

FELADATKIÍRÁS

A feladatkiírást a **tanszék saját előírása szerint** vagy a tanszéki adminisztrációban lehet átvenni, és a tanszéki pecséttel ellátott, a tanszékvezető által aláírt lapot kell belefűzni a leadott munkába, vagy a tanszékvezető által elektronikusan jóváhagyott feladatkiírást kell a Diplomaterv Portálról letölteni és a leadott munkába belefűzni (ezen oldal HELYETT, ez az oldal csak útmutatás). Az elektronikusan feltöltött dolgozatban már nem kell megismételni a feladatkiírást.



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

Sásdi András

KÉT TENGYELŰ FORGÓASZTAL GYORSULÁSMÉRŐ SZENZOROK KARAKTERIZÁLÁSÁHOZ

KONZULENS

Dr. Stumpf Péter Pál

BUDAPEST, 2025

Tartalomjegyzék

Összefoglaló	6
Abstract.....	7
1 Bevezetés	8
1.1 A munka indoka és célja	8
1.2 Specifikáció	8
1.3 Feladatok áttekintése.....	8
1.4 Projekt arculata	9
1.5 Szakirodalmi áttekintés, elméleti összefoglaló és tervezési irányelvek.....	9
1.5.1 MEMS szenzorok [1] [2] [3]	9
1.5.2 Biztonságtechnikai lépések [5] [6]	10
1.5.3 kéttengelyű forgatómechanizmusok	11
2 Modulok	13
2.1 Mikrovezérlő.....	13
2.2 Motorok	14
2.2.1 Motorvezérlő IC [8]	14
2.2.2 Léptetőmotorok sebességprofil vezérlése [9]	15
2.2.3 Trigonometrikus sebességprofil.....	16
2.2.4 Numerikus módszerek	19
2.2.5 Módszer implementálása beágyazott eszközre	22
2.2.6 Enkóderek [11]	23
2.2.7 Driver	24
2.3 Kezelőfelület.....	25
2.3.1 Állapotgép.....	26
2.3.2 Számítógépes vezérlés [13]	27
2.4 Kijelző [14] [15]	27
2.4.1 Nyomtatott áramkör	28
2.4.2 Driver	30
2.4.3 Grafikus kezelői felület.....	31
2.5 Gomb panel.....	31
2.6 Numerikus billentyűzet.....	34
2.7 Referenciaszenzor	36

2.7.1 Driver	37
2.8 Alaplap.....	38
2.8.1 Táp áramkör.....	38
2.8.2 Vészkapcsoló	40
2.8.3 Külső trigger jel	43
2.8.4 SPI lehallgató.....	44
2.8.5 EEPROM	45
2.8.6 EEPROM driver.....	45
2.8.7 Mikrovezérlő és további modularitási/bővítési megfontolások.....	46
3 Összefoglalás/eredmények értékelése.....	50
3.1 Eredmények	50
3.2 Javaslatok/Következtetések/Tanulságok.....	50
4 Irodalomjegyzék.....	51
5 Summary.....	53
Függelék.....	54

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. 04.

.....
Sásdi András

Összefoglaló

Abstract

The goal of the thesis is to design and implement a controlled two-axis rotary table that is adequate for the characterisation of MEMS accelerometers. As discussed in **chapter ...** 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 upon startup tares itself). Among others, the characterisation 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 characterisation engineers and shortening the runtime of these measurements

1 Bevezetés

1.1 A munka indoka és célja

A feladat MEMS gyorsulásszenzor karakterizálásához felhasználható vezérelhető kéttengelyes forgóasztal tervezése és megvalósítása. A MEMS szenzorok ahogy azt a ... **fejezetben is említtem** 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. A jelenlegi eljárás 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 karakterizálómérnök munkája könnyítése és a mérési idők csökkentése.

1.2 Specifikáció

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. Ezután körvonalazódott, hogy milyen modulok szükségesek az eszköz megvalósításához. A modulok 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 utolsó lépésre, mely az eszköz mechanikájának megvalósítása, már nem volt időm.

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] [2] [3]



1-1. ábra BMA508 [4]

A MEMS szenzorok a mérendő fizikai mennyiségeket valamilyen mikromechanika segítségével alakítják mérendő elektromos jellé. Ezt a jelet továbbítás előtt erősíti, esetleg digitalizálja a szenzor jeltovábbítás előtt. Az érzékel 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 és magas pontosságuk. A MEMS gyorsulásérzékelők népszerűsége vetekszik az egykor domináns piezoelektromos gyorsulásérzékelőkkel. Felhasználási területük közt szerepelnek az okos eszközök és a gépjárművek is, megbízhatóságukat alátámasztja, hogy ez utóbbiban biztonsági rendszerek részeként is felhasználásra kerülnek.

A MEMS gyorsulásérzékelő szenzorok felépítésükből és működési elvükből adódóan belső mechanikai feszültség következtében (mely akár hőmérsékletváltozás, illetve beforrasztáskor is kialakulhat) DC offsettel 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 nem csak a szenzor bekapcsolásakor, de az esetleges hőmérsékletváltozások miatt a szenzor működése közben is üzemel.

1.5.2 Biztonságtechnikai lépések [5] [6]

A mérőberendezés használatánál jelen van a személyi sérülés kockázata, melyet megszüntetni vagy csökkenteni kell. A személyi sérülés forrása a forgó elemek melyek lehetséges következménye az ütés, nyírás [5] (3.2 táblázat).

Ezeknek a kockázatoknak a leg egyszerűbb minimalizálása egy védőburkolat beépítése. A védőburkolatot tervezésekor a veendő szempontokat a következőkben részletezem. Feladata a veszélyforrás emberrel való érintkezését megakadályozni. Általános szabályai, hogy legyen szilárd, ne idézzon elő további veszélyeket, ne legyen egyszerűen megkerülhető vagy hatástalanítható, tartson a veszélyes tértől kellő távolságot, a munkafolyamat megfigyelését csak a lehető legszükségesebb módon korlátozza [6]. Továbbá fontos elvárás, hogy a védőburkolat ne lassítsa feleslegesen a munkát, 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 veszélyes 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.

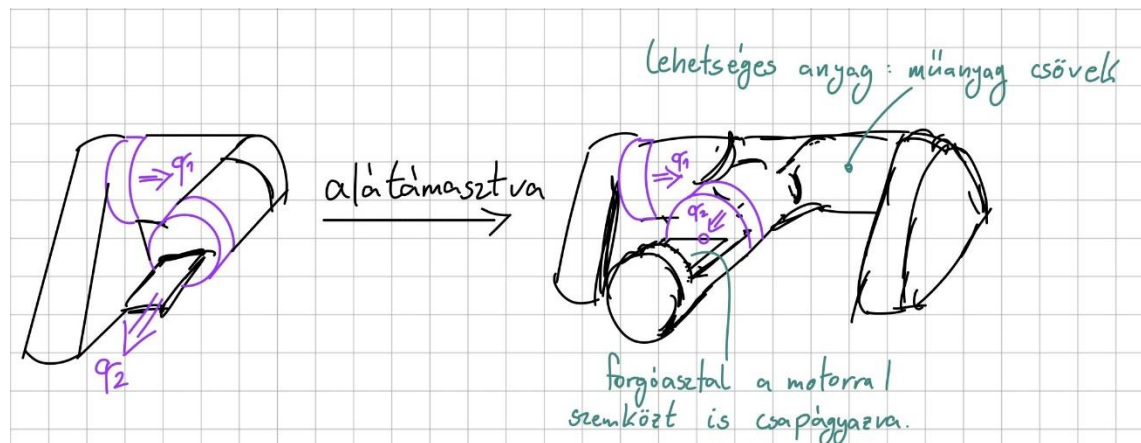
A biztonsági berendezések vizsgálhatóságát is figyelembe kell venni a tervezéskor. A berendezés által kiadott jelet, valamint a megszakítók állapotát is vizuális visszajelzéssel kell ellátni, például egy piros és egy zöld LED-del.

1.5.3 kéttengelyű forgatómechanizmusok

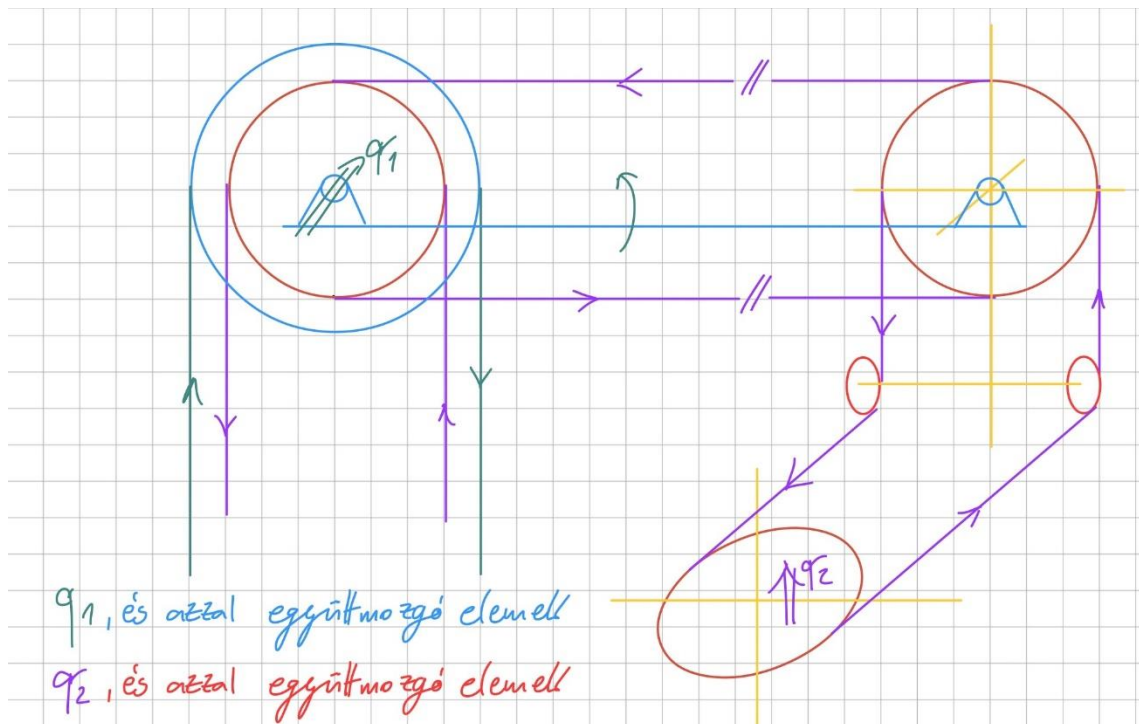
A kéttengelyű forgatómechanizmus alapvetően két egymásra merőleges forgatási tengelyből áll. Amennyiben a két tengely egymással párhuzamos lenne a gimbal lock jelenség lépne fel, ezért a forgatás csak a közös tengely mentén lehetne lehetséges.

A mechanizmusnak a háromtengelyű gyorsulásmérő szenzor bármely tengelyét a gravitációs térrel párhuzamossá kell tudnia tenni. Ezt minél jobb ismételhetőséggel, 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 rögzített tengelynek (q_1) a gravitációs térre merőlegesnek kell lennie.



1-2. ábra Kézi vázlatok az első megoldásra



1-3. ábra Kézi vázlat a második megoldásra

Alapvetően két megközelítés lehetséges, a mechanikailag leg egyszerűbb megoldás esetén egy rögzített és egy forgatott aktuátor felelős a tengelyek forgatásáért (1-1. ábra). Ebben az első esetben az egyik kihívás a forgatott motor jeleinek és tápvezetékeinek átvitele forgócsatlakozókon a mérőasztalra menő jeleken túl. A másik kihívás abból ered, hogy a forgatott asztalon kívül egy aktuátor tömegét és ebből adódó tehetetlenségét is gyorsítani kell az első motornak. A második kivitelezés esetén a motorok rögzítettek, így nem szükséges a második motornak sem mozgatása, sem annak jeleinek és tápvezetékeinek átvitele. Ebben az esetben viszont bonyolult fogaskerékes vagy (mint a vázlaton is látható) hajtósíjas mechanizmus szükséges (1-3. ábra).

2 Modulok

Az eszköz megvalósításához szükséges egységek fejlesztését úgynevezett deszkamodellen végeztem, ami a gyors prototípusgyártás egyik elterjedt módja. Az elektronikai alkatrészek próbapanelen vannak, nincs nyomtatott áramkör, próbanyákon vannak a csatlakozók kialakítva. Így a kiválasztott eszközök (pl. OLED modul) megfelelőségét tudtam ellenőrizni kis előkészülettel és megkezdhettem a modulokhoz szükséges kezelő szoftver megvalósítását. A fejlesztői környezet, amit a szoftver megírásához használtam az STM32CubeIDE. A választott STM mikrokontrollerre érdemes ebben a fejlesztői környezetben szoftvert fejleszteni.

Miután megbizonyosodtam a választott alkatrészek megfelelőségéről és összeállt a deszka modell, megterveztem a modulokhoz szükséges nyomtatott áramköröket. Az áramkörök tervezéséhez a KiCAD szoftvert használtam.

2.1 Mikrovezérlő

Az eszköz központi egysége, amin az ezt működtető szoftver fut az maga a mikrovezérlő. A projektjeimet (szakdolgozatom, önálló projekt stb.) általában STM32 mikrovezérlőt használok. Ezekhez a mikrovezérlőkhöz nagyon jól használható fejlesztői környezet tartozik, amiben meg vannak valósítva az eszköz perifériáit kezelő osztályok, így nem nekem kell megírni például az I2C drivert. 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 modullal vannak szerelve.

A következő feladat az STM32 mikrokontroller családból kiválasztani a megfelelőt. A szempontok minél több GPIO, legalább 3 SPI-csatorna, legalább 4 DMA, legalább 4 időzítő, Ethernet és USB kommunikáció. Ezen megfontolások alapján a választásom az STM32F767ZIT6-re esett. Ennek főbb tulajdonságai [7]:

- Arm® 32-bit Cortex®-M7 CPU
- Akár 2Mbyte flash memória
- Akár 512 Kbyte ram
- Általános felhasználású DMA: 16 csatorna

- Akár 18 időzítő
- Akár 168 IO port
- Akár 4 I2C
- Akár 4 USART
- Akár 6 SPI
- USB2.0
- Ethernet

2.2 Motorok

Abból, hogy a mérőeszköz gyorsulásmérő szenzorokat pozícionál következnek az elvárás, hogy a mérőeszköz miután felvette a kívánt pozíciót azt tartsa. Tehát nulla szögsebességnél is legyen nyomatéka, valamint oszcilláció lehetősége ne állhason fenn. 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, mikrostep felbontást 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.

2.2.1 Motorvezérlő IC [8]

A léptetőmotorok vezérléséhez, mint azt feljebb is említettem, a TMC2209 IC-t használtam. Az IC a szükséges passzív áramköri elemekkel ellátva is kapható, beépítve a SilentStepStick elnevezésű panelen. Ez egy elterjedt léptetőmotor meghajtó IC, 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 végállaskapcsoló nélkül is lehet alaphelyzetet találni a motorokkal, valamint képes kihagyott lépések detektálására. A lehető leg optimálisabb motorvezérlés érdekében a meghajtó kalibrálható. Ezeket a funkcionálisait egyelőre még nem használtam ki, de az alaplap tervezésekor úgy hoztam létre a kapcsolást, hogy az UART csatorna használható legyen.

Ez a motorvezérlő kétfázisú bipoláris léptetőmotorok vezérlésére alkalmas 2.8A maximális árammal. A pontos motorválasztásra akkor kerülhet sor, amikor a mechanikai terv már elkészült. Mivel a motor mérete nem befolyásolja az alapvető működési és vezérlési elvet, ezért ez nem jelent akadályt a fejlesztés során.

A motorvezérlő legfinomabb mikrolépés beállítása 1/32. Ez 1.8° felbontású léptetőmotorok esetén $\alpha_{min} = 0,05625^\circ$ -os felbontást jelent.

2.2.2 Léptetőmotorok sebességprofil vezérlése [9]

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.

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

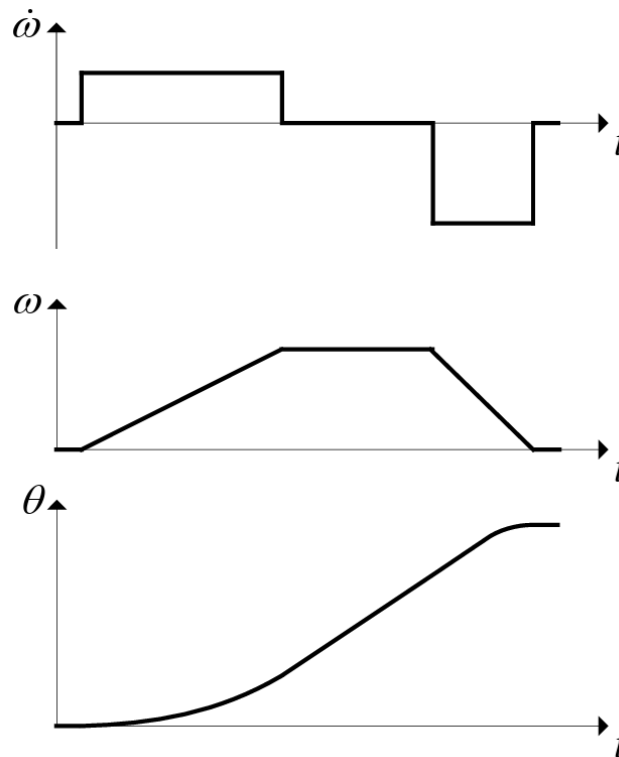
$$\omega = \frac{\alpha}{\tau} \quad (2.2)$$

Az egyenletekben a α az egy lépésimpulzus hatására tett elfordulást, n az impulzusok számát, Θ a mozgás során tett teljes elfordulást, τ az impulzusok között eltelt időt, ω pedig a szögsebességet jelöli.

Belátható viszont, hogy nem vezérelhető a motor oly módon, hogy álló helyzetében azonnal egy kívánt szögsebességhez tartozó τ 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 2-1. ábra 1-1. ábra is mutatja.



2-1. ábra Gyorsulás ($\dot{\omega}$), sebesség (ω), pozíció (θ) [9]

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

Ezen kívül a forrásban leírt megoldásban 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 forgás közben ne használjak számítási kapacitást.

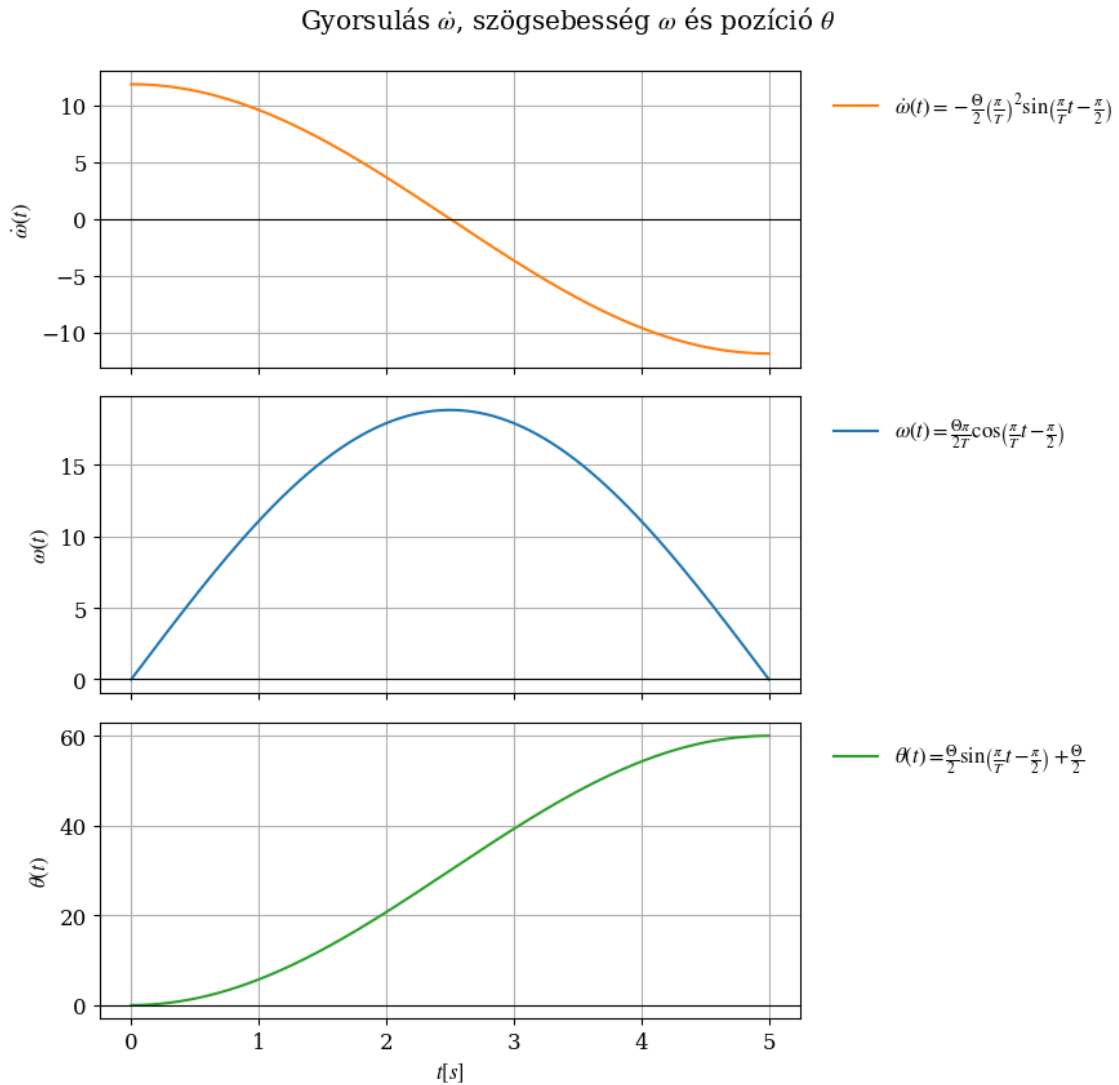
2.2.3 Trigonometrikus sebességprofil

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

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

Ennek deriváltja pedig az (2.4) függvény:

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



2-2. 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 először számoltam a sebességprofil, majd ezt kiértékeltem $\frac{\Theta}{\alpha} = N$ db helyen, majd az így kapott ω_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 ω_{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.

$$\theta(t) = \frac{\Theta}{2} \cdot \sin\left(\frac{\pi}{2} \cdot t - \frac{\pi}{2}\right) + \frac{\Theta}{2} \quad (2.5)$$

Kifejezve a $t(\theta)$ függvényt:

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

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

$$t_k = \left(\arcsin\left(\left(k \cdot \alpha - \frac{\Theta}{2}\right) \cdot \frac{\Theta}{2}\right) + \frac{\pi}{2} \right) \cdot \frac{T}{\pi}, \left\{ k \mid 1 \leq k \leq \frac{\Theta}{\alpha}, k \in \mathbb{Z} \right\} \quad (2.7)$$

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 ω_{max} között. A szögsebesség maximuma könnyen belátható, hogy $t = \frac{T}{2}$ pontban található. Az (2.4) függvénybe behelyettesítve ebben a pontban:

$$\omega\left(t = \frac{T}{2}\right) = \frac{\pi \cdot \Theta}{2T} \cdot \cos\left(\frac{\pi}{T} \cdot \frac{T}{2} - \frac{\pi}{2}\right) \quad (2.8)$$

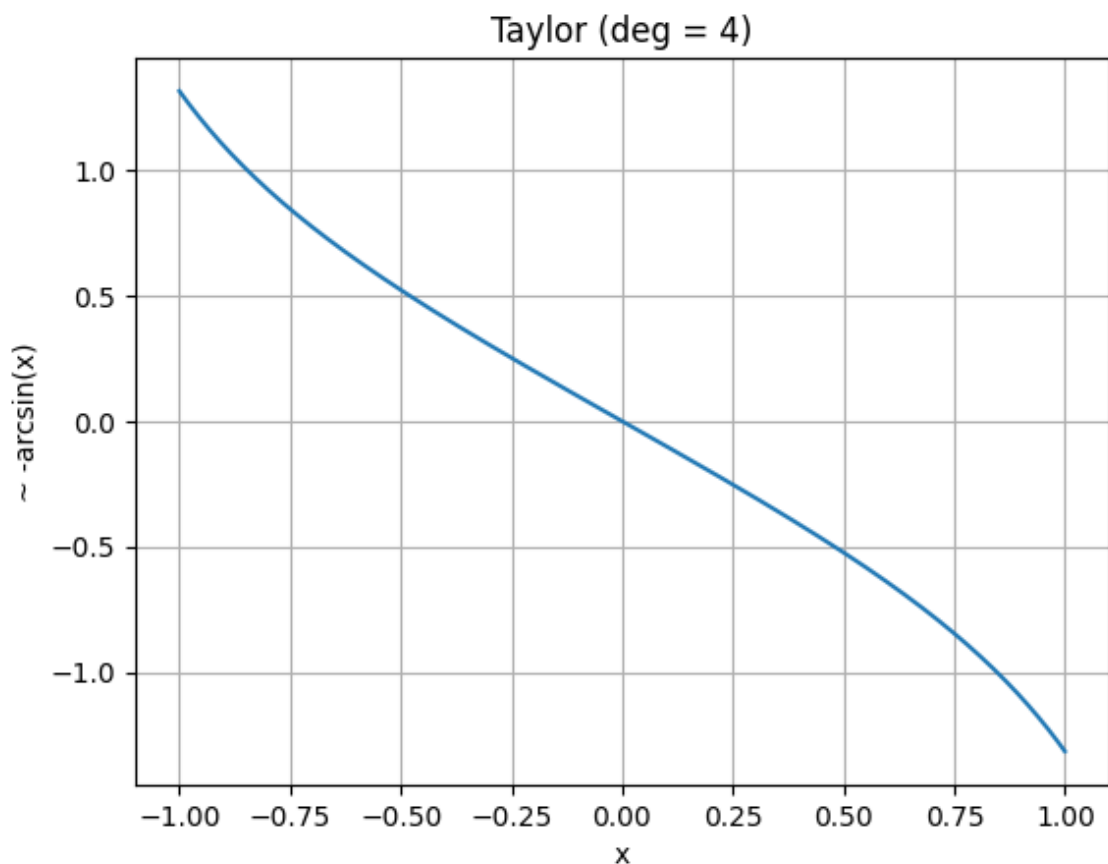
majd kiértékelve, az (2.9) összefüggést kapjuk. Ebből akár visszaszámolható a szükséges T a kívánt ω_{max} alapján.

$$\omega_{max} = \frac{\pi \cdot \Theta}{2T} \quad (2.9)$$

2.2.4 Numerikus módszerek

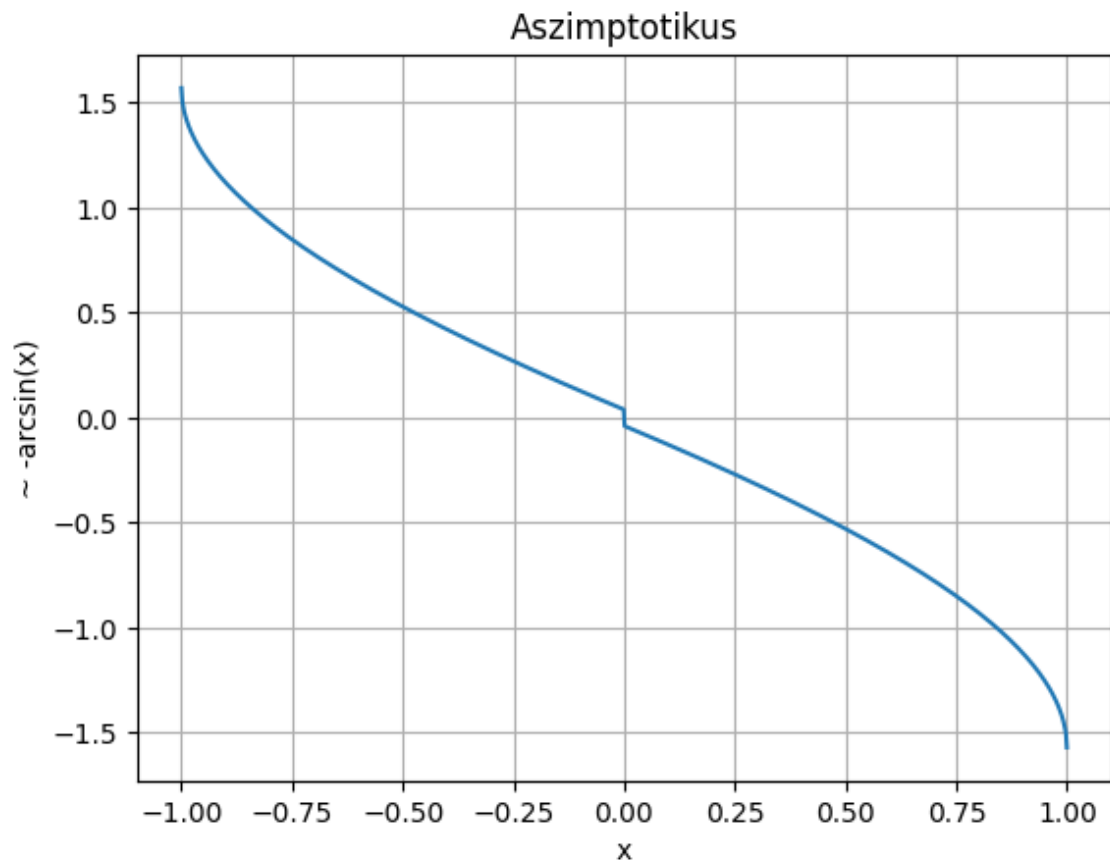
A következő felmerülő probléma, hogy az arcsin függvény számítása. Erre több megoldás is lehetséges: Taylor sor használata, aszimptotikus közelítés ± 1 -körül [10] vagy keresőtábla lineáris interpolációval. A numerikus módszereket először Python környezetben valósítottam meg, majd összehasonlítottam az eredményeket és ez alapján döntöttem, hogy melyiket implementáljam az eszközön.

Az összehasonlításnál negyedfokú Taylor sort használtam. Ezzel a $-\arcsin$ függvény közelítését a 2-3. ábra mutatja. Mint látható, nullától távolodva egyre jobban torzul a függvény közelítése.



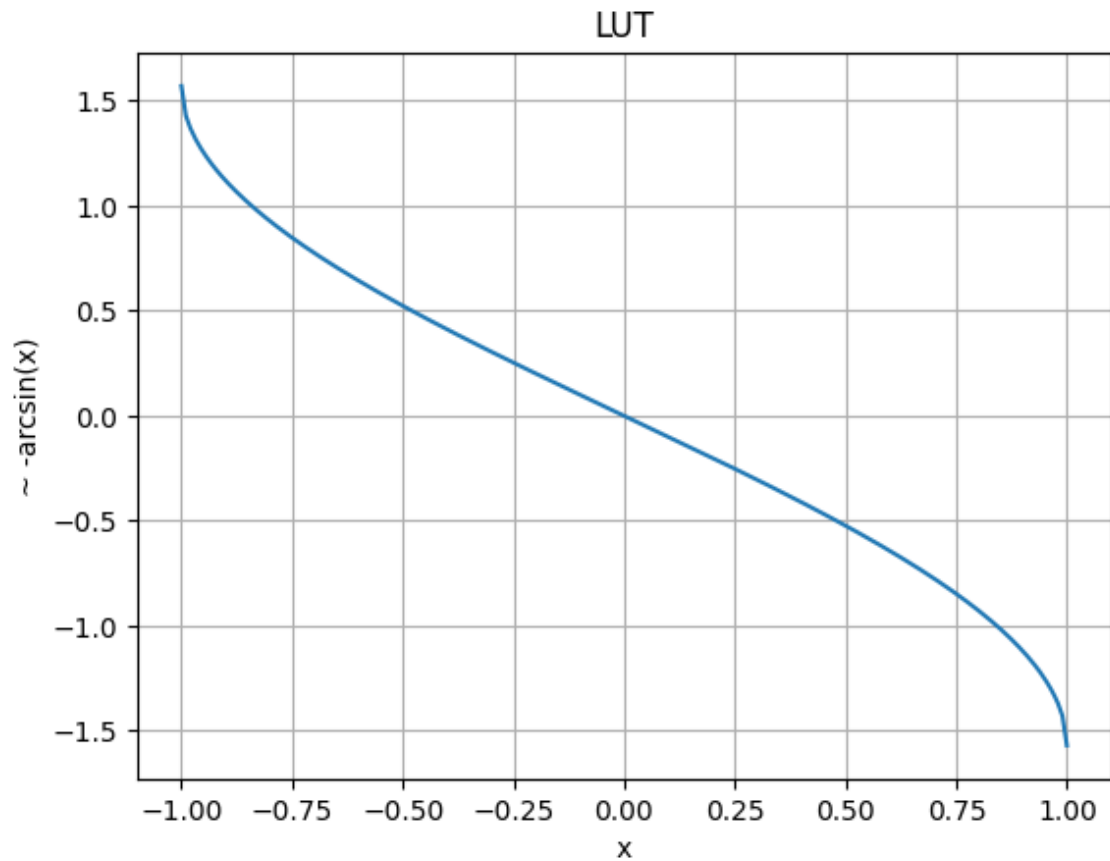
2-3. ábra $-\arcsin$ közelítése negyedfokú Taylor sorral

A következő vizsgált megoldás az aszimptotikus közelítés, mellyel 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 2-4. ábra mutatja. Látható, hogy itt törés van nullánál. Magasabb fokú kifejtésnél ez az effektus csökken, de a nem folytonosságból adódó hiba sosem tűnik el.



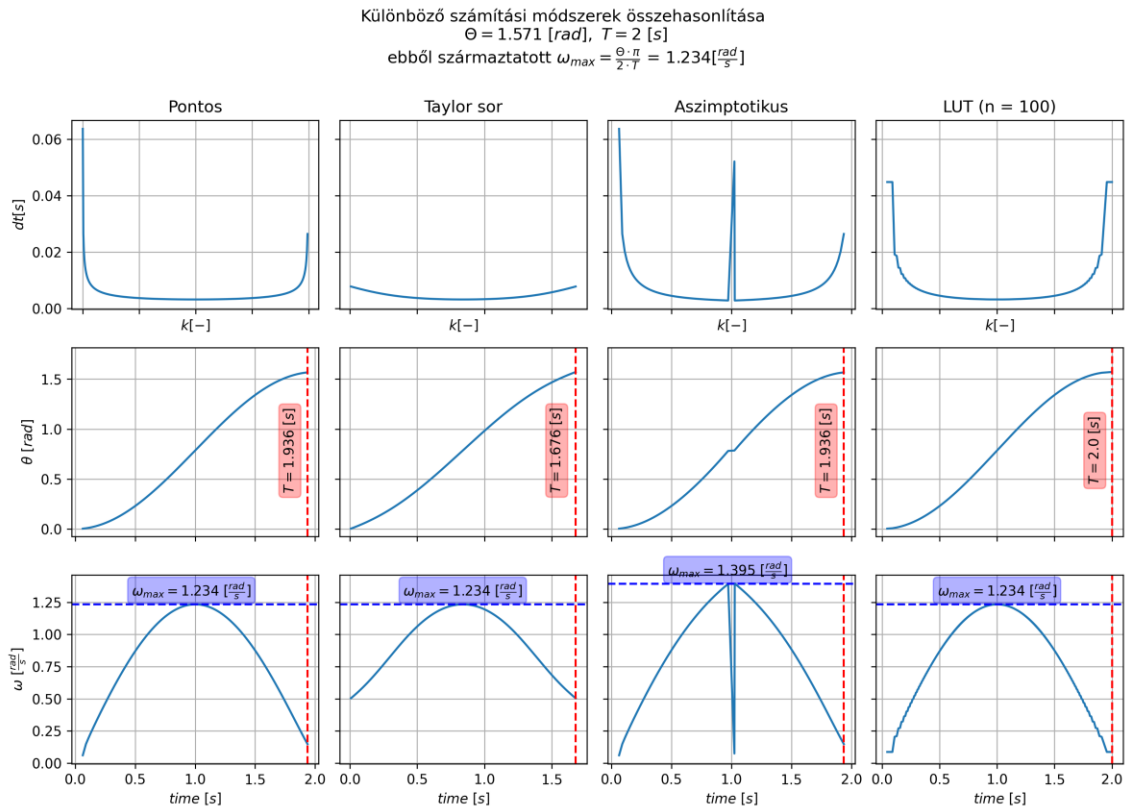
2-4. ábra Aszimptotikus közelítés

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, ezért valójában 200 elemes tábla felbontását adja. Az így kapott $-\arcsin$ közelítést a 2-5. ábra mutatja.



2-5. ábra -arcsin közelítése keresótáblával

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 2-6. ábra foglalja össze.



2-6. ábra közelítési módszerek

A megadott T beállási időt a Taylor soros megoldás közelítette a leg rosszabbul, 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 nem foglalkoztam. 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 érték 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

Mivel kis számokkal kellene számolni, könnyen bekövetkezhetne, hogy számolás közben nullává változik 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 figyelniem kellett.

2.2.6 Enkóderek [11]

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

Ezen megfontolások alapján mágneses alapú enkódert választottam. Ezek az eszközök a hall effektuson alapulnak, a tengelyre rögzített mágnes terének abszolút helyzetét képesek megállapítani. A pontos választott szenzortípus az AMS által gyártott „AS5147U” 14 bites mágneses elfordulásmérő szenzor. 14 biten az elméleti felbontása az eszköznek $\frac{360}{2^{14}-1} \cong 0.022 \left[\frac{\text{deg}}{\text{LSB}} \right]$, ami kisebb, mint az elérhető α_{\min} . A szenzor beszerezhető beépítve az AMS által gyártott „AS5x47U Motor Board” panelbe. Ez a kész eszköz különböző tengelyátmérőkre rögzíthető, megfelelően polarizált mágnesekkel érkezik, valamint különböző méretű motorokkal kompatibilis rögzítőfuratokkal el van látva.



2-7. ábra AS5x47U Motor Board [12]

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 kiolvasható a szenzorból. További előnye, hogy diagnosztikai adatok is kiolvashatóak, így a megfelelő beépítés ellenőrizhető.

2.2.7 Driver

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

Az eszközök driverét C++ nyelven írom, objektum orientált módon a könnyű használat, modularitás és bővíthetőség érdekében. Külön header fájlba kigyűjtöttem az eszköz regisztereit, majd létrehoztam az eszközt kezelő osztályt. Jelenleg csak a számomra szükséges, leg alapvetőbb funkciókat valósítottam meg.

A kommunikációhoz létrehoztam az üzeneteket küldő- és fogadó metódusokat a HAL könyvtár SPI függvényeinek segítségével. Ezekre a metódusokra azért volt szükség, mert 16 az eszköznek a 16 bites kommunikációját használtam, amihez CRC számítás is szükséges, valamint így a hibakezelésre is több lehetőség van.

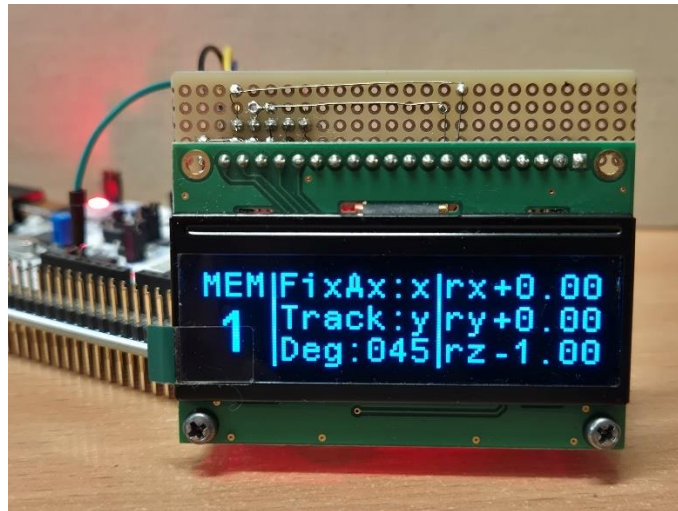
Az inicializáló metódust a konstruktortól külön hoztam létre, így annak logikai visszatérési értéke jelezheti az inicializálás sikerességét. Az inicializálás során az eszköz szűrőjének K értékének minimumát az adatlapban megadottak szerint a kis szögsebességeknél használatosan 0-ra állítom, mivel a pozíció 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 még nem implementáltam hibakezelést, de a jövőben implementálni fogom, hogy a felhasználó pontosabb visszajelzést kapjon a hiba típusáról.

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 módszert is. Ez az érték a mágneses tér erősségével arányos.

2.3 Kezelőfelület



2-8. ábra OLED modulon megjelenített kezelőfelület (OLED próbapanelen, fejlesztés alatt)

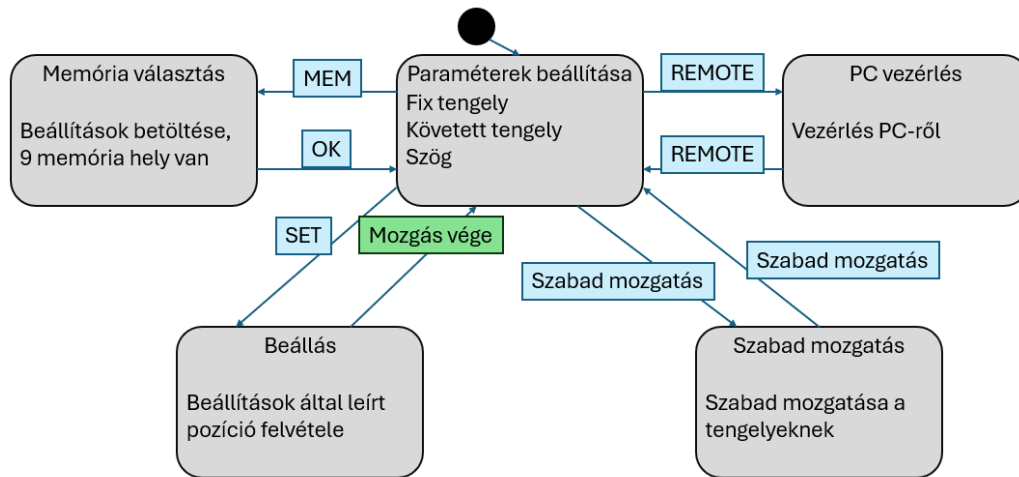
Mivel cél, hogy az eszköz számítógép nélkül is intuitívan használható legyen meg kellett terveznem a kezelőfelületét. Céлом az volt, hogy a laboratóriumi tápegységek kezelőfelületével hasonlóan egyszerű és egyértelmű interfészt alkossak. Itt is (akárcsak a labortápoknál is) a felhasználó kiválaszt 2 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 szeretnénk 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



2-9. ábra állapotgép

Az eszköz vezérlését egy egyszerű állapotgéppel valósítottam meg (2-9. ábra). A felhasználó a gép bekapcsolásakor a „Paraméterek beállítása” állapotba lép (2-8. ábra). Ebben 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 mozgató 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 objektum orientáltan. A switch-case megoldást csak a beállítások állapoton belüli alállapotgép kezelésére használtam.

Az állapotokat egy közös ősosztályból származtatom le. 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 kezelhetőek az egyes állapotok és azok tranziens viselkedései.

Az állapotgép maga is egy külön osztály. Az állapotgép konstruktorának meg kell adni a már le példányosított állapotok pointerait. Ez mind statikusan történik, ezért nem áll fenn a memóriaszivárgás veszélye. Az állapotgépnek a konstruktorán kívül 2 további kettő, egy futás (run) és egy állapotváltás (change state) függvénye van. Az osztály változói: minden típusú állapot pointer típusú változók, a jelenlegi állapotot szamon tartó jelenlegi (current) állapotra mutató ősoosztály 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 működik.

2.3.2 Számítógépes vezérlés [13]

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.

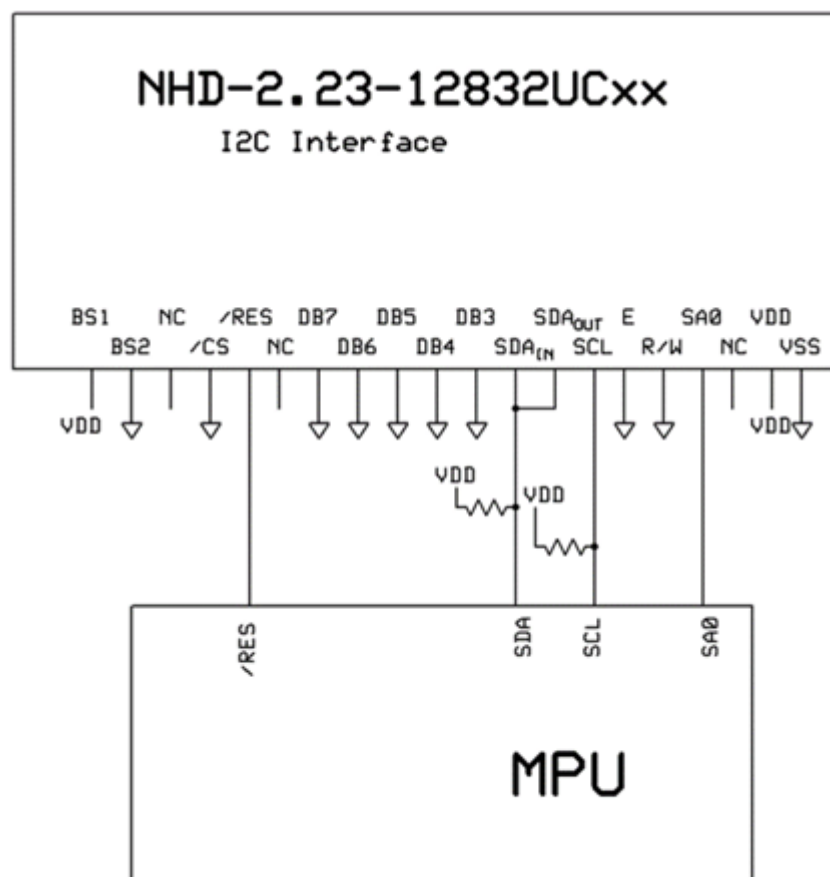
2.4 Kijelző [14] [15]

A kijelző kiválasztásánál a következő szempontokat vettem figyelembe: képpontonkénti kezelhetőség (például nem 8 szegmenses), kontraszt, megfelelő méret, I2C vagy SPI támogatása, jó dokumentáció elérhetősége, professzionális megjelenés.

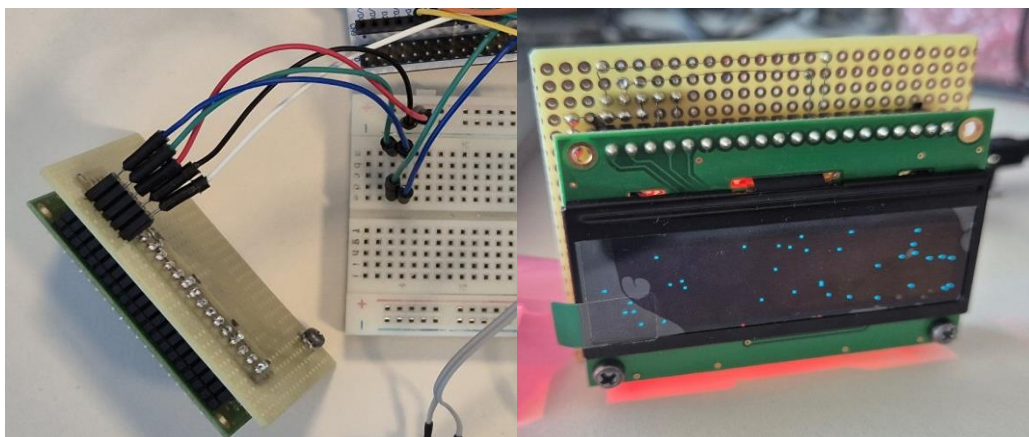
A fenti elvárásaimat tökéletesen teljesítette egy, a Newhaven Display International által gyártott 128x32 felbontású kék OLED panel (NHD-2.23-12832UCB3). Ez az OLED panel az SSD1305 típusú vezérlő IC-vel van szerelve. Ez a vezérlő IC támogat párhuzamos és soros kommunikációt is, és maga az OLED panel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

2.4.1 Nyomtatott áramkör

Az OLED panel egy nyomtatott áramkörre szerelve kapható, amin 20 csatlakozási furat 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 (2-10. ábra).



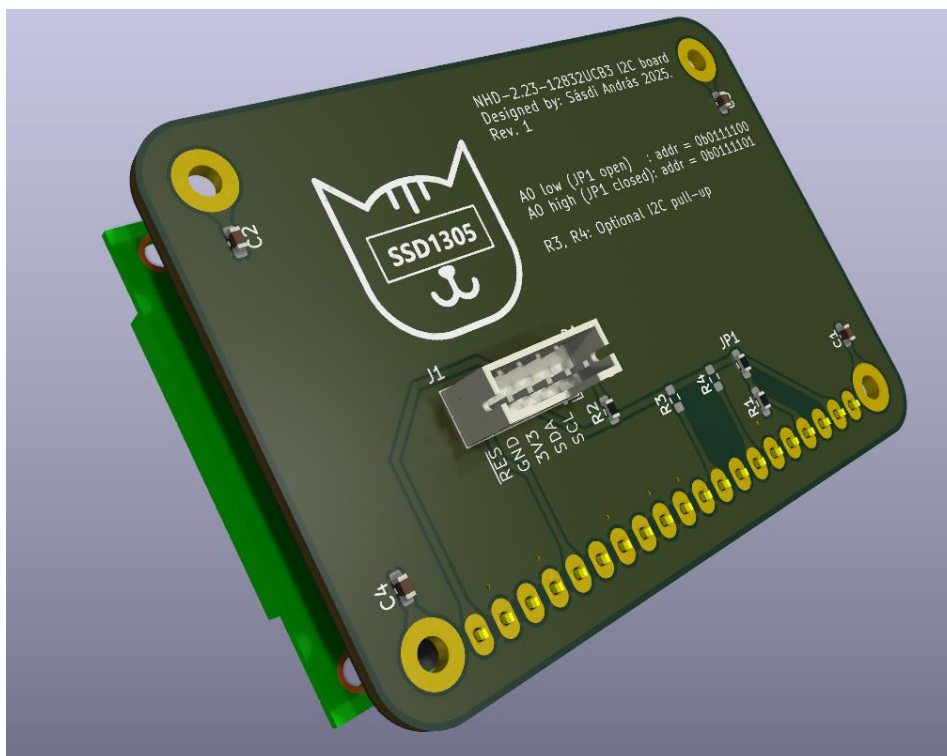
2-10. ábra OLED panel adatlapja I2C konfigurációhoz [15]



2-11. ábra OLED modul próbapanelen

Amíg a panel funkcionalitását 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 vezetékkel 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).



2-12. 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

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.

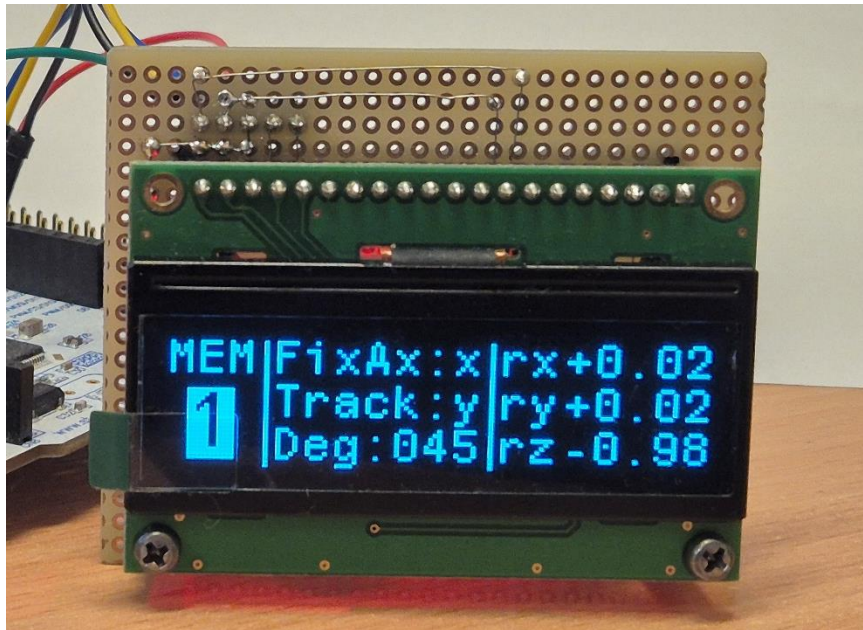
Ezt a drivert is C++ nyelven valósítottam meg, objektum orientált módon. A driver megírásához a következő linken elérhető MIT licenszes driverből merítettem ötleteket néhány megoldásra, a betűk bitmapjeit viszont teljes egészében innen másoltam (<https://github.com/4ilo/ssd1306-stm32HAL>). A driver megírása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

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 OELED kezelő objektum metódusai a parancsok megvalósításai, illetve a kirajzolás logikája.

Az inicializáló függvény itt is a konstruktoról 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 osztálynak rendelkezik egy bitmap puffer tömbbel. A felhasználó ebbe töltheti bele 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, ez egy jövőbeli potenciális fejlesztési lehetőség, ahogyan a jobb hibakezelés is.

2.4.3 Grafikus kezelői felület



2-13. ábra Grafikus kezelői felület, memóriaválasztás állapotban

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

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.

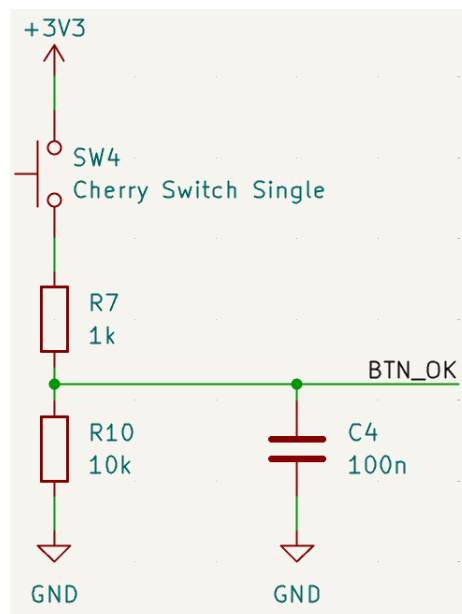
2.5 Gomb panel

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 érik biztosra, hogy lenyomták-e a gombot, van, hogy túl erősen nyomják és összességében „olcsó” hatást adnak az eszköznek. Ezzel szemben a mechanikus gombok, amiknek 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

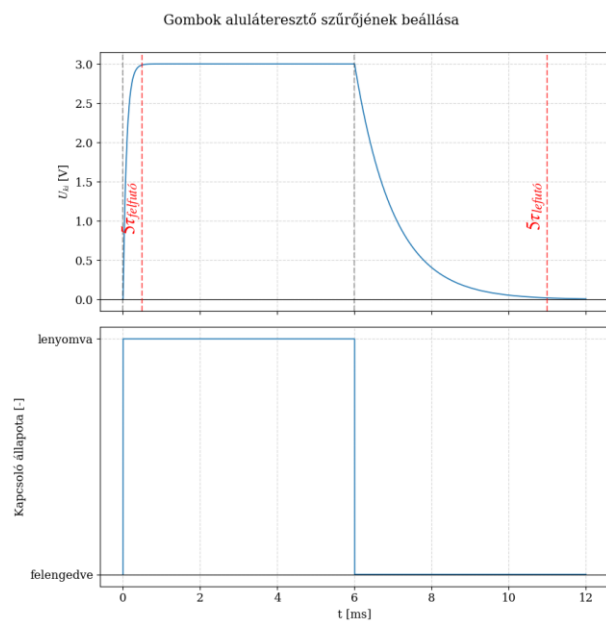
gombok egyértelmű aktivációs ponttal és érezhető kattanással és hanggal rendelkeznek. A lenyomásról a felhasználó biztos visszajelzést kap.

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 [16]. 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 RC tag, illetve ennek ötvözése egy hiszterézises kapcsolóval (schmitt triggerrel). Én schmitt trigger nélküli, egyszerű RC tagos megoldást választottam (2-14. ábra), ami lényegében egy alul áteresztő szűrő. Első iterációban ezt 0.5ms-os feltöltési és 5ms kisütési idővel terveztem. Ezt az áramkör megérkezésekor tesztelni fogom, hogy megfelelő-e és ez alapján szükség esetén módosítom az ellenállás és kondenzátor értékeket.



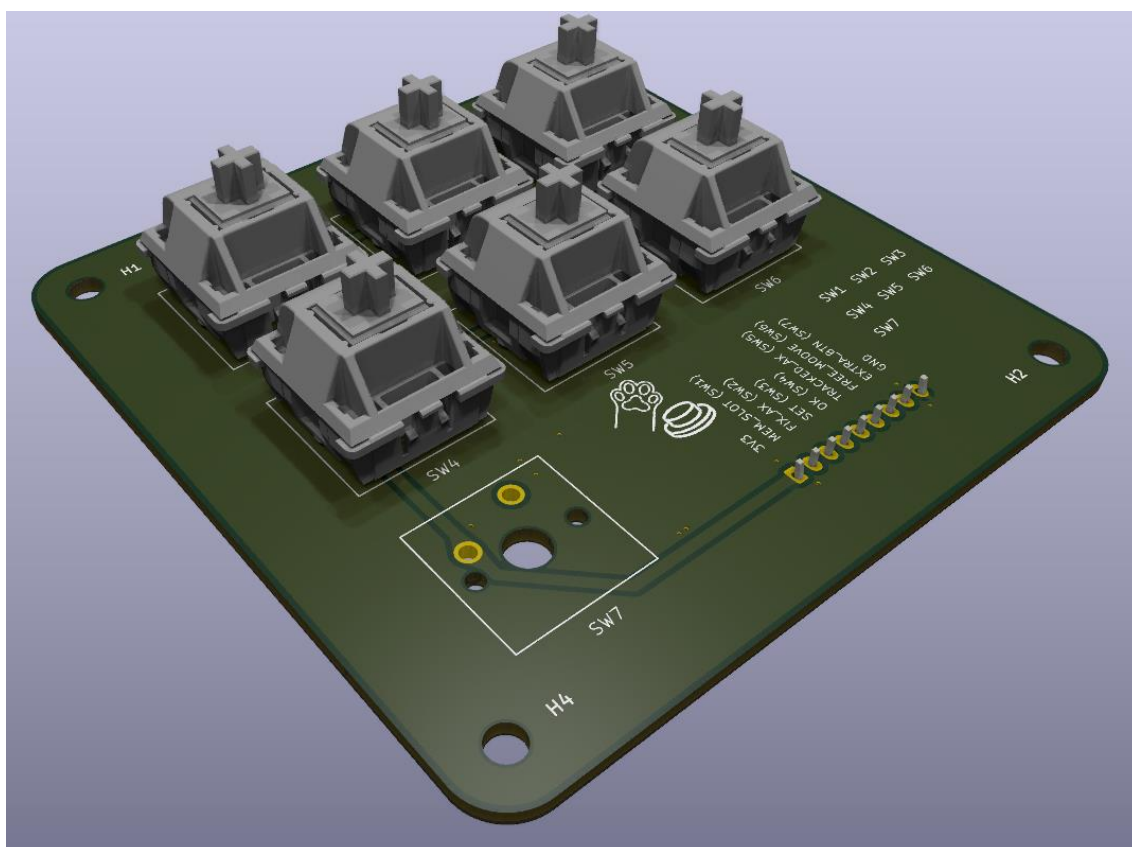
2-14. ábra Nyomógomb pergésmentesítése

A szűrő választ a gomb lenyomására és felengedésére (pergést nem szimulálva) a 2-15. ábra mutatja



2-15. ábra RC kör beállása

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



2-16. 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 mozgatás állapotban a mozgatás is ezzel valósítható meg. A számlap gombjaival szemben is a korábban támasztott elvárásaim voltak, ezért itt is Cherry Blue nyomógombokat használtam.

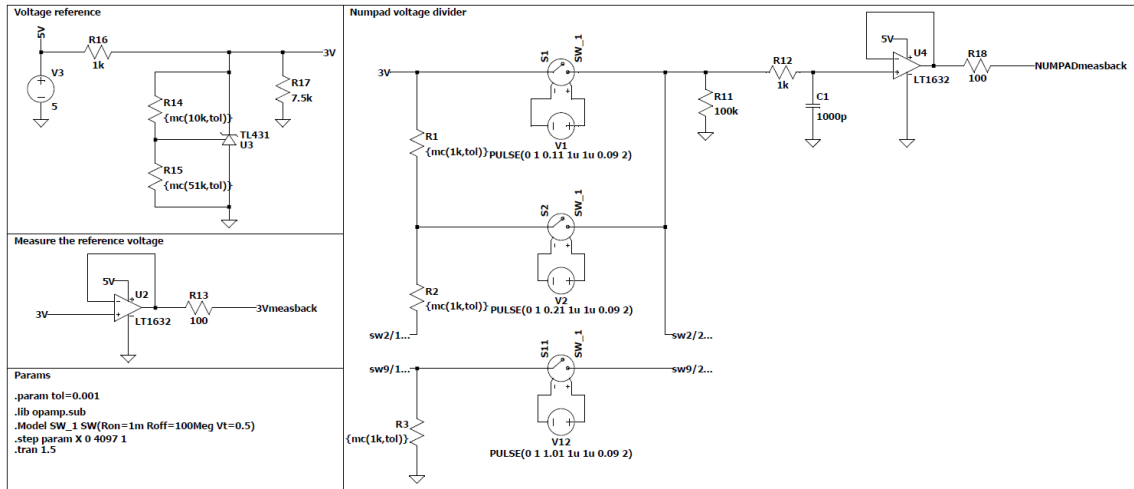
Mivel az eszköznek sok perifériát kell kezelnie és szempont a bővíthetőség, Minimalizálni szerettem volna a számlap kezeléséhez felhasznált GPIO-k számát. Egy lehetséges megoldás erre, és amivel tovább haladtam az egy analóg megközelítés. A kimeneti feszültség a lenyomott gombbal arányos. Ezt a megoldást a vezetékes fülhallgatók gombjainak kivitelezéséből merítettem [17].

A feszültségosztós megoldásnál 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.

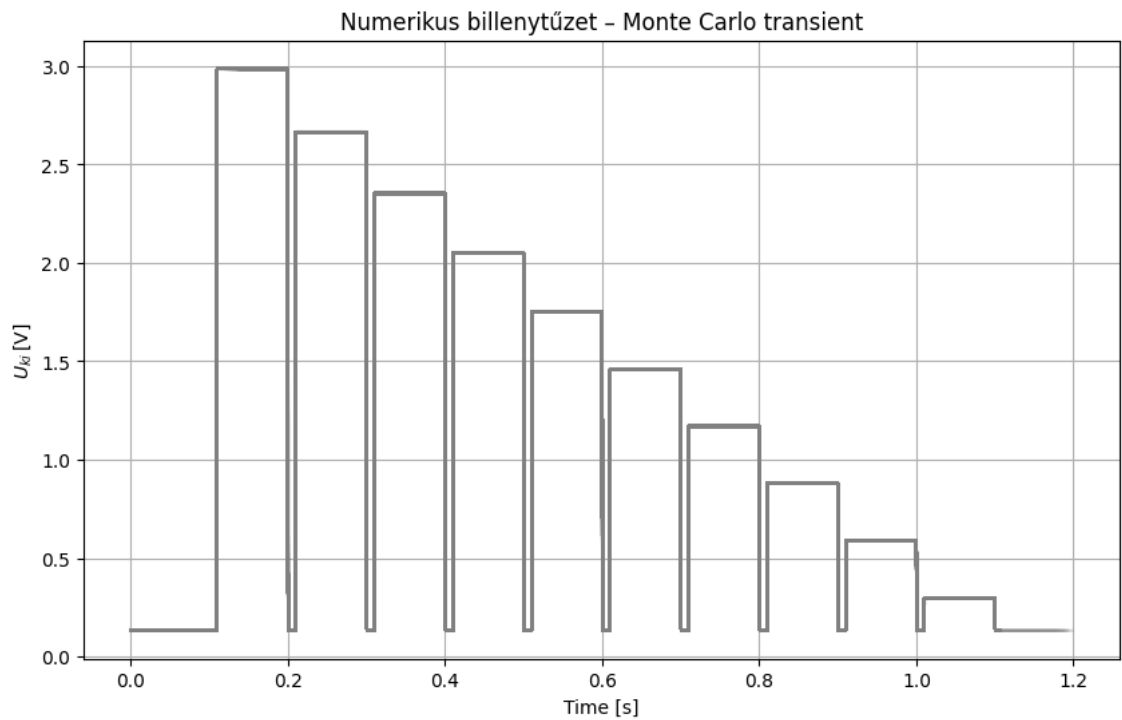
Mivel az eszköz tervezése során a modularitás és rugalmasság központi szempont a két megközelítést ötvöztem. A bemeneti 5V-ból 3V-os referenciafeszültséget hozok létre, de mind a bemeneti, mind az osztott feszültség vissza van vezetve a mikrovezérlőhöz. A feszültségek visszamérését követőerősítőkön keresztül vezetem vissza, de erősítésnek a lehetőségét kialakítottam: a visszacsatolás 0R 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 értéknek 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 (2-17. ábra), Monte Carlo legrosszabb esetek elemzéssel. Ennek a lényege, hogy végig iterálja a tűréssel ellátott passzív

komponenseknek az adott tűréssel megadott leg rosszabb értékeit, így minden legrosszabb kombinációt lemodellez. A tűréssel ellátott elemek a feszültségreferenciát beállító ellenállások, illetve a feszültségosztó ellenállásai.



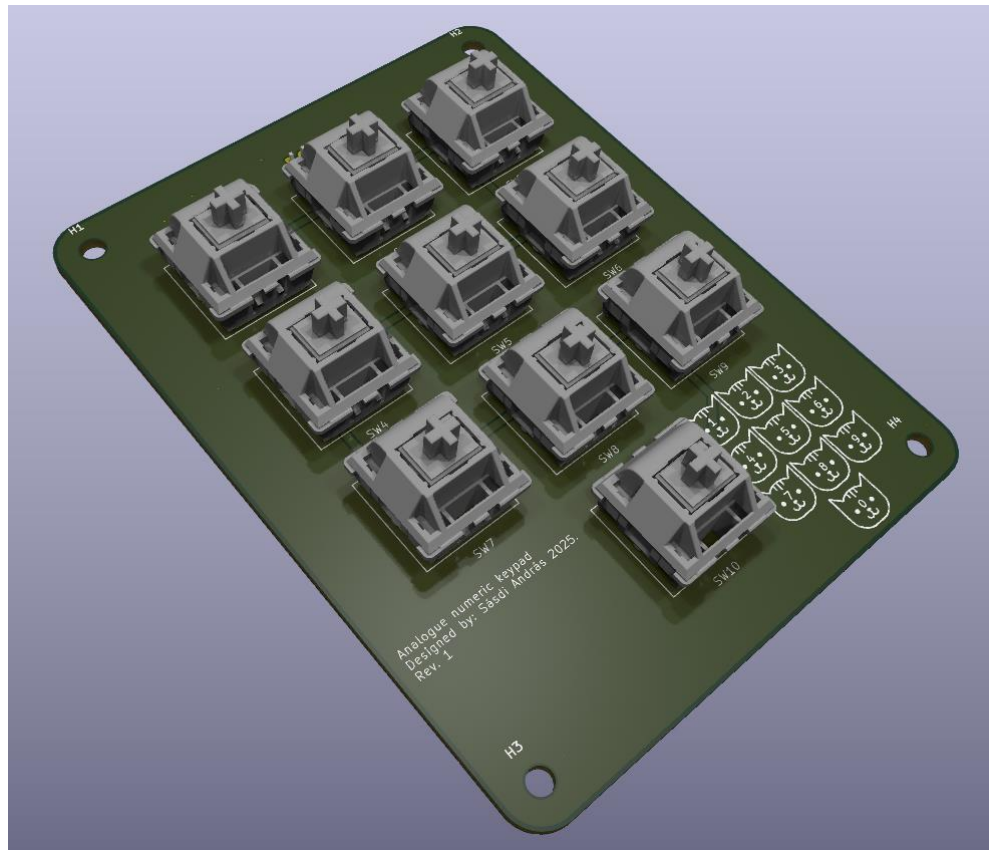
2-17. ábra numerikus billentyűzet LTspice modellje



2-18. ábra Monte Carlo szimuláció eredménye

Ahogy azt a 2-18. ábra mutatja, a 0.1%-os tűréssel rendelkező ellenállások által okozott szórás a lépések magasságánál nagyságrendekkel kisebb, az ábrán nem is értelmezhető.

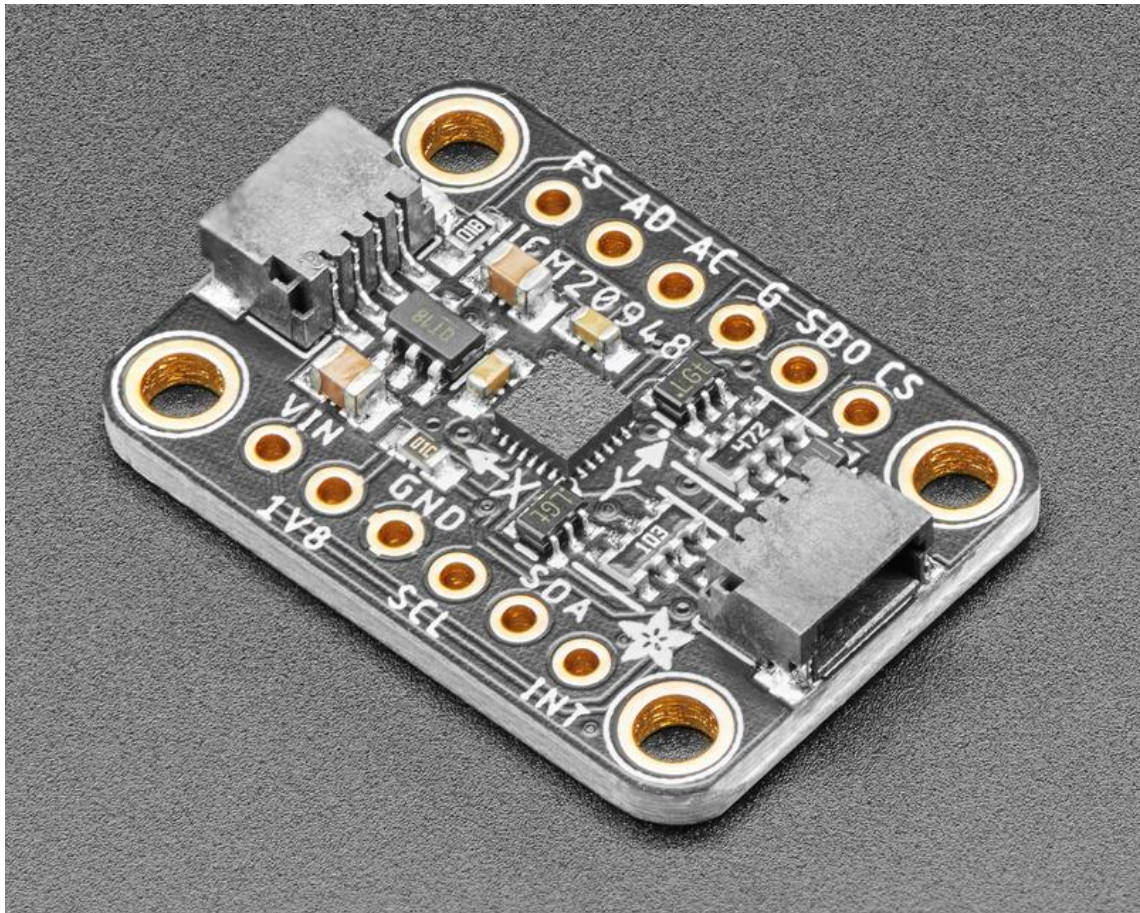
Az áramkör tervezésénél a visszamérés előtt a feszültségosztó kimenetét egy alul áteresztő szűrőn keresztül vezetem. Ennek az értékei egyelőre 1k ohm és 1nF, de amennyiben szükséges, a panel viselkedésének kimérése után ez módosítható.



2-19. ábra Numerikus billentyűzet PCB panelje

2.7 Referenciaszenzor

A mért szenzor pozíciójának visszamérhetőségét egy referenciaszenzorral biztosítom. A választott szenzor a TDK által gyártott ICM 20948 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 emellett az adott szenzor mellett az volt továbbá, 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. Sajnos nem találtam számomra megfelelő kezelő könyvtárat a szenzorhoz, ezért nekem kellett azt megvalósítanom.



2-20. ábra Adafruit TDK InvenSense ICM-20948 9-DoF IMU [18]

2.7.1 Driver

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.

Ez a 9tengelyű szenzor lényegében egy 6 tengelyű gyorsulásmérő és giroszkóp szenzorból és egy AK09916 típusú 3 tengelyű iránytűből áll. Az utóbbi integrálva van az IC-be és a belső I2C buszon keresztül lehet vele kommunikálni. Sajnos nem sikerült megbízhatóan kommunikálnom ezzel a belső szenzorral. Mivel jelenlegi felhasználáshoz ez nem szükséges nem töltöttem időt ennek megoldásával és nem implementáltam az iránytű kezelését.

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, amiktől örökööl 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 örökööl.

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

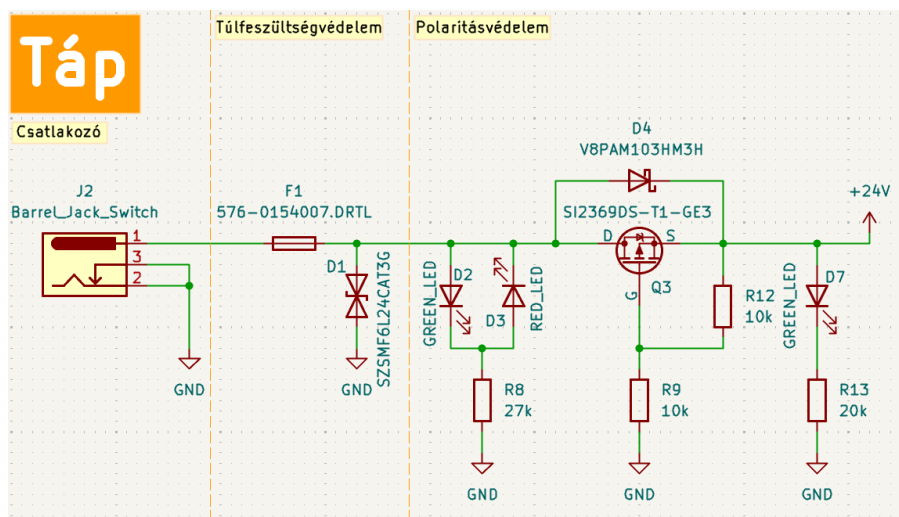
2.8 Alaplap

Az eszköz leg komplexebb áramköre, ami a modulokat összeköti az alaplap. A tervezés közben a központi megfontolás a könnyű javíthatóság, átalakíthatóság és modularitás volt.

2.8.1 Táp áramkör

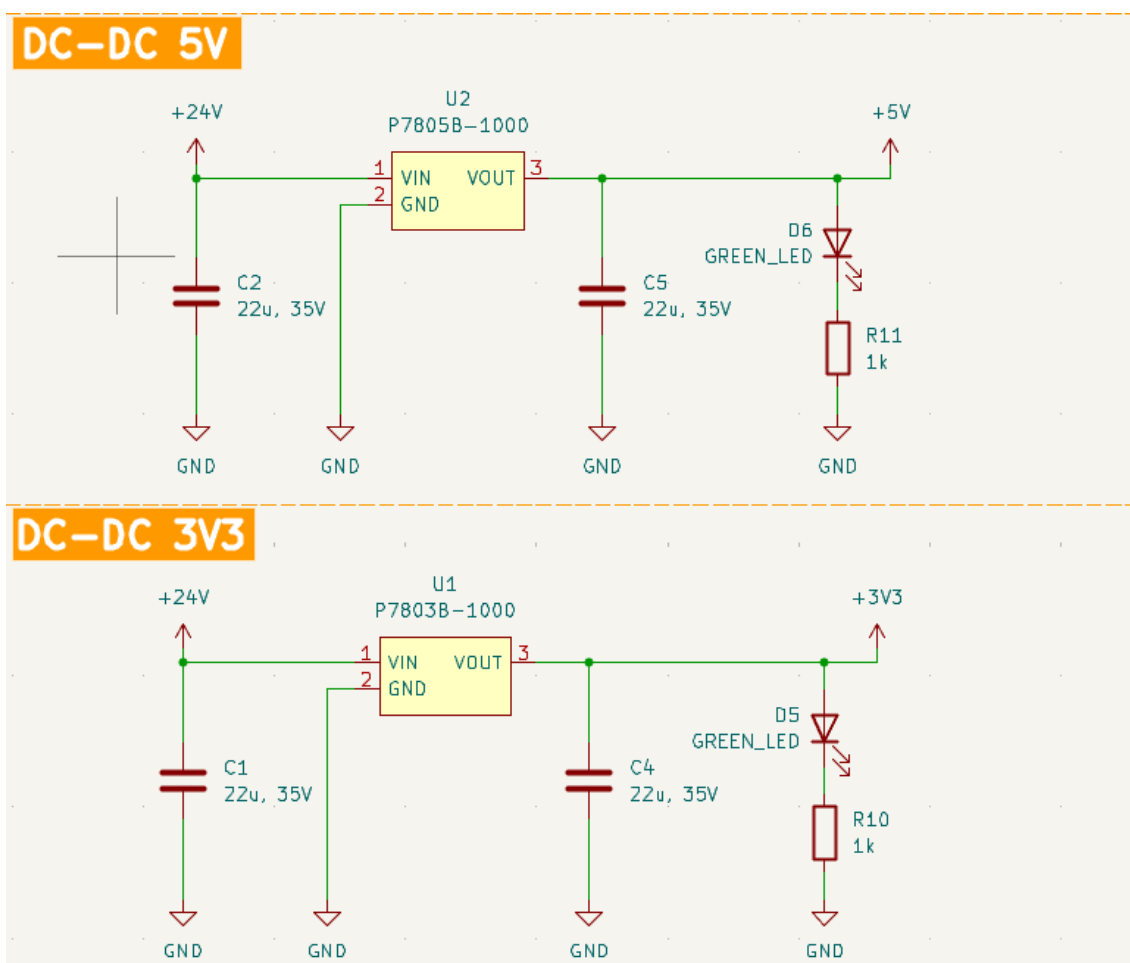
Az alaplapot 24V-os tápellátással terveztem, mivel a motorvezérlők akár 28V-os motor tápfeszültség kapcsolására is alkalmasak.

A túlfeszültség elleni védelmet egy két irányú zénerdiódával és olvadóbiztosítékkal valósítottam meg. A polaritás védelemhez MOSFET tranzisztort alkalmaztam, ami a megfelelő polaritásnál nyit csak ki. Ezzel párhuzamosan egy teljesítménydiódát is bekötöttem, hogy nyitásnál a nagy áram ne a MOSFET test diódáján folyjék. A megfelelő polaritást és a tápfeszültség jelenlétét LED-ek jelzik vissza.



2-21. ábra Tápfeszültség előkészítése

A panelen szükség van továbbá 3.3V, illetve 5V feszültségre is. Ezeket a feszültségeket kapcsolóüzemű DC-DC buck konverterekkel állítom elő (2-22. 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 P7803B-1000, míg 5V-ot P7805B-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, alacsony és alacsony kimeneti zajjal rendelkeznek. További előnyük, hogy pusztán 2 (egy be- illetve egy kimeneti) kondenzátorra van szükségük a működéshez. [19]



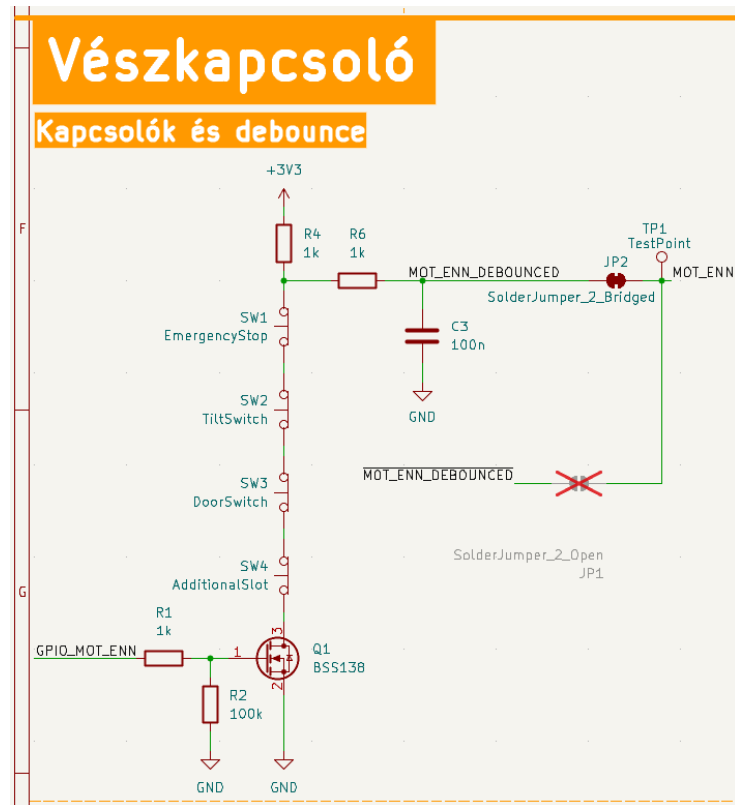
2-22. ábra Alacsonyabb feszültségek előállítása

2.8.2 Vészkapcsoló

Ahogy azt a Biztonságtechnikai lépések fejezetben is említettem az eszközt el kell látni vészleállító berendezéssel, ami az aktiátorokat deaktiválja. 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.

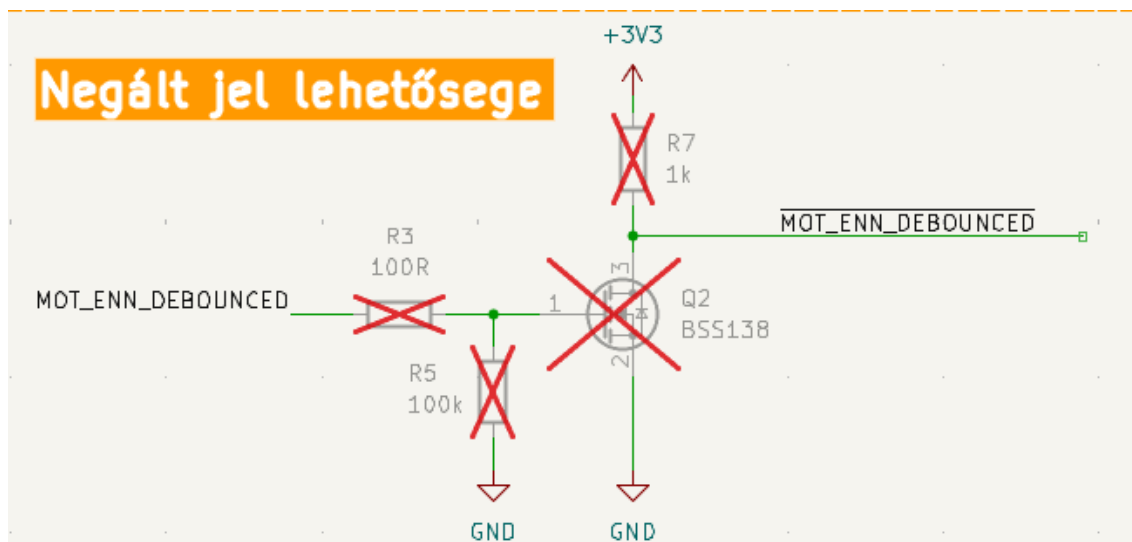
A vészleállító jel legalább 3 helyről érkezhethet (vészleállító gomb, burkolat nyitás kapcsoló, döntéskapcsoló). Ezeken kívül a modularitás és fejleszthetőség érdekében egy további kapcsoló lehetőségével és szoftveresen vezérelhető jellel is kiegészítettem a vészleállító kapcsolást. Alapesetben zárt 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 2-23. ábra mutatja.



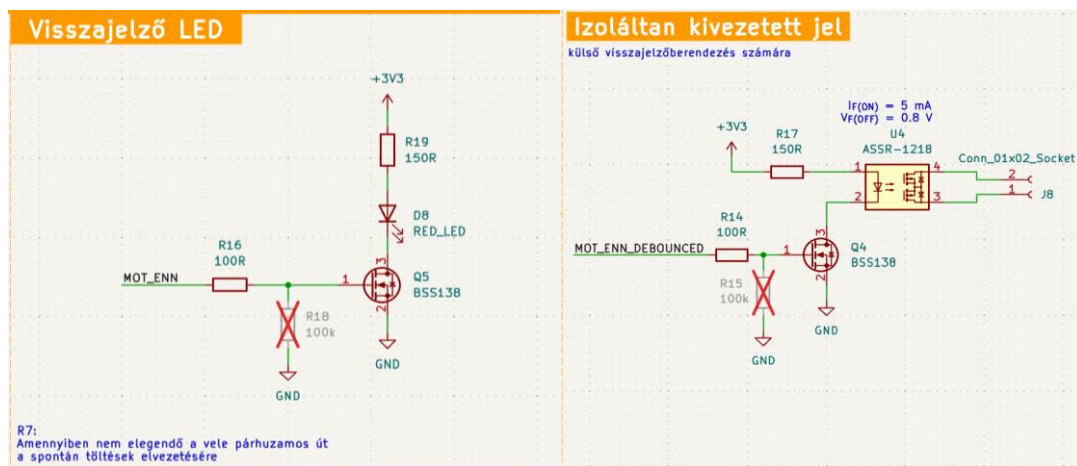
2-23. ábra Leállító kapcsolók jelei

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 2-24. ábra mutatja, a piros x-ek a szoftverben a nem beépítendő alkatrészeket jelölik.



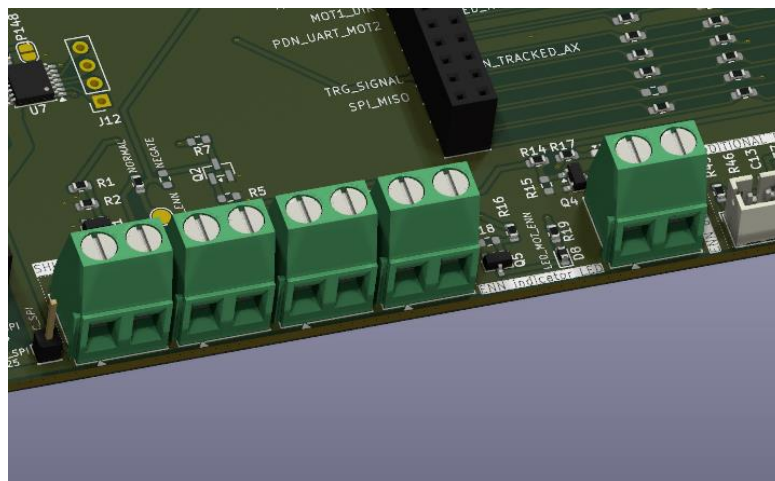
2-24. ábra negált jel létrehozásának lehetősége

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.



2-25. ábra Visszajelző LED és külső jelet kapcsoló szilárdtest relé

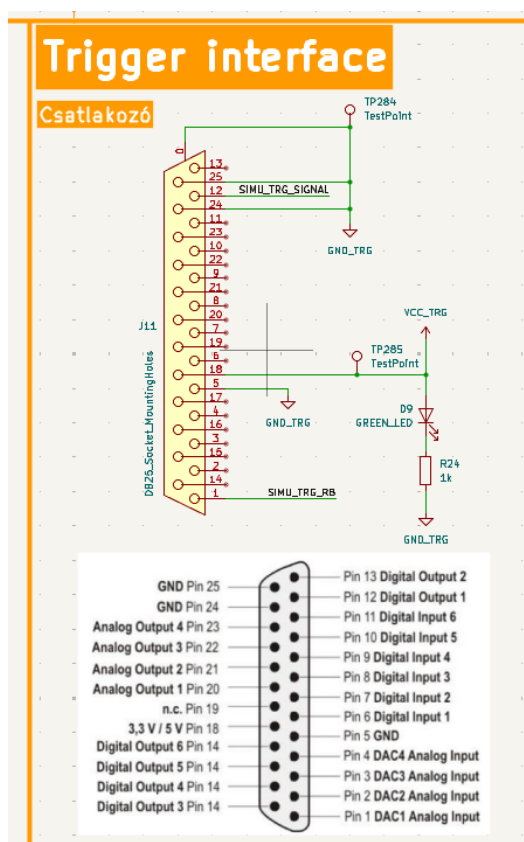
A kapcsolók jeleit és a külső jelet is csavaros csatlakozóval lehet bevezetni (2-26. ábra). Így sokféle vezetékezési megoldás egyaránt beköthető az alaplapba. Amennyiben ez hosszú távon rossz megközelítésnek bizonyul, nem nehéz ezeket más csatlakozóra cserélni.



2-26. ábra Jelek csatlakozói

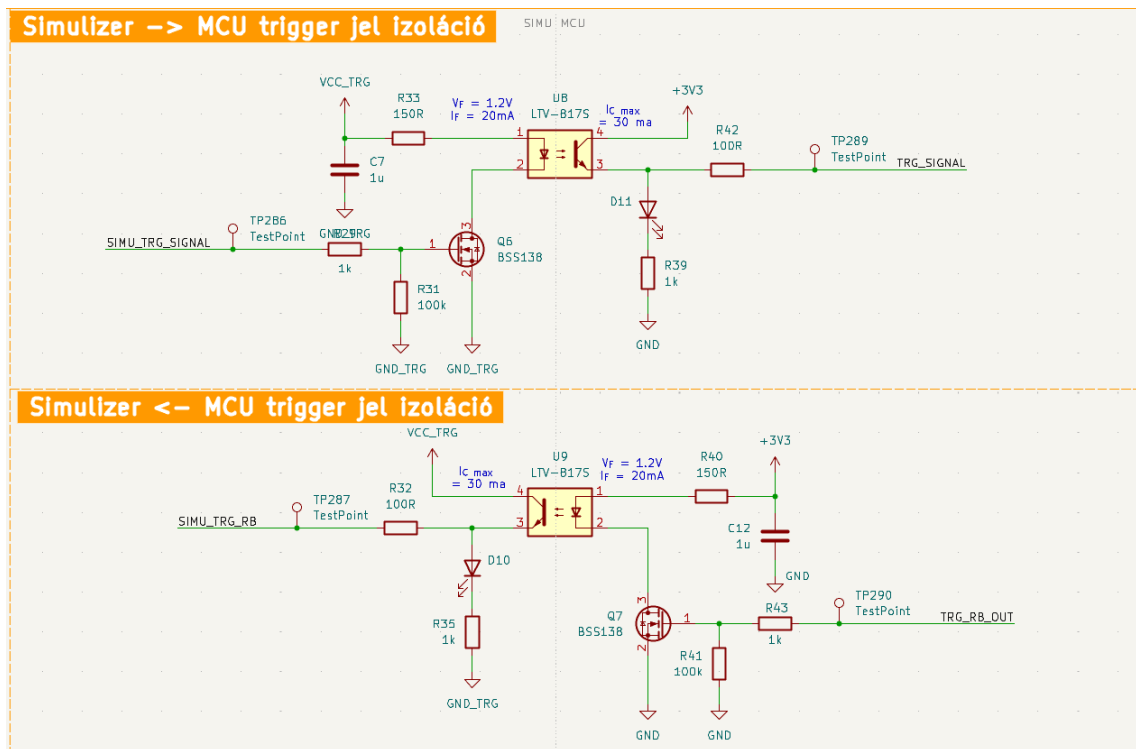
2.8.3 Külső trigger jel

Elvárás volt, hogy kívülről lehessen a beállást időzíteni egy trigger jellel, ami specifikált csatlakozón keresztül érkezik (2-27. ábra). Az időzítő jelnek és csatlakozójának galvanikusan leválasztottnak kell lennie az eszköztől, illetve az időzítést vissza is kell jelezni a külső eszköz felé.



2-27. ábra külső időzítő jel

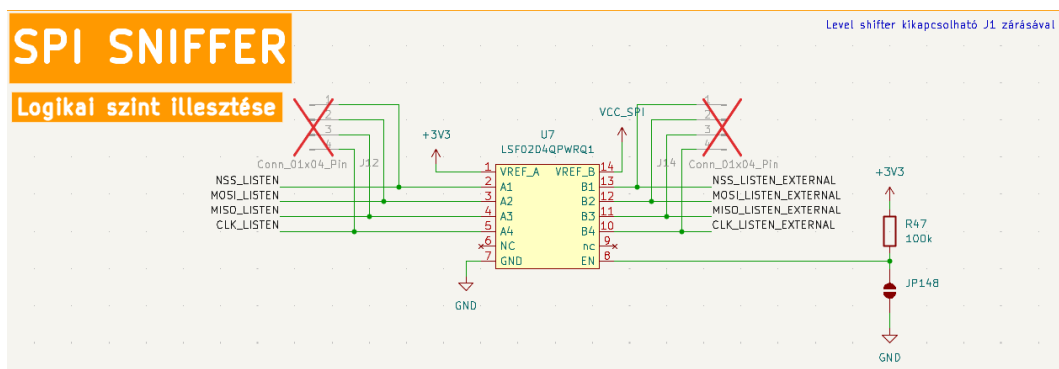
A külső időzítőjel csatlakozóján lévő tápfeszültséget LED fénnel visszajelzem, így könnyen megmondható, ha be van kapcsolva a külső eszköz. A kívülről érkező jelet optocsatolón keresztül viszem a mikrovezérlőhöz és a mikrovezérlőtől a külső eszközhöz is optocsatolón keresztül vezettem a visszajelző jelet.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



2-28. ábra külső időzítőjel és visszajelzés galvanikus leválasztása

2.8.4 SPI lehallgató

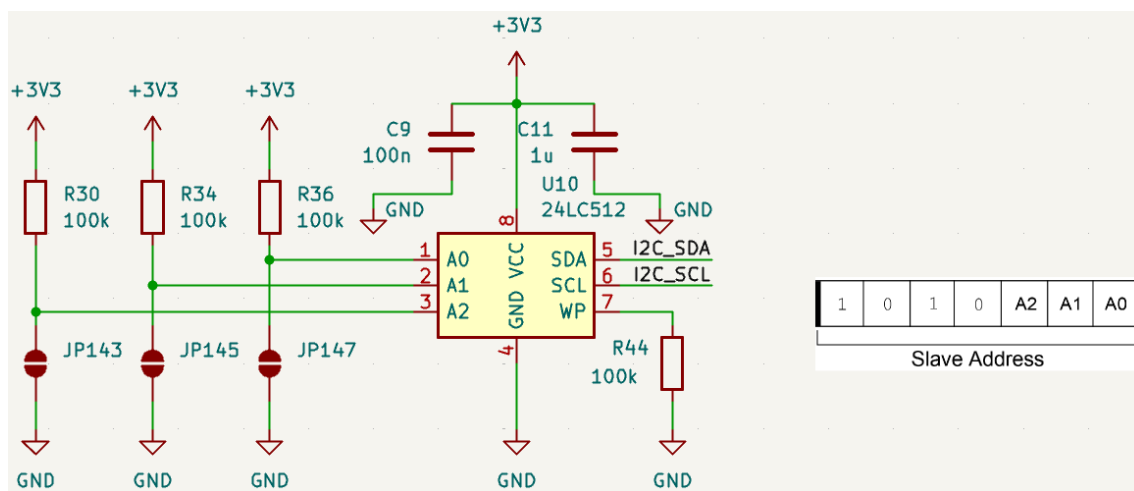
A tesztelt gyorsulásérzékelők lehetnek SPI kommunikációval konfiguráltak. Ebben az esetben érdemes lehet lehallgatni az SPI-csatornát, és a forgatást akár bizonyos SPI üzenetekhez időzíteni. Mivel ennek a kommunikációnak a feszültségszintje nem ismert ezért a jelet illeszteni kell. Erre két irányú, több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



2-29. 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 írású driverem, valamint a fejlesztéshez próbapanelre szerelt ilyen típusú IC-m, amivel az alaplap elkészültéig is ellenőrizhettem a rendszerbe illeszthetőséget. A modularitás és későbbi fejlesztetőség jegyében könnyen konfigurálhatóvá tettem az EEPROM címét.



2-30. 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

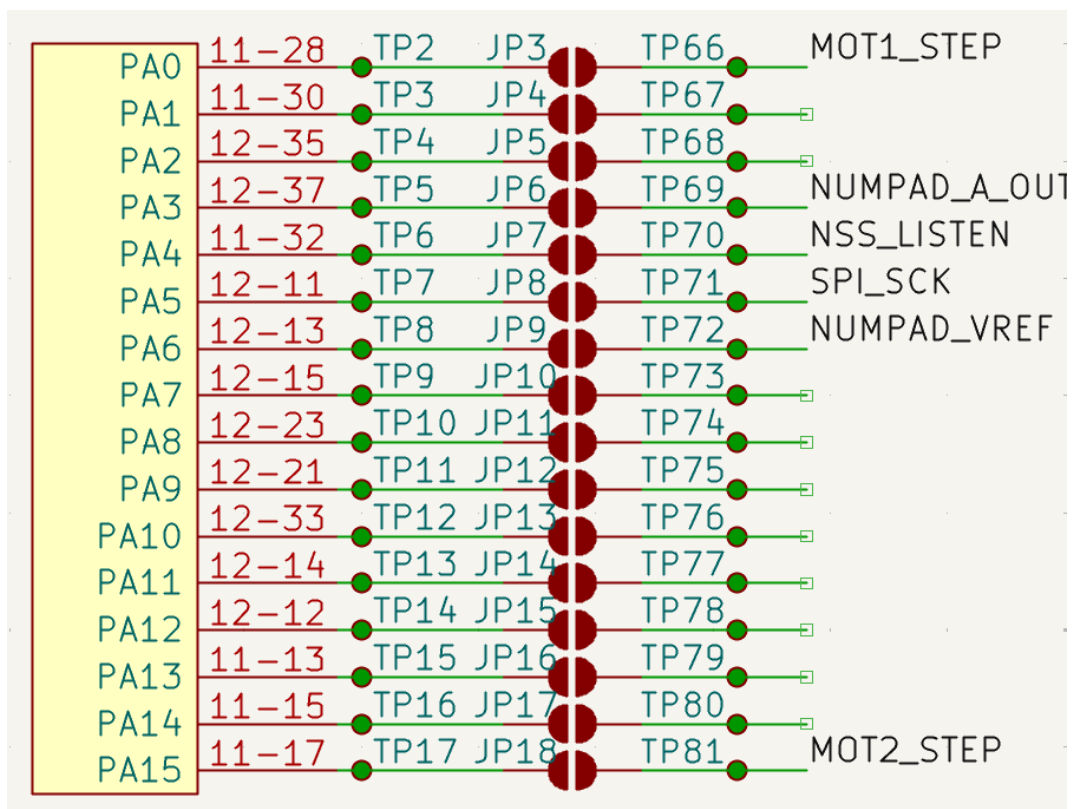
byte-os lapra van tagolva. Az első 55 oldalon 440 „header” tárolható (kulcs, érték címe, érték típusa), majd az 55. oldaltól a kulcsokhoz tartozó értékek laponként vannak tárolva, azaz 445 oldal áll rendelkezésre, így 5 oldal marad kihasználatlan, és 440 adat tárolható el.

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.

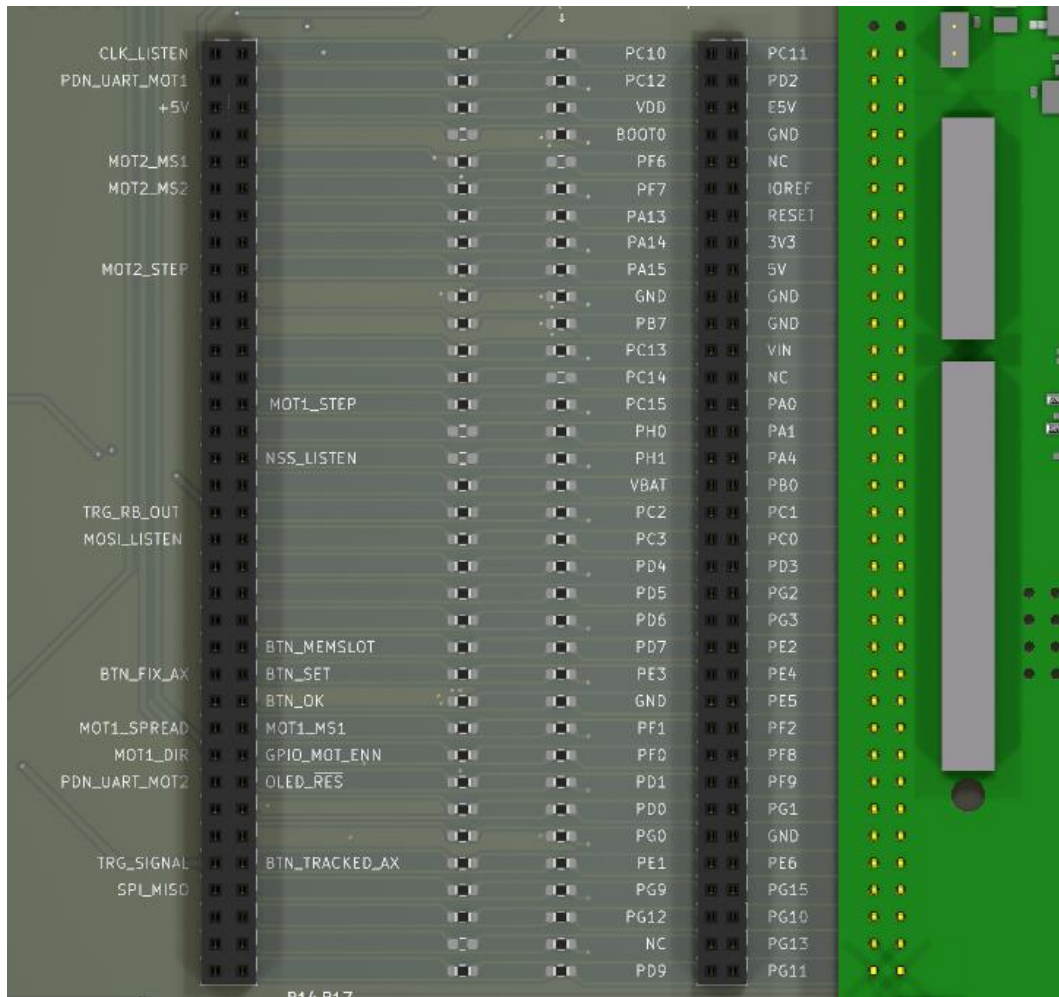
Jelen alkalmazásban a memóriát a beállítások mentésére használom. Minden memóiahelyhez 3 kulcsot használok fel: fix tengely, követett tengely, beállított szög.

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 a könnyű követhetőség érdekében. Ezeket a megoldásokat a 2-31. ábra és a 2-32. ábra mutatja be.

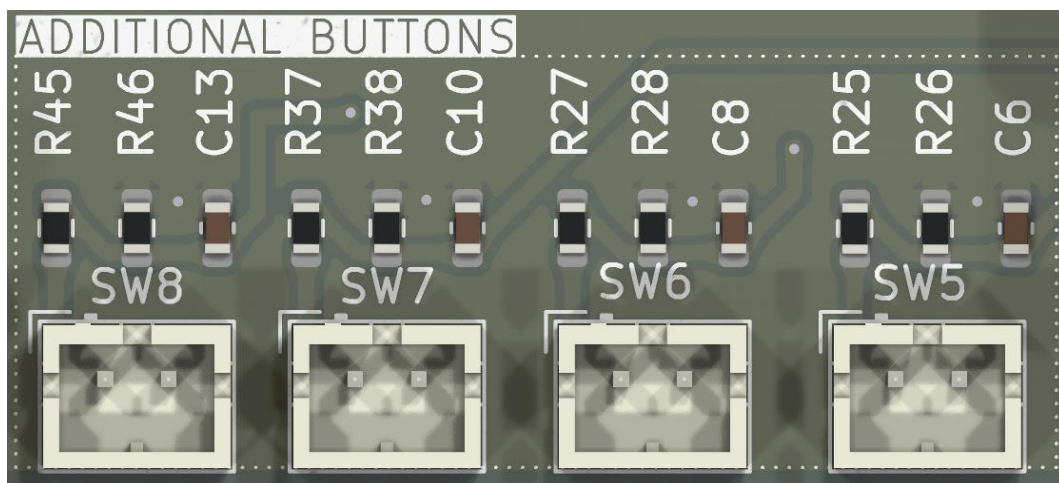


2-31. ábra Mikrovezérlő "A" portjának csatlakozóinak kivezetése

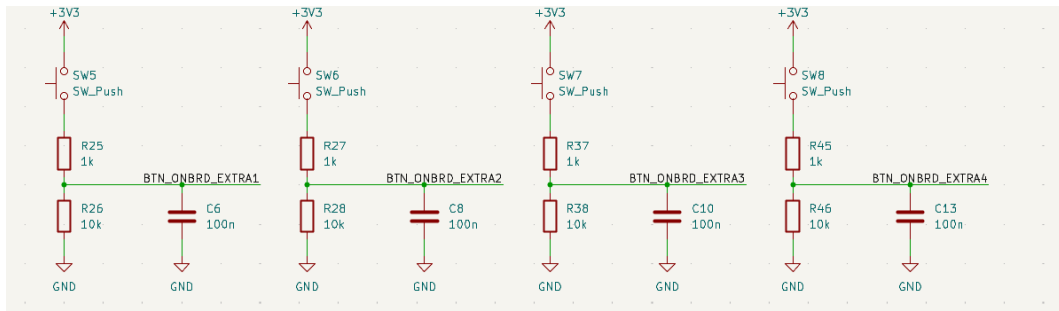


2-32. ábra Mikrovezérlő jeleinek kivezetése a nyomtatott áramkörön

A bővíthetőséget szem előtt tartva az alaplapon kialakítottam 4 további gomboknak is a bekötési lehetőségét (2-33. ábra és 2-34. ábra). Ezeket a gombokat a panelen pergés mentesítem, így nagyon egyszerűen beköthetőek ezek a további gombok.



2-33. ábra További gombok lehetőségének kialakítása a nyomtatott áramkörön



2-34. ábra Kiegészítő gombok pergésmentesítése

3 Összefoglalás/eredmények értékelése

3.1 Eredmények

Az eredeti kiírással ellentétben nem jutott sor a mechanikai tervezésre. Ennek oka, hogy a feladat adag sokkal nagyobb volt, mint amit realizztikusan meg tudtam volna ennyi idő alatt valósítani. Ennek ellenére az eszköz leg fontosabb részei elkészültek és bebizonyosodott, hogy tudnak együtt működni. Az eszköz megvalósítása közben a leg nagyobb kihívást a motorok folytonos vezérlés jelentette. Ezen túl megvalósítottam egy drivert az SSD1305 IC-hez, mai 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

4 Irodalomjegyzék

- [1] „All about MEMS sensors,” © Bosch Sensortec GmbH, 2024.
[Online]. Available: <https://www.bosch-sensortec.com/about-us/our-company/mems/>. [Hozzáférés dátuma: 10 05 2025].
- [2] C. Murphy, „Why MEMS Accelerometers Are Becoming the Designer’s Best Choice for CbM Applications,” Analog Devices, 01 02 2021. [Online]. Available: <https://www.analog.com/en/resources/technical-articles/why-memes-acceler-are-best-choice-for-cbm-apps.html>.
[Hozzáférés dátuma: 10 05 2025].
- [3] *PCB design guidelines for MEMS sensors*, STMicroelectronics, 2023.
- [4] „Acceleration sensor: BMA580,” © Bosch Sensortec GmbH, 2024.
[Online]. Available: <https://www.bosch-sensortec.com/products/motion-sensors/accelerometers/bma580/>. [Hozzáférés dátuma: 10 05 2025].
- [5] Z. Bende és A. Rávai, *Gépek biztonsága*, Budapest: Akadémiai Kiadó, 2018.
- [6] Technológiai és Ipari Minisztérium, *MUNKAESZKÖZÖK BIZTONSÁGA*, Technológiai és Ipari Minisztérium, 2022.
- [7] STMicroelectronics, „STM32F767ZI,” STMicroelectronics, 2025.
[Online]. Available: <https://www.st.com/en/microcontrollers-microprocessors/stm32f767zi.html>. [Hozzáférés dátuma: 19 június 2025].
- [8] TRINAMIC Motion Control GmbH & Co. KG, *TMC2209 Datasheet*, Hamburg: Analog Devices, 2023.
- [9] ATMEL, „AVR446: Linear speed control of stepper motor,” Atmel Corporation, San Jose, CA 95131, USA, 2006.

- [10] „ASN Inverse Sine,” Algorithms Project, 1 augusztus 2003. [Online]. Available: <https://algo.inria.fr/esf/function/ASN/ASN.html>. [Hozzáférés dátuma: 18 09 2025].
- [11] AMS, *AS5147U/AS5247U*, ams OSRAM Group, 2020.
- [12] AMS, *AS5x47U Motor Board*, AMS, 2021.
- [13] D. Hajnal, „Shellminator,” 2025. [Online]. Available: www.shellminator.org. [Hozzáférés dátuma: 13 szeptember 2025].
- [14] Solomon Systech Limited, *SSD1305*, Solomon Systech Limited, 2008.
- [15] Newhaven Display International, Inc., *NHD-2.23-12832UCB3*, 2661 Galvin Court, Elgin, IL 60124 USA: Newhaven Display International, Inc., 2024.
- [16] P. Stumpf, *Áramkörtervezés - Mikrovezérlők és digitális áramkörök környezete és illesztése*, Budapest: Automatizálási és Alkalmazott Informatikai Tanszék, 2022.
- [17] „android.com,” 12 február 2025. [Online]. Available: <https://source.android.com/docs/core/interaction/accessories/headset/plug-headset-spec>. [Hozzáférés dátuma: 15 szeptember 2025].
- [18] 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].
- [19] CUI inc., *P78B-1000*, Portland: CUI inc., 2025.

5 Summary

Függelék