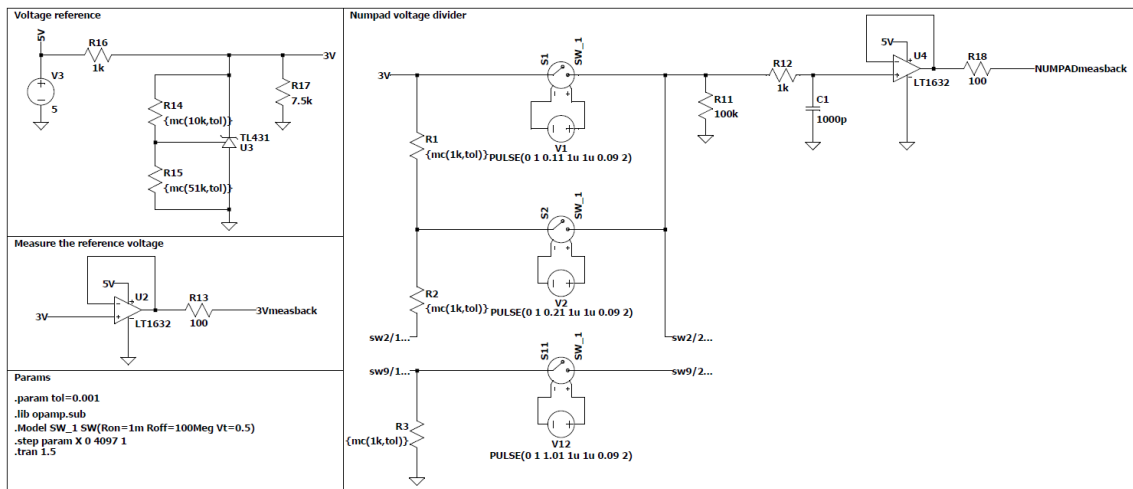
2-25. ábra Feszültségosztó 4 kapcsolóval

Egyszerre több gomb lenyomása esetén a regisztrált gombnyomás az lesz, aminek lenyomásakor létrehozott feszültségértékhez legközelebb áll a több gom lenyomásakor kapott feszültség.

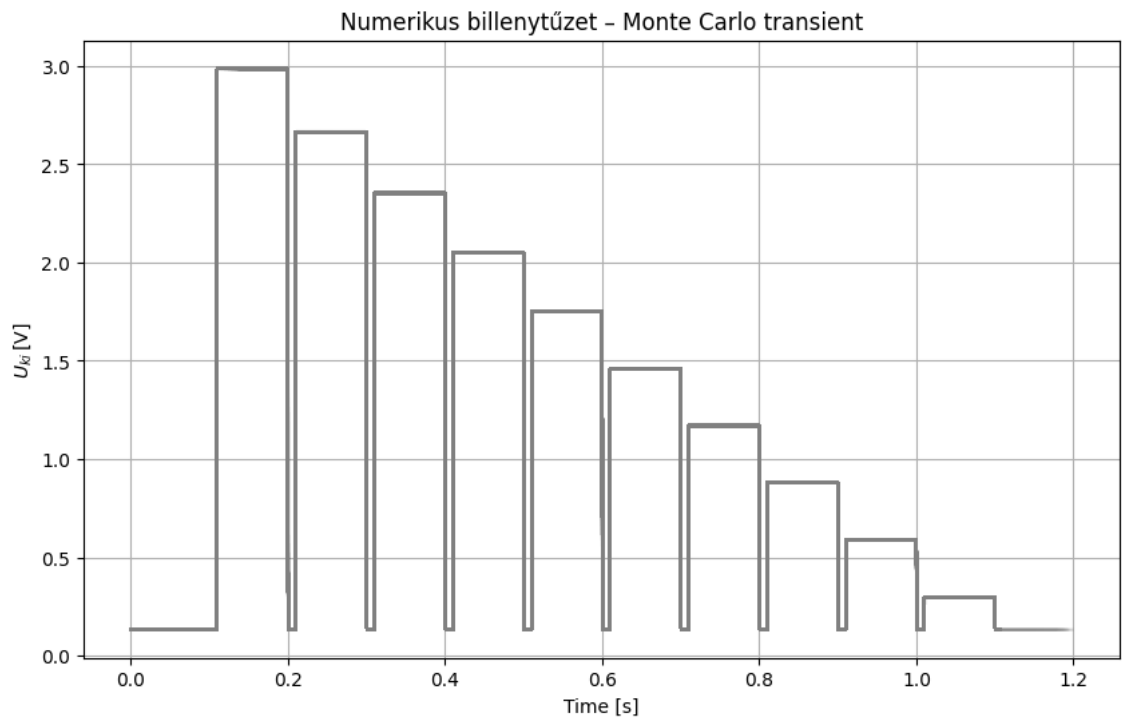
A lenyomott gomb megállapításához szükséges feszültségosztás mérésére alapvetően két megközelítés lehetséges. Az első, hogy a be- és kimeneti feszültséget is méri a mikrovezérlő és ennek kettőnek a hányadából lehet a lenyomott gombot megállapítani. Ennek előnye, hogy a bemeneti feszültség változása nem befolyásolja a módszer eredményét, hátránya viszont, hogy 2 analóg-digitális átalakító is szükséges hozzá. A második megközelítés ismert referenciafeszültség létrehozása, majd ennek az osztása. Ebben az esetben elegendő egy ADC használata, a mért jelet ismert szintekhez hasonlítva.

Mivel az eszköz tervezése során a modularitás és rugalmasság központi szempont, mindkét kivitelezést megvalósítottam. A bemeneti 5V-ból 3V-os referenciafeszültséget hozok létre, de mind a bemeneti, mind az osztott feszültség vissza van vezetve a mikrovezérlőhöz. A feszültségek visszamérését követőerősítőkön keresztül vezetem vissza, de erősítésnek a lehetőségét kialakítottam: a visszacsatolás OR ellenálláson keresztül, valamint nem beépített 0603-as beforrasztási lehetőség passzív alkatrésznek, amivel erősítés vagy szűrés megvalósítható. Tehát akár az is megoldható, hogy a bemeneti 5V-ot vezetem a feszültségosztóra (kiveszem a referenciafeszültséget biztosító állítható referenciát), majd visszamérés előtt ezt visszaskálázom a mikrovezérlőnek megfelelő feszültségtartományra.

Ez a megközelítés felépítéséből adódóan érzékeny a passzív komponensek értékek szórására. A 0.1%-os pontosságú ellenállások bőségesen elérhetőek és nem számítanak drága, precíziós ellenállásnak. Azt, hogy ez a pontossági kategória elegendő lesz-e LTspice szoftverben ellenőriztem (2-26. ábra), Monte Carlo legrosszabb esetek elemzéssel. Ennek a lényege, hogy végigiterálja a tűréssel ellátott passzív komponenseknek az adott tűréssel megadott szélső értékeit, így minden legrosszabb kombinációt lemodellez. A tűréssel ellátott elemek a feszültségreferenciát beállító ellenállások, illetve a feszültségosztó ellenállásai. Így tehát összesen  $N = 12$  tűréssel ellátott komponenssel rendelkezik a modell. Ez összesen  $2^N + 1 = 2097$  futtatást jelent. [22]



2-26. ábra numerikus billentyűzet LTspice modellje

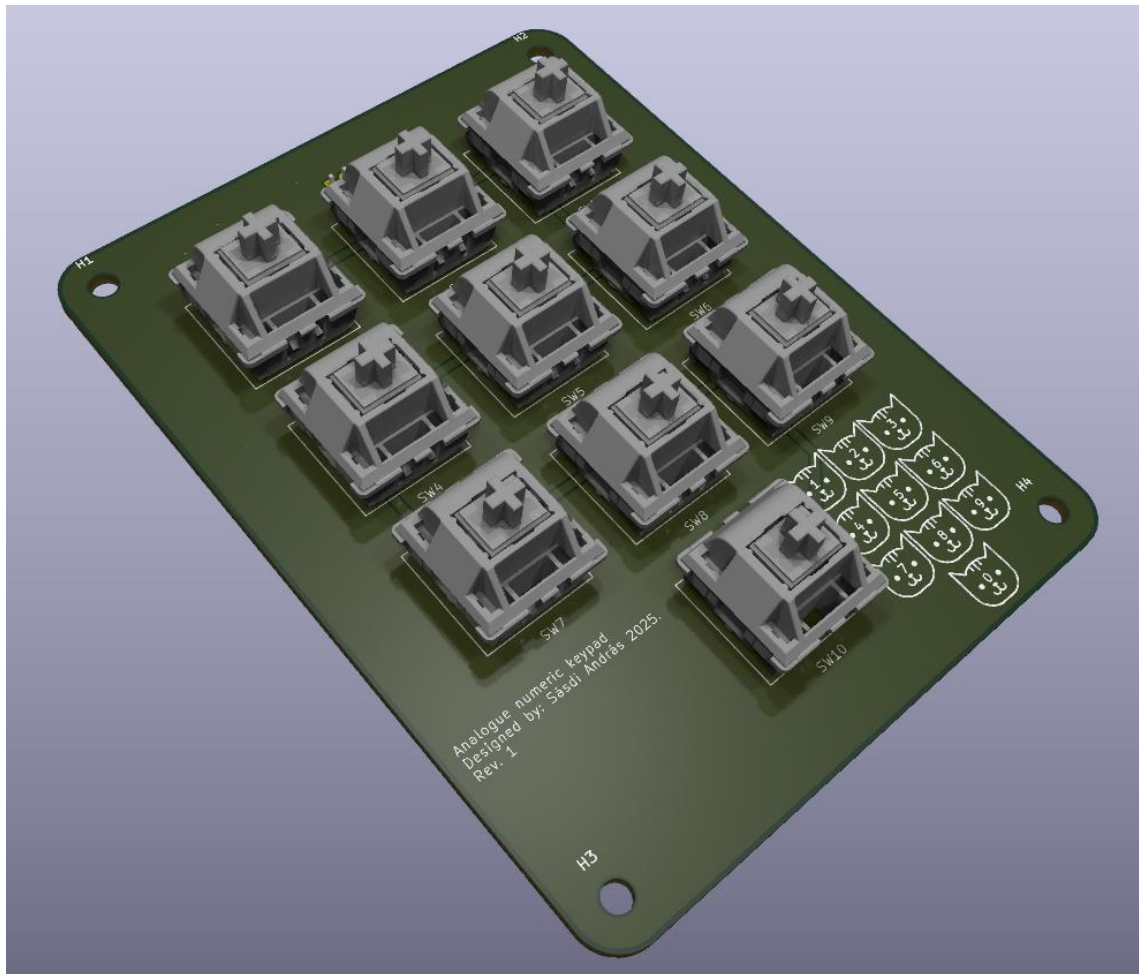


2-27. ábra Monte Carlo szimuláció eredménye

A 2-27. ábra szemlélteti a Monte Carlo szimuláció eredményét. A szimuláció során sorban az valamennyi gomb lenyomása szimulálva lett. Ennek eredményeként a felfebb tárgyalt működésnek megfelelően a 10 gombnak megfelelően 10 diszkrét feszültségérték látható. A kimenő feszültség egy adott lépcsője egy adott gomb lenyomásakor mérhető kimeneti feszültséget mutatja.

Ahogy azt a 2-27. ábra mutatja, a 0.1%-os tűréssel rendelkező ellenállások által okozott szórás a lépések magasságánál nagyságrendekkel kisebb, az ábrán nem is figyelhető meg.

Az áramkör tervezésénél a visszamérés előtt a feszültségosztó kimenetét egy aluláteresztő szűrőn vezetem keresztül. Ennek az értékei egyelőre  $1\text{k}\ \Omega$  és  $1\text{nF}$ , de amennyiben szükséges, a panel viselkedésének kimérése után ez módosítható.

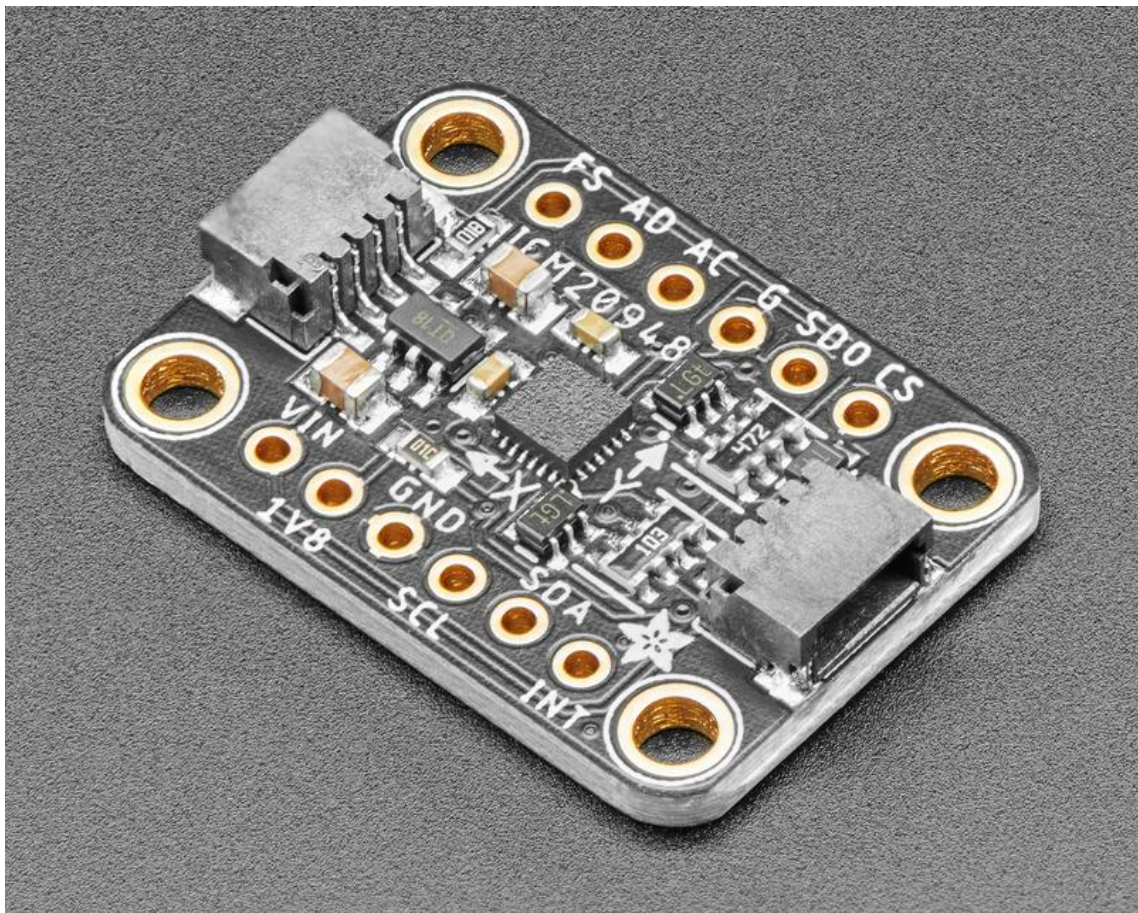


**2-28. ábra Numerikus billentyűzet PCB panelje**



## 2.7 Referenciaszenzor

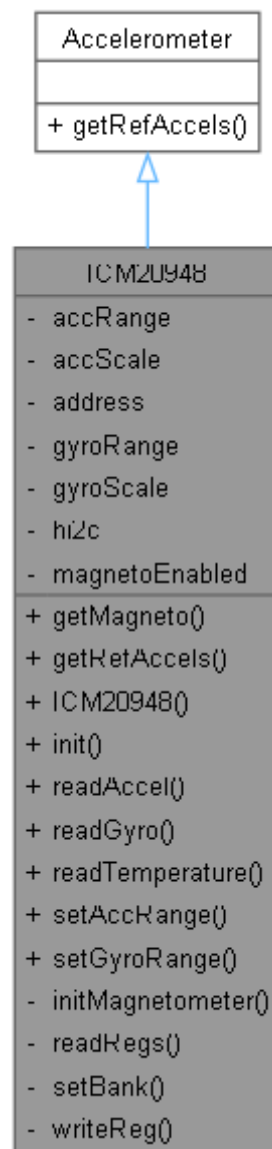
A mért szenzor orientációjának visszamérhetőségét egy referenciaszenzorral biztosítom. A választott szenzor a TDK által gyártott ICM 20948, ami egy 9 tengelyű gyorsulásmérő, giroszkóp, magnetométer. Egyelőre csak referencia gyorsulásmérőként van használva. Azért esett a választásom olyan szenzorra, ami nem csak gyorsulásmérő, mivel az eszköz tervezése és kivitelezése közben szempont volt a bővíthetőség és a modularitás. További indok volt emellett az adott szenzor mellett, hogy az Adafruit cég árulja ezt a szenzort PCB-vel együtt, amin minden szükséges passzív alkatrész szerepel, így nem szükséges külön áramkört terveznem a szenzorhoz.



2-29. ábra Adafruit TDK InvenSense ICM-20948 9-DoF IMU [23]



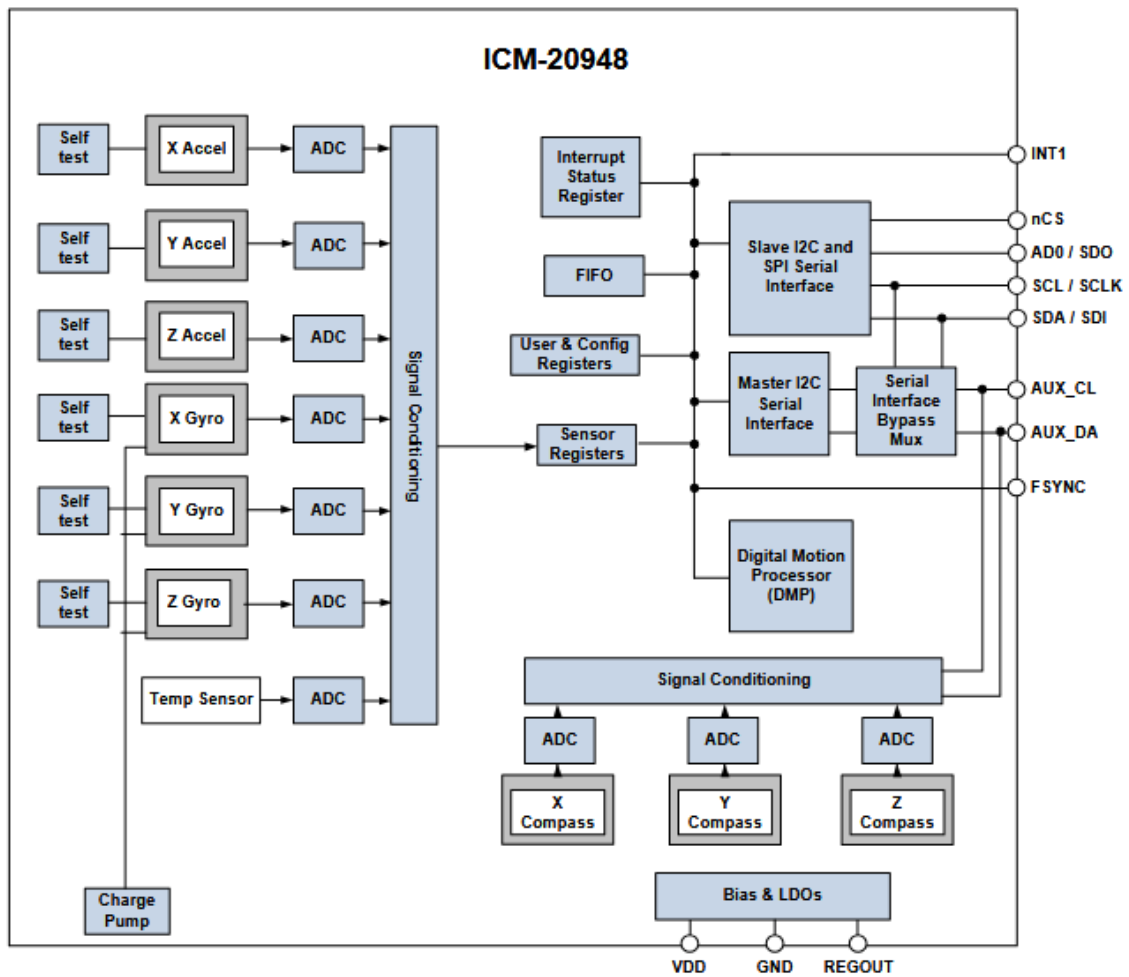
### 2.7.1 Driver



2-30. ábra ICM20948 kezelőosztályának UML diagramja

Mivel nem találtam számomra megfelelő drivert ehhez a szenzorhoz, ezt a drivert is én fejlesztettem magamnak a szenzor adatlapja alapján.

Ez a szenzor is I2C-csatornán kommunikál, tehát itt is a HAL könyvtárat használtam, valamint a korábbiakkal azonos módon itt is objektum orientált drivert írtam.



2-31. ábra ICM20948 blokkdiagrammja az adatlapból [23]

Ez a 9 tengelyű szenzor egy 3 tengelyű giroszkópból, egy 3 tengelyű gyorsulásmérőből és egy AK09916 típusú 3 tengelyű digitális iránytűből áll. A giroszkóp és a gyorsulásmérő egy integrált céláramkörön (ASIC) osztoznak, míg a mágneses szenzor külön ASIC-en van. Ez a két egység közös tokban találhatóak és közös I2C busz köti őket össze. Mivel jelenlegi felhasználáshoz ez nem szükséges, nem implementáltam a mágneses szenzor kezelését, így a driver csak gyorsulás és giroszkóp adatok lekérdezését támogatja.

Itt is, mint az enkóder esetében is, külön header fájlba írtam ki az IC fontosabb regisztereinek címeit, illetve itt is csak a számomra szükséges funkcionalitásokat implementáltam.

Az inicializáló függvényt a korábbiakhoz hasonlóan a konstruktortól függetlenül írtam meg. Az egyértelmű függvényeken kívül, mint a gyorsulás és giroszkóp kiolvasása

még a beépített hőmérséklet szenzornak a kiolvasását és a giroszkóp és gyorsulás mérő szenzor mérési tartományait beállító metódusokat valósítottam meg.

Megjegyezném továbbá, hogy annak érdekében, hogy az állapotgép más típusú referenciaszenzorral is tudjon működni, létrehoztam egy gyorsulásmérő őssosztályt, amitől örököl az ICM20948 osztály is. Ennek az osztálynak a referenciagyorsulásokat megadó függvényét írja ez felül. Így amennyiben később más típusú referenciaszenzor kerül beépítésre, az is kompatibilis lesz az eszközzel, ha ettől az őssosztálytól örököl.

Természetesen ezt a drivert is lehet még bővíteni. Nem csak a korábban említett digitális iránytű kiolvasásának implementálásával, de például jobb hibakezeléssel, vagy jelfeldolgozó metódus implementálásával.

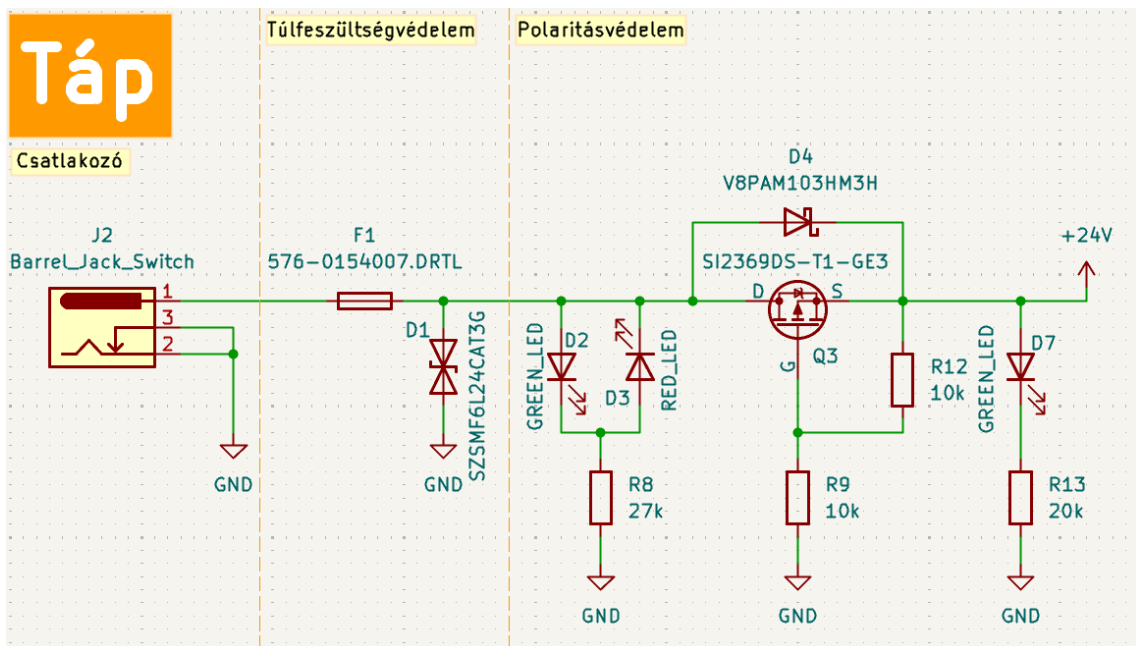
## **2.8 Alaplap**

Az eszköz legkomplexebb áramköre, ami az összes modult összefogja, az alaplap. A tervezés közben a központi megfontolás a könnyű javíthatóság, átalakíthatóság és modularitás volt.

### **2.8.1 Táp áramkör**

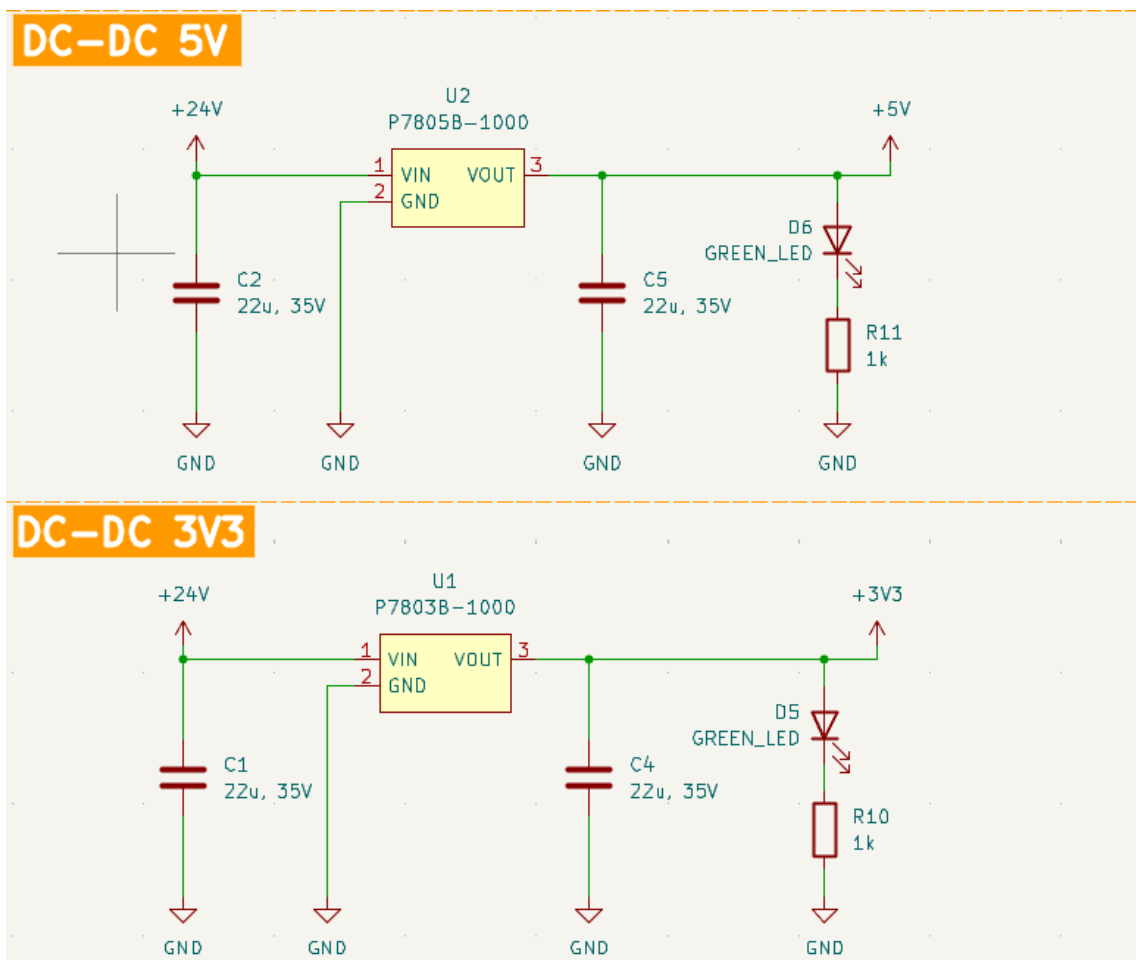
Az alaplapot 24V-os tápellátással terveztem, mivel a motorvezérlők akár 28V-os motor tápfeszültség kapcsolására is alkalmasak.

A túlfeszültség elleni védelmet egy kétirányú Zéner-diódával és olvadóbiztosítékkal valósítottam meg. A polaritás védelemhez MOSFET tranzisztort alkalmaztam, ami a megfelelő polaritásnál nyit csak ki. Ezzel párhuzamosan egy teljesítménydiódát is bekötöttem, hogy a kapacitások miatt magas bekapcsolási áram ne a MOSFET test diódáján folyjék. A megfelelő polaritást és a tápfeszültség jelenlétét LED-ek jelzik vissza.



2-32. ábra Tápfeszültség előkészítése

A panelen szükség van továbbá 3.3V, illetve 5V feszültségre is. Ezeket a feszültségeket kapcsolóüzemű DC-DC buck konverterekkel állítom elő (2-33. ábra). Itt is LED-ek jelzik vissza a kimeneti feszültség jelenlétét. A felhasznált DC-DC átalakítók P78B-1000 sorozatúak. Ezek az átalakítók fix kimeneti feszültségűek, 3.3V-ot P7805B-1000, míg 5V-ot P7803B-1000 állít elő. Az 1000 a sorozatszám végén az 1A maximális kimeneti áramot jelzi. Létezik 2A változat is, amennyiben ez szükséges lenne, a csere nagyon egyszerű. Azért ezt a konvertert választottam, mert a témavezetőm tapasztalat alapján javasolta, majd az adatlap elolvasása után megfelelőnek tartottam. A bemeneti feszültségi tartományukba beleillik a 24V tápfeszültség, rövidzárlat elleni védelemmel és alacsony kimeneti zajjal rendelkeznek. További előnyük, hogy pusztán 2 (egy be- illetve egy kimeneti) kondenzátorra van szükségük a működéshez. [24]



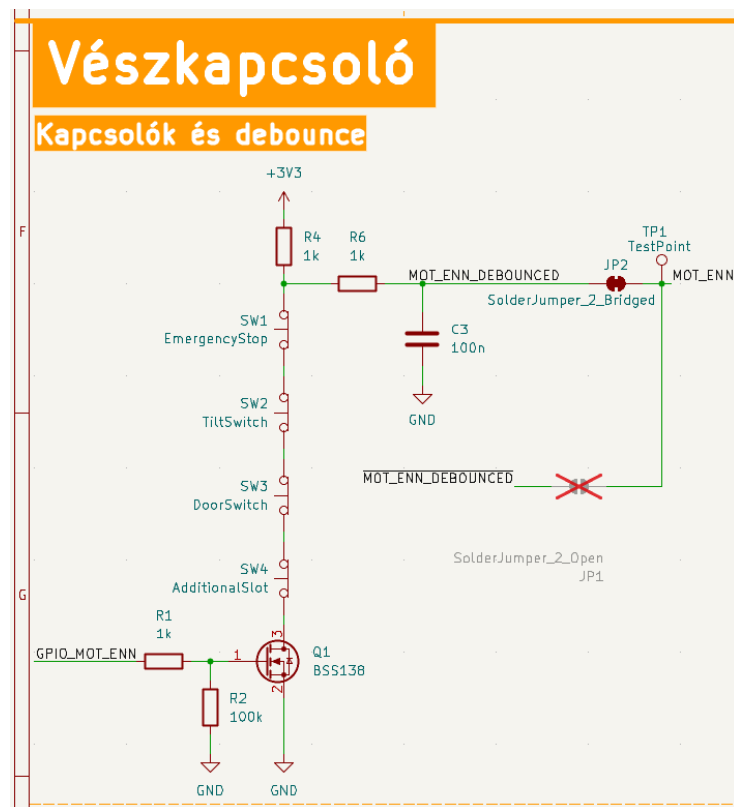
2-33. ábra Alacsonyabb feszültségek előállítása

## 2.8.2 Vészkapcsoló

Ahogy azt a Biztonságtechnikai lépések (1.5.2) fejezetben is említettem, az eszközt el kell látni vészleállító berendezéssel, ami az aktuátorokat deaktiválja. Ahogy itt is látható és ahogy azt a biztonsági megfontolásoknál említettem a vészleállító nem áramtalanítja az egész rendszert, csupán a motorokat. Mivel a motorvezérlő modulokon van „Enable Not”, azaz „nem engedélyez” bemenet, elegendő ezt magas logikai szintre kapcsolni, ekkor a motorok áramtalanítása megtörténik és megszűnik a nyomaték. Ez az engedélyező jel tehát nem a mikrovezérlőn keresztül érvényesül, hanem közvetlenül a motorvezérlőkbe van bekötve. Így a mikrovezérlőtől függetlenül szinte azonnal leállítja a motorokat.

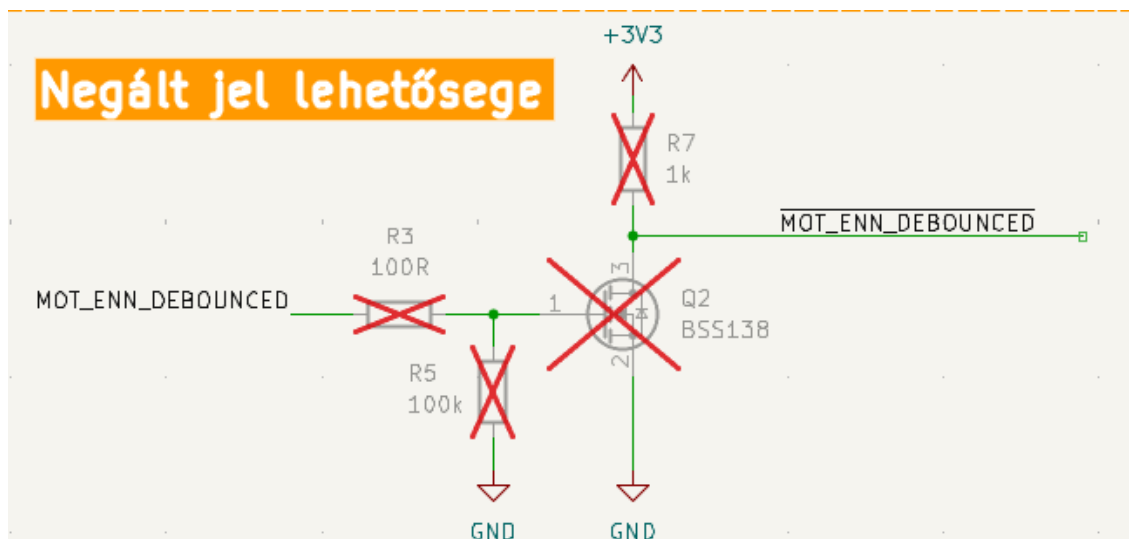
A vészleállító jel legalább 3 helyről érkezhetsz (vészeállító gomb, burkolat nyitás kapcsoló, döntéskapcsoló). Ezeken kívül a modularitás és fejleszthetőség érdekében egy további kapcsoló lehetőségével és szoftveresen vezérelhető jellel is kiegészítettem a vészleállító kapcsolást. Alapesetben zárt kapcsolókkal terveztem a kapcsolást, hiszen

biztonságtechnikai szempontból ez ideálisabb: vezeték szakadás vagy kilazulás esetén is aktiválódik a leállítás, azaz meghibásodás esetén biztonságos állapotba kerül. A mikrovezérlőtől érkező jellel egy BSS138 MOSFET kapcsolok. Ezek a kapcsolók sorba vannak kötve, így bármelyik nyitott állapotba kerül a motorvezérlők leállítják a motorokat. Ezt a jelet is pergésmentesítettem egy RC taggal mielőtt tovább vezetném a motorvezérlőkhöz, illetve a mikrovezérlőhöz. A kapcsolást a 2-34. ábra mutatja.



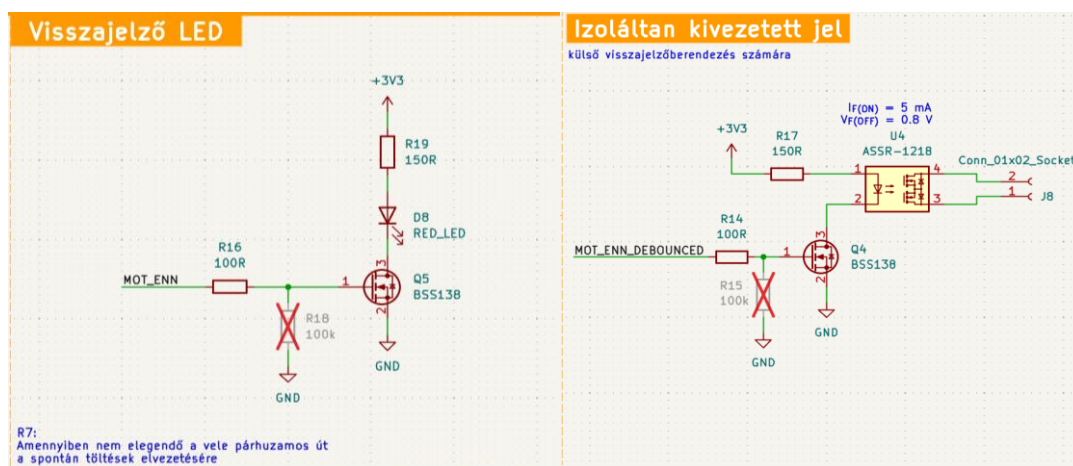
2-34. ábra Leállító kapcsolók jelei

Annak érdekében, hogy amennyiben a jövőben olyan motorvezérlőre lennének cserélve a jelenlegiek, aminek fordított az engedélyező logikája, a jel negálásának a lehetőségét is kialakítottam. Ezt a 2-35. ábra mutatja, a piros x-ek a KiCad szoftverben a nem beépítendő alkatrészeket jelölik.



2-35. ábra negált jel létrehozásának lehetősége

A vészleállító jelet egy visszajelző LED segítségével vizualizálom az alaplapon. Külső visszajelző berendezés lehetőségét úgy alakítottam ki, hogy egy szilárdtest relét kapcsolok az engedélyező jellel. Így az alaplaptól teljesen külön üzemelő visszajelző berendezés kapcsolható.



2-36. ábra Visszajelző LED és külső jelet kapcsoló szilárdtest relé

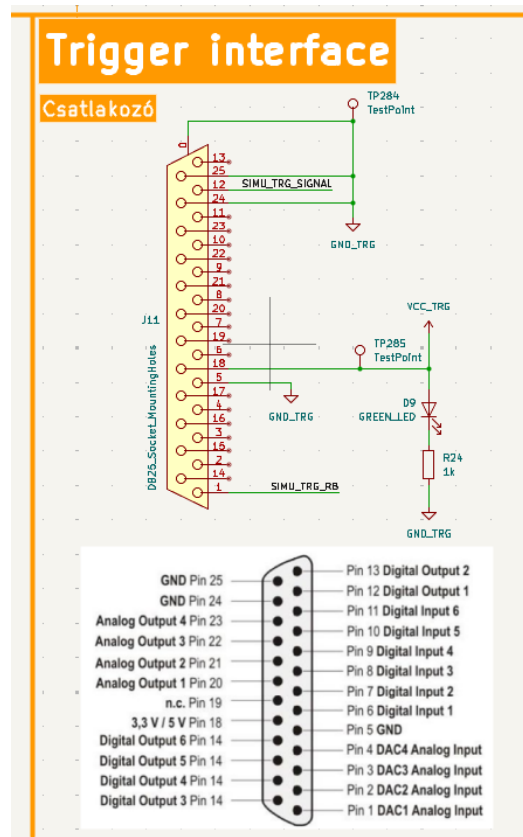
A kapcsolók jeleit és a külső jelet is csavaros csatlakozóval, úgynevezett sorkapoccsal lehet bevezetni (2-37. ábra). Így sokféle vezetékezési megoldás egyaránt beköthető az alaplapba. Amennyiben ez hosszú távon rossz megközelítésnek bizonyul, nem nehéz ezeket más csatlakozóra cserélni.





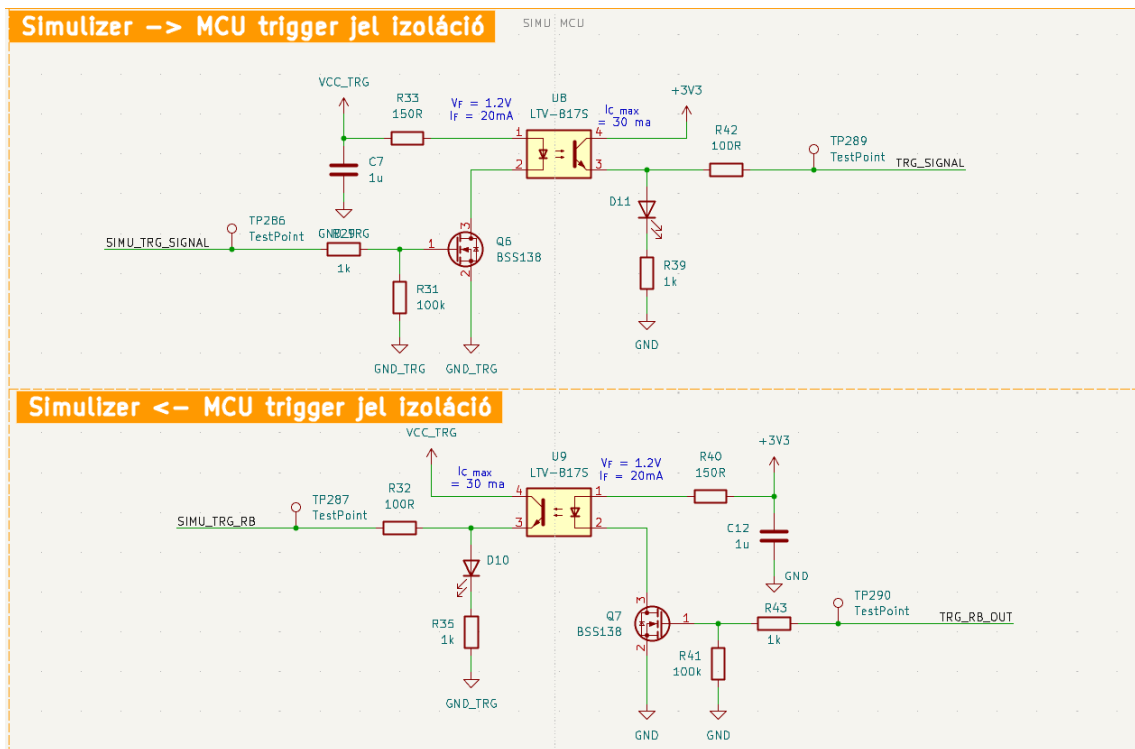
### 2.8.3 Külső trigger jel

Elvárás volt, hogy kívülről lehessen a beállást időzíteni egy trigger jellel, ami specifikált csatlakozón keresztül érkezik (2-38. ábra). Az időzítő jelnek és csatlakozójának galvanikusan leválasztottnak kell lennie az eszköztől, illetve az időzítést vissza is kell jelezni a külső eszköz felé.



2-38. ábra külső időzítő jel

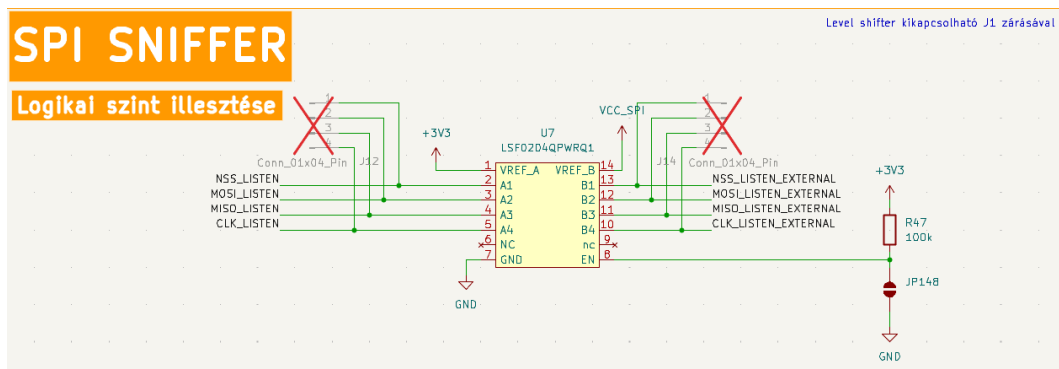
A külső időzítőjel csatlakozóján lévő tápfeszültséget LED fénnel visszajelzem, így könnyen megmondható, ha be van kapcsolva a külső eszköz. A kívülről érkező jelet optocsatolón keresztül viszem a mikrovezérlőhöz és a mikrovezérlőtől a külső eszközhöz is optocsatolón keresztül vezettem a visszajelző jelet. A jeleket LED-ek segítségével vizualizáltam.



2-39. ábra külső időzítőjel és visszajelzés galvanikus leválasztása

## 2.8.4 SPI lehallgató

A tesztelt gyorsulásérzékelők lehetnek SPI kommunikációval konfiguráltak. Ebben az esetben érdemes lehet lehallgatni az SPI-csatornát, és a forgatást akár bizonyos SPI üzenetekhez időzíteni. Mivel ennek a kommunikációnak a feszültségszintje nem ismert ezért a jelet illeszteni kell. Erre két irányú, több feszültségszint illesztésére alkalmas 4 csatornás kommunikációs szint illesztő IC-t használtam.

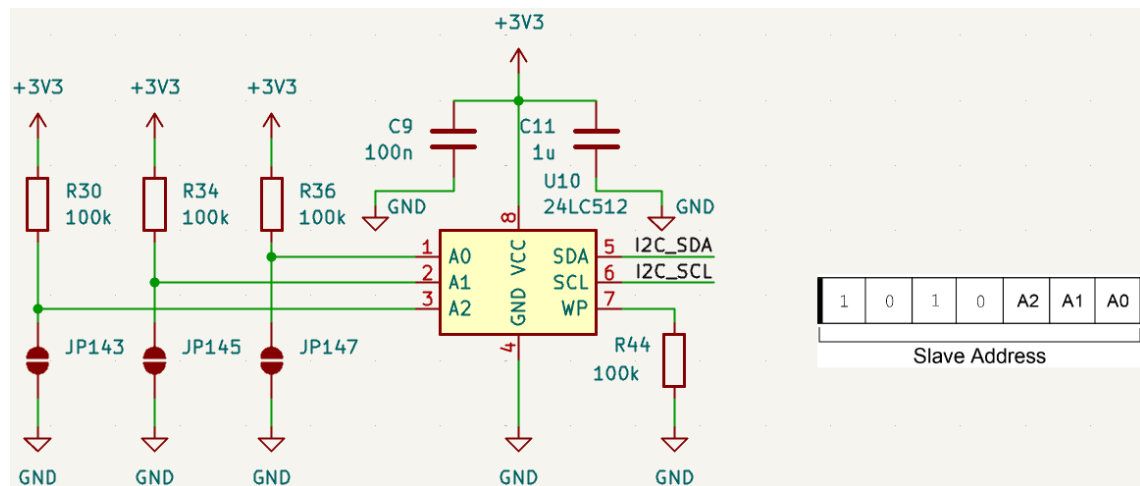


2-40. ábra SPI logikai szint illesztése

## 2.8.5 EEPROM

A Kezelőfelület fejezetben már említettem, hogy a beállításokat egy EEPROM-ra le lehet menteni. Az EEPROM IC az alaplapon kapott helye. Ez egy 24LC512 típusú

EEPROM, amire azért esett a választás, mert korábbi projektjeim során is használtam már és volt hozzá saját driverem, valamint a fejlesztéshez próbapanelre szerelt ilyen típusú IC-m, amivel az alaplapp elkészültéig is ellenőrizhettem a rendszerbe illeszthetőséget. A modularitás és későbbi fejleszthetőség jegyében könnyen konfigurálhatóvá tettem az EEPROM I2C címét.



2-41. ábra EEPROM

## 2.8.6 EEPROM driver

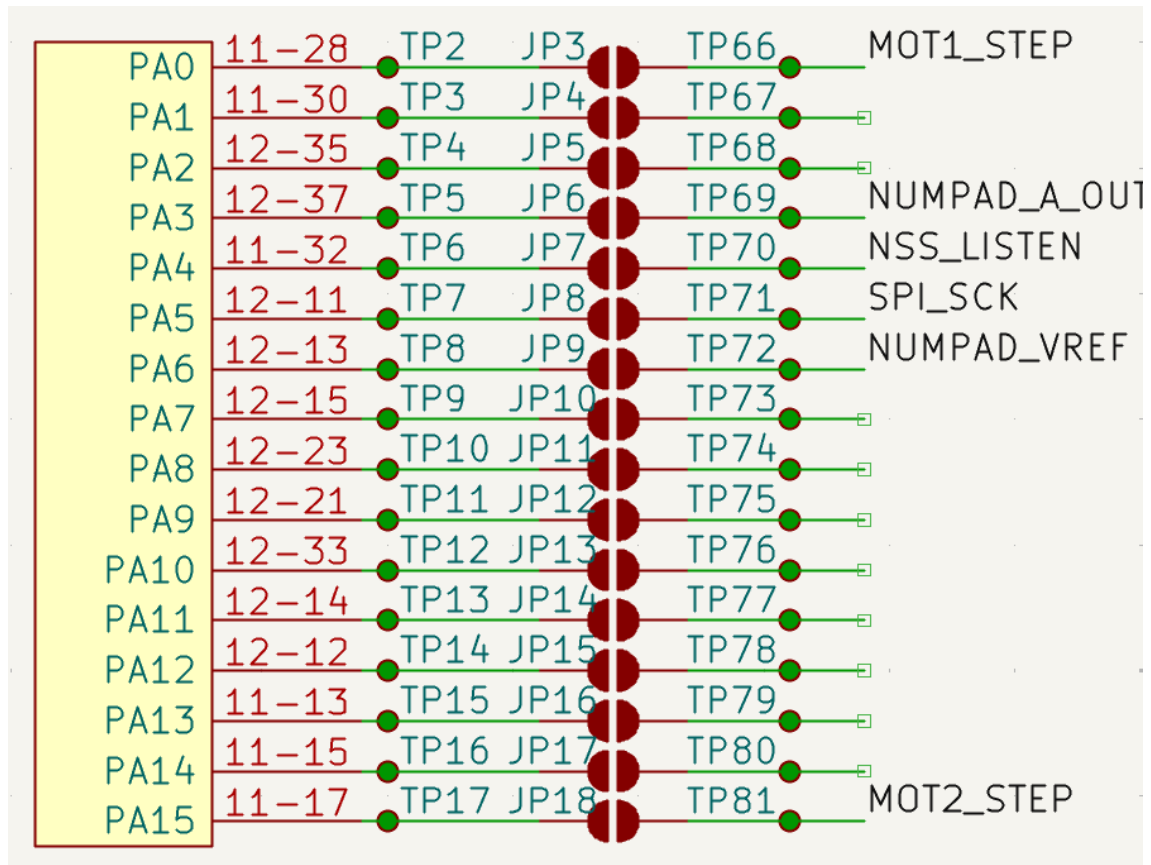
Erre az EEPROM-ra már korábbi projektjeim során is írtam drivert, így a szoftverfejlesztési idő jelentősen rövidült. Ennek a drivernek a lényege, hogy kulcs-érték adatpárokat tárol. Használata így nagyon egyszerű, és támogatja 8, 16 és 32 bites számok, valamint karakterláncok eltárolását (maximum 128 byte per adat). A memória 500db 128 byte-os lapra van tagolva. Az első 55 oldalon 440 „header” tárolható (kulcs, érték címe, érték típusa), majd az 55. oldaltól a kulcsokhoz tartozó értékek laponként vannak tárolva, azaz 445 oldal áll rendelkezésre, így 5 oldal marad kihasználatlan, és 440 adat tárolható el.

A felhasznált kulcsokat inicializálás előtt deklarálni kell. Inicializáláskor a memórián tárolt kulcsokat összeveti a megadottakkal, egyezés esetén elmenti, melyik memória címen található az adat, eltérés esetén felszabadítja a memória helyet és felülírja a kulcsot. Az adattípusok változtatására futás közben is lehetőség van.

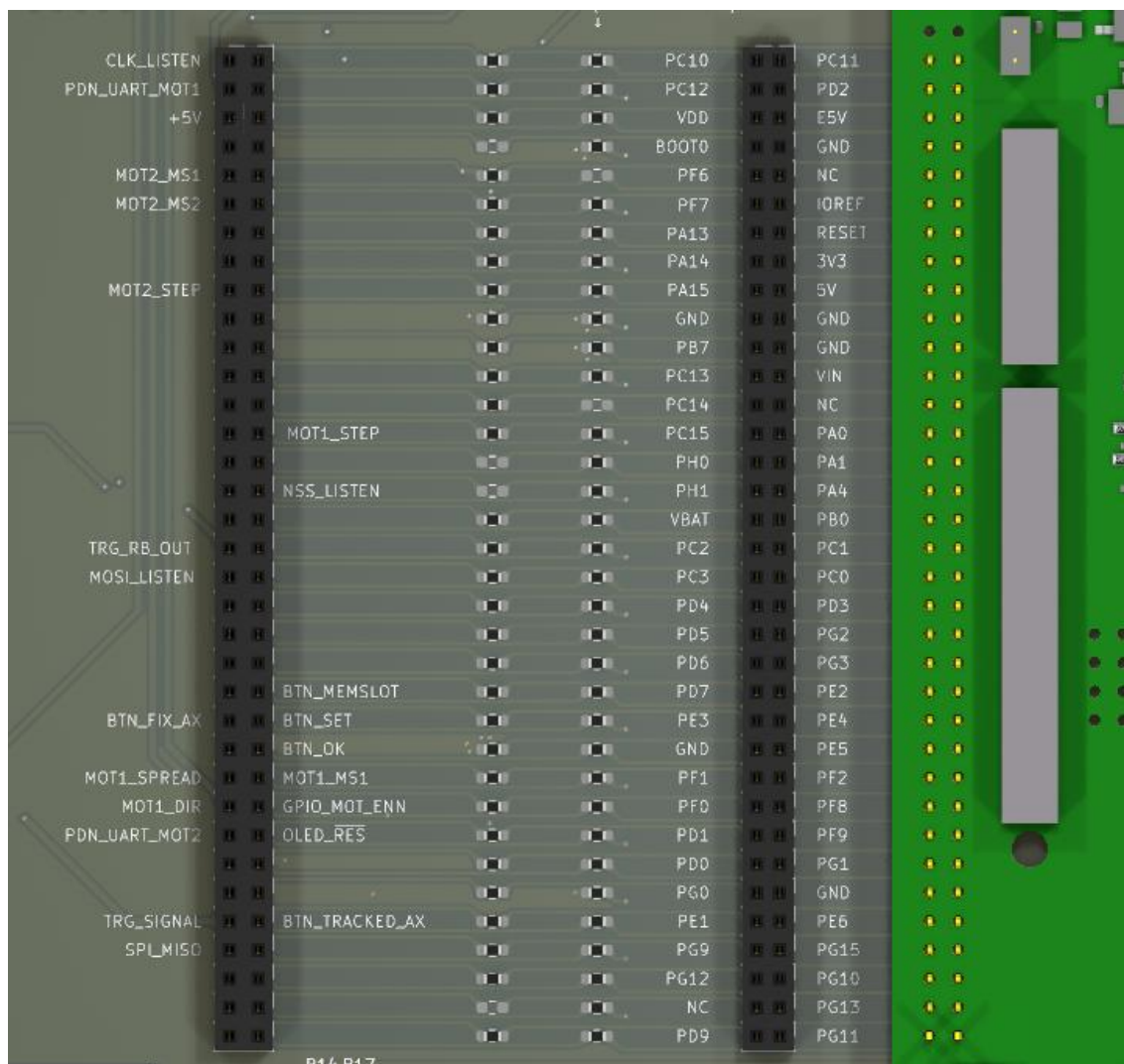
Jelen alkalmazásban a memóriát a beállítások mentésére használom. Minden memóiahelyhez 3 kulcsot használok fel: fix tengely, követett tengely, beállított szög.

## 2.8.7 Mikrovezérlő és további modularitási/bővítési megfontolások

A mikrovezérlőt, mint korábban is említettem fejlesztői panellel együtt terveztem bele az eszközbe. Így nem kell a mikrovezérlő környezetét kialakítanom, valamint meghibásodás esetén is könnyen cserélhető. A modularitás és későbbi könnyű bővíthetőség érdekében a Nucleo összes csatlakozóját kivezettem. Mindegyik csatlakozót jumperen keresztül vezettem ki, a jumper előtt és után is tűkesor került. Így könnyedén újra konfigurálható a kivezetés, valamint új modulok könnyen beilleszthetőek. A nyomtatott áramkörön pedig minden jelet feliratoztam, ami a Nucleo-ba van bekötve a könnyű követhetőség érdekében. Ezeket a megoldásokat a 2-42. ábra és a 2-43. ábra mutatja be.

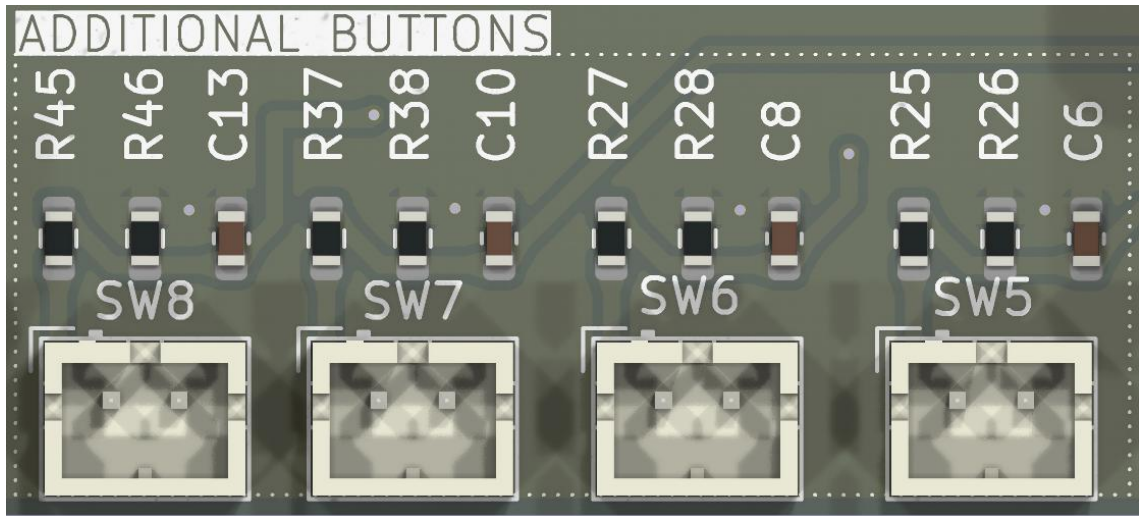


2-42. ábra Mikrovezérlő "A" portjának csatlakozóinak kivezetése

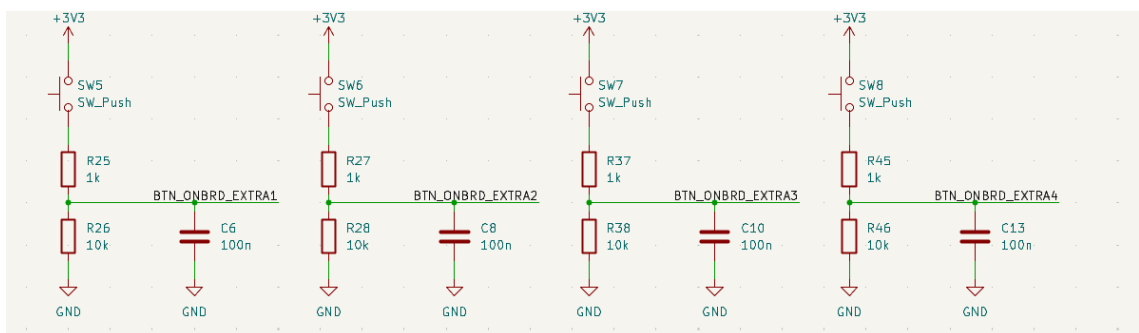


2-43. ábra Mikrovezérlő jeleinek kivezetése a nyomtatott áramkörön

A bővíthetőséget szem előtt tartva az alaplapon kialakítottam 4 további gomboknak is a bekötési lehetőségét (2-44. ábra és 2-45. ábra). Ezeket a gombokat a panelen pergés mentesítem, így nagyon egyszerűen beköthetőek ezek a további gombok.



2-44. ábra További gombok lehetőségének kialakítása a nyomtatott áramkörön



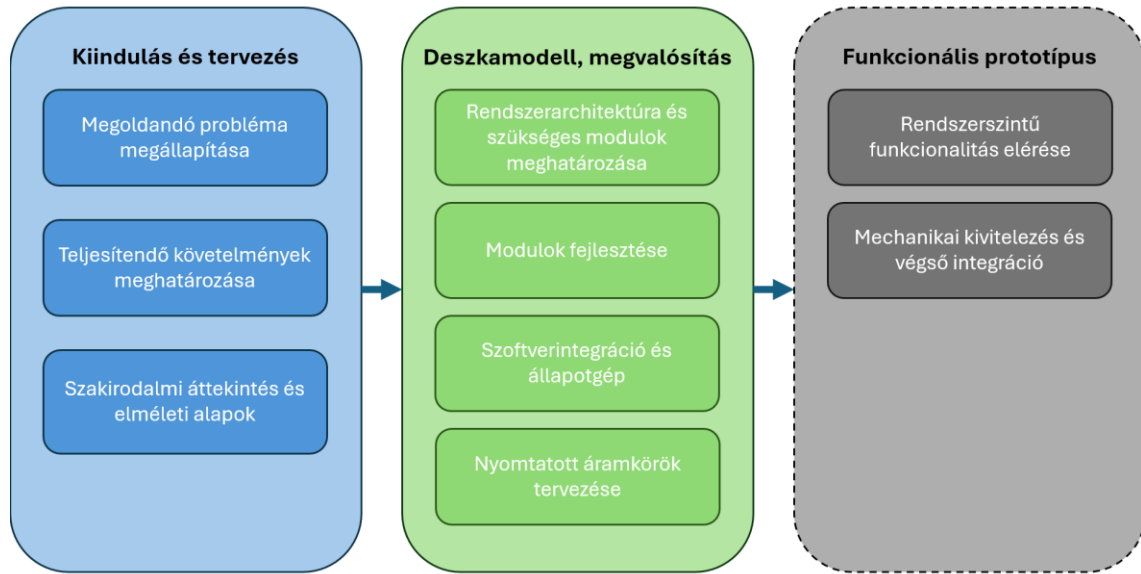
2-45. ábra Kiegészítő gombok pergésmentesítése

## 3 Összefoglalás/eredmények értékelése

### 3.1 Eredmények

A mechanikai design kivitelezése még folyamatban van. Ennek ellenére az eszköz legfontosabb részei elkészültek és bebizonyosodott, hogy tudnak együttműködni. Az eszköz megvalósítása közben a legnagyobb kihívást a motorok folytonos vezérlés jelentette. Ezen kívül megvalósítottam egy drivert az SSD1305 IC-hez, ami annak legtöbb funkcionalitását lefedi, valamint egy olyan nyomtatott áramkört terveztem az ezt használó NHD-2.23-12832UCB3 kijelző modulhoz, amivel egy későbbi fejlesztő egy másik projektbe könnyedén beillesztheti. Megvalósítottam egy, az alapvető funkcióit lefedő drivert az AS5174U enkóderhez valamint az ICM20948-hoz. Létrehoztam továbbá egy későbbi projektekbe is könnyedén beilleszthető numerikus billentyűzetet, ami korlátozott GPIO számú mikrovezérlőkhöz is könnyedén felhasználható, illetve egy könnyen beszerelhető gomb panelt. Ezeken túl az eszköz alaplapját is megvalósítottam, ahol a modularitást és későbbi szerelhetőséget igyekeztem tudásom szerint maximalizálni.

## 3.2 Javaslatok/Következtetések/Tanulságok



3-1. ábra Projekt ütemezése

A projekt ütemezését a 3-1. ábra szemlélteti. A diplomamunka keretein belül az első két lépést kellett megvalósítanom. Munkám során elkészült a tőlem elvárt funkcionális (deszka) modell. Ennek segítségével bebizonyosodott, hogy a vele szemben támasztott követelményeket az eszköz teljesíti. A laborban beüzemelhető prototípus következik, melynek a fejlesztése és komponenseinek gyártása folyamatban van (PCB-k stb.) A deszkamodell fejlesztése során gyűjtött tapasztalatok a felhasználói prototípus fejlesztése során figyelembe lettek véve és ezek alapján folyik jelenleg is a tesztelés és gyártás.

Miután megbizonyosodok a rendszer megfelelő működéséről, a szoftveres megoldásoknál említett kiegészítéseket kell megvalósítanom, valamint egy Python kezelő osztályt, amivel könnyen szkriptelhető az eszköz, és Matlab környezetből is kezelhetővé teszi.



## 4 Irodalomjegyzék

- [1] „Acceleration sensor: BMA580,” © Bosch Sensortec GmbH, 2024. [Online]. Available: <https://www.bosch-sensortec.com/products/motion-sensors/accelerometers/bma580/>. [Hozzáférés dátuma: 10 május 2025].
- [2] Micromega Dynamics, „MEMS vs Piezoelectric Accelerometers: Which One to Choose?,” 2025. [Online]. Available: <https://micromega-dynamics.com/mems-vs-piezoelectric-accelerometers-which-one-to-choose/>. [Hozzáférés dátuma: 16 december 2025].
- [3] „All about MEMS sensors,” © Bosch Sensortec GmbH, 2024. [Online]. Available: <https://www.bosch-sensortec.com/about-us/our-company/mems/>. [Hozzáférés dátuma: 10 május 2025].
- [4] C. Murphy, „Why MEMS Accelerometers Are Becoming the Designer’s Best Choice for CbM Applications,” Analog Devices, 01 február 2021. [Online]. Available: <https://www.analog.com/en/resources/technical-articles/why-memes-acceler-are-best-choice-for-cbm-apps.html>. [Hozzáférés dátuma: 10 május 2025].
- [5] *PCB design guidelines for MEMS sensors*, STMicroelectronics, 2023.
- [6] Z. Bende és A. Rávai, Gépek biztonsága, Budapest: Akadémiai Kiadó, 2018.
- [7] Technológiai és Ipari Minisztérium, MUNKAESZKÖZÖK BIZTONSÁGA, Technológiai és Ipari Minisztérium, 2022.
- [8] evogimbals, „How Does a 3 Axis GoPro or DSLR Gimbal Work?,” 20 október 2017. [Online]. Available: <https://www.evogimbals.com/blogs/evo-blog/how-does-a-3-axis-gopro-or-dslr-gimbal->

work?srsltid=AfmBOorvBczEIqaMZhcBiq\_msvbJ5f39ZO5wl-6lc-  
ebMVzyROUqKVBI. [Hozzáférés dátuma: 16 december 2025].

- [9] STMicroelectronics, „STM32F767ZI,” STMicroelectronics, 2025. [Online]. Available: <https://www.st.com/en/microcontrollers-microprocessors/stm32f767zi.html>. [Hozzáférés dátuma: 19 június 2025].
- [10] „electronics.stackexchange.com,” [Online]. Available: <https://electronics.stackexchange.com/questions/70643/how-to-reverse-rotation-direction-of-stepper-motor>. [Hozzáférés dátuma: december 12 2025].
- [11] TRINAMIC Motion Control GmbH & Co. KG, *TMC2209 Datasheet*, Hamburg: Analog Devices, 2023.
- [12] ATMEL, „AVR446: Linear speed control of stepper motor,” Atmel Corporation, San Jose, CA 95131, USA, 2006.
- [13] „ASN Inverse Sine,” Algorithms Project, 1 augusztus 2003. [Online]. Available: <https://algo.inria.fr/esf/function/ASN/ASN.html>. [Hozzáférés dátuma: 18 szeptember 2025].
- [14] Eltra-encoder, „Magnetic rotary encoder. Working Principle,” 4 augusztus 2022. [Online]. Available: <https://eltra-encoder.eu/news/magnetic-rotary-encoder-working-principle>. [Hozzáférés dátuma: 13 december 2025].
- [15] AMS, *AS5x47U Motor Board*, AMS, 2021.
- [16] AMS, *AS5147U/AS5247U*, ams OSRAM Group, 2020.
- [17] D. Hajnal, „Shellminator,” 2025. [Online]. Available: [www.shellminator.org](http://www.shellminator.org). [Hozzáférés dátuma: 13 szeptember 2025].
- [18] Solomon Systech Limited, *SSD1305*, Solomon Systech Limited, 2008.

- [19] Newhaven Display International, Inc., *NHD-2.23-12832UCB3*, 2661 Galvin Court, Elgin, IL 60124 USA: Newhaven Display International, Inc., 2024.
- [20] P. Stumpf, *Áramkörtervezés - Mikrovezérlők és digitális áramkörök környezete és illesztése*, Budapest: Automatizálási és Alkalmazott Informatikai Tanszék, 2022.
- [21] „android.com,” 12 február 2025. [Online]. Available: <https://source.android.com/docs/core/interaction/accessories/headset/plug-headset-spec>. [Hozzáférés dátuma: 15 szeptember 2025].
- [22] J. Spencer és G. Alonso, „LTspice: Worst-Case Circuit Analysis with Minimal Simulations Runs,” 8 június 2017. [Online]. Available: <https://www.analog.com/en/resources/technical-articles/ltpice-worst-case-circuit-analysis-with-minimal-simulations-runs.html>. [Hozzáférés dátuma: 15 október 2025].
- [23] S. Bryan és R. Kattni, „Adafruit TDK InvenSense ICM-20948 9-DoF IMU,” Adafruit, 05 augusztus 2020. [Online]. Available: <https://learn.adafruit.com/adafruit-tdk-invensense-icm-20948-9-dof-imu/overview>. [Hozzáférés dátuma: 29 november 2025].
- [24] CUI inc., *P78B-1000*, Portland: CUI inc., 2025.
- [25] Oriental Motor U.S.A. Corp., „Oriental Motor,” Oriental Motor U.S.A. Corp., [Online]. Available: <https://www.orientalmotor.com/stepper-motors/technology/unipolar-bipolar-connections.html>. [Hozzáférés dátuma: 11 december 2025].

# Függelék

A függelék a következő dokumentumokat tartalmazza:

1. Gombpanel kapcsolási rajza
2. Numerikus billentyűzet kapcsolási rajza
3. OLED modul kapcsolási rajza
4. Alaplap kapcsolási rajza