



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

Juhász Ákos István

**TÖBBSZABADSÁGFOKÚ  
UNIVERZÁLIS ROBOTKAR  
TERVEZÉSE ÉS ÉPÍTÉSE**

KONZULENS  
**Dr. Stumpf Péter Pál**

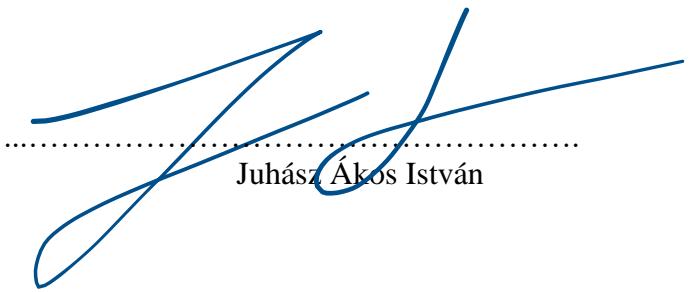
BUDAPEST, 2023

## HALLGATÓI NYILATKOZAT

Alulírott **Juhász Ákos István**, 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 átvettettem, egyértelműen, a forrás megadásával megjelöltetem.

Hozzájárulok, hogy a jelen munkám alapadatait (Juhász Ákos István, Többszabadságfokú univerzális robotkar tervezése és építése, angol és magyar nyelvű tartalmi kivonat, 2023, Dr. Stumpf Péter Pál) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzé tegye. 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, 2023. 06. 01.



A handwritten signature in blue ink, reading "Juhász Ákos István", is placed over a dotted line. The signature is fluid and cursive, with a large loop on the letter 'a'.

# Összefoglaló

A diplomatervem keretein belül egy 5 tengelyes *robotkar* teljeskörű megtervezését és megvalósítását kísérlem meg. Ez egyrészt egy nyílt forráskódú kész berendezést, másrészt egy kiváló támponot eredményez a témaban más hasonló házi projektek kivitelezéséhez.

A robotkar modellezéssel és 3D nyomtatással előállított szerkezeti elemekből épül fel. Az ezeket összekötő csuklópontokban - szintén saját tervezésű - hullámhajtóműveket alkalmazok. A hardver egyedi elektronikai áramkörrel és tápegység modullal egészül ki. Végül mindenek működtetésére egy érintőképernyőn megjeleníthető felhasználói felületet és a motorok mozgatásához szükséges programkódot mutatok be. A végcél, hogy az említett alkatrészhalma - megannyi kereskedelmi kötőelemmel és termékkel együtt - egyetlen berendezéssé álljon össze.

A produktum funkcionális célja, hogy a *soros robotkar* végpontját egyszerűen vezérelhetővé tegyem digitális gombok aktiválásával, térbeli koordináták megadásával.

A megvalósítás során megannyi méretezési, elektrotechnikai és szoftveres problémával találkoztam, melyek jelentős munkaközi kísérletezést, tesztet és próbát igényeltek. Szintén kihívást okozott a témaban történő kezdeti tájékozódás és a tervezési csapásirány kiválasztása, ugyanis egy robotkar megalkotását rendkívül sokféleképpen lehet kezdeni.

A mű sokszínű tematikáját tekintve jól illeszkedik a mechatronikai interdiszciplinák tudományterületei közé, ugyanis egyaránt érinti a műszaki mechanika, a gyártástechnológia, az áramkörtervezés, a szoftverkészítés és a jelfeldolgozás területeit. Ugyanakkor a gazdaságosi szempontok figyelembevétele mellett érinti a tervezéselméleti alapokon nyugvó konstrukcióképzés és termékéletút témaköreit is.

A tervezési és építési folyamat egy működőképes prototípus megalkotásában és demonstrációjában teljesedik ki, végül a munkát a diszfunkciók és a hibák okainak feltárásával és kiértékelésével zárom.

# **Abstract**

For my thesis the Reader can follow the complete design process and execution of a 5-axis robotic arm. On the one hand it provides an open-source functional device and serves as a jumping off point for those interested in the development of similar projects.

The robotic arm consists of components which were designed then created via a 3D printer. The joints utilize a self-designed strain wave gearing module. A unique electric circuit and a power supply were installed as well. Finally, I demonstrate the touch screen compatible user interface and the program code used to operate the machinery. The end goal is for the beforementioned parts to come together as a singular product.

The functional goal of the robotic arm is to enable easy to use control with digital buttons and via providing space coordinate values.

During development I've encountered various electrotechnics, measuring and software related difficulties which entailed major testing sessions. Getting the necessary information and deciding the course of action for development proved rather challenging as there are various ways of starting such a project.

The work with its multi-faceted aspects fits well into the multi-disciplinary field of mechatronics, since it includes engineering mechanics, production-technology, circuit modelling, software development and signal processing. Aspects of economic relevance, product life-cycle and design theory were touched upon as well.

The process of design and development culminates in the creation and demonstration of a prototype device. Finally, I close my work with the examination and evaluation of issues and disfunctions found in the end product.

# Tartalomjegyzék

|   |           |
|---|-----------|
| <b>1 Bevezetés .....</b>  | <b>3</b>  |
| 1.1 Robotok definíciói, csoportosítási lehetőségeik .....           | 3         |
| 1.2 Robotok konvencionális és innovatív alkalmazási területei ..... | 6         |
| 1.3 Hasonló projektek és termékek, feladatkitűzés pontosítása.....  | 7         |
| <b>2 A robotkar rendszerterve.....</b>                              | <b>9</b>  |
| <b>3 Gépezeti modellalkotás .....</b>                               | <b>11</b> |
| 3.1 A robotkar geometriájának meghatározása.....                    | 11        |
| 3.1.1 Mobilitási fok.....   | 12        |
| 3.1.2 A 3D nyomtatott alkatrészek előnyei.....                      | 13        |
| 3.2 Nevezetes mozgástartományok.....                                | 15        |
| 3.3 Csapágyazás.....  | 17        |
| 3.4 A csuklópontok kialakítása .....                                | 19        |
| 3.4.1 Az általam alkalmazott hajtómű .....                          | 20        |
| 3.4.2 Kardinális szempontok és tesztek .....                        | 21        |
| 3.4.3 Megvalósítható áttétel.....                                   | 22        |
| 3.5 A megfogó egység .....  | 24        |
| 3.6 Összeszerelési lépések .....                                    | 26        |
| <b>4 Villamos vezérlés tervezése .....</b>                          | <b>27</b> |
| 4.1 Hardverkomponensek és kommunikációs protokollok.....            | 27        |
| 4.2 A mikrokontroller .....   | 29        |
| 4.3 Léptetőmotorok és vezérlőik .....                               | 30        |
| 4.4 Szenzorika és vezetékezés .....                                 | 35        |
| 4.5 Központi vezérlőpanel .....                                     | 37        |
| 4.6 Tápkör és elektronikai doboz egység.....                        | 41        |
| <b>5 Demonstráció; tesztek és validált részfunkciók .....</b>       | <b>45</b> |
| 5.1 Kinematikai modell és validációja.....                          | 45        |
| 5.2 Vezérlőjelek összehasonlítása.....                              | 49        |
| 5.3 Mozgatás általános demonstrációja .....                         | 51        |
| <b>6 Működtető szoftver implementálása .....</b>                    | <b>53</b> |
| 6.1 Programnyelv és fejlesztői környezet .....                      | 53        |

|   |           |
|---|-----------|
| 6.2 Adatszerkezet és algoritmusok logikája .....                    | 53        |
| 6.3 Léptetőmotorok programvezérlése .....                           | 55        |
| 6.3.1 Forgalás megvalósítása .....                                  | 55        |
| 6.3.2 Kalibráció.....   | 57        |
| 6.3.3 Egyéb metódusok.....  | 58        |
| 6.3.4 Paraméterek tárolása .....                                    | 59        |
| 6.4 GUI Backend .....   | 59        |
| 6.4.1 Végpontok leírása .....                                       | 59        |
| 6.4.2 Metódusok definiálása .....                                   | 59        |
| 6.5 GUI Frontend .....  | 60        |
| 6.5.1 Futtatási képernyő.....                                       | 60        |
| 6.5.2 Naplózás megvalósítása .....                                  | 62        |
| 6.5.3 Inicializálás a berendezés kezelője által .....               | 62        |
| 6.6 Élesztési terv .....  | 64        |
| <b>7 Hibák és optimalizálási lehetőségek .....</b>                  | <b>65</b> |
| 7.1 Áramellátás és feszültség-stabilizálás.....                     | 65        |
| 7.2 A megfogó szorítóerejének növelése .....                        | 66        |
| 7.3 Karkinyúlás, szaggatott mozgás és a delaminációs jelenség ..... | 66        |
| 7.4 Szoftveres finomhangolás .....                                  | 68        |
| <b>8 Jövőbeli fejlesztési lehetőségek .....</b>                     | <b>69</b> |
| <b>9 Költségreálizáció .....</b>                                    | <b>70</b> |
| <b>10 Konklúzió; eredmények kritikája .....</b>                     | <b>71</b> |
| <b>11 Köszönetnyilvánítás .....</b>                                 | <b>73</b> |
| <b>12 Irodalomjegyzék.....</b>                                      | <b>74</b> |
| <b>13 Függelék.....</b>   | <b>78</b> |

# 1 Bevezetés

## 1.1 Robotok definíciói, csoportosítási lehetőségeik

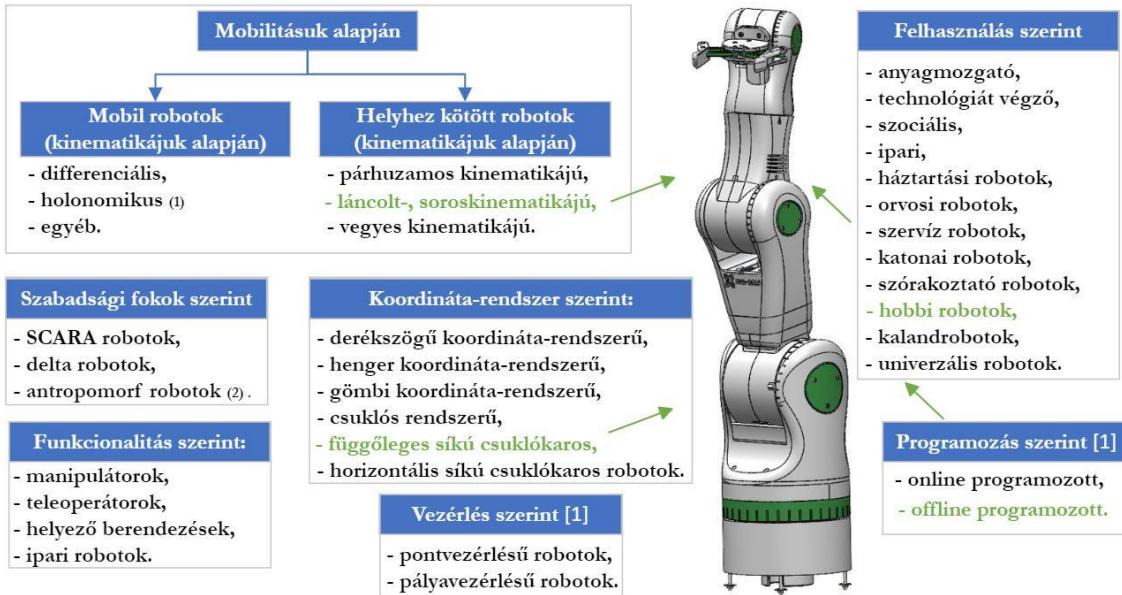
A *robotok*, olyan tetszőleges - általában kvázi-ideális - szerkezetű elektromechanikai berendezések, melyek szerkezettel, szenzorikával, beavatkozó szervekkel vannak felszerelve, valamint a mechanikus, emberi beavatkozás - átprogramozhatóság - nélküli autonóm programbefolyásolásra képesek [1] [2] [3]. Kifejtve ezen szabadon munkára fogható berendezések univerzálisan állítható, többtengelyű mozgó automaták, melyeknek általában utak és szögek mentén történő mozgássegymásutánisága valamilyen szempontrendszer alapján determinált. A konkrét munkavégzéshez gyakran valamilyen megfogóval, szerszámmal vagy valamilyen egyéb céleszközzel vannak felszerelve, így változatos anyagkezelési és technológiai feladatokra felhasználhatók [3]. Tehát a robotok funkcionális egységeit tekintve a következőkből állnak; *robothardver* és *robotszoftver*. A robothardveren belül megkülönböztetjük a robotmechanikát (*mechanizmusokat*), a hajtó egységet (*erőgép*), a robot szerszámát, szenzoregyüttesét, valamint az irányító rendszerét (*kontrollerét*).

Tehát manapság például egy reptéri futószalag, ami képes a gyanús bőröndöket automatikusan szelektálni, éppen úgy robotnak tekinthető, mint egy gyerek távirányítható jáékautója, ami jelzi ha lemerülőben van az elem. Mindezeket a különböző komplexitású *beágyazott rendszerek* és *kontrollerek* teszik lehetővé [4].

Létrejöttüket elsősorban a fejlett államok gyártási volumenének növekedését gátló munkaerőhiány, a minőségre és reprodukálhatóságra való törekvés, a termelési igény előterbe kerülés, valamint az emberi egészségre ártalmas munkahelyeken az emberi munka kiváltására irányuló igények segítették elő [2] [3] [5].

A robot megnevezést a cseh „*robota*” szóból származik, ami munkát jelent [3]. Szemantikai és historikus kitekintés, hogy ezen jelentések kapcsolódnak a feudalizmus korabeli módosult "*ingyelmunka*" kifejezéshez és a munkából eredő kiszolgáltatottság asszociációjához. Ezen jelentések nyelvben megjelenő ellenreakciója a robot szó pejoratív megítéléséhez is vezetett, mely részben megalapozta a későbbi, oly népszerű "robotok fellázadása" szcenáriókat, ami napjainkban a mesterséges intelligencia térnyerésével az "öntudatra ébredő gépek és komputerek" gondolatvilágával egészültek ki.

A robotok sokfélék, a célhoz kötött tevékenységük széles spektrumon mozoghat, éppen ezért a robotok csoportosítása is számos szempont szerint lehetséges, melyhez az **1. ábra** nyújt támpontot [1] [3] [6] [7]:



**1. ábra - Robotok csoportosítási lehetőségei**

A diplomatervemben szereplő robotkar (**4. ábra**) egy soroskinematikájú, helyhez kötött, *függőleges síkú csuklókaros*<sup>1</sup>, offline programozható, hobby robotkar.

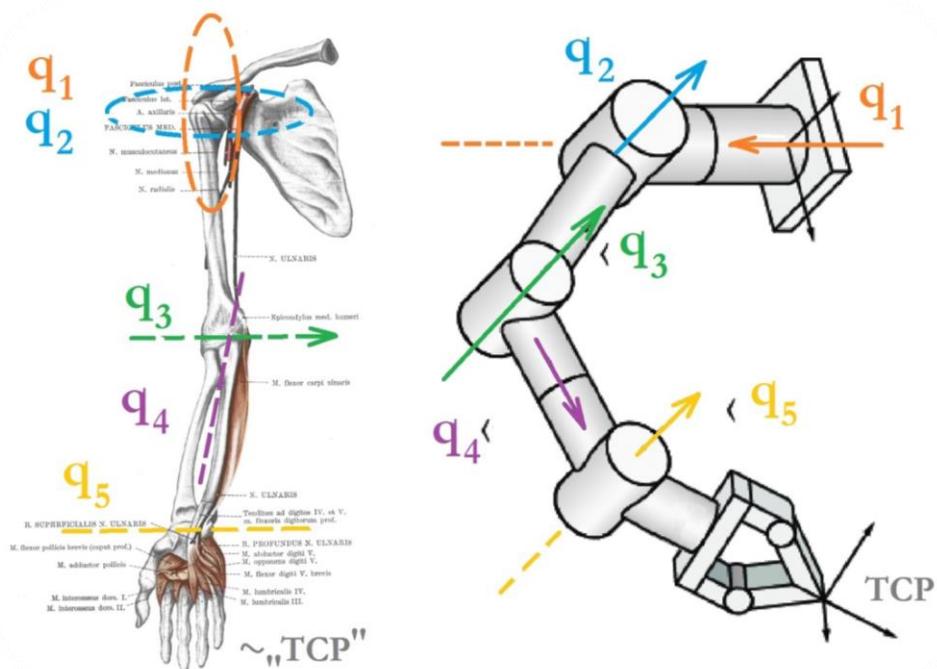
"A csuklókaros robotok legelterjedtebb változata, a hat tengelyes robotok, melyek valamennyi vizsgált cég kínálatában, nagy számban és méretváltozatban megtalálhatók. A feladatokat tekintve ez a típus kerül alkalmazásra a legváltozatosabb módon" [8].

A *robotkarok* mechanizmusait tekintve a népszerű csuklókaros robotok, ún. robotkarok mechanikai modellje az emberi kar mozgását, illetve az ember munkavégzés mozgásciklusát leképző gépszerkezetek. Mechanizmusuk egymáshoz sorosan kapcsolódó mechanikai testekből (*tagokból*) állnak, melyek kinematikai kényszerekkel (*ízületekkel*) kapcsolódnak egymáshoz [3] [8].

Bár léteznek csupán az elnevezésük szerint *humanoid robotkarok*, az általam célként kitűzött jellegű konstrukció (**2. ábra**) munkatere egy emberi kar mozgáspályáinak összességét képes bejárni [9]. Amennyiben képzeletben a következő néhány összevonást

<sup>1</sup> Csuklókaros robot: Olyan rotációs kényszerekkel rendelkező robot, amelynél a kényszerek közül kettő tengelye egymásra merőleges és metszi egymást, míg a harmadik az előző kettő valamelyikével párhuzamos [3].

elvégezzük, az az emberi kar szabadságfokaival azonos szabadságfokú mechanizmust fogja eredményezni. Ha a kettő egymásra merőleges csuklót ( $q_1$  és  $q_2$ ) összevonjuk, jó közelítéssel megkapjuk az emberi vállízület (*scapulohumerális ízületnek*) a mozgástartományait. A  $q_4$  tengely körüli rotáció az orsócsont és singcsont közötti mozgást modellezte le. Végül pedig a könyök ( $q_5$ ) precesszió jellegű forgatását szűk szögtartománya végett elhanyagolhatónak tekintem. Különbség így csupán abban marad, hogy az emberi kéz állványzata, az emberi törzs a robotkarhoz képest elforgatott, függőleges felfüggesztésű. Az így kapott szerkezet az evolválódásunk során kialakult emberi kar mozgását megközelítőleg leírja.



2. ábra - A 7 szabadságfokú emberi végtag és egy humanoid robotkar [10]

Bizonyos aspektusból a humanoid szó jelentését az emberi felépítéssel nagy hasonlóságot mutató robotok merítenék ki leginkább. Ilyen produktumok például a *Boston Dynamics<sup>©</sup>Atlas*, a *Honda<sup>©</sup>ASIMO*, vagy az *Aldebaran Robotics<sup>©</sup>NAO* robotjai. Egy másik kontextusban az *emberi viselkedés* lemodellezhetőségének mértéke is lehetne a humanoid szó ismérve.

## 1.2 Robotok konvencionális és innovatív alkalmazási területei

Az előző fejezetben ismertetett csoportosítási lehetőségek kiegészítéseként most a kezelő személye szerint áttekintem a robotok két újabb általános családját, ami szerint a lehetnek *civilek* vagy *szakemberek* a végfelhasználók. Ezen elgondolás mentén az alkalmazott robotok is újra csoportosíthatók a hétköznapi polgár számára is elérhető és megfizethető *konvencionális* és *ipari/speciális robotok* berendezéseire.

Az ipari robotokon belül - az általában magasabb anyagi és szellemi tőkét is igénylő - speciális technológiákat és szaktudást alkalmazó robotok között felfedezhető néhány kuriózum, melyek egyúttal teret engednek az *innovatív* robotikai eljárások térhódításának.

Ilyen jellegű innováció például a munkaerőpótására irányuló fejlesztések a kommunikációs robotok terén. Ennek néhány ékes páldánya Japánban már előrehaladott stádiumban járó *kórházi* és *szociális robotok*, melyek a gépi beszédfelismerés és beszéd fejlődésével képesek enyhíteni a kórházakban fekvő vagy elmagányosodott idősebb korosztály egyedüllétéét. Ilyen robotok például a *SoftBank Robotics* által tervezett *Pepper robot*, de kisebb cégek már Magyarországon is kezdtek fejlesztésbe, mint például az *Enjoyrobotics* [11].

Hasonló tevékenységet végeznek a veszélyes munkakörökben alkalmazott *robbanóanyag-mentesítő automaták*, mint például a *Telerob Explosive Ordnance Disposal (EOD)*. Néhány gyártó, aki hasonló robotokat hozott forgalomba: *iRobot*, *QinetiQ* vagy a *Northrop Grumman*.

A veszélyes atmoszférához kapcsolódóan léteznek sugárzásgyanús környezetre vagy akár atomkatasztrófák sújtotta területeken bevethető *sugárvédelmi-robotok* is, mint például a *Mitsubishi Heavy Industries "MARS"* vagy a *Toshiba Robotics "Scorpion"*.

A legtöbb embert érintő modern robottechnológia alkalmazás mégis az úgynevezett *robot asszisztált sebészeti eljárás (Robotically Assisted Surgery)*, mely segítségével világszerte már milliókat operáltak meg. Az egyik legnépszerűbb rendszer a *da Vinci® Surgical System (3. ábra)* már a világ több, mint 60 országában 4400 telepített rendszerrel rendelkezik, melyekkel lehetséges akár távolról is ellátni vagy megműteni egy páciens. Ezzel a módszerrel jelentős időt és költséget lehet megspórolni azoknak a kórházaknak és sebészeknek, akik rendelkeznek a praktizáláshoz szükséges robotokkal és licence-ekkel.

Hihetnénk, hogy ezek a gépek csupán "imitálják, ismétlik" azt a mozdulatsort, amit néhány ország vagy nem ritkán akár egy-két kontinenssel arrébb a műtétek végrehajtó sebész cselekszik, azonban ez távolról sem igaz. A valóságban jelentékeny mértékben képesek beavatkozni az aktuális cselekménybe, például a mozgás stabilizálásával vagy az orvos kezében esetlegesen ébredő fisiológiai remegések csökkentésével [12]. Ezen beavatkozásokat is mind a fent említett szenzorikával és szoftverekkel lehetséges elérni.

További orvosi asszisztens-robotokat forgalmazó gyártók a *Stryker Mako System* vagy a *Zimmer Biomet ROSA Surgical System* [13].

A konvencionális robotok a fentiek alapján már behatárolhatók, kifejtésüktől eltekintek, viszont a jólérhetőség jegyében néhány példa:

- robotporszívó (**3. ábra**),
- önvezető autó,
- "okos otthonok" berendezései, okos eszközei,
- vagy akár egy többfunkciós érzékelőkkel ellátott turmixgép.



3. ábra - Példák civil felhasználású és speciális szakembert igénylő robotokra [Google képek]

### 1.3 Hasonló projektek és termékek, feladatkitűzés pontosítása

A modern robotok áttekintését követően egy olyan általános célú robotkar elkészítésé mellett döntöttem, amit akár a háztartásban is hasznosítani lehet. Ennek a feltétele az, hogy a berendezés elérhető áron - például egy általános rendeltetésű személyi számítógép áráért - beszerezhető legyen, valamint, hogy az univerzalitás kritériumát

teljesítve egyáltalán lehetséges legyen az otthoni felhasználás. Ehhez valamilyen egyszerű mozdulatsor betanulása, végrehajtása volna kielégítő feladat, mint például a ruhahajtogatás vagy egy kis méretű tárgy odébb helyezésé.

A demonstrációs célok pontosítva a *Fanuc*<sup>©</sup>, *ABB*<sup>©</sup>, *KUKA*<sup>©</sup>, és *Boston Dynamics*<sup>©</sup> általános ipari robotkarjainak áttekintése után figyelmem a hasonló célkitűzésű cégek gyártmányaira és projektjeire terelődött, mint például az *Elephant Robotics*<sup>©</sup> *myCobot* 280-as könnyen programozható robotkarja. Ez a modell rendkívül felhasználóbarát módon, szoftver nélkül, manuálisan tanítható [14]. Ez azt jelenti, hogy a kézzel célhelyszínre koordinált kar képes kiváló pontossággal reprodukálni a betanult mozdulatsort.

A tárgyak mozgatásának vonatkozásában említendő a '97 májusi sakk világajnokság, amikor az *IBM*<sup>©</sup> *Deep Blue* sakkmotorja először aratott győzelmet ember felett [15]. Egy sakkkfigurával tett lépés megtétele megfelelő cél lenne. A sakktábla ötletét a *Millennium Europa*<sup>©</sup> által forgalmazott sakktábla készletek ihlettek. Ebből következik, hogy léteznek tisztán szoftveres, táblás és az általam preferált robotkaros sakkrobotok. Több sakközö robotkart is megvizsgáltam, és általában 4-5 szabadságfokú humanoid manipulátorokkal vagy 4 szabadságfokú *scara robotokkal* találkoztam [16] [17] [18] [19]. Azonban a megfogók többségét nem találtam elég kiforrottaknak. Amennyiben kizárolag sakközösségra szánnám a berendezést, akkor a *scara elrendezésű* vagy a legkevesebb, a tábla fölé telepített 3 tengelyes, egyszerűen koordinálható változatok lennének a legoptimálisabbak.

A diplomamunka számára egy hasonló, a BCN3D Technologies<sup>©</sup> által közzétett projekt nyújt kiindulási támpontot [20]. **A diplomamunka elsődleges célja az, hogy ezen támpont alapján elkészüljön egy működőképes és felhasználóbarát robotkar prototípusa.**

A berendezés áttervezése során a külső formatervet és tengelytávolságokat leszámítva minden fizikai komponenst áttervezek, beleértve a mozgatáshoz szükséges érintett mechatronikai részegységeket is. Az *ARDUINO MEGA* helyett egy *RASPBERRY PI 3 B*, a *TB6560* vezérlők helyett pedig a *RDV8825* típusok kerülnek beépítésre. Ezen kívül az eszközt egy 7'-os *DSI* érintőképernyővel tervezem ellátni. Az elektronikai egységeket egy általam tervezett áramkör kapcsolja össze. A mozgatást pedig az eredeti beavatkozóktól eltérő típusú és méretű motorok fogják végezni.

## 2 A robotkar rendszerterve

A tervezés kezdőlétéseként szükséges minden teljesíteni kívánt tervezéselméleti peremfeltételt pontosan definiálnunk, amik a berendezés működését övező követelményeket és elvárásokat leírják. Összességében e folyamat eredményeként adódik a *rendszerterv* és az ún. *követelményjegyzék*. Utóbbi a tervezési folyamat egyik legfontosabb dokumentuma, amely biztosítja a tervezési folyamatok koherenciáját, szisztematikusan lehetővé téve a termék követelményeinek jövőbeli teljesülését. Ennek segítségével a követelmények és azok teljesülése külön-külön is elemezhetők [21].

Ugyan a dokumentumban szereplő specifikációkat általában egy *megrendelő* igényei alapján kell összeállítani, ezúttal mégis a robotkar célkitűzéseit és hobbi jellegét tekintve, ezen feltételeket én szándékozom meghatározni. Máskor természetesen történhet egy ügyfél, alfelhasználó vagy általános piaci igények becslése alapján is.

A koncepcióképzés kezdeti szakaszában számos koncepcióváltozatot dolgoztam ki, viszont a gyakorlatot felülírva a *dolgozat célja* -, univerzális robotkar tervezése - meghatározás miatt eltekintek a változatok részletezésétől.

A követelményjegyzéknek az összes funkciót tartalmaznia kell, valamint az azokhoz tartozó elvárásokat a következő három főbb minősítési osztályba<sup>2</sup> kell sorolnia; *alapkövetelmények*, *szintkövetelmények* és *óhajok* [21].

A robotkar teljes rendszertervét a **4. ábra**, míg követelményjegyzékét a függelék [1. melléklete](#) mutatja be.

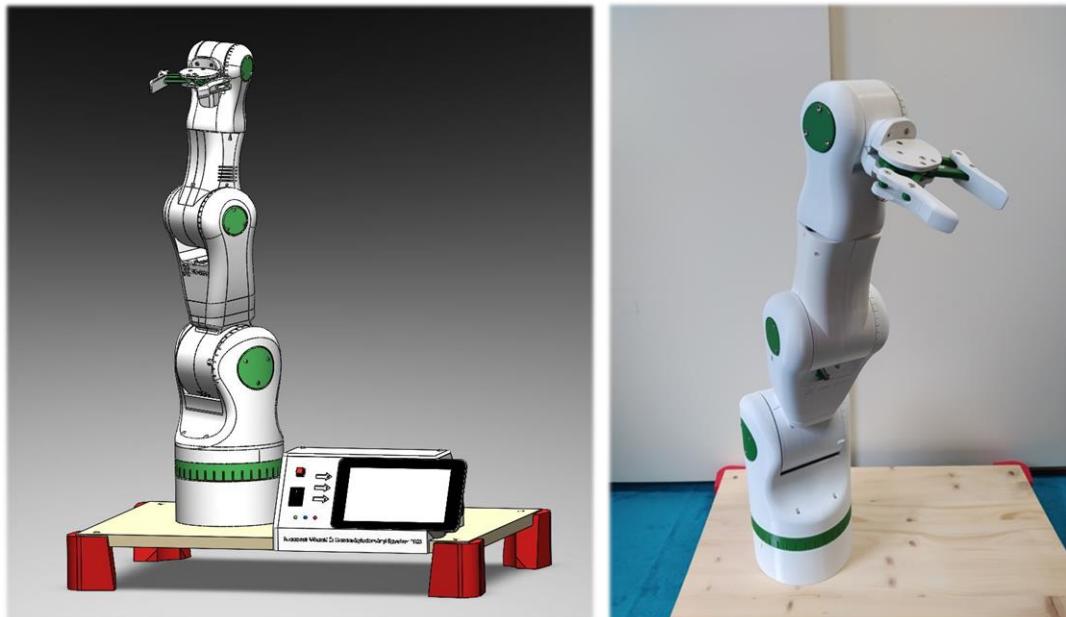
A fenti elképzéléseimet figyelembe véve néhány komponensspecifikus jellemző rajzolódik ki előttem. Például a 14.,15.,16. követelmények a megfogó fejegységre vonatkozó, míg a 9. - 13. követelményekig pedig a karrendszer mechanizmusára vonatkozó jellemzők. Ezen felül természetesen külön kezelhetnénk szoftver- és hardverkövetelményeket is, viszont esetben ez nem tekintettem szükségesnek.

A későbbiekben látható ([4.3-es fejezet](#)), hogy többek között például a léptetőmotorok tartónyomatéka a hullámhajtás alkalmazása miatt nem került be a követelmények közé.

---

<sup>2</sup> Minősítő - vagy másképp súlyozó - osztályok definiálása az 1. számú mellékletben kerül kifejtésre.

Az is megfigyelhető, hogy a jegyzék a sebességszabályzásra szintén nem tér ki. A sebességet csupán egyes *sebességszintű* számításokhoz használom, hiszen a sebességértékek a funkcióbiztonságot nem érintik.



**4. ábra - A robotkonstrukció modellje és építménye**

Amennyiben a megfelelőségi követelmények listaelemei egyidejűleg, a jegyzéknek megfelelő mértékben teljesülnek, a robotkar működőképes szerkezetként fog viselkedni vezérelhetőségének megtartása mellett.

### 3 Gépezeti modellalkotás

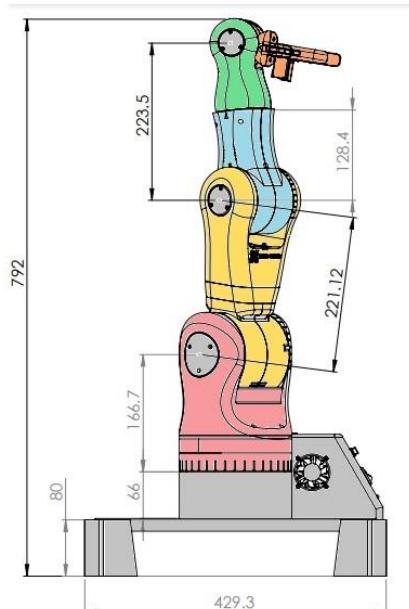
A modellt *SolidWorks2021* programban alkottam meg, mely összesen 271 komponensből és kötőelemből épül fel. Ezen modellt a *modell.zip* nevű [1. csatolmányban](#) lehet megtekinteni.

A BCN3D MOVEO eredeti modelljéből csekély mennyiségű műszaki paraméter került megtartásra, minden összes méreteket, rádiuszokat és tengelytávolságokat tartottam meg, amelyek viszont a mozgástartományokat jól behatárolták.

#### 3.1 A robotkar geometriájának meghatározása

Az általam konstruált robot (**5. ábra**) egy 5 csuklós, 5 szabadságfokkal rendelkező, magas autonómiai fokra képes robotkar, mely kizárolag csuklókkal összekapcsolt soros tagokból épül fel, melyeket 3D nyomtatással hozok létre.

Az alkotó tagok szerelési fotót a függelék [4. csatolmány](#) tartalmazza.



5. ábra - A robotkar szegmentumai és befoglaló méretei

A *BCN3D Technologies* egy 3D nyomtatókat is fogalmazó cég, ami közzétett egy nyílt forráskódú robotkar tervet, ami a saját nyomtatójukra került optimalizálásra és - a beszerzést megkönnyítendő - előre összeállított *BOM*<sup>3</sup>-al rendelkezik. Az erre a karra

---

<sup>3</sup> BOM: Az angol "Bill Of Material", magyarul tételejegyzéket jelent.

tervezett léptetőmotorok, az *Arduino Mega* kontroller, a *TB6560* vezérlők, valamint a *Marlin* grafikus felhasználói felület minden kiváltásra kerültek.

Az eredeti kar felhasználóbarát termékesítése az alábbi módosításokkal történik:

- a motorok elrejtése,
- a külső kábelkorbácsok eltüntetése,
- letisztult külső formaterv készítése,
- 3D nyomtatott hullámhajtóművek alkalmazása a csuklókban,
- az önsúly csökkentése érdekében történő kikönnyítések kialakítása,
- az összes szíjhajtás eltávolítása,
- csapágyazások csuklóterhelésekhez történő igazítása,
- elektronikai elemek egyetlen dobozban történő integrálása,
- új megfogó-mechanizmus tervezése/meglévő átalakítása.

A fenti szempontokat minden könnyű szerelhetőség megtartása mellett igyekszem egyesíteni.

### 3.1.1 Mobilitási fok

Minden csuklónak egy szabadsági foka és öt kötöttségi foka van. Ez azt jelenti, hogy egy ilyen mechanikai kényszer korlátoz 3 pozíció és 2 rotáció típusú elmozdulási lehetőséget, egyet pedig szabadon hagy. A diplomamunka címében szereplő "többszabadságfokú" kifejezés arra utal, hogy a huroktól mentes robotkar több szabadságfokkal rendelkezik. Az egység mobilitási vizsgálata a kinematikai láncolat alapján a következő [2]:

$$DoF = n \times s - \sum_{i=1}^5 (n_i - i) = 5 \times 6 - 5 \times 5 = 5 \quad (1)$$

ahol,

- $n$  - a merev testek, tagok száma (állványzat  $n = 0$  indexű) [-],
- $s$  - egy merev test szabadsági fokának száma téren [-],
- $n_i$  - kényszerkapcsolatok, kinematikai párok száma [-],
- $i$  – alkalmazott típusú kényszer osztályszáma, kötöttségi foka [-],

Ezen 5 *DoF* mentén kialakuló 5 változót szándékozom motorokkal pozicionálni. Az 5 szabadságfokú robotkarok alkalmazása az iparban, szereldékben is gyakori. Ennek

oka az, hogy ezen robotkarok megközelítik a kiváltani kívánt emberi erőforrás 7 szabadsági fokkal rendelkező végtagjainak mobilitását.

"Ha a *DoF*-ok száma nagyobb, mint a feladat végrehajtásához minimálisan szükséges, akkor az inverz kinematika esetén maradnak *extra szabadsági fokok*." Amik kinematikai értelemben redundánsak, ezeket *redundáns robotoknak* nevezzük. [22]. Az **1. egyenletből** következik, hogy ennél a konstrukciónál ez nem áll fenn.

### 3.1.2 A 3D nyomtatott alkatrészek előnyei

Ezen additív alakadó technológia napjainkban egyre nagyobb hangsúlyt kap - kiváltképp a gyakori újratervezést igénylő prototipizálás szakaszában, - ahol költséghatékony megoldást nyújthat azokra az esetekre, melyeknél például meg kívánjuk spórolni az új fröccsöntő szerszámok készítését [23] [24] [25]. A robotot alkotó összes műanyag elemet *FDM technológiájú<sup>4</sup> Prusa MINI+ típusú nyomtatóval hoztam létre*.

Ez a gyártási mód befolyásolja már a tervezés szakaszát is, ugyanis lehetőséget biztosít egyedi geometriájú alakzatok létrehozására. Ilyenek a *belő hűtési csatornák*, az *egyedi vezetékjáratok* és *kábelcsatornák (32.ábra)*, a menetes *belő betétek* magfuratai, vagy a hajtásláncban résztvevő *fogazott elemek (7.ábra)*. A menetes betétek lehetnek alakzáró, a nyomtatás során vagy után behelyezendők, és lehetnek az alakadást követően beolvastandók (**6. ábra**).

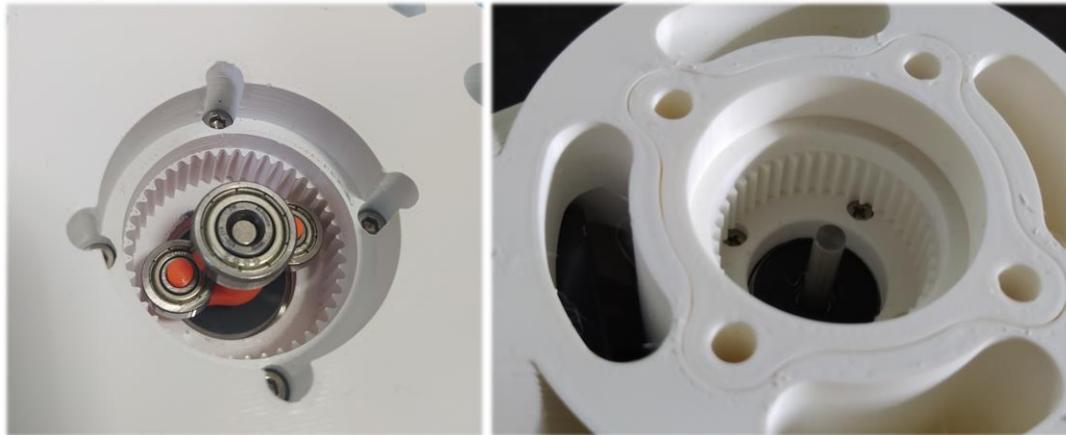


**6. ábra - Különböző betétemek és azok beszerelt változatai [[www.3djake.hu](http://www.3djake.hu)]**

---

<sup>4</sup> FDM: Fused Deposition Modeling, azaz szálolvasztásos alakadó technológia. A legköltséghatékonyabb és elterjedtebb eljárás, viszont lassú és gyakran nem elég pontos és finom a létrehozott felület.

Az M4-es betéteket  $\varnothing 5.8$ -as, az M3-as betéteket pedig  $\varnothing 4.2$ -es "magfuratokba" helyezem el. Tapasztalataim szerint az általam használt *Ultimaker*<sup>®</sup> és *Fritz*<sup>®</sup> márkatűjű filament alapanyagok a szerkezet terhelhetőségéhez mért kellő szakító- és nyomószilárdság értékeit képesek nyújtani.



**7. ábra - Szegmentumokba nyomtatott elemek különböző motorok estében**

A csuklópontok mentén  $10^\circ$ -os felbontású rovátkázott skálákat helyeztem el, amikkel a szögelfordulás nyomon követhető szemrevételezéssel.

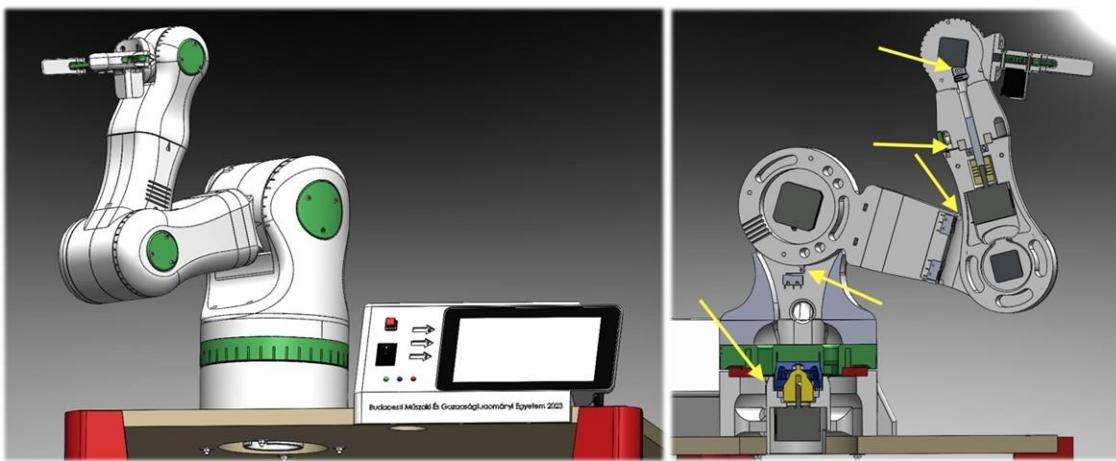
Alapvető irányelvnek vettem továbbá, hogy az állványzathoz közelebb lévő tagok kitöltési tényezőjét (*infill-t*) nagyobbra válasszam, mint a soron következő, alapzattól távolabbi tagokét. Ezzel az alacsonyabb indexű tagok tömegének és szilárdságának növelését igyekeztem elérni, ami elősegítheti a robotkar stabilitását.

3D nyomtatás esetén rendszerint maguk a modellek szolgálnak kiviteli tervként.

## 3.2 Nevezetes mozgástartományok

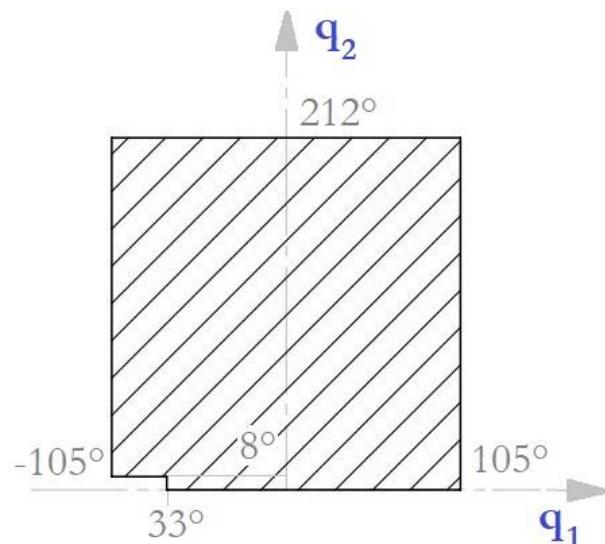
A robotkarra jellemző egyedi tartományok és intervallumok alatt a *HOME*-pozíciót, a *csuklóteret*, valamint a *munkateret* értem.

A robotkar *HOME*-pozíciója (**8. ábra**), azaz referencia- vagy alaphelyzete meghatározza a kibillentetlen  $t = 0$  időpillanatban a tagok egymáshoz viszonyított pozícióját és ezzel a szenzorok elhelyezkedését is. A valóságban a kapcsolók érintését követően az inaktiváláshoz  $4^\circ$ -ot visszaléptetnem szükséges.



**8. ábra - HOME-pozíció és érintőkapcsolók elhelyezkedése**

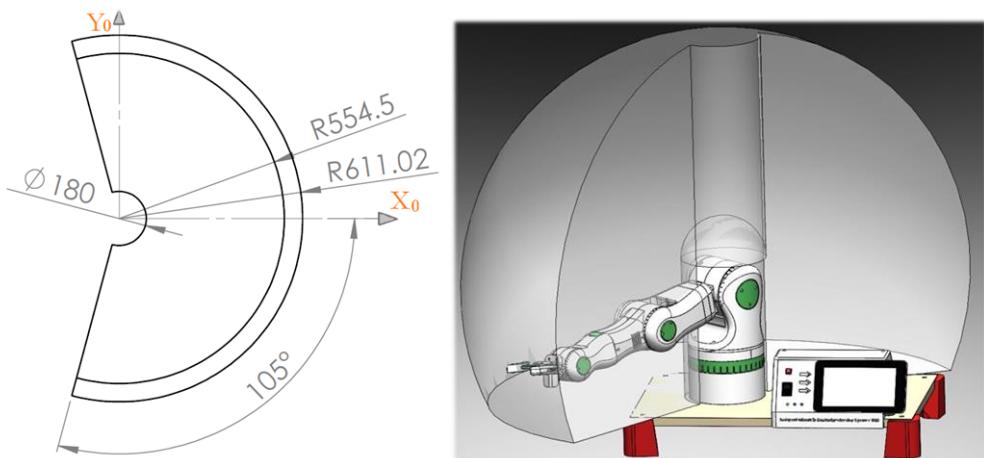
A *csuklótér* (**9. ábra**) a csuklók megengedett paramétereinek mozgástartományát ábrázolja. A vázolt csuklótér hiányos részét az elektronikai doboz jelenléte magyarázza, amibe nem szándékozom belevezetni a kart, így ott az értékfelvétel nem megengedett.



**9. ábra - A robotkar q1 és q2 szomszédos csuklópárjának csuklótere**

A munkatér (**10.ábra**) az összes lehetséges pont halmaza a világ koordináta-rendszerben értelmezve, amit a *TCP*<sup>5</sup> felvehet. Amennyiben ezt csak és kizárálag a csuklótér határozza meg, *direkt kinematikai*, fordított esetben pedig *inverz kinematikai* rendszerről beszélünk.

A TCP által elérhető pontok síkbeli kirajzolásával, majd  $\pm 105^\circ$ -os megforgatásával meghatározom a munkateret. A munkatér így egy *hiányos gömbszeletre* hasonlít leginkább [26] [2].



**10. ábra - A robotkar munkaterének felülnézeti ábrája és az általa megközelíthető ponthalmaz**

Ezen a robotot jellemző tér- és szögintervallumok ismeretében azt tudom leolvasni **10. ábrráról**, hogy a jelenlegi megfogóval a 611.02 mm-es rádiusz végpontja már a munkatéren kívül, azonban még az ún. tiltott zónán belül helyezkedik el. Ezen felül a világ koordináta-rendszerben értelmezett z irányban megakadályozom a negatív értékek felé történő manőverezés, mert azok az *alaplap* alatt helyezkednek el és megközelítésük fizikai ütközéshez vezetne.

---

<sup>5</sup> TCP: Tool Center Point, azaz szerszám középpont.

### 3.3 Csapágyazás

A forgó mozgások megvezetésére használt csapágyazások célja az egymáshoz viszonyított elemek közötti súrlódó hatás minimalizálása [27]. Az általam választott megoldások igazodnak a tagok jellegéből adódó követelményekhez, valamint az általános csapágyazással kapcsolatos irányelvekhez<sup>6</sup>.

A robotkar tervezésénél 3 különböző féle csapágytípust (**11.ábra**) használok.

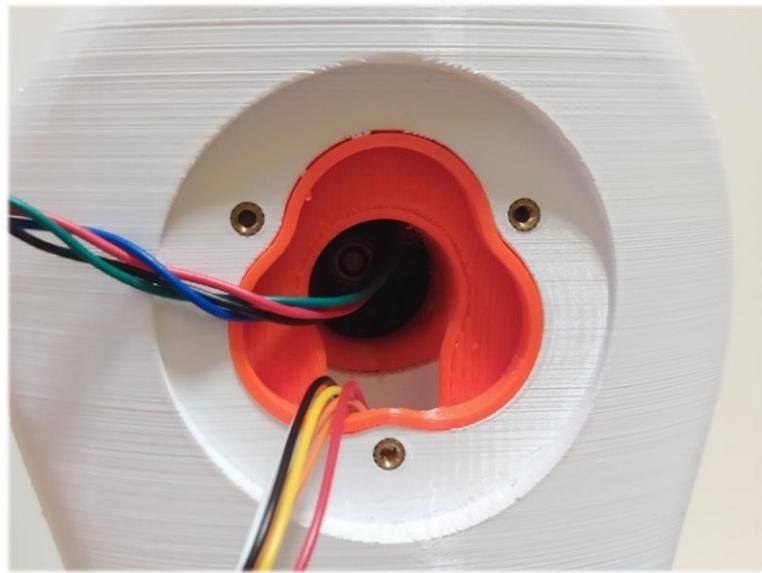


**11. ábra - Vegyes, radiális és axiális terhelésű csapágyak és azok beépítési helyei**  
[felső sor: katalógus képek]

A kialakítás során a helytakarékos kihasználás és a költségmegtakarítás okán a vezetékezést koaxiálisan nyomtatott csőtengelyekbe (**12.ábra**) rejttem, hogy elkerüljem a forgócsatlakozók használatát. Ezzel a módszerrel igyekszem elkerülni a vezetékek megfeszülés okozta szakadásos károsodást is.

---

<sup>6</sup> Csapágyazások kialakításának irányelvei: Terhelési irány választása, alátámasztás kialakítása, kenési mód kiválasztása, forgástengelyre illesztés, szilárdsági tartomány választása.



12. ábra - A forgástengelyekbe tervezett csőtengelyek mentén kialakított vezeték-üregek

Törekedtem arra, hogy a vertikális tengelyeket *kétoldali* csapágyazással lássam el, hogy az önsúlyból és terhelésekkel következő reakcióerőket "berágódás" nélkül legyenek képesek felvenni. Az ilyen típusú forgástengelyek esetén az egyik oldalon a vezetékezést rejtiő csapágy, a másikon oldalon pedig a hajtómű csapágya veszi fel a terhelést. A  $n$ -edik tag felé közelítve kisebb méretű és tömegű elemeket használok, mint például polimer tengelyek és csapágyak.

A kísérletezés és irodalomkutatás során megismerkedtem potenciálisan használható termékekkel, mint a *Franke*<sup>®</sup> típusú huzalcsapággyal és kipróbtam saját készítésű *Igus*<sup>®</sup> márkarjú poralapú *iglidur*<sup>®</sup> J260-PF típusú filamentből készült siklócsapágyanyag nyomtatását is. Ezeket később a hagyományos termékekhez viszonyított magas költség- és időigény miatt elvetettem [23] [28].

### 3.4 A csuklópontok kialakítása

A robot működtetése során a csuklókban 3 helyen alkalmazok közvetlen meghajtást, 3 helyen pedig az ipari robotokban is gyakran használt *hullámhajtóművek* alkalmazása mellett döntöttem. A közvetlenül meghajtott alkatrészeket tengelykapcsolókkal (**13.ábra**) kötöm a motorcsövekhez.



**13. ábra - Az alkalmazott merev és flexibilis, hasított tengelykapcsolók**

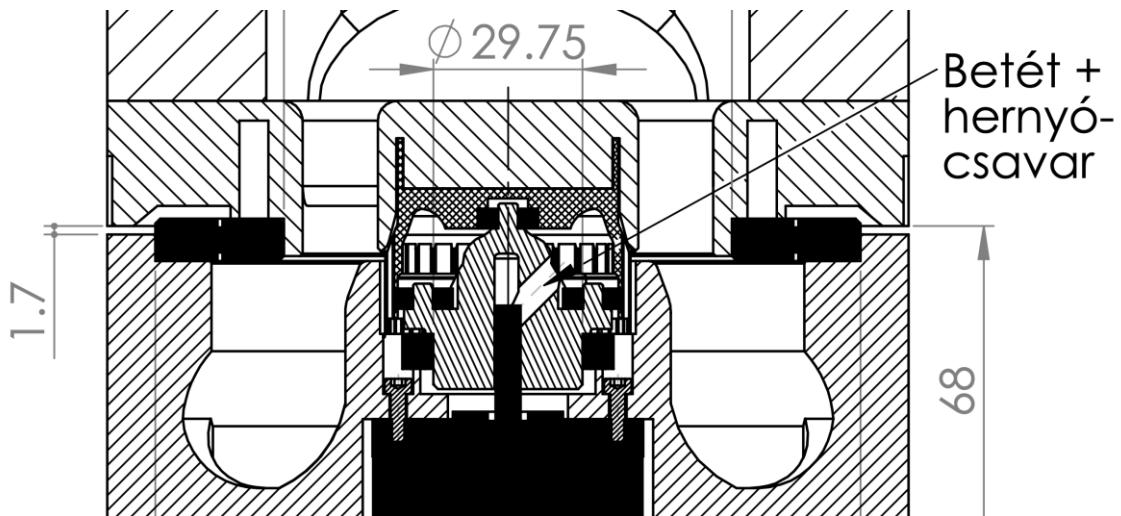
A *hullámhajtás* létrehozására alkalmas hullámhajtómű forgó mozgás és nyomaték folyamatos átvitelére alkalmas speciális fogazott hajtás. Alapvetően abban különbözik a konvencionális fogaskerék hajtásuktól, hogy ebben a rendszerben az egyik elem rugalmas alakváltozása szolgál a működés alapjául. Ezt az ún. hullámgenerátor állítja elő, amin működés közben az elfordulás mellett deformációs hullám is végigfut [29].

A hajtás legfőbb előnye a kis térigényű nagy áttétel megvalósítása. Ez lehet akár például 1:150-hez is, ami lehetővé teszi a beavatkozók teljesítményének hatékonyabb kihasználását és a robotkar nagyobb erővel, pontossággal történő manőverezését. Az alkalmazás további előnyei közé tartozik a könnyű tömeg, a csendes működés, a megbízhatóság és a kisebb fogyasztású motorok alkalmazhatóságának lehetővé tétele.

Esetben a hullámhajtómű minden alkatrésze saját tervezésű és nyomtatással előállított. Azért a nagyobb terhelésű alacsonyabb indexű tagok csuklóiba helyezem ezeket, mert egyszerűbben áll rendelkezésre elegáns hely, másrészről szeretném kihasználni a hullámhajtómű önzáró tulajdonságát.

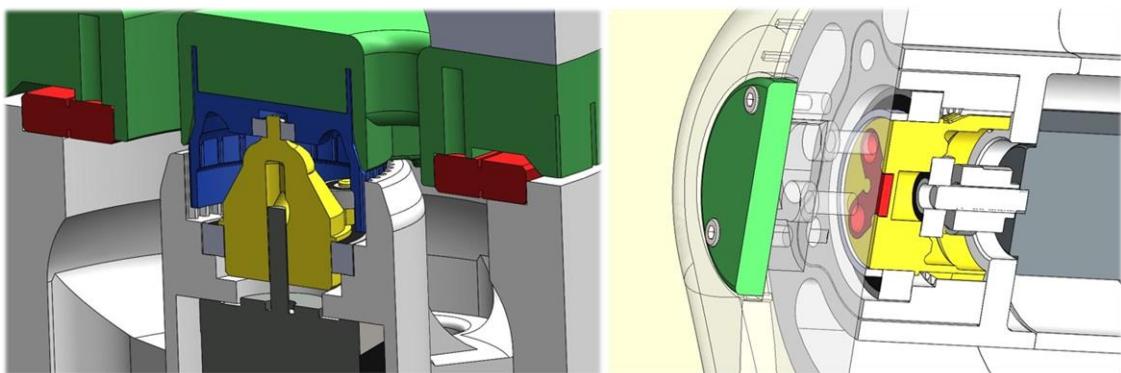
### 3.4.1 Az általam alkalmazott hajtómű

Mindegyik csuklópont úgy lett kialakítva, hogy belső részében elférjen a forgatást létrehozó léptetőmotor (**14. ábra**). A méretkülönbségek miatt kétfélel alakítottam ki (**15. ábra**).



14. ábra - Az első csuklópont környezetét jellemző metszeti sík

A fogak kialakítását egy általános fogazat profiljának megalkotásával, majd azon skicc körkörös kiosztásával végeztem. A hullámhajtómű áll egy merev gyűrűből, egy rugalmas hüvelykből, egy hullámgenerátorból, valamint az ezeket csapágyak segítségével összefogó merev keretrendszerből. A működőképességhez figyelmem kellett a merev és flexibilis elemek közötti megfelelő mozgástér kialakítására.



15. ábra - Különböző kialakítású hajtóművek metszetei

A beavatkozók zárt térben történő kialakítása miatt szellőző nyílásokat alakítottam ki az érintett tagokon.

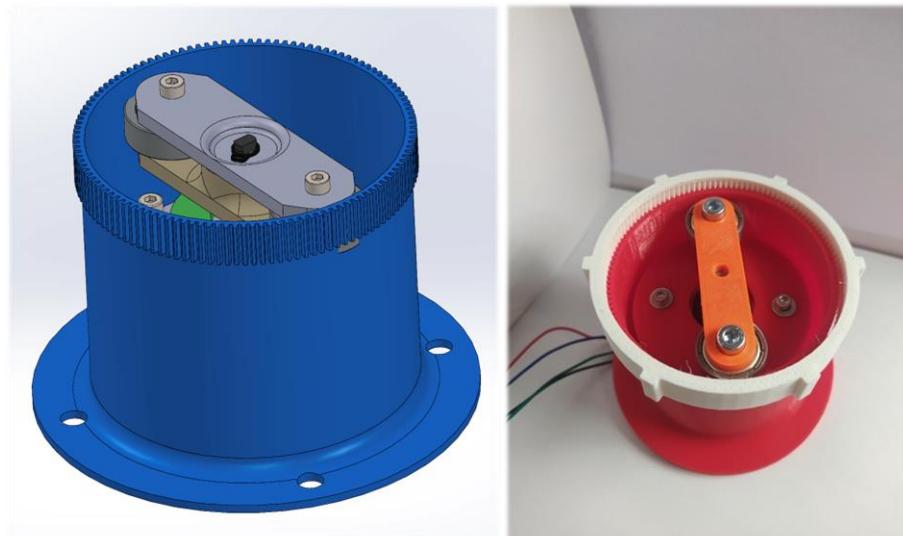
### 3.4.2 Kardinális szempontok és tesztek

A hajtóművet alkotó komponensek "szűk keresztmetszete" a rugalmas hüvely, melynek a tesztek alapján a következő paramétereit lehetséges módosítani:

- falvastagság,
- anyagminőség,
- fogmagasság.

Ezen - ugyan egymástól is függő - szempontok közül is vízválasztó az anyagminőség. Azt a kérdést, hogy hagyományos szilárdságú vagy valamilyen flexibilis anyagból nyomtassam ezt az elemet, tesztekkel döntöttem el.

Az **16. ábrán** látható PLA anyagú teszt-hajtómű 1:70-es lassító áttétel megvalósítására képes, viszont nagy mérete és áttetele miatt koncepcióját végül elvettem.



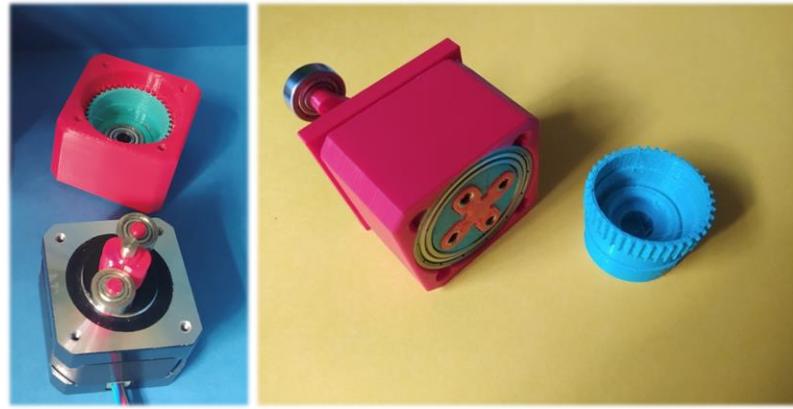
**16. ábra - A fejlesztés kezdeti szakaszának hajtóműve és próbája**

Egy bizonyos méret alatt és egy bizonyos relatív falvastagság fölött nem képes a keményebb, kevésbé rugalmas PLA<sup>7</sup> anyagú hüvely megfelelően alkalmazkodni a deformációs hullámhoz, valamint korábban produkál fáradásos törést. Ennek alternatívája a flexibilis anyagú kivitel (**17. ábra**), ami könnyebben alakváltozik, cserébe pontosabb tervezést igényel és rövidebb élettartamot produkál a radiális reakcióerők okozta összepréselhető fogak és a *delaminációs jelenség*<sup>8</sup> okán.

---

<sup>7</sup> PLA: politejsav anyagú filament.

<sup>8</sup> Delamináció: a rétegszétválás vagy rétegleválasztódás folyamata.



**17. ábra - A hullámhajtómű tesztelésére gyártott motoros és kézi egység**

A fenti szempontok minden a kereskedelemben kapható fém hüvelyes hullámhajtóművek műanyag változatai. Amennyiben a hullámhajtómű nem lenne megfelelő az általam kívánt méretben és/vagy paraméterekkel, így lehetséges a szintén nagy megvalósítható áttételekre képes cikloid hajtással történő kiváltása [23].

### 3.4.3 Megvalósítható áttétel

Általánosságban kimondható, hogy a hullámhajtóművek esetében a *módosítás* számítási módja és az, hogy a hajtó és hajtott oldal forgásirányára megváltozik-e, attól függ, hogy a komponensek közül melyiket rögzítem és melyiket választom forrás, illetve fogadó oldalnak. Három komponensről lévén szó ez  $3! + 2 = 8$  különböző módon lehetséges [29]. Ezen módok közül a robot geometriájához és a motorok elhelyezéséhez leginkább illeszkedő módot választottam, amikor a hullám generátor a bemenő, a flexibilis hüvely a kimenő oldal és a merev gyűrű pedig a rögzített elem.

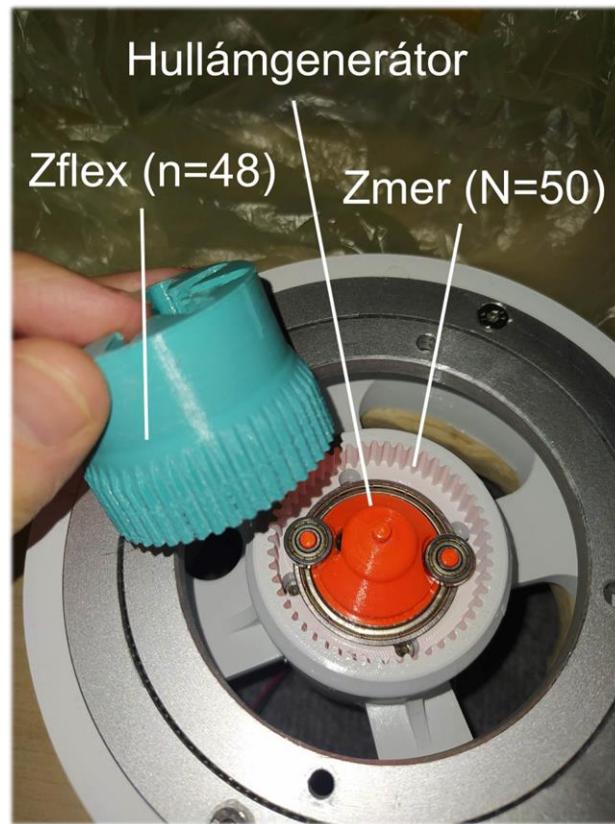
A hullámhajtómű áttételének univerzális számítása a következőképpen alakul:

$$i = \frac{z_{flex}}{z_{flex} - z_{mer}} \quad (2)$$

ahol,

- $i$  : az áttétel [-],
- $z_{flex}$  : a flexibilis elem fogszáma [-],
- $z_{mer}$  : a merev elem fogszáma [-].

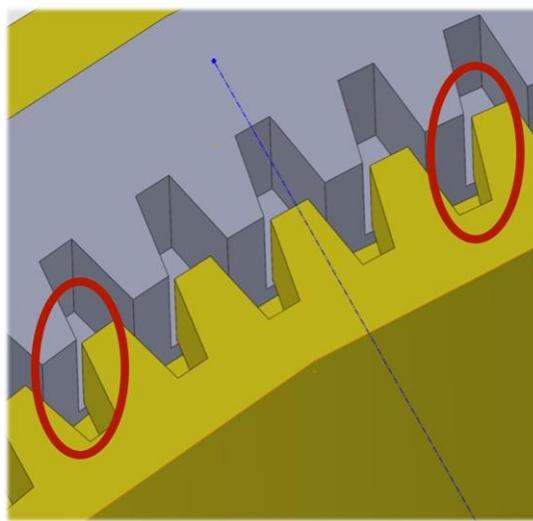
A 2. egyenletet felhasználva méreteztem a 18. ábrán látható elemeket.



18. ábra - Az 1. tag rotációját végző hajtómű fogparaméterei

A programozást leegyszerűsítendő behelyettesítve  $n_1 = 48$   $n_{2,3} = 40$   $N_1 = 50$ , és  $N_{2,3} = 42$  értékeket  $i_1 = 24$ ,  $i_{2,3} = -20$  lassító áttételeket jelentenek. A nevezők okozta előjelváltozás miatt a kisebbik hajtóműveknél megváltozik a forgásirány.

A méretezés során 2 fog eltolást alkalmaztam, hogy a hullámgenerátor okozta elliptikus torzítás nélkül a fogazat-párok egymáshoz képest folyamatos eltolásban legyenek azonos osztás esetén is (**19. ábra**).



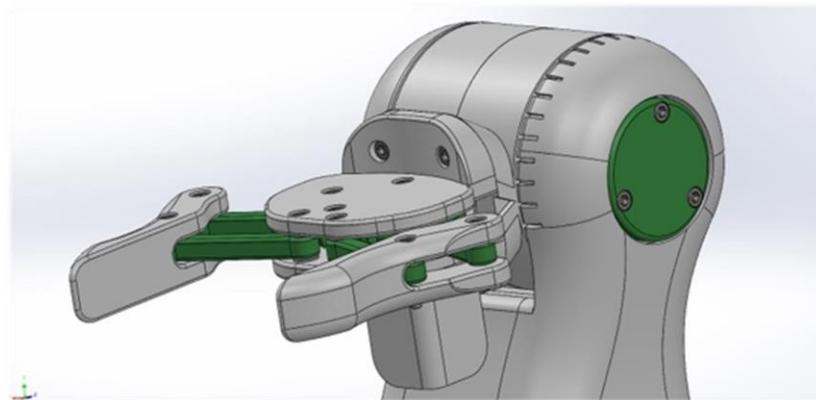
19. ábra - A fogszámkülönbség okozta rés lassú növekedése

Megfigyelés; a külső, merev és belső flexibilis darabok modellezéstechnikai szempontból szimpla eltolás és offset parancsok segítségével nem állíthatók elő.

A hajtás választása egyúttal azt is jelenti, hogy az első 3 csuklóban nem lesz szükséges - a későbbiekben részletezett - ún. *mikroléptetési mód* alkalmazására, valamint, hogy az ezek mentén történő rotációs szöghiba is az áttétellel arányosan osztódik majd.

### 3.5 A megfogó egység

A robotkar cserélhető végberendezéseként kétféle megfogót is elkészíték, egy univerzális párhuzamszorítású megfogó egységet (**20. ábra**) és egy a speciálisan a demonstrációhoz tervezett sakkfigura-emelő pofákkal szerelt egységet. Az előbbiek az elkészült változata a **21.ábrán** látható.



**20. ábra - A végberendezés modellje**



**21. ábra - Az felszerelt általános rendeltetésű megfogó**

A CAD modell alapján a két különböző megfogó lokális koordináta-rendszerben vett helyvektora különbözik:

$${}^5\underline{r}_{TCP} = \begin{bmatrix} 155 \\ 0 \\ 0 \\ 1 \end{bmatrix} \text{ és } {}^5\underline{r}_{TCP} = \begin{bmatrix} 98 \\ -85 \\ 0 \\ 1 \end{bmatrix} \quad (3)(4)$$

Bár gyakran a hasonló megfogók működtetését szervomotorok segítségével oldják meg, én egy Nema8 léptetőmotorral szereltem őket fel.

Az irodalomkutatás során a következő technológiájú megfogókkal találkoztam:

- mechanikus szorítópofák<sup>9</sup>,
- vákuumos megfogók,
- többkarú emelők ("sneak peek gripper"),
- polip-megfogók ("octopus gripper"),
- "jamming" szorítók<sup>10</sup>,
- *magnum megfogók*<sup>11</sup>,
- *servoelektronos megfogók*,
- *forgókaros* és *pantográf* karos robotmegfogók [30].

A párhuzamos megfogó pofák sakk-figura emeléshez alkalmas vízszintesen történő működtetését szoftveresen  $K_5$  és  $K_6$  bevezetésével és helyvektoraiak Z komponenseinek egyenlővé tételevel lehet elérni. A tesztek alapján a motorok tekintetében 90°-os szögfordulás van a teljesen nyitott és zárt állapotok között.

Felmerült, hogy érzékelővel látom el a végberendezést, hogy a mikroprocesszor visszajelzést kaphasson szorító-állapot felvételéről vagy akár a sikeres objektum mozgatásáról, viszont végül ezt a projekt egészét illetően elhanyagolhatónak ítétem meg, így elvetettem.

<sup>9</sup> Robot megfogó egység vagy berendezés: angol gripper.

<sup>10</sup> A *jamming megfogó* egy speciális légyomással vezérelt rugalmas anyagú megfogó egység, ami kiválóan felveszi az akár törékeny és szabálytalan tárgyak alakját is a megemelésük során [30].

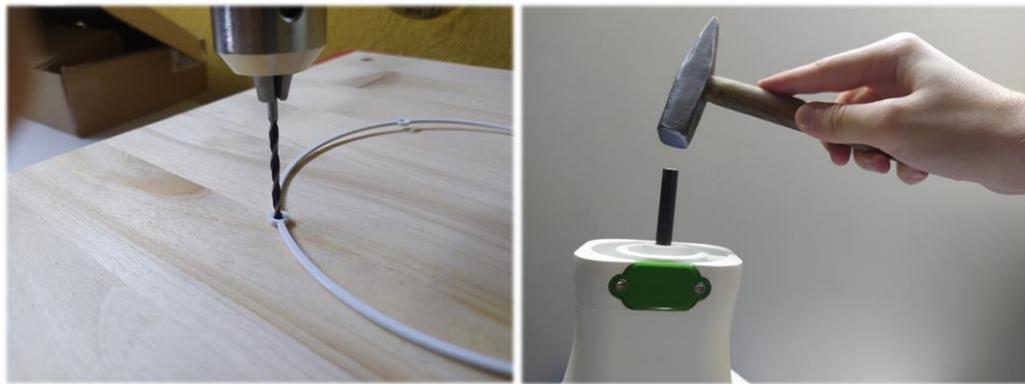
<sup>11</sup> A *magnum megfogók*, olyan precízen megmunkált mechanikus szerkezetek, amiknél a szorító erőt egy dupla ellenetés spirállal ellátott tengely hozza létre.

### 3.6 Összeszerelési lépések

A teljes hardverrendszer összeszerelési lépései a következő sorrendiséget követi:

- A léptetőmotorok elhelyezése a robotkar nyomtatott elemek üregeiben.
- A hullámhajtóműveket alkotó elemek összeállítása/felhelyezése.
- A szegmensek csavarkötésekkel történő szerelése.
- Az elektronikai doboz összeállítása.
- Az elektronikai doboz és a robotkar felszerelése az "alapelemre".
- A vezetékezés sematikus ábra szerinti bekötése.
- A szoftver élesztése és próbafüttatása.

Néhány komponens (**22. ábra**) előmunkálást, felszabályozást vagy egyéb eszközt igényel az összeállításhoz.



**22. ábra - Nyomtatott sablon használatával és szilárd illesztéssel történő lépések**

A leginkább sérülékeny elemek a flexibilis hüvelyek, melyek behelyezése a már forgástengely mentén rögzített hullámgenerátor lassú forgatása közben javallott. A mechanizmus állványzatát egy nyomtatott lábakon álló alaplap biztosítja, melybe minden vezeték alulról csatlakozik (**23. ábra**).



**23. ábra - Az alapra történő rögzítés műanyagbetétes csavarbiztosító anyákkal**

## 4 Villamos vezérlés tervezése

Ebben a fejezetben az elektronikai komponensek működtetésért felelős egységeket mutatom be, melyek részben kereskedelmi, részben saját tervezésű komponensekből épülnek fel.

### 4.1 Hardverkomponensek és kommunikációs protokollok

A robotkar működtetéséhez szükséges *hardver* gyakran különböző protokollokat használ a működése során.

A *kommunikációs protokoll* egy olyan szabványosított kommunikációs eljárás, amely meghatározza az elektronikus eszközök közötti kommunikációt. Ezek a protokollok standardizált formában rögzítik az adatok átviteli módját, formátumát, kódolását, dekódolását, valamint a kinyert információra vonatkozó integritási előírásokat [31].

Az általam alkalmazott egységek listája a következőképpen alakul:

| Egység  | Típus  |
|---|--|
| Kapacitív érintőképernyő                                    | FT5406   |
| Képernyő adapter panel                                      | E305654  |
| ↑ külső egységek ↑<br>Mikroprocesszor<br>↓ belső egységek ↓ | ↑ külső egységek ↑<br>RASPBERRY PI 3 B<br>↓ belső egységek ↓ |
| Motorvezérlő  | DRV8825  |
| Bipoláris léptetőmotorok                                    | 42HS48, 20HS33,<br>28HS32, 17HS40                            |
| Végálláskapcsolók   | DM-03S-7P, MSW 13  |
| Tápegység   | 5.1 V / 2.5 A és 12 V / 2.5A                                 |

1. táblázat<sup>12</sup> - A robotkar hardverkomponensei

A 24. ábra a hardverkomponenseket és a jelfolyamok kapcsolatát ábrázolja, ahol a robotkar elemeket és a motorvezérlőket duplikálva kell értelmezni. A kommunikáció egyszerűsítése érdekében került a tápegységtől (0.) (3.) vagy belső egységek<sup>13</sup> által

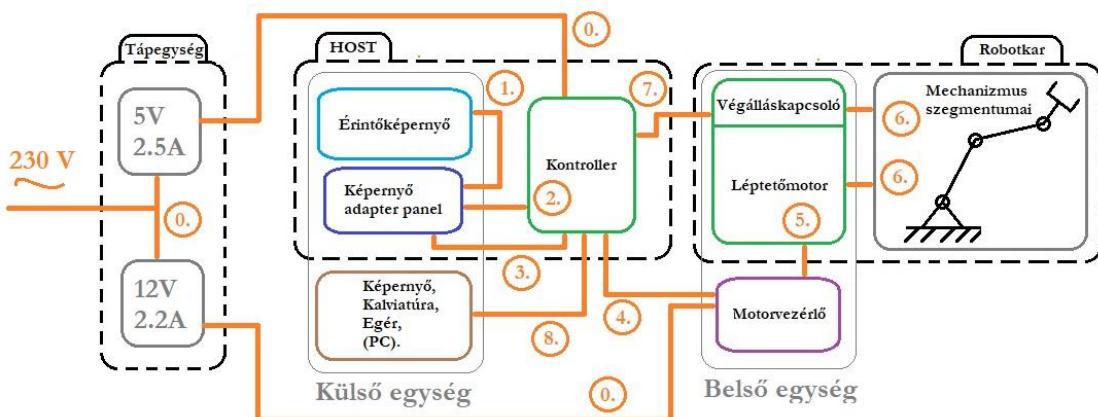
<sup>12</sup> Táblázatban a "↓, ↑" szimbólumok a külső és belső egység felé történő csatlakozás irányát jelölik.

<sup>13</sup> Belső egység: az irányítási körben a mikrokontrollertől - jelfolyam szerint - hátrébb lévő komponensek.

használt digitális jelektől különböző működtetést. A főcsoportok közül a *Tápegység* vagy a *HOST* elemeinek meghibásodása esetén működésképtelen válik a *robotkar*.

A belső komponensek között általánosságban véve digitális jelek (7.) segítségével történik a kommunikáció, ami nem igényel külön kódolást és dekódolást, közvetlenül hat például az impulzus-áram a léptetőmotor (5.) mechanikai elemeire. A kontroller panel és a vezérlő között is csupán időben változó szélességű egyenáramú impulzusszélesség-modulált vezérlőjelekkel kommunikál (4.). A beavatkozók a szegmentumokkal pedig minden össze (6.) mechanikai kapcsolatot alkot.

A kontrollerbe befutó külső egységek<sup>14</sup> azonban már használnak kommunikációs protokollokat. A képernyő és a mikrokontroller között (1.) DSI<sup>15</sup> protokoll, míg az adapter panel és a RASPBERRY PI között (2.) pedig a SDA1<sup>16</sup> és SCL1<sup>17</sup> GPIO-okon megvalósított I2C<sup>18</sup> protokoll működik. Ezeken felül az esetlegesen csatlakoztatott egér és klaviatúra (8.) USB-protokollt<sup>19</sup> alkalmaz. Továbbá lehetséges még - kifejezetten a fejlesztés szakaszát elősegítendő - tetszőleges képernyő (8.) csatlakoztatása HDMI-protokoll<sup>20</sup> segítségével.



## 24. ábra - Hardverkörnyezet blokkdiagramja

<sup>14</sup> Külső egységek: az irányítási körben a mikrokontrollertől előrébb lévő komponensek.

<sup>15</sup> DSI: Display Serial Interface, soros, szinkron kommunikációs protokoll, képernyőkhöz.

<sup>16</sup> SDA1: Serial Data Line, azaz szerializált adat vonal.

<sup>17</sup> SCL1: Serial Clock Line, azaz szerializált órajel vonal.

<sup>18</sup> I2C: Inter-Integrated Circuit, kétirányú kommunikációs protokoll.

<sup>19</sup> USB: Universal Serial Bus, azaz univerzális soros busz.

<sup>20</sup> HDMI: High-Definition Multimedia Interface. magasfej

## 4.2 A mikrokontroller

A piacon számos modell elérhető különböző specifikációkkal, kiegészítőkkel, bővítő és adapter panelekkel. Az általam alkalmazott mikroprocesszor kiválasztásánál a következő szempontokat vettet figyelembe:

- viszonylag olcsón beszerezhető,
- PWM jelek egyidejű előállításának képessége,
- minimum 500 MB RAM,
- DSI képernyő csatlakozó,
- rendelkezzen a megfelelő törpefeszültségű kimenetekkel,
- HDMI kimenet a programalkotáshoz,
- Python programozási nyelv támogatása,
- megfelelő számú GPIO kimenet.

Az irodalomkutatáskor számos mikrovezérlő kártyát, CNC-vezérlőt és 3D nyomtató processzorát számításba vettetem, mint például az *Arduino Duo*, *Nano*, *Mega*, *Duet2*, *Adafruit*, *BeagleBone PRU*, *Wio Terminal D51R*, *STM32F042*, *Duet 2 Maestro*, *Smoothieboard*, *ESP32* vagy a *BigTreeTech* vezérlőit. Ezek közül azonban a legtöbb túlzottan szűk felhasználásra lett tervezve, így praktikussági okokból érdeklődésem hamar átterelődött a hobbisták körében elterjedt két márka, a *RASPBERRY PI*<sup>®</sup> termékekre és az *ARDUINO*<sup>®</sup> fejlesztőkörnyezetre.

Az *ARDUINO*-val végzett munka esetén könnyen programozhatunk a C++ nyelv használatával és jó teljesítményt érhetünk el a valós idejű feladatokban [32].

A *RASPBERRY PI* valamennyivel erősebb és költségesebb, mint az *ARDUINO*, és egy teljes értékű számítógépet képvisel, amely Linux-alapú operációs rendszert futtat. A *RASPBERRY PI*-t általában Python vagy más magas szintű nyelven érdemes programozni, melynek előnye, hogy könnyen kezelhető és jól hasznosíthatók az *IoT*<sup>21</sup> projektekben. A *RASPBERRY PI*-ok sokkal rugalmasabbak, mint az *ARDUINO*-ok. Számos be- és kimenetek támogatása mellett, támogatják az Ethernet, a Wi-Fi és a Bluetooth funkciókat is. Ezenfelül számos bővítő panel beszerezhető hozzájuk, mint például kijelzők, I2C portok érzékelőkhez vagy multiplexerek [33].

---

<sup>21</sup> IoT: a "Internet of Things" szakterület angol nevének rövidítése.

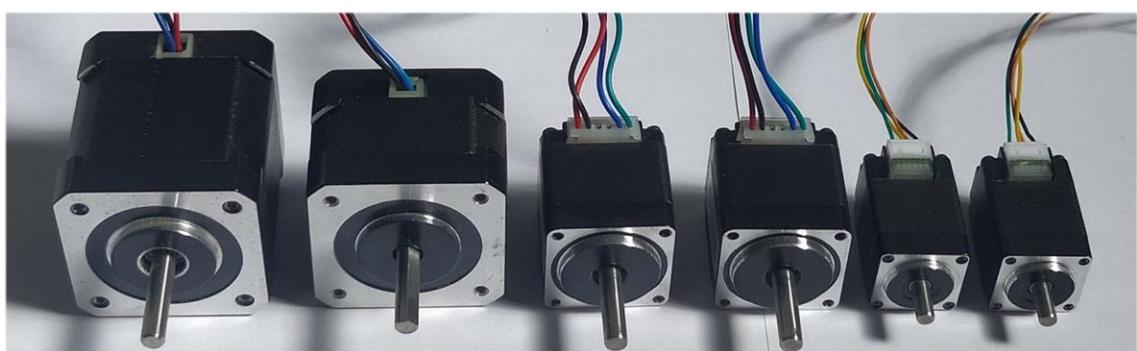
Mindezeket summázva, a külső egységekkel történő kompatibilitás, a RAM és a magasszintű nyelvek támogatása miatt az általam választott mikroprocesszor egy *RASPBERRY PI 3 Model B*. A kontroller 40 pin-es GPIO porttal rendelkezik, melynek részletes feliratozása a függelék [4. mellékletében](#) olvasható, azonban az összehasonlítás során az generálható PWM jelek száma volt a *leginkább meghatározó tényező*.

A fejlesztés során az eszközre a *Raspberry Pi Foundation* saját operációs rendszerét, a *Raspbian*-t és a szintén *Linux* alapú *Ubuntu Mate*-et installálok. Kiterjesztve ezen operációs platformok egyúttal a robotkar potenciális *firmware*-ét is jelentik.

A RASPBERRY PI 3 B interfészait a függelék [2. melléklete](#), míg az előző fejezetben tárgyalt képernyő adatait a függelék [3. melléklete](#) tartalmazza.  
[<https://www.rs-components.com>] [[hestore.hu](https://hestore.hu)]

### 4.3 Léptetőmotorok és vezérlőik

A szegmentumok mozgatásához szükséges forgómozgás előállításához a pontos pozícionálás és a széles *felhasználási területek*<sup>22</sup> miatt *léptetőmotorokat* választottam beavatkozó gyanánt. A méretezés során a Nema17, Nema11 és Nema8 motorok<sup>23</sup> kerültek számításba a csuklópontokból eredő különböző hely- és teljesítményigények miatt [34]. Fontos szempontom volt még a tengelycsunk lapolással rendelkező kialakítása, mely hernyócsavaros felfogatás esetén praktikus. Az általam választott kétfázisú bipoláris motorok a **25. ábrán** láthatók.

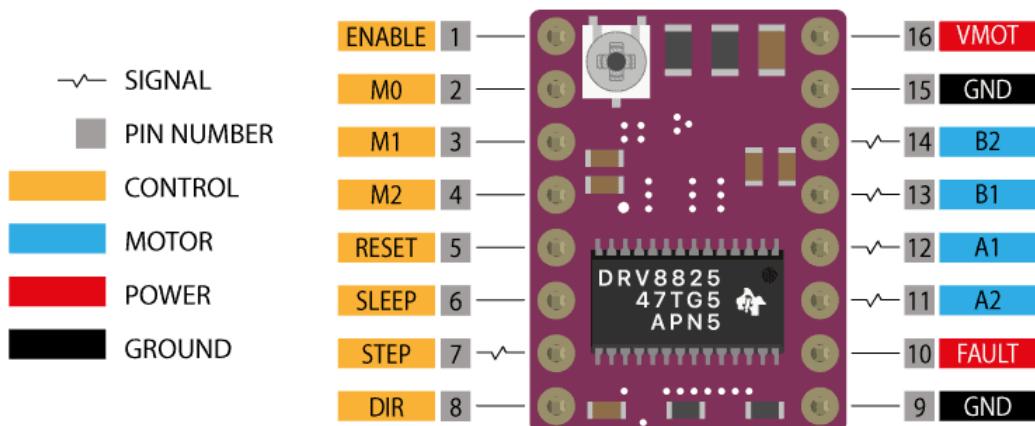


25. ábra - NEMA 17-42HS48, NEMA 17-20HS33, NEMA 11-28HS32, NEMA 8-17HS40 egységek

<sup>22</sup> Léptetőmotorokat gyakran alkalmazzuk CNC megmunkáló egységekben, lézervágókban és 3D nyomtatókban, ahol a pontosság alapvető elvárás.

<sup>23</sup> NEMA: National "Electrical Manufacturers Association", ami meghatározza a méretet és a felfogatást.

Ezen motor fajta a nagy nyomaték, pontosság mellett viszonylag költséghatékony, viszont magas fogyasztással rendelkezik és vezérlést igényel. Hárrom népszerű motorvezérlő modellt vizsgáltam meg, a A4988, TMC2130, és a DRV8825 vezérlőket, melyek közül az utóbbira esett választásom. A A4988 és a DRV8825 hasonló képességekkel rendelkeznek, míg a modernebb TMC2130-at csupán érzékelőzárás, csökkentett zajszint és motorhatékonyság növelés esetén lenne érdemes használni [35]. A **26. ábrán** a motorvezérlő kezelhető tüskéi találhatók, melyekhez tartozó funkciókat hosszasan teszteltem (**28. ábra**).



26. ábra - RDV8825 lábkiosztása - [<https://www.studiopieters.nl>]

A motor tápellátását a VMOT és a GND (15) (16) lábak biztosítják. A B1, B2 és A1 A2 (11-14) kivezetések<sup>24</sup> a bipoláris motor tekercseinek kivezetései. Az ENABLE és FAULT (1) (10) pinek a léptetőmotor tiltására vagy engedélyezésére, valamint a hibajelzési és védelmi funkciók kezelésére valók. Mindkettő negált jelekkel operál és logikai szintjelekkel vezérelhetők. Amíg a Fault a kontrollerről kap 5.5 V-ot, addig az ezen áramkör földje a GND (9). Az M0, M1, M2 (2-4) tüskékkal a léptetési felbontás, azaz az ún. *mikroléptetési mód* adható meg. A RESET és a SLEEP (6) (7) a vezérlő alaphelyzetbe állítását, valamint az energiatakarékos alvó állapotba helyezését teszik lehetővé. Ezeket egy *jumper*-el áthidalva, egyszerre, egy logikai szinttel vezérlem. A STEP (7) a mikrovezérlő panelről érkező PWM jeleket fogadja a léptetés vezérléséhez. A DIR (8) a forgás irányát határozom meg szintén logikai szintekkel.

A megbízhatóságán, széles feszültségspektrumán, és helytakarékos kialakításán kívül még a fent említett számos *mikroléptetési módozata* (*mikrostepping*) miatt is

---

<sup>24</sup> A Bipoláris léptetőmotor kivezetéseinek párosítása tekercsenként: fekete-zöld, vörös-kék.

népszerű. A finomabb felbontású léptetési funkció (**2. táblázat**) lehetővé teszi a léptetőmotorok számára az egész számú lépésekkel eltérő -, akár két szögperces - pozicionálást, valamint az üzemelés során a finomabb rezgést és zajszintet. A léptetési módot gyakran úgy implementálják, hogy szoftveresen akár egyetlen forgatási folyamat közben is módosítani lehessen azt [36].

| M2       | M1       | M0       | Lépési mód                             |
|----------|----------|----------|--|
| <b>0</b> | <b>0</b> | <b>0</b> | Teljes lépés ( $1.8^\circ/1$ )         |
| <b>0</b> | <b>0</b> | <b>1</b> | 2 mikrolépés/lépés ( $1.8^\circ/2$ )   |
| <b>0</b> | <b>1</b> | <b>0</b> | 4 mikrolépés/lépés ( $1.8^\circ/4$ )   |
| <b>0</b> | <b>1</b> | <b>1</b> | 8 mikrolépés/lépés ( $1.8^\circ/8$ )   |
| <b>1</b> | <b>0</b> | <b>0</b> | 16 mikrolépés/lépés ( $1.8^\circ/16$ ) |
| <b>1</b> | <b>0</b> | <b>1</b> | 32 mikrolépés/lépés ( $1.8^\circ/32$ ) |

2. táblázat - A DRV8825-tel megvalósítható mikroléptetési módok [[5. csatolmány](#)]

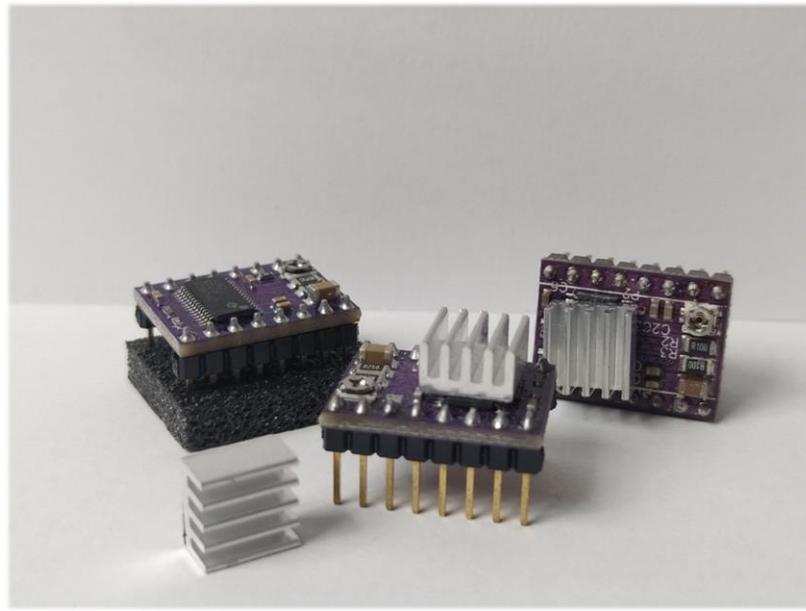
Megfigyelésem szerint a vezérlőn történő léptetési módozat nyákon történő megadására -, mely M0, M1 és M2 szintjelezésével történik - három lehetőségem van. A vezetőszigetek kialakításával, ami később nem módosítható. Ritka módosítási igények esetén DIP kapcsolók használatával. Ebben az esetben módosítás esetén a léptetések számát is módosítanom szükséges a kód részletekben. Illetve tisztán GPIO-okon keresztül, amikor bármikor adott a szoftveres modifikáció lehetősége. Én a nyák tervezése során csuklótól függően minden háróm módszert alkalmazom.

A vezérlést igénylő léptetőmotorok mozgatása technikailag H-híd segítségével valósul meg. A *Texas Instruments*® által gyártott, *teljes hidkapcsolással* szerelt DRV8825-ös IC tokozását, működését, és funkcionális blokkdiagramjait a függelék [5.csatolmánya](#) szemlélteti.

Az IC számára szükséges vezérlőjelet általában egy időzítő (*timer*) modul vagy PWM generátor modul állítja elő a mikrokontrollerben, amely meghatározza a jel periódusának hosszát, az adott időintervallumon belüli impulzusok számát és a jel kitöltési tényezőjét [36].

Célszerűen minden chipre és IC-re - köztük a vezérlőkre is - szereltem az áramlási iránynak megfelelő hűtőbordázatot (**27. ábra**), előkészítve őket az esetleges hődisszipációból eredő hőmennyiség passzív hűtések elvezetésére. A tesztek során a

hűtőbordákon keletkező hőmérséklet ellenőrzésére *K*-típusú hőelemet<sup>25</sup> használtam. A vizsgált időszakban ez sosem haladta meg a 52 °C-ot.



**27. ábra - Vezérlő modulok a tokozásra ragasztott hűtőtestekkel**

A vezérlők bal felső sarkában egy *potenciométer* található, ami az áramkorlátozó képesség (5) beállítására szolgál. Ez a limit biztosítja, hogy a léptetőmotor áramfelvétele ne haladja meg a maximális értéket, így gondoskodva a motort és a vezérlő túlmelegedéssel és túlárammal szembeni védelméről.

Az áramkorlátozó képesség beállítása úgy zajlik, hogy amultimétert a potenciométer és a földelés közé kötöm, majd a potenciométer jobbra forgatása növelem a referenciaáramot, míg balra forgatásával csökkentem azt. Bekapcsolom a tápot majd addig növelem az áramkorlátozó értéket, amíg az a motor adatlapján megadott referencia áramból számolt feszültségértékét el nem éri a következő egyenlet alapján:

$$v_{ref} = \frac{I_{max}}{2} \quad (5)$$

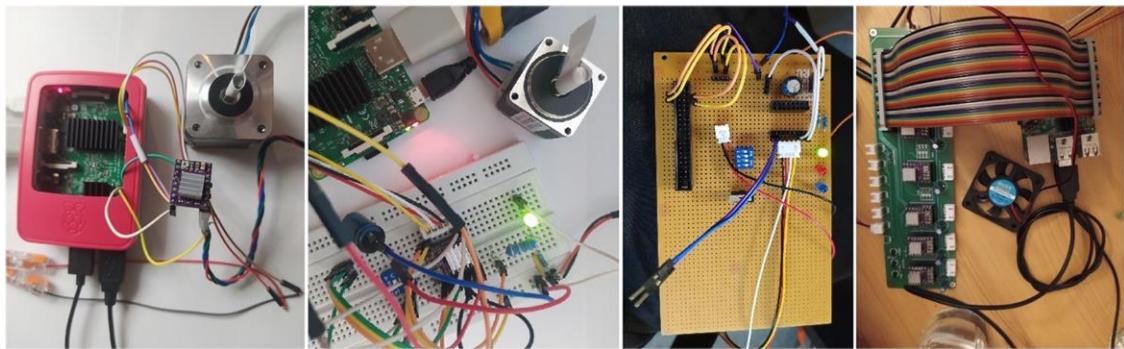
Ahol:

- $I_{max}$  - a motor típusától függő áramfelvétel [A],
- $v_{ref}$  - a beállítandó megengedett feszültségérték [V].

---

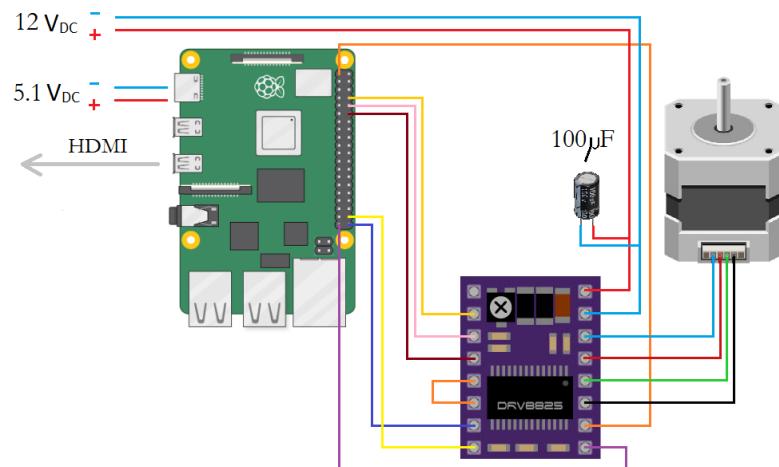
<sup>25</sup> K-típusú hőelem: NiCr Ni ötvözettel kétvezetékes termoelektronos hatáson alapuló hőmérséklet-érzékelő.

A vezérlő rendelkezik magas árammal szembeni védelemmel, ezért szerencsére a tesztek során (**28. ábra**) nem sikerült egy áramkört sem túlterhelnem.



**28. ábra - A beavatkozó-elektronika teszteinek "evolúciója" a vezérlőtől az központi panelig**

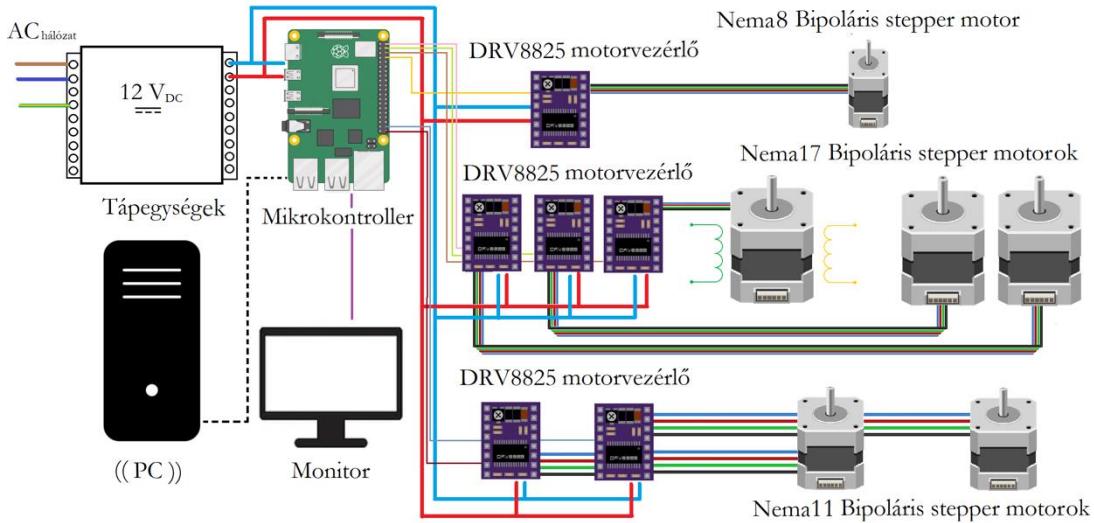
Eredményül adódott, hogy a motorok csupán egy viszonylag szűk áramtartományon belül működnek. A teszteket követően a bekötések a **29. ábra** alapján valósultak meg. [37]



**29. ábra - Példa a vezérlő egy lehetséges bekötésére**

Leggyakoribb bekötési hiba az volt, hogy a különböző referenciaáramok motoronként rosszul voltak beállítva, vagy a RESET és ENABLE tüskék nem a megfelelő bináris jelet kapták. Ezek zöme méréssel és újraindítással kezelhetők voltak, azonban kiderült, hogy a rendszer érzékeny lehet a működés közben végrehajtott változtatásokra és nem is egyformán reagál rájuk. Továbbá úgy találtam, hogy a túl magas frekvenciájú vezérlőjelek általi rotációt már nem képes kezelní és végrehajtani az eszköz.

A berendezés összes motor-csoporttal és vezérlőikkkel a **29.ábra** szerinti kapcsolatban állnak.



30. ábra - Berendezés teljes kapcsolati sémája

## 4.4 Szenzorika és vezetékezés

Egy adott szegmentum állásának ismerete - és ezzel a végpont tetszőleges pozicionálása - elengedhetetlen aspektus a robotkar működéséhez. Ehhez a robotokban használatos különböző szenzorokat kellett áttekintenem és tesztelnem.

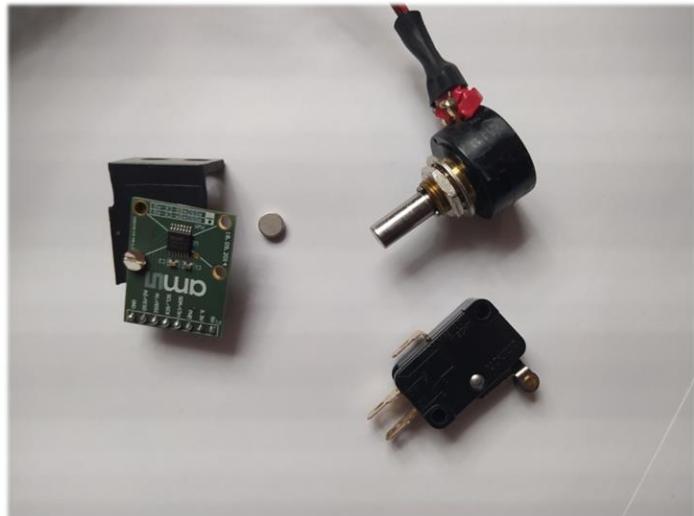
Az alkalmazni kívánt szenzor kiválasztását a csoportosítási lehetőségek áttekintésén keresztül közelítettem meg az irodalomkutatás során. A környezettől való információ felvételének módja szerint lehetnek *érintkezéses* és *érintkezés nélküli*, míg a kiszolgált technológiához való kapcsolódásuk szerint pedig lehetnek *belső* vagy *külső* szenzorok. Belső szenzorok például az *útmérők*, *fordulatszám-érzékelők*, *erő-* és *nyomatékszenzorok*, külső szenzorok pedig a mechanikai *kapcsolók*, *optikai kapcsolók* (*optokapuk*) és egyéb érintkező nélküli szenzorok. Az átalakító elemtől és a fizikai hatáselvétől függően lehetnek *aktív* és *passzív* átalakítók [3]. Az információ továbbítása történhet elektronikus vagy elektromechanikus úton.

A fentiek alapján én három különböző szenzor típust vizsgáltam meg (31. ábra):

- *mágneses pozíció szenzor (Hall-szenzor)*,
- *forgójeladó (abszolút enkóder)*,
- *mechanikai végálláskapcsoló*.

A mágneses pozíószenzorok hátránya, hogy jellemzően dekódolást igénylő PWM kimenettel rendelkeznek, valamint, hogy a chip és a mágnes nagy pontosságú egylengelyűséget igényel. Ezt biztosítani a nyomtatott alkatrészek tűrései mellett nehéz feladat lenne.

A forgójeladók közül egy abszolút pozíció megadására képes szenzor nyújtaná a legtöbb információt, amivel a HOME-pozíció nullázási protokollja is elhagyható lenne. A forgójeladók output jelei jellemzően bináris jelek vagy Gray-kód<sup>26</sup>.



**31. ábra - Vizsgált érzékelők (mechanikai kapcsoló, mágneses Hall-jeladó és abszolút enkóder)**

Az általam tervezett berendezés peremfeltételei nem indokolnák kimondottan bonyolult érzékelőrendszeret és jelfeldolgozást, így az általam választott érzékelő típus a végálláskapcsoló. Ez aktív, külső érzékelőnek minősül, azon belül is mechanikai kapcsoló, ami ún. bináris érzékelést tesz lehetővé. Azt jelzi, hogy egy adott mozgási tartományban a vizsgált tag elérte-e a tervezett végállapotát vagy sem.



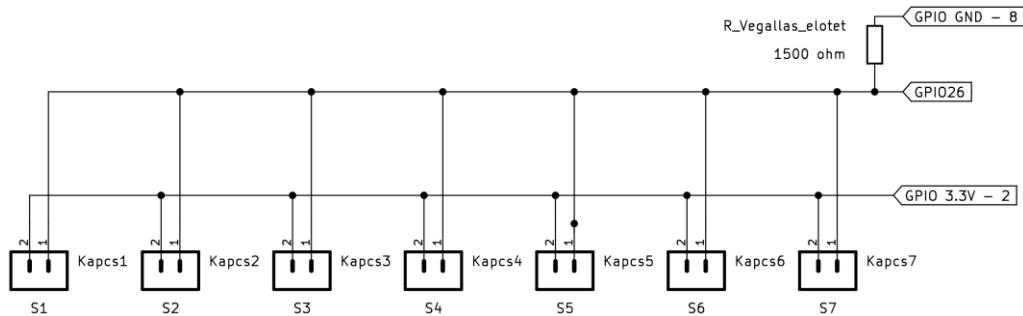
**32. ábra - A kapcsolókhöz és mozgatószervekhez vezető kábelezés nyomvonala**

---

<sup>26</sup> A Gray-kód egy egylépéses kód, ami azt jelenti, hogy az egyik pozícióból a következőbe, vagy megelőzőbe lépve a kódban csak egy bit változik. Az így kódolt tárcsák kiolvasásánál esetlegesen fellépő hiba lényegesen kisebb, mint a bináris kódolásúnál, tehát megbízhatóbb pozíciómérést eredményez.

Ezen érzékelő nem működőképes egyéb - például a HOME-pozíció - szoftveres megvalósítása nélkül. Továbbá az egyszerű kapcsolás és az alacsony GPIO port igény miatt kevés vezetékezés mellett megépíthető, amivel a szerelés is tovább egyszerűsíthető. A **32. ábra** bemutatja a vezetékezés számára modellezett utakat a robotkaron belül.

A kapcsolás a **33. ábrán** vázolt *huzalozott vagy kapcsolattal* lehetséges [26].



**33. ábra - A végálláskapcsolók kapcsolása [5. melléklet]**

Ezzel a kapcsolással a kontroller használni kívánt GPIO-k száma is csökkenthető, mivel egyetlen vezetékről valósítottam meg a végálláskapcsolók aktiválását. Ez azonban ahhoz vezet, hogy egyrészt egyszerre nem nullázhatók a különböző csuklók, másrészt - mivel nem lehet tudni melyik tag kerül végállásra, szoftveres kombinációval<sup>27</sup> lehet csak beazonosítani a magas jelszintre került kapcsoló kilétét.

## 4.5 Központi vezérlőpanel

A előzőekben bemutatott elektronikai komponensek összekötésére egy állandó kapcsolatot biztosító hordozó létrehozása szükséges.

A geometriai viszonyok jellegéből és a beavatkozók számából kiindulva jónéhány készen kapható vezérlő panelt megvizsgáltam, melyeket 3D nyomtatókhoz, kisebb méretű CNC berendezésekhez és lézervágókhöz forgalmaznak. Ezek azonban egyenesvonalú és nem pedig szögelfordulásokban mérhető mozgásokra lettek elsősorban tervezve, valamint számos számomra körülmenyesen hasznosítható interfésszel és funkcióval kerültek forgalomba. Továbbá van közöttük olyan, ami készre tervezettségből adódóan csupán korlátozottan támogatnak programnyelveket és keretrendszerket. A kutatás szakaszában általában ezeknél a termékeknél a következő kész platoformokkal és *firmware*-ekkel találkoztam: *Marlin*, *DualWebControl*,

---

<sup>27</sup> "Szoftveres kombináció": a végálláskapcsolók és a léptetőmotor GPIO-k END kapcsolatának vizsgálata.

*RepRapFirmware*, *OctoPrint*. A kérdéses interfészek alatt a szabályozható "hotend-hűtőventilátort", hőszenzorokat, BLTouch szenzorokat, fűtőpatronok csatlakozóit vagy épp a WiFi és Bluetooth egységeket értem, amikre többnyire a robotkarhoz nincs szükség.

Ilyen konkrét egységek például a *RUMBA32*, a *SKI 3*, a *Manta M8P* vagy az "*Octopus*" panelek, melyeket a *BigTreeTech*<sup>©</sup> gyárt és forgalmaz.

Összességében a már korábban kiválasztott DRV8825 vezérlőre való tekintettel, valamint, hogy ezen panelek egyike se lett volna költséghatékony vagy megfelelően kihasznált, a piacon kapható HAT<sup>28</sup> bővítő kártyák helyett, az összes szükséges foglalatot és aljzatot magába foglaló bővítő - pontosabban: integráló - nyák áramkör (*PCB*) megalkotása mellett döntöttem.

Az áramkör az alábbi elemek csatlakoztathatóságát hivatott megvalósítani:

- a tápegység (12 V DC),
- visszajelző LED-sor,
- érintőképernyő kijelző,
- végálláskapcsolók,
- RASPBERRY PI 3 B +,
- DRV8825 motorvezérlők,
- bipoláris léptetőmotorok,
- DC ventilátor,
- vészhelyzeti / RESET<sup>29</sup> nyomógomb.

A visszajelző LED-sor színjelei:

- zöld: rendszer áram alatt, hűtés aktív (*System-standby LED*),
- kék: processzor aktív, robot mozgatásra kész (*Kontroller-standby LED*),
- vörös: motorvezérlők egyike diszfunkciós (*Motorhiba-LED*).

A *motorhiba-LED* arról ad visszajelzést, hogy valamelyik motor forgatása nem valósult meg, nem detektálja sem a hiba mivoltát, sem pedig, hogy melyik motor produkál diszfunkciót. Az ilyen jellegű problémát három módon dolgozhatjuk fel. A manuális vezérlő-RESET gomb megnyomásával, a szoftveres digitális *REBOOT-gomb* megnyomásával, vagy esetenként a program újbóli futtatásával.

---

<sup>28</sup> HAT: *Hardware Attached on Top* rövidítése, magyarra fordítva hardverre csatlakoztatható bővítő panel.

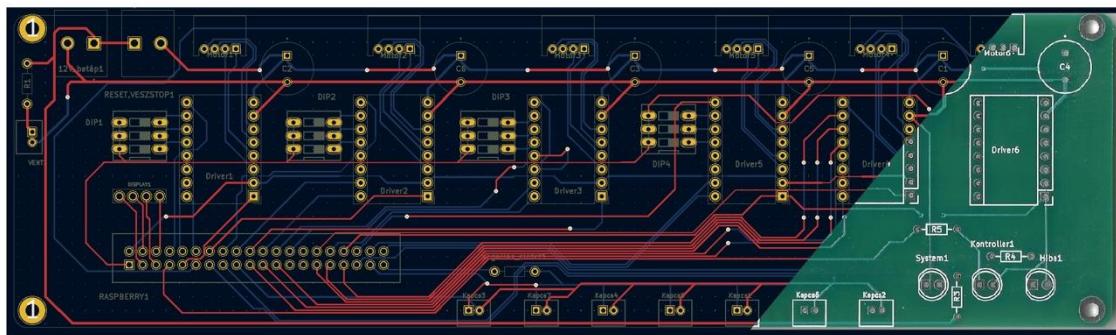
<sup>29</sup> RESET gomb bekötése: A léptetőmotorok vezérlőjének tápellátását szakítja meg.

Mindenekelőtt az áramköri tervet<sup>30</sup>, majd a nyomtatott áramkör huzalozási nyáktervét<sup>31</sup> (**34. ábra**) készítettem el.

A robotkar teljes sematikus ábrája a függelék [5. mellékletében](#) megtekinthető, melyben a kapcsolások a kód szerinti `GPIO.setmode(GPIO.BCM)` megadásban értelemezendők. A GPIO-k számának alacsonyan tartásával nem válik szükségessé a multiplexerek<sup>32</sup> használata.

A nyomtatott áramkör kialakítása a különböző csuklók motorjaihoz különböző mikroléptetési módok rögzítését teszi lehetővé, aszerint, hogy közvetlen vagy közvetett a hajtás, illetve, hogy előreláthatólag mennyire súlyos szempont az adott csuklópont pontosságát garantálni. Az M<sub>1</sub> M<sub>2</sub> M<sub>3</sub> M<sub>5</sub> motorok esetén DIP típusú kapcsolókat tettem az áramkörbe. Az M<sub>4</sub> közvetlen (áttétel nélküli) kialakítása miatt, szoftveresen megadható a módozat, míg M<sub>6</sub> - a megfogó fejhez tartozó motor - állandó 1/16-os léptetést valósít meg.

A mikroprocesszorként használt RASPBERRY PI nem kerül rá közvetlenül a nyákra. Azt a képernyő vezérlőpanelének hátoldalán helyezem el, és egy 2x20-as szalagkábellel kötöm őket össze. A rövidtávú feszültségingadozások kiegyenlítésére minden vezérlő tápellátásával párhuzamosan egy 25V-os és 100 µF-os kapacitást telepítetek.

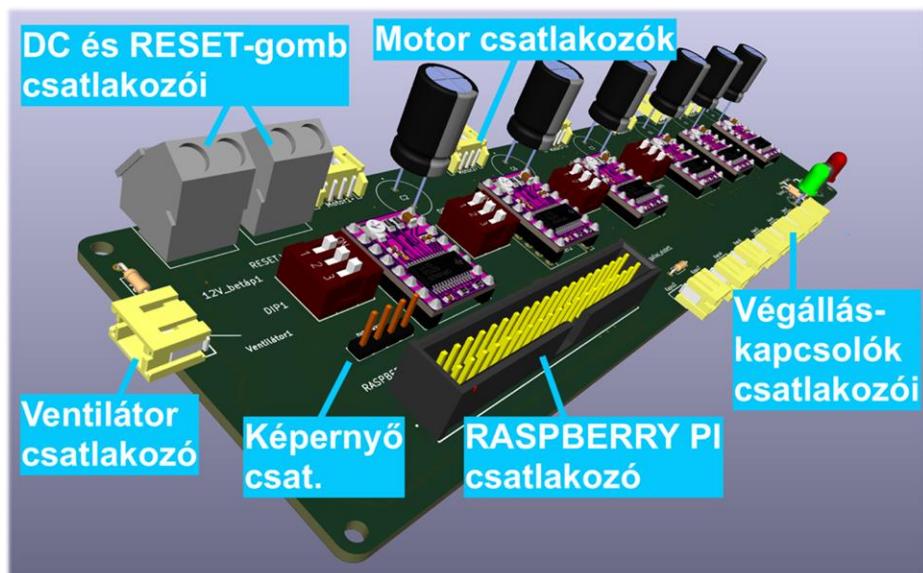


A kétrétegű furatszerelt alkatrészeket magába foglaló vezetékezés kialakítását először papíron kézzel, majd szoftveresen alakítottam ki. A szerelést elősegítendő a nyákon az alkatrészek referenciáit a "silkscreen" réteg segítségével feltüntettem. A legkisebb szigetelési távolságot a gyártástechnológiához igazítva 0.25 mm-re állítottam.

A komponensek pozicionálást megkönnyítette, hogy az áramkör számára a hely nem volt limitált. A végső kialakítás során általános irányelvként kerültem a 90°-os, preferáltam a 45°-os vezetékezést [38] [39].

A vezetősáv szélességének méretezéséhez a *KiCad - PCB Calculator*-át használtam. Túlbiztosítás gyanánt a tápcsatlakozók vezetőrétegeit felfelé kerekítettem, vastagítottam. A nyák két oldala közötti átjárásról furatfémmezett átmenet gondoskodik.

A tervező szoftver alkalmas a gépészeti tervezést megkönnyítő 3D modell (**35. ábra**) generálására, melybe - a nyomtatási nyomvonalakhoz hasonlóan - az elektronikai elemek lenyomatát a [www.snapeda.com](http://www.snapeda.com) weboldalról szereztem be vagy magam készítettem el.



**35. ábra - Az áramkör modellje és interfészei**

A standard 2.54 mm rászterezésű nyomtatott huzalozási lemezt a Műegyetem jóvoltából, az *UniPCB* gyártotta le. Ehhez a gerber fájlokat<sup>33</sup> ([2. csatolmány](#)) a *KiCad*

---

<sup>33</sup>A gerber fájlok (.gbr) a PCB tervezőprogramokból exportált RS274X formátumú fájlok, amelyek a nyomtatott áramkör rétegeinek geometriját, helyzetét és méretét tartalmazzák (, például szigetelő és vezetőszigetek, furatok pozíciói stb.) [39].

6.0-ás verziójával konstruáltam és generáltam. A panel FR4-es epoxigyanta-üvegszövet anyagú kompozit, amin litográfiai eljárással előállított réz vezetőszigetek alakítják ki az áramkör vezetékezését. Tekintve a kis sűrűségű lábkiosztást, a komponenseket hagyományos kézi lágyforrasztással rögzítettem (**36. ábra**).

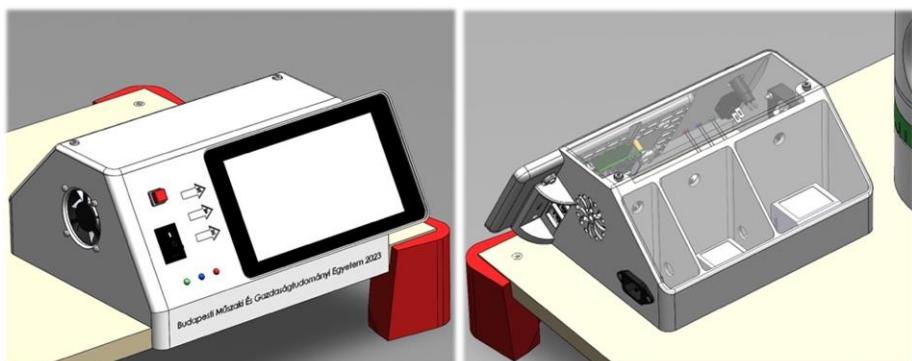


**36. ábra - Az összeállított központi vezérlőpanel**

Tapasztalatom szerint a tervezést érdemes a szükségesnek vélt alkatrészek beszerzésével kezdeni, és később ezen elemek tokozásához illeszkedő lábkiosztások alapján könyvtárat létrehozni a panel tervezéséhez. A beültetés során ezzel a módszerrel a lábak pozíciói és méretei nagy valószínűséggel megfelelőek lesznek. Az elkészült PCB kipróbálását az áramkör validációjának is tekintem egyben.

## 4.6 Tápkör és elektronikai doboz egység

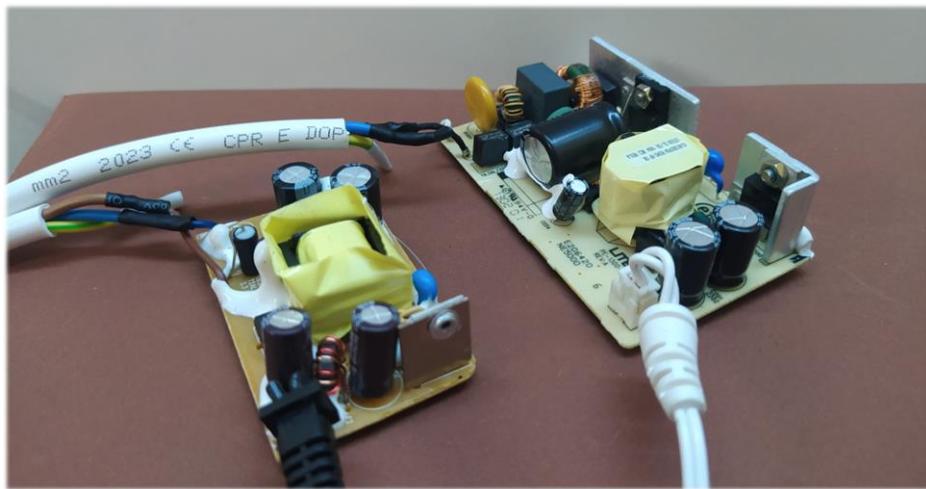
Az előzőekben meghatározott hardverkomponensek fizikailag is szükséges egyetlen szigetelt és védett helyre installálni, hogy egyetlen hálózati aljzatról is működtethető legyen. Mindezek a robotkartól elkülönített, leszedhető fedelű elektronikai dobozba (**37. ábra**) helyezett két tápegységgel (**38. ábra**) valósítottam meg, melyhez hasonlókat az az ipar is alkalmaz.



**37. ábra - Elektronikai doboz modellje**

Az egységen a következő komponensek kerültek összeköttetésre:

- tápegységek (2 db),
- RASPBERRY PI 3 B +,
- képernyő bővítő panel,
- érintőképernyő,
- hűtő ventilátor,
- gombok és visszajelző diódák,
- központi vezérlőpanel,
- kábelkorbács a robotkar felé.



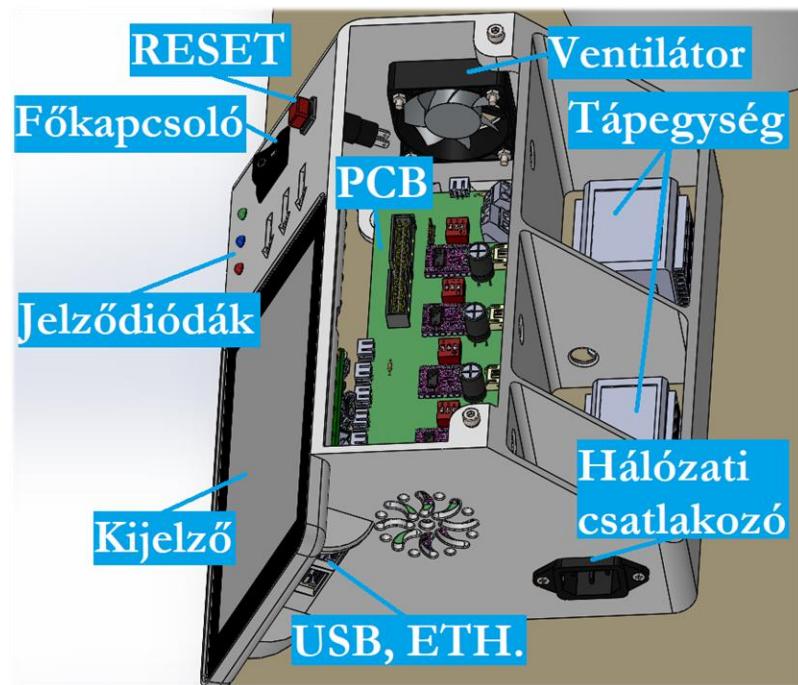
38. ábra - Az alkalmazott tápegységek

A két - kontrollert és vezérlőket ellátó - párhuzamosan kapcsolt, külső tápegység bontásból származik 5.1V/2.5A, valamint 12V/2.5A kimeneti értékekkel. Kimeneteik forrasztott, teknős tükesoros, illetve USB mikro-B típusúak.

A rendszer becsült áramigénye a fogyasztók egyidejű működtetését feltételezve:

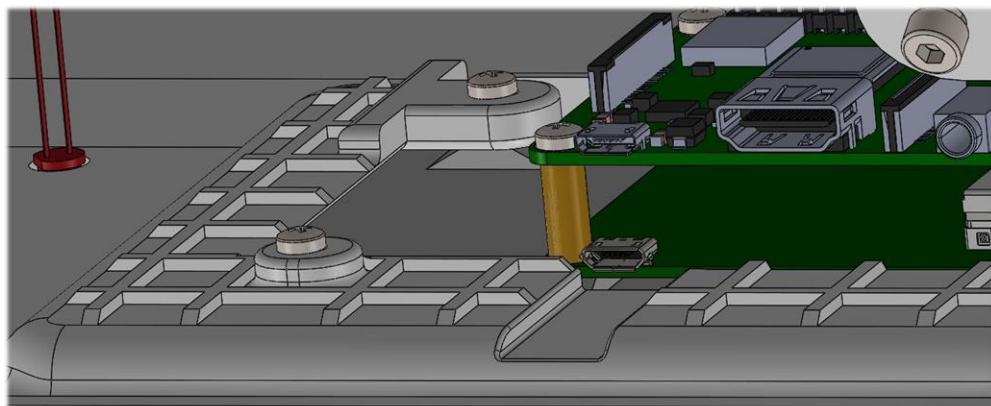
$$A_n = 3 \times 0.8A + 1 \times 1.1A + 1 \times 0.4A = 3.9A \quad (6)$$

A motorokat azonban a gyakorlatban nem fogom egyszerre működtetni. A vezérlők adatlapjuk alapján 8.2 - 45V közötti feszültségtartományt képesek kezelni. A dobozban (**39. ábra**) elkülönül egymástól a hálózati feszültségű kör és a vezérlőpanelhez tartozó törpefeszültségű kör.



39. ábra - Elektronikai dobozban található részegységek

A dobozt helyenként - például a képernyő felfogatásának mentén - nem lehetett a megfelelő falvastagságúra kialakítani. Ezekben a zónákban, melyeket a **40. ábra** mutat be, a merevségről plusz statikai elemek beépítésével kellett gondoskodnom.



40. ábra - A 4 mm-nél vékonyabb elemek merevítő bordázása

Annak érdekében, hogy a berendezéshez a doboz felnyitása nélkül is csatlakozni lehessen, fizikailag portokat tettem hozzáférhetővé rajta (**41. ábra**).



**41. ábra - Kivezetések további szoftveres fejlesztéshez és beavatkozáshoz**

Az egyedi geometriai alakzatot -, amit finoman igyekeztem a robotkar formavilágához illeszteni - szintén FDM 3D nyomtatásos technológiával állítok elő.

A főkapcsoló felnyomását követően a berendezés fényjelzést küld a felhasználónak, amiből a felhasználó feltételezi a megfelelő működést, valamint automatikusan elindul a légkeringtetéshez alkalmazott ventilátor.

A képernyő mikrovezérlő panelhez történő csatlakoztatásához szükséges információk a függelék [3. mellékletében](#) találhatók.

## 5 Demonstráció; tesztek és validált részfunkciók

### 5.1 Kinematikai modell és validációja

A 5 tengelyes *síkcsukló ízületekkel* tervezett robotkonstrukció mozgásának mindenkorai leírására számos modell rendelkezésre áll. Én ezek közül a *Denavit-Hartenberg konvenciót*<sup>34</sup> választom, melynek lényege, hogy a robotmechanizmus koordináta-rendszereinek (**42. ábra**) felvétele során bizonyos egyszerűsítő szabályokat tartok be, aminek következtében a tagokhoz tartozó transzformációs mátrixok 6 helyett, mindössze 4 változóval leírhatóvá válnak. A betartandó szabályok a következők [40]:

- Az (i+1). rendszer x tengelye metszi az i. rendszer z tengelyét.
- Az (i+1). rendszer x tengelye merőleges az i. rendszer z tengelyére.

A konvencióval kapcsolatban a következő észrevételek tehetők:

- Egy csukló z tengelye legyen az elfordulás vagy a transzláció tengelye.
- A koordináta-rendszerek x tengelyei lehetőség szerint legyenek párhuzamosak egymással.
- Az y tengelyek jobbkéz szabály szerint adódnak [2] [41] [40] [42].

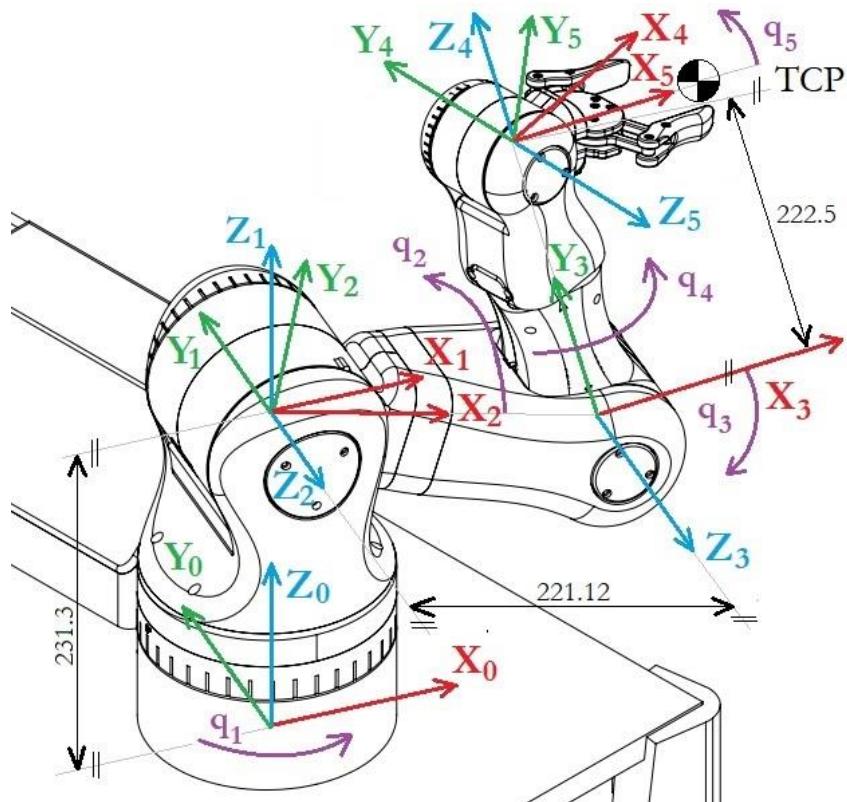
Céлом a végberendezés megfelelő koordinálása, ehhez pedig - az iparban is gyakran használatos - inverz kinematikai leírás<sup>35</sup> szükséges. Számításaim során csupán a végpozíció elérését tekintem célnak, például így a szögsebességek vagy a szögggyorsulások leírásától és meghatározásától jelenleg eltekintek.

A robotkart pont-pont irányítási rendszert hivatott megvalósítani, ami azt jelenti, hogy "a robotkar a csuklók eredő szabályozási tranzíenseitől függő trajektória mentén fog mozogni" [43].

---

<sup>34</sup> Denavit-Hartenberg konvenció: A képzés tudásanyagára alapozva a Khalil & Dombre félé konvencióval számoltam [2].

<sup>35</sup> Inverz kinematikai probléma: abban az esetben alkalmazzuk, amikor a TCP kívánt koordinátái ismertek, míg a csuklóparaméterek vektora - esetünkben  $Q$  - pedig az ismeretleneket tartalmazza.



42. ábra - A robotkar tagjaihoz rögzített ízületi koordináta-rendszerek

A koordináta-rendszereket a szegmentumokon belül helyeztem el. Az általános csuklókoordinátákat ( $q_i$ ) a vázolt referencia-pozícióban zérusnak tekintem. A csuklóparaméterek értékeit a **3. táblázat** mutatja *mm*-ben és *fokban*.

| i (tag) | $\alpha_i$ | $a_i$  | $\theta_i$        | $d_i$ | $q_i \text{ min}$ | $q_i \text{ max}$ |
|---------|------------|--------|-------------------|-------|-------------------|-------------------|
| 1.      | 0°         | 0      | $q_1$             | 231.3 | -105°             | 105°              |
| 2.      | 90°        | 0      | $-16^\circ + q_2$ | 0     | 0°                | 212°              |
| 3.      | 0°         | 221.12 | $32^\circ - q_3$  | 0     | 0°                | 244°              |
| 4.      | -90°       | 0      | $q_4$             | 223.5 | -90°              | 90°               |
| 5.      | 90°        | 0      | $-20^\circ + q_5$ | 0     | 0°                | 220°              |

3. táblázat - A robotkarra vonatkozó DH-paraméterek táblázata

Ahol,

- $\alpha_i$ : a  $z_i$  és  $z_{i-1}$  tengelyek közötti szög az  $x_{i-1}$  körül forgatva [fok],
- $a_i$ : a  $z_i$  és  $z_{i-1}$  tengelyek távolsága az  $x_{i-1}$  mentén mérve [mm],
- $\theta_i$ : a  $x_i$  és  $x_{i-1}$  tengelyek közötti szög az  $z_i$  körül forgatva [°],
- $d_i$ : a  $x_i$  és  $x_{i-1}$  tengelyek távolsága az  $z_i$  mentén mérve [mm],
- $q_i$ : n-edik taghoz tartozó csuklóváltozó [°], ahol  $[q_i] \in \mathbb{R}^3$ ,
- $q_i \text{ min}$ : csuklókoordináta értelmezési tartományának minimuma [°],
- $q_i \text{ max}$ : csuklókoordináta értelmezési tartományának maximuma [°].

A DH-konvencióból adódó **(6)** behelyettesítve megkapom a szomszédos koordináta-rendszerek közötti transzformációs mátrixokat. Azok szorzataként pedig a  ${}^0\underline{T}_5$  mátrixot kapom eredményül, ami a globális koordináta-rendszerből az  $n$ -edikbe képez [41] [26].

$${}^{i-1}\underline{T}_i = \begin{bmatrix} C\vartheta_i & -S\vartheta_i & 0 & a_i \\ Ca_i S\vartheta_i & Ca_i C\vartheta_i & -Sa_i & -d_i Sa_i \\ Sa_i S\vartheta_i & Sa_i C\vartheta_i & Ca_i & d_i Ca_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow {}^0\underline{T}_1, {}^1\underline{T}_2, {}^2\underline{T}_3, {}^3\underline{T}_4, {}^4\underline{T}_5 \quad (6)$$

Ahol,  $S\vartheta_i \cong \sin(q_i)$  és  $Ca_i \cong \cos(q_i)$  időfüggő változók szögfüggvényei.

Adottak a végberendezés lokális koordináta-rendszerben értelmezett helyvektorok, amik a két általam tervezett megfogó esetén a fent említett **(6)** és **(6) egyenletekből** következnek.

Ahhoz, hogy csupán ( $q$ ) paramétereket tartalmazó kifejezést kapjak,  ${}^0\underline{r}_{TCP}$  felírásához a helyvektorok valamelyikével és a fenti transzformációk meghatározott rendű mátrixszorzatával szükséges mátrixszorzást végeznem:

$${}^0\underline{r}_{TCP} = \left( \prod_{i=0}^n {}^i\underline{T}_{i+1}(q_i) \right) {}^5\underline{r}_{TCP} \quad (4)$$

A láncszabály szerinti deriválás szabályai szerint "a csuklósebességek és a sebességvektorok közötti kapcsolat a sebességekben lineáris" [22]. Felhasználva, hogy a helyvektor komponensek deriváltjai sebességkomponensek, az **5. egyenletet** kapom:

$$\frac{\partial {}^0\underline{r}_{TCP}(q)}{\partial q} \dot{q} = {}^0\dot{\underline{r}}_{TCP}^d(t), \text{ azaz } {}^0\underline{J}_{TCP} \dot{q} = {}^0\underline{v}_{TCP}^d, \quad (5)$$

Így ugyan elkerülöm a nemlineáris egyenletrendszer megoldását, azonban ez a formula az integrálásból következő numerikus hibák halmozódása ellen nem véd, valamint a tesztek során nehezen birkózott meg vele a robotkar processzora. A **6. egyenlet** szerint alakul a *Jacobi-mátrix*, azonban a **7. egyenlet** szerinti inverz számítását tekintve látható, hogy számomra egy ettől eltérő, kvadratikus Jacobi-mátrix lenne szükséges.

$${}^0\underline{J}_{TCP} = \begin{bmatrix} \frac{\partial {}^0\underline{r}_{TCP}_x}{\partial \dot{q}_1} & \dots & \frac{\partial {}^0\underline{r}_{TCP}_x}{\partial \dot{q}_l} \\ \vdots & \ddots & \vdots \\ \frac{\partial {}^0\underline{r}_{TCP}_m}{\partial \dot{q}_1} & \dots & \frac{\partial {}^0\underline{r}_{TCP}_m}{\partial \dot{q}_l} \end{bmatrix} \quad (6)$$

A **(5)** rendezésével és a **(6) egyenletek** felhasználásával végül adódik:

$$\dot{\underline{q}} = \underline{J}_{TCP}^{-1} \underline{v}_{TCP}^d \rightarrow \mathbf{Q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{bmatrix} \quad (7)$$

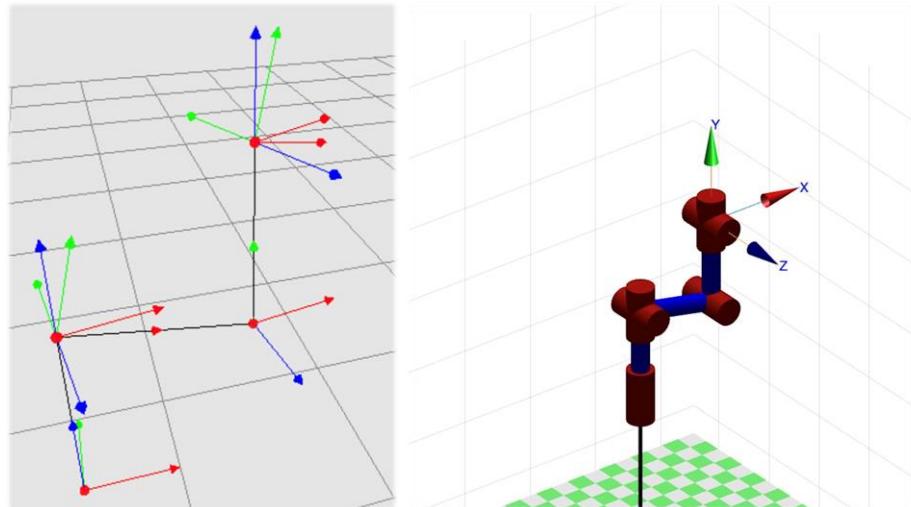
További egyszerűsítéseket eszközölve, amennyiben külön kezelem  $q_4$  csuklóváltozót, akkor  $q_4 = 0^\circ$  esetén a szerszám középpont az  $O_2, O_3, O_5$ <sup>36</sup> által meghatározott síkot nem hagyja el és így zárt alakos megoldás is lehetséges a **8. egyenlet** bevezetésével.

$$\tan(q_1) = \frac{Y_{TCP}^d(t)}{X_{TCP}^d(t)} \quad (8)$$

A világ koordináta-rendszerben értelmezett értékek számítása - egy koordinátageometriai megközelítést alapul véve - így már lineáris egyenletrendszerként 3 ismeretlenre egyszerűen megoldható a **9. egyenlet** felhasználásával [5] [41]:

$${}^0\underline{r}_{TCP}(q) = {}^0\underline{r}_{TCP}^d(t) \rightarrow q_2, q_3, q_5 \quad (9)$$

A validáció során -, melyet a **43. ábra** szemléltet, - a szerszám középpont **7. egyenlet** alapján felvett geometriai realizáció eredményét rajzolom ki és megvizsgálom a megkívánt  ${}^0\underline{r}_{TCP}^d$  vektor koordinátáival való egyezőséget. A csuklóváltozók és a *DH-paraméterek* ellenőrzéséhez, ábrázolásához a MATLAB "Robotics Toolbox for MATLAB" bővítményét használtam.



**43. ábra - Koordináta-rendszer viszonya és a validációhoz használt soros modell**

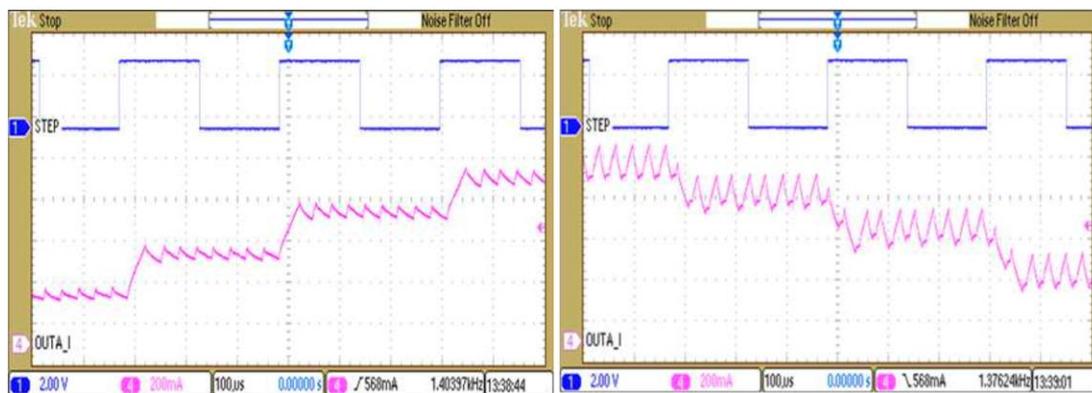
<sup>36</sup>A  $O_n$  a  $K_n$  koordináta-rendszer origóját jelöli.

Ezzel a módszerrel leellenőriztem, hogy a szegmentumok kívánt mozgása mely forgatások kompozíciójával állítható elő, és hogy a robotkar szerszám középpontja a megfelelő koordinátákat veszi-e fel.

## 5.2 Vezérlőjelek összehasonlítása

Az előállítható PWM jelek vizsgálata egyfajta validáció, mert egyrészt össze tudom hasonlítani a *DRV8825 adatlapján* feltüntetett kimeneti PWM jelekkel (**44. ábra**), másrészt ellenőrizhető a PWM jel motor forgatásra gyakorolt hatása.

A méréshez a Texas Instruments *Electronics Explorer DIGILENT<sup>®</sup>* integrált munkaállomását használom, mely tartalmaz oszcilloszkópot, multimétert, tápegységkört és forrasztásmentes tesztpáncelt ("breadboard-ot").

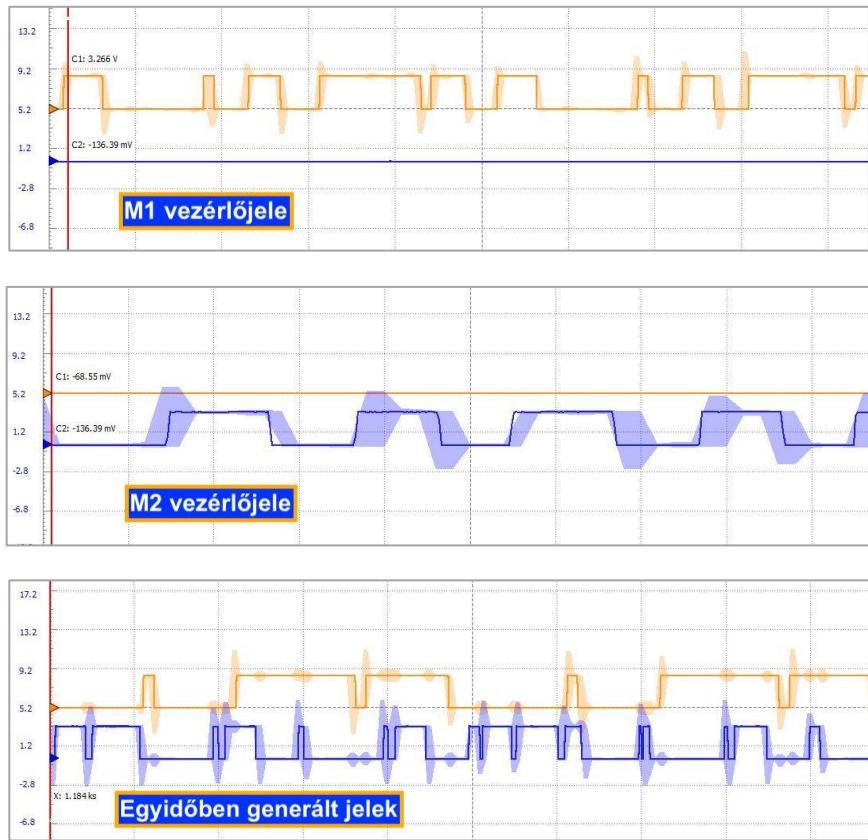


**44. ábra - A vezérlő adatlapján megadott PWM jelalakok [vezérlő adatlap]**

Először a motor forgatásához szükséges PWM jelet egy egyszerű *for ciklussal* igyekeztem előállítani, azonban erre elegánsabb szintaktika is rendelkezésre áll [36].

Az adatlapon megadott jelalakok a vezérlő be- és kimeneti oldalain mérhető rózsaszínű analóg és kék színű helyettesítő digitális impulzussorozat-jelalak (*PWM - STEP\_pin*) [44]. Az áramköri kiosztások során nem kellett számoljak az időzítők ("timer-ek") által egyidejűleg kezelhető vezérlőjelek számával, a vizsgálat során viszont igen.

Az általam generált jelalakokat a **45. ábra** szemlélteti, amiket az alaprogram szerint 0.008 másodpercig tartok magas állapotban 3.3V amplitúdóval. A kitöltést igyekszem 40-60% közötti értékre állítani úgy, hogy a periódus idő legfeljebb 0.01 s legyen.



45. ábra - Néhány egyenként futtatott és párhuzamosan generált PWM jelalak képe

A motorok tekintetében a mérések során általam megfigyeltek a következők:

- Kisebb impulzusszélességgel és kitöltéssel gyorsabb forgást tudok elérni.
- A forgatás sebessége kifejezhető az egy forgatáson belüli impulzusok számával → kevesebb impulzus gyorsabb de pontatlanabb forgatást eredményez.
- Az 50%-os kitöltés csendesebb, mint a 0 vagy 100% irányába eltoltak.
- Kettő vagy több jel egyidejű előállítása mindig kevésbé szabályos jeleket eredményez.
- Alacsonyabb frekvencia akusztikai értelemben véve csendesebb motorjárást eredményez.
- Túlzottan, (2-3 nagyságrenddel) alacsonyabb frekvencia esetén a kontroller alacsonyabb bitszámú pontosságú jelet képest csak generálni.

### 5.3 Mozgatás általános demonstrációja

Az eddig bemutatott részfunkciók validációján felül, szükség van egy az előző modellekkel és mérést egyesítő valós tesztre is, hogy a részegységek egyidejű összehangolt működését is kipróbáljam.

Úgy tekintem, hogy egy sakkfigura mozgatása megfelelő lenne erre a célra, valamint megalapozná egy későbbi, a már a végfelhasználó számára is hasznosítható sakk-szoftver és hardvercsomag megalkotását, melyhez a következők szükségesek:

- mágneses érzékelőkkel felszerelt sakktábla,
- egységes sakk-készlet, figurák megfelelő kialakítása,
- sakkfigura-megfogó egység,
- sakk szabályainak program általi ismerete,
- sakkfigurák alapállásának ismerete.

A figurák megfelelő kialakítása az alsó felületükbe készített furatokban elhelyezett mágnesekkel történik, mely ezután az érzékelőkkel ellátott táblára helyezve aktiválják az adott mezőt. A sakkfigura-megfogó egység pedig a figurák törzsének alakjához igazodó pofa szerkezettel emelhető meg hatékonyan. Ennek mozgatásánál figyelni kell a párhuzamos emelések és visszahelyezések műveletére a többi környezetben lévő figura érintése, feldöntése nélkül.

A pontosság, ismételhetőség felderítésekor a könnyebb megfogás végett a hagyományos megfogót szereltem fel a berendezésre, melyet a **46. ábra** szemléltet.



46. ábra - A robotkar működésének és a végberendezés pályavetületeinek vizsgálata

Tekintve, hogy a validációk külön-külön csupán szükséges, de nem elégséges feltételei a követelményjegyzékben rögzítetteknek, éles teszteket végeztem síkban és térben a mozgás ellenőrzéséhez. A síkbeli működés vizsgálatát a sakkjáték jellege indokolta kiváltképp, melyet milliméterpapíron igyekeztem ábrázolni.

Az ismételhetőség szempontjából átlagosan 8-14 mm különbség keletkezett a holtjátékok okozta bizonytalanság okán.

A vizsgálat eredményei alapján tett következtetésekre a [10. fejezetben](#) térek ki, míg a teljes demonstrációhoz használt tesztszoftver a [3. csatolmányában](#) van.

# 6 Működtető szoftver implementálása

A működtető szoftver a különböző szoftverkomponensek átfogó, szinergikus együttműködését igyekezik garantálni a robotfunkciók biztosítása érdekében. Más projektek áttekintésekor megfigyeltem, hogy a működtetéshez szükséges felprogramozás széles spektrumon; a manuális, akár bitszintű alkotástól a szimulációt követő ún. *embedeed coding*<sup>37</sup> procedúra során, számítógéppel generált kódokig is terjedhet. Én az előbbi módon, egy egyszerűbb implementáció mellett döntöttem.

A szoftver szerkezetileg két részre osztható: a felhasználói környezetre, ami az *GUI* mappában és fejezetekben található, és a motorok vezérléséért felelős *Script* mappa tartalmára. A *Script* mappán belül a *Modell* almappában található a kinematikai modell megvalósítása. A teljes moduláris architektúrájú mapparendszer és program pedig a [3. csatolmányában](#) olvasható. A dolgozat terjedelmi határai miatt csak a meghatározó, lényegi kódrészletek és az azokat övező gondolatiság kerülnek bemutatásra. [45] [46] [37].

## 6.1 Programnyelv és fejlesztői környezet

Ugyan a motorvezérlők közvetlen működtetése magas és alacsony jelekkel bitszinten történik, alkalmazok több leírónyelvet is, mint a CSS vagy HTML. A mozgatás megvalósításának többségét eddig ismereteimre és a képzés anyagára alapozva *Python programnyelv* segítségével készítem el. A *JavaScript nyelvet* pedig a felhasználói felület interaktívvá tételehez használom fel.

Ezen program elkészítéséhez a *Thonny* és a *Geany IDE*-ket használtam, valamint felhasználtam az előző fejezetekben készített modelleket.

## 6.2 Adatszerkezet és algoritmusok logikája

Általánosságban véve az adatszerkezet alatt a szoftver által kezelni kívánt adatok tárolásának és rendszerezésének módját értem. A RASPBERRY PI korlátolt számítási

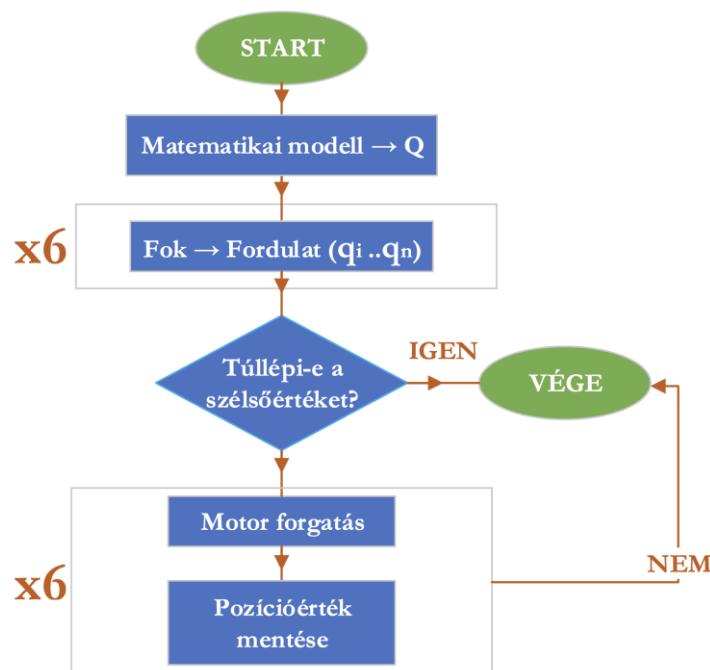
---

<sup>37</sup> Embedeed coding: magyarul beágyazott programozás, amit hatékonyan lehet vezérlők és szenzorok programozásához használni például MATLAB segítségével. A kódot a működési modell alapján számítógép generálja.

kapacitására való tekintettel adatbázis helyett JSON fájlformátumban<sup>38</sup> tárolom az adatokat. Ezen adatok a későbbiekben mind a felhasználói felület, mind pedig a vezérlésért felelős script-ek számára könnyen felhasználhatóvá válnak. A mapparendszer *kontroller* része az új Django projekt létrehozásával jön létre. A program működése során a GUI<sup>39</sup> gombnyomásra küldött adatát a Django FrameWork API<sup>40</sup>-ok segítségével továbbítja a motorvezérlő scriptek számára.

A számos alkalmazott könyvtár és modul (*admin, urls, models, test, apps*) között van a különálló Python Math matematikai könyvtárcsomag is, ami a matematikai modell algebrai műveleteit képes megfelelően kezelní.

A programozás során alkalmazott logikai és vezérlési szerkezetet a **47. ábra** mutatja be:



**47. ábra - Robotkar alapciklusának folyamatábrája**

A kezdéskor a matematikai modell visszaadja nekünk az egyes motoronként szükséges *elfordulást* (fokban vagy radiánban a modelltől függően). Ezt követően a fokot átváltjuk minden motor esetében a végezni kívánt *fordulatok számára*, ami az adott fokban mért általános csuklókoordinátáknak megfelelnek. Ezután a validáció következik.

<sup>38</sup> JSON: JavaScript Object Notation fájltípus.

<sup>39</sup> GUI: "Graphical User Interface", magyarul grafikus felhasználói felület.

<sup>40</sup> API: "Application Programming Interface", amik adott szoftverkomponensel történő kommunikációra használható.

Amennyiben a szélsőértékeken bármelyik motor túlfutna, akkor az adott tag forgatását azonnal letiltom. Amennyiben nem lépi túl, akkor a motort az adott fordulatok számával mozgatom. Ezen végpozíciókat elmentem. A *contorllerApp* az URL-eket adja meg, hogy az adott URL mit futtasson az *URLcalibration*-en belül. A motorok forgatása kezdetben egymás után történik a könnyebb tesztelés és programozás érdekében.

Az *algoritmusok logikai kapcsolatai* alapvetően soros folyamatokból és néhány bináris kimenetelű döntési és vizsgálati ciklusból épülnek fel.

## 6.3 Léptetőmotorok programvezérlése

E fejezet a "script" mappa tartalmát mutatja be. minden motor esetében végérényesen azt kell meghatároznom, hogy *melyik motor, melyik irányba és hány lépést* tegyen. Ennek a módját a következő alfejezetekben ismertetem [47].

### 6.3.1 Forgatás megvalósítása

A mozgatás megvalósításáról a *motor\_rotate.py* és a *calculate\_rotations.py* gondoskodik az alábbi változókat felhasználva:

| Programban használt megnevezés | Melyik kódrészben található? | Változó jelentés                                       |
|--------------------------------|------------------------------|--|
| <i>sleep_time</i>              | <i>rotate_motor.py</i>       | A PWM jel periódusideje                                |
| <i>limit_switch_signal</i>     | <i>rotate_motor.py</i>       | A végálláskapcsolók állapotát tarolom benne            |
| <i>degree_list</i>             | <i>move_robot.pyban</i>      | Az elfordulások szögértékeinek listája                 |
| <i>motor_properties</i>        | <i>move_robot.pyban</i>      | A motorok tulajdonságainak listájas                    |
| <i>rotations_list</i>          | <i>move_robot.pyban</i>      | Az elfordulásokhoz tartozó léptetések számának listája |
| <i>log</i>                     | <i>write_log.py</i>          | A naplóbejegyzések listája                             |

4. táblázat- A kódban használatos szignálok és változók táblázatának részlete

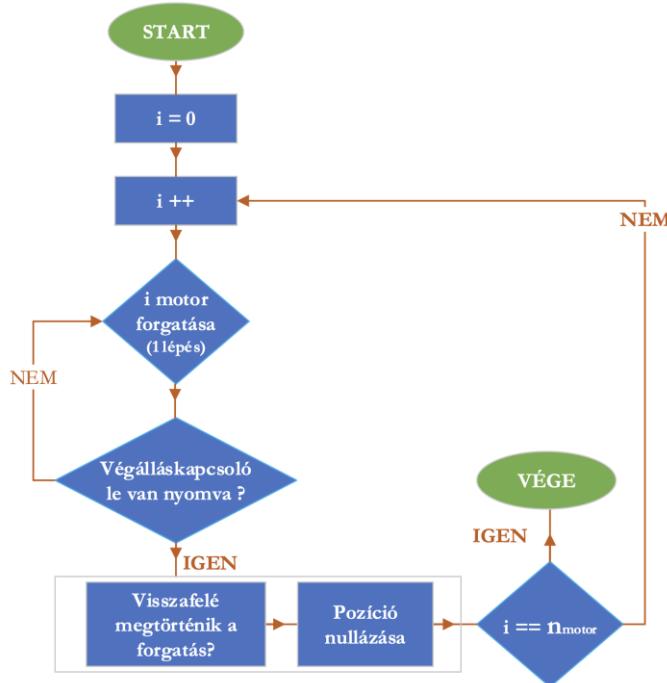
A *motor\_rotate.py* a motor forgatását végző esszenciális Python fájl, melynek egyik forgatási példája a következő [48]:

```
# Kimenetek szintjeleinek megadása
if n_of_rotations < 0:
    GPIO.output(direction, GPIO.LOW)
    n_of_rotations = n_of_rotations * (-1)
elif n_of_rotations > 0:
    GPIO.output(direction, GPIO.HIGH)
else:
    return
GPIO.output(step, GPIO.HIGH)
GPIO.output(enable, GPIO.LOW)
GPIO.setup(21, GPIO.IN)
GPIO.setup(20, GPIO.OUT)
GPIO.output(20, GPIO.HIGH)

# Innentől forgatunk

if transfer == 1:
    sleep_time = 0.008
elif transfer == 50:
    sleep_time = 0.0008
else:
    return
```

A *motor\_rotate.py* azonban csak a pozíciótűrésekkel, és nem pedig a biztonsági holtterekkel tud számolni. A folyamatábra (**48. ábra**) bemutatja a futó folyamat állapotváltozásra gyakorolt menetét és algoritmusát, mely mentén a *40-pines port* kiosztása is megtervezésre került.



**48. ábra - HOME-pozíció elérésének folyamatábrája**

A script indításakor motoronként lefuttatom az alábbi folyamatot. Egészen addig futtam a motort ameddig a végállást el nem éri. Amikor eléri a forgatást leállítom, illetve  $4^\circ$ -al visszaforgatom a csuklót, majd a motor pozíciót nullázom és elmentem.

Az első kettő motorhoz tartozó adatokat az **5. táblázat** ismerteti, míg az összes rendszerben résztvevő motor jellemzőjét tartalmazó teljes táblázat a [7. mellékletében](#) tekinthető meg.

|                                     | MOTOR 1     | "BCM"<br>megadása | "BOARD"<br>megadása | MOTOR 2     | "BCM"<br>megadása | "BOARD"<br>megadása |
|-------------------------------------|-------------|-------------------|---------------------|-------------|-------------------|---------------------|
| <b>Forgásirány<br/>változik-e?</b>  | 1           | -                 | -                   | -1          | -                 | -                   |
| <b>Áttétel:</b>                     | 24          | -                 | -                   | 20          | -                 | -                   |
| <b>ENABLE<br/>pin:</b>              | -           | 14                | 08                  | -           | 23                | 16                  |
| <b>STEP pin:</b>                    | -           | 15                | 10                  | -           | 24                | 18                  |
| <b>DIR pin:</b>                     | -           | 18                | 12                  | -           | 25                | 22                  |
| <b>M0:</b>                          | MANUÁLIS    | -                 | -                   | MANUÁLIS    | -                 | -                   |
| <b>M1:</b>                          | MANUÁLIS    | -                 | -                   | MANUÁLIS    | -                 | -                   |
| <b>M2:</b>                          | MANUÁLIS    | -                 | -                   | MANUÁLIS    | -                 | -                   |
| <b>Forgásirány<br/>nullázástól:</b> | -1          | -                 | -                   | 1           | -                 | -                   |
| <b>Szélsőállás<br/>nullázástól:</b> | $210^\circ$ | -                 | -                   | $212^\circ$ | -                 | -                   |

**5. táblázat - Példák az első két beavatkozóhoz tartozó specifikációkra**

A *while ciklus* kondíciója a kapcsoló magas, illetve alacsony jelszintjeit vizsgálja. A kapcsolókat fizikailag a (26)-os GPIO-ra csatlakoztatom. Amennyiben a kapcsoló végállásra jut, magas jelszintet kap, így a folyamat megszakításra kerül.

### 6.3.2 Kalibráció

A kalibrációt a *CalibrateMotor.py* és a *RunCalibration.py* fájlok végezzik.

Az első függvény egyetlen motor konkrét forgatását végzi, míg az utóbbi függvény pedig az ahhoz tartozó motor adatainak inicializálását hajtja végre, továbbá a *CalibrateMotor.py* is ez által kerül futtatásra.

### 6.3.3 Egyéb metódusok

A megfelelő működéshez szükséges további részszoftverek a következők:

- *CalibrateMotor.py*,
- *RunCalibration.py*,
- *Calculate\_rotations.py*,
- *Get\_date.py*,
- *Load\_position.py*,
- *Save\_position.py*,
- *Validate\_position.py*,
- *Write\_log.py*.

Ezen metódusok garantálják, hogy a motorok összhangban és a megfelelő paraméterekkel tudjam mozogni.

Például a *save\_position.py* *save\_position.json* függvénye elmenti a motor pozícióját, annak érdekében, hogy a pozíció validációjánál a tervezett pozíció felvételük előtt összehasonlíthatóak legyenek a szélső pozíciókkal. A *write\_log.py* pedig egy új sort hoz létre a *log.json* fájlon belül. A dátumokat a mikroprocesszor beállításai alapján automatikusan generálja.

A *calibrate\_motor.py*-on belül a *calibrate\_motor* függvény addig végez forgatást, amíg az adott motor végállását el nem éri. A kapcsoló aktiválásakor előre meghatározott mértékben visszafordul majd a pozíciót nullázza és a mozgatásból eredő változtatást a *motor\_positions.json* fájlba elmenti.

Minden motor specifikációjának megfelelően a hullámhajtómű áttetele, a pozíciója, az azonosításhoz szükséges motorszám is a *RunCalibration.py*-ban kerül elmentésre és ezek alapján kerül futtatásra a *calibrate\_motor* függvény.

A *calculate\_rotations.py* a radián és fok közötti átválthatóságot biztosítja és kiszámolja, hogy az adott motornak mennyit kell forognia, hogy a modellnek megfelelő elfordulást realizálhassam.

A *get\_date.py* a naplózáshoz (**52. ábra**) szükséges időpontot képes megadni év/hónap/nap/óra/perc/másodperc formátumban:

```
now.strftime('%Y.%m.%d. %H:%M:%S')
```

A *load.position.py* a pozíciók betöltéséért felelős, melyeket a fent kifejtett *save\_position* ment el.

A *validate\_position.py* - validálja az aktuális pozíciót a *position-limits.json* figyelembevételével annak érdekében, hogy elkerüljem a végálláson túlmutató navigációt.

### 6.3.4 Paraméterek tárolása

A paraméterek tárolásában az alábbi fájlok vesznek részt:

- a *position.json* az aktuális pozíciókat tárolja,
- a *position-limits.json* a  $q_n$  koordináták határainak értékét őrzi,
- a *properties.json* a motorok specifikációit, jellemzőit tartalmazza.

## 6.4 GUI Backend

Ebben a fejezetben mutatom be a felhasználó felület mögötti adatok kezelését és logikai folyamatait. Összefoglalva a GUI és a motorvezérlő szoftverrészleteket köti össze.

### 6.4.1 Végpontok leírása

Az általam alkalmazott struktúra esetén az URL linkek tartalmai a végpontok. Az alábbi végpontok esetén a felhasználói felület különböző funkcionálitásai kerülnek végrehajtásra. A robotkar esetében a végpontok az *urls.py* fájlban találhatók.

### 6.4.2 Metódusok definiálása

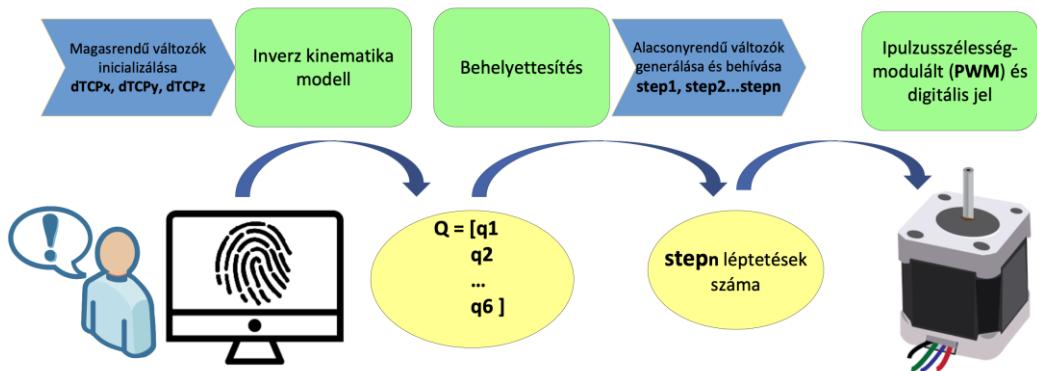
A *views.py* felelős a metódusok megvalósításáért.

HTML-ről lévén szó szabványos HTML válaszokat használok, mint például a 200-as ("ok" válasz) vagy az 500-as ("Server Error" válasz).

A kalibráció futtatásáról a "run\_calibration\_script" kódrész gondoskodik. Lekapcsolását - szoftveres LogOut - a "go to shutdown" oldal betöltése végzi. A berendezés tényleges kikapcsoláshoz pedig "Kikapcsolás" gomb aktiválása a *subprocess run* python környezetből futtatja a "sudo shutdown linx" nevezetű linux parancsot.

## 6.5 GUI Frontend

A kezelő felületen leadott magas absztrakciós szinten értelmezhető bemeneti értékek és változók átfogóan az ún. *magasszintű* változók. Az ezek által elindított reakciólánc blokkábráját a **49.ábra** mutatja:



49. ábra - A változók kezelésének lépései a felhasználótól a beavatkozóig

### 6.5.1 Futtatási képernyő

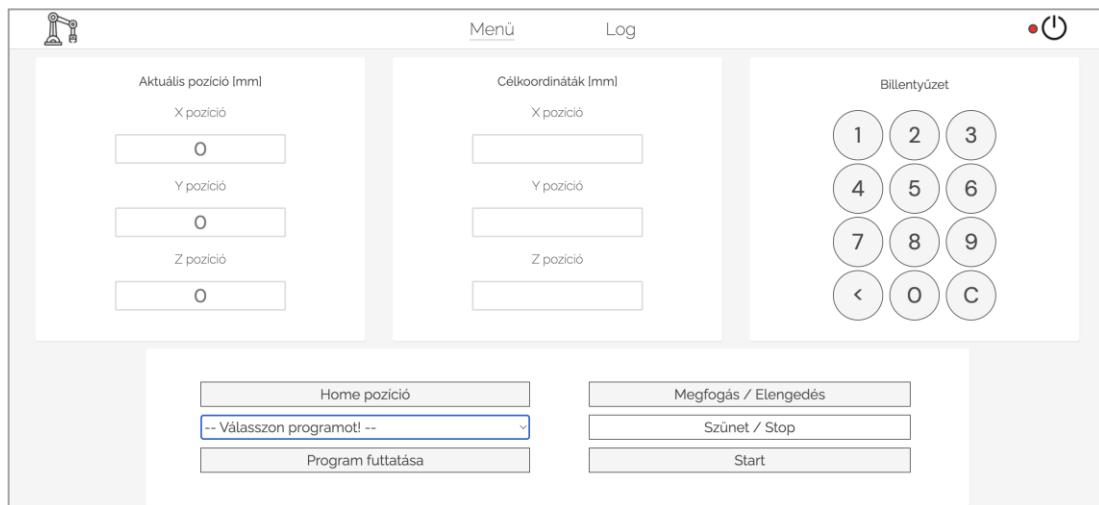
A felhasználói felület feladata, hogy a hardverrendszer aktuális állapotát kijelezzé és hogy a felhasználó azon a robot vezérlésére tett változtatásokat tudjon eszközölni.

Tekintve, hogy a berendezés a háztartásokba tervezem, az elsődleges felhasználói felület egy 800 x 480 felbontású 7 inch-es érintőképernyő, így a reszponzív megjelenítés nem került fókuszból. A fejlesztés szakaszában tetszőleges PC-ről is lehetséges teszteket végezni a teljes kód könnyebb átláthatósága mellett. Sőt a WiFi-n történő megosztást követően, minden azonos hálózatra csatlakozó eszközről lehetséges a robotkar vezérlése, mint például egy mobiltelefon.

A futtatott képernyő létrehozását az *ui.html* renderelésével végzem, kikapcsolását pedig a *shutdown.html* behívásával. A HTML-en belül a *head* jelölő alatt a *Google Fonts* betűtípus gyűjteményt, a *style* jelölő alatt pedig a szín és stílusjegyekre vonatkozó előírásokat adtam meg. A grafikus felületet *JavaScript* és *jQuery* segítségével működtetem. A futtatás során a digitális LED zöld színjelet ad, ellenkező esetben vöröset.

Az egységen a következő funkciók (**50. ábra**) érhetők el:

- aktuális TCPx, y, z robotkar végkoordináták,
- kívánt TCPx, y, z robotkar végkoordináták,
- HOME-pozíció,
- start, Szünet és Stop gombok,
- lenyíló ablak programválasztáshoz (**51. ábra**),
- kiválasztott program futtatása gomb.



**50. ábra - A felhasználói felület funkció-palettája**

A karakterbevitel eszközöként egy numerikus billentyűzet-modult is elhelyezek a felületen. *Első program*, *Második program* ... *n-edik program* egy előre kiválasztott, vagy betanított program végrehajtására lehetne a jövőben alkalmas, amik fixen telepített robotok és/vagy repetitív műveletek esetén lehetnek kifejezetten hasznosak.



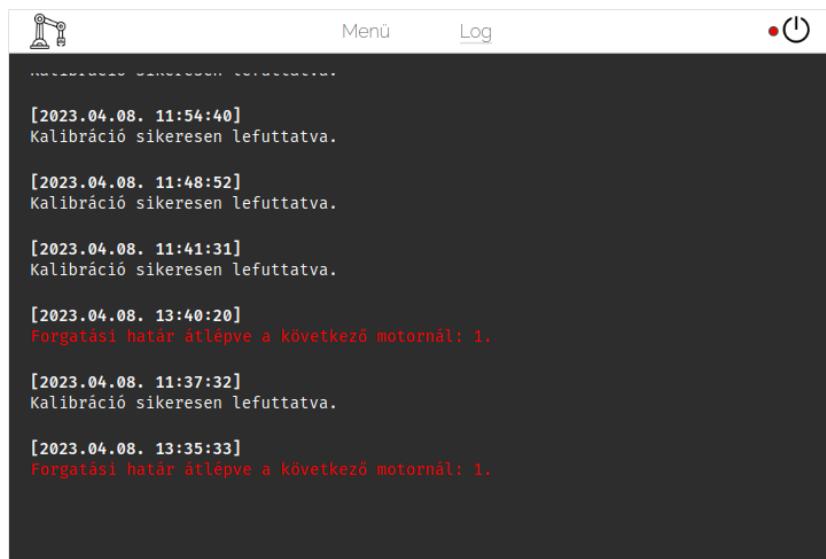
**51. ábra - A programválasztás lenyíló fülei**

A tervezés során kezdetben kész grafikus felületek és tesztkörnyezetekre terjedt ki mások munkáinak áttekintése, viszont végül nem azok mellett döntöttem. Ilyenek voltak például a *ROS*, *ESP3D-WEBUI*, *Seeduino®* gyártmányú *Wio terminal D51R* vagy éppen a *BCN3D MOVEO* saját felülete, a *RepRapFirmware*.

## 6.5.2 Naplózás megvalósítása

A naplózás célja, hogy a jövőben minden vezérlésben bekövetkezett lényeges tevékenységet regisztrálják és egy helyen vissza tudjak tekinteni (**52. ábra**), az utólagos javítások és módosítások érdekében.

Ezen adatokat a *write\_log.py* függvény regisztrálja, majd *log.html* felületen láthatjuk viszont. A napló legördülő menüt az érintőképernyőn belül a *LOG gomb* aktiválásával hívhatjuk elő.



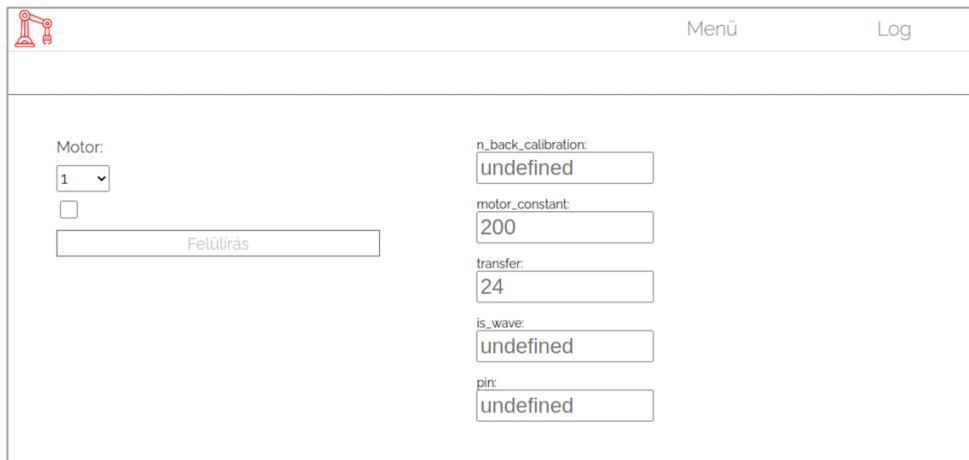
52. ábra - A generált napló kinézete

Technikailag a naplózás úgy történik, hogy az *open json* fájlt betöltöm egy listába, majd ezt a listát kiegészítem egy naplóbejegyzéssel a *log.append* függvény segítségével, ami még egy *lista elemet* hoz létre az eddigiek mögött. Végül ezt az elemet a *log.json* tárolja el. Előny, hogy így nem szükséges a felhasználónak manuálisan naplóznia, valamint, hogy így klaviatúra modul létrehozása továbbra sem szükséges a felhasználó számára különböző mozdulatsorok kialakítása során sem.

## 6.5.3 Inicializálás a berendezés kezelője által

Biztonsági okokból a *motor paramétereket* (**5. táblázat**) - ugyan könnyen elérhetővé szükséges tenni - nem lehet felülírni egykönnyen azokat, módosításuk pedig nagyfokú műszaki fejelmet igényel. Az itt félreütött értékek nem hibaként jelentkeznek, hanem a matematikai modell alapján mindenkorral alapjául szolgálnak a futtatásnak.

Ezen művelet elvégzéséhez egy *initializáló-menü funkciót* hozok létre, mely a *debug.html*<sup>41</sup>-ben (**53. ábra**) testesedik meg. A lehetséges módosításokhoz egy jelölőnégyzet bepipálása és 5 másodperc várakozás szükséges, így elkerülhetők az elírások és az illetéktelen módosítások.



**53. ábra - Az inicializáláshoz szükséges rejtett menü kinézete**

A "for motor in data" részletben látható, ahogyan a legördülő menük a data.json fájlból dinamikusan frissítem be, melyről az "updateFields()" függvény gondoskodik és tölti fel a mezőket.

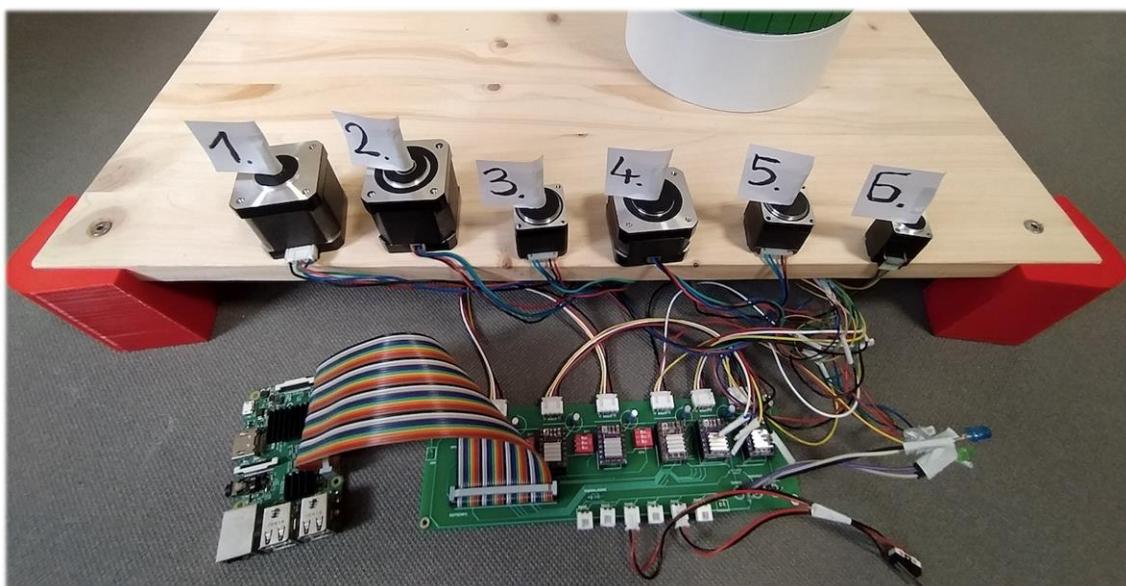
---

<sup>41</sup> "debug.html": Az inicializálás folyamatában a hibák javíthatóságát teszi lehetővé, innen a megnevezés.

## 6.6 Élesztési terv

Az *alfa1.0* (első működő szoftververzió) élesztését a robotkar élesítésével egyidejűleg hajtom végre. Először a forgatás funkciót alakítottam ki, majd teszteltem ezeknek a modell alapján történő egyidejű végrehajthatóságát.

Az élesztési folyamat első tesztfázisa az **54. ábrán** látható:



**54. ábra - A hardver robotkaron kívüli motorokkal történő futtatása**

Egyéb kikötéseket és teendőket nem vonatkoztatok az élesztés folyamatára.

## 7 Hibák és optimalizálási lehetőségek

Ezen fejezetben az elkészült robotkar működésének teszteredményei alapján történő újraértékeléssel próbálok reflektálni a saját munkámra.

Általánosságban a vezetékezésből és szerelési sorrend szükségszerű betartásából adódóan a szerelés tartott az elvárhatónál több ideig.

### 7.1 Áramellátás és feszültség-stabilizálás

Általánosságban elmondható, hogy a léptetőmotorok működtetése során nem állandó feszültségértékű kimenettel rendelkező tápegység hiányában a léptetések során hibák jelentkezhetnek. Ez akár lépések kihagyásán keresztül rossz pozíciók felvételében is alakot ölhet, aminek az oka gyakran a motor által kifejtett a szükségesnél kisebb nyomaték létrehozása [49]. Erre a mozgatás során történő hűtőventilátor belassuló hangja, vagy olyankor a *Standby-LED* bekapsolást követő villogása hívta fel a figyelmet.

Ahhoz, hogy a terhelések változásaitól függetlenül közel konstanst értéken tartsam a tápegységek kimeneteit, *kapcsolóüzemű DCDC* tápegység alkalmazása jött szóba.

Az energiaigény csökkentése érdekében aktiválhatnám a vezérlő kapcsolás szerint rövidre zárt alvó módot, azonban ez a közvetlen hajtású csuklókban a tartónyomaték megszűnését eredményezné, ami a mozgatott objektumok elejtéséhez is vezethetne.

Lehetőség van az egyszerre működtett léptetőmotorok számának korlátozására is, azonban ez éppen a robotkar mozgásának esszenciáját, sőt a függetlenre tervezett motorok létjogosultságát venné el. Köztes megoldásként jelenleg úgy lett implementálva, hogy egyszerre legfeljebb 4 léptetőmotor kerül működtetésre és egy időpillanatban csupán 2 indulhat egyszerre.

Végül fórumok tapasztalati szerint egyszerűen egy magasabb kimeneti feszültséggel rendelkező tápegység is gyakran megoldja a hasonló problémákat.

A mikrokontroller a fejlesztés elején a saját tápegységét illetően is hibát, alacsony feszültséget észlelt és jelzett, azonban a cseréje megoldotta ezt a problémát.

## 7.2 A megfogó szorítóerejének növelése

A [megfogó egységet](#) működtető *Nema8* szabványú, névleges 0,018 Nm nyomatékú motor nehezen produkálja a megfelelő szorítóerőt, valamint túlzottan finom illesztést igényel. Továbbá egy esetleges áramkimaradás vagy feszültségingadozás esetén elveszítheti tartónyomatékát oly mértékben, hogy a mozgatni kívánt objektumtól függően akár el is ejtheti azt.

Tekintve, hogy alapesetben a pofák és a motor között nincsen áttétel sem nyomatékhataroló elem, így a használat könnyen megviselheti a motorcsomkhoz való közvetlen rögzítést is. Esetleges kiváltására egy kis méretű nagy áttételű DC motor lenne a legmegfelelőbb, ami viszont jelentős átalakítást igényelne.

## 7.3 Karkinyúlás, szaggatott mozgás és a delaminációs jelenség

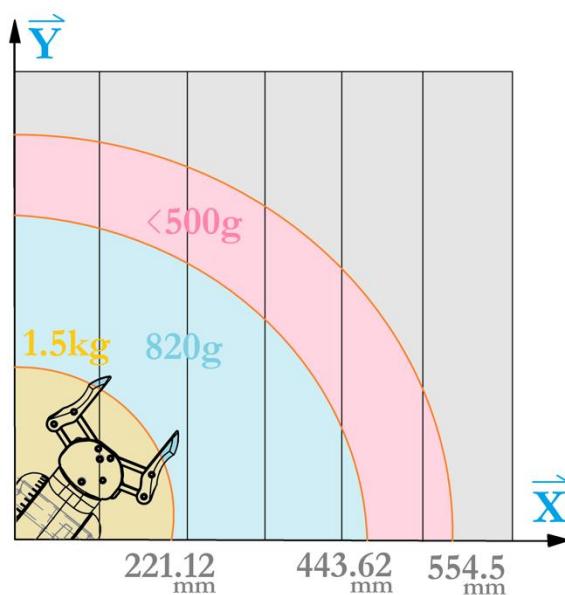
A robotkar mechanikai igénybevételnek kitett elemei közül a delaminációra hajlamos *Polymaker PolyFlex™TPU90* típusú *termoplastikus poliuretanból* nyomtatott flexibilis anyagú hajtómű hüvely bizonyult a legkevésbé terhelhetőnek. Ezen rétegleválasztódásra (**55. ábra**) való hajlamosság a nyomtatás technológiájának anizotróp tulajdonságából ered. Ezen folyamat mentén érdemes a *maximális karkinyúlást* is vizsgálni. Az általam alkalmazott nyomtató és szeletelő program esetén a különböző kitöltési struktúra és tényező függvényében végbemenő szilárdsági értékek áttekintéséhez a [50] irodalom anyagának feldolgozása nyújtott támponot.



**55. ábra - A megerősítés előtti delamináció "áldozatául" esett hüvely tesztelem**

A rugalmas elem szakadása általában a legvékonyabb keresztmetszet vagy a technológiai hibák mentén történik.

Tesztjeim alapján bebizonyosodott, hogy a csuklópontok nyomatékának növekedésével nő a leválásos károsodás esélye. Ezen hatások mérséklésére a javításhoz szükséges megfelelő cserealkatrész biztosítása szükséges. A kisebb nyomaték értékek érdekében a mozgatható objektum tömegét korlátozom a végpozíció függvényében. A tesztek során meghatároztam azokat a dedikált sávokat, melyek mellett a berendezés üzembiztos működését várhatóan megőrzi az adott terhelés mellett. Az így keletkező terhelhetőségi diagramot a **56. ábra** mutatja.



**56. ábra - Terhelhetőségi diagram**

A robotkar maximális terhelhetőségét maximális karkinyúlás mellett a tesztek alapján így ~450 grammra becsülöm.

Ezenfelül a forgáspontokban lévő hüvely rugalmas alakváltozásából adódóan a mozgás nagyobb terhelések és gyorsulások esetén szaggatottá válik. Ez egy akár koncepcionális hibaként is felfogható, ugyanis még a kezdeti, kézzel végzett tesztek ([3.4.1.-es fejezetben bemutatott kézi kerék](#)) során helyes értékelés mellett, helyén kellett volna kezelnem a rugalmas alakváltozás okozta jelenséget és vagy a flexibilis nyomtatvány lecserélése mellett dönten vagy nagyobb erőkarok mellett végzett fizikai terheléses teszteket is végezni. A csere esetén az említett cikloid vagy szíjhajtás is megfelelt volna.

A pontatlanság legfőbb oka az első ( $K_1$ ) vertikális tengelyű csapágy közel 0.5-1 mm-es ketyogása, ami révén a robotkar végberendezésében több tíz milliméteres holtjáték keletkezett. A csapágyazás előfeszítése oldaná meg a problémát, azonban ez az állványzati elem komplex átalakítását hozná magával.

## 7.4 Szoftveres finomhangolás

Az előző fejezetben bemutatott szoftver hibáit nem a nemmegfelelő implementáció, hanem inkább a megvalósított funkciók vagy azok hiányának tükrében vizsgálom.

Elméleti szinten ugyan figyelembe vettet a munkatér elhelyezkedését, azonban ezekkel a programvezérlés megalkotása során nem számoltam, csupán a felhasználói felületen megadható koordináták nagyságrendjét határoztam meg. A robotkar biztonsági zónáival és az érinteni nem kívánt csuklótér értékekkel, szintén nem számoltam.

Az vizsgált irodalom alapján ismert; "a szinguláris helyzetet az adja meg, amikor a Jakobi-mátrix determinánsa nulla" [41]. Ezen értékek létezését nem vizsgáltam és szintén nem vettet figyelembe.

A felhasználói felületen ugyan digitális LED-el jelzem, hogy a robot mozgási folyamatot végez vagy sem, azonban célszerű lenne megakadályozni új folyamat szoftveres indítását ameddig az egyik futtatása nem fejeződött be. Egy folyamatközi, ideje előtt történő, újabb adatbevitel során a program "összezávarható".

A világ koordináta-rendszerben vett egyes térfolycadokban a robot nem képes minden pozíciót megfelelően kiszámítani/felvenni a matematikai leírás vagy annak implementálásának eddig még azonosítatlan hibájából adódóan.

A RESET-gomb manuális aktiválása esetén a berendezés nem kap semmilyen információt arról, hogy a forgatás részben vagy teljesen megszakításra került. Ennek következtében egy esetleges ellentétes értelmű rotáció esetén az aktuális pozíció valós idejű, szenzoros követés hiányában, hibásan rossz aktuális pozícióval fog kalkulálni. A referencia-pozíció újból felvétele nélkül a limitált szögeken (*position-limits.json*) túlmutató mozgást is le fog futtatni, ami ütközéshez vezeti a berendezést.

A programozás során jelenős energiákat vont el tőlem az, hogy gyakran nem lehetett eldönteni, hogy egy aktuális kódrészlet esetében vajon a mikrokontroller számítási kapacitása bizonyul túl alacsonynak vagy valami egyéb hiba áll a háttérben.

## 8 Jövőbeli fejlesztési lehetőségek

Ebben a fejezetben a [feladatkitűzés pontosítása](#) alfejezet tartalmát újragondolva vizsgálom, mik azok, amiket még meg lehetett volna valósítani, viszont hiányukat semmiképpen nem minősítem [hibának](#), hiszen megalkotásuk eredetileg sem volt cél.

Hála a DH-paraméterek táblázatának és a parametrizált 3D modelleknek, a robotkar tengelytávolságainak mentén módosítható méretekkel rendelkezik egyéni igény vagy feladat esetén történő módosításhoz.

A működtetés tekintve egyszerűsítést jelentene egy saját készítésű joystick-modul megalkotása a TCP irányításához. Ehhez egy 3D-joystick lenne a legalkalmasabb, melynek billenőkarját fel-le - Z irányban - is lehetséges lenne mozgatni és vezérelni.

Mérnöki és vizuális értelemben egy szimulációs környezetbe történő integrálás nyújtana nagy előrelépést, például *MATLAB Simulink* vagy *SimScape* segítségével.

Nem kevés munkával akár gépi látással támogatott interakcióra is rá lehetne bírni a robotot. Ehhez a szoftveres munkán túl dedikált fizikai pontokban elhelyezett kamerarendszerre vagy valamilyen *MoCap-rendszerre*<sup>42</sup> lenne szükség.

A matematikai modell továbbfejlesztése szükséges a  $q_4$  működtetéséhez, vagy akár a pont-pont vezérlést kiváltó pályavezérlés megvalósításához.

Betanított mozgássorozatok által megvalósított funkció kidolgozása, mint például egy ruhadarabok összehajtогatására képes program megalkotása és kipróbálása.

A szenzorika fejlesztése mind a kar csuklópontjaiban, mind az elektronikai dobozban előnyös lenne. Az elsőre valamilyen enkóder- például a **31. ábrán** látható - az utóbbira pedig például a hűtőbordák közelébe elhelyezett hőmérsékletszenzor nyújthatna megoldást.

A tömeg csökkentése érdekében a motorok karon kívüli helyezése is jelentős munkával járó, de hasznos fejlesztés lenne. Az erőátvitelt például *bowden-rendszer* végezhetné.

---

<sup>42</sup> Mocap: *Motion Capture System*, mely segítségével valós személyek és tárgyak mozgását lehet digitalizálni.

## 9 Költségrealizáció

Tekintve, hogy akár az átlagos háztartások számára is elérhető árképzésű termék létrehozása volt a célom, most itt az ideje számot adni a közel 300 darab alkatrész becsült költségvonzatáról. A diplomaterv jellegéből adódóan munkadíjakat nem, amortizációt viszont tartalmaz a táblázat.

Az alábbi (**6. táblázat**) az ún. egyszerűsített költségtáblázat, ami a nagyobb részegységek összértékeit tartalmazza, hozzávetőlegesen a fejezeteknek megfelelő sorrendben.

| Sorszám   | Részegység megnevezése                                   | Összeg              |
|-----------|--|---------------------|
| 1.        | Fa alapzat   | 4.380 HUF           |
| 2.        | 3D nyomtatott szegmensek                                 | 58.768 HUF          |
| 3.        | Nagy méretű nyomtatványok (e-box) bérnyomtatása          | 46.000 HUF          |
| 4.        | 3D nyomtató karbantartása                                | 8.690.000 HUF       |
| 5.        | Külföldi termékek szállítatása és vámkezelése            | 3.400 HUF           |
| 6.        | Menetes betétek  | 5.710 HUF           |
| 7.        | Gépészeti kötőelemek (csavarok és csapágyak)             | 18.295 HUF          |
| 8.        | Elektronikai komponensek + léptetőmotorok                | 69.284 HUF          |
| 9.        | Tápegységek  | 6.820 HUF           |
| 10.       | RASPBERRY PI kontroller                                  | ~31.900 HUF         |
| 11.       | 1 db PCB gyártása  | 11.600 HUF          |
| 12.       | Érintőképernyő   | 39.980 HUF          |
| 13.       | Kötelező tesztelemek, fogyó eszközök, szerszámok + egyéb | ~18.000 HUF         |
| Összesen: |  | <b>281.427. HUF</b> |

**6. táblázat - Egyszerűsített költségtáblázat**

A tesztelések, kísérletezés során károsodott, "elfüstölt" elemek miatt a fenti végösszeghez viszonyítva további 10%-ot, redundáns tervezést alkalmazva pedig a feleslegesen berendelt elemek és filament alapanyagok további 35-40%-ot emésztettek fel.

Ez a kiindulási alapként szolgáló BCN3D MOVEO eredeti robotkar hozzávetőleges ~204.000 forintos árához viszonyítva megfizethető. Azonban látszik, hogy a RASPBERRY PI kontroller választása egy viszonylag költséges döntés volt.

## 10 Konklúzió; eredmények kritikája

Ebben a fejezet a dolgozat célja alapján ítélem meg az elvégzett munkát.

A talán kissé távoli célokat megfogalmazó [feladatkitűzés](#) fejezethez képest hiányosan, ám a diplomamunka elvárt terjedelméhez megfelelően kerültek megvalósításra az elvégzendő feladatok.

A robotkar végérőlénnyesen *sikeresen* elnyerte tervezett **fizikai alakját**, melyhez sikeresen elkészült a működtető **elektronikai** komponensek összessége. Ezenfelül a **szoftveres** alapfunkciók is implementálásra és kipróbálásra kerültek. A robotkart alkotó részegységeket viszonylag konzekvens módon igyekeztem **tesztek** és **validációs** technikáknak alávetni. Végül kimondható, hogy a világ koordináta-rendszerben vizsgálva a robotkar sikeresen eléri a tervezett és megadott pozíciót, tehát az **alapfunkció** is megvalósításra került.

A csuklópontok mozgatása *részben sikeresnek* mondható, mert végül csupán a q4-es csukló manuális megadásával/nullázásával lehetséges a modell megfelelő működtetése. A számítások elvégzésére a RASPBERRY PI jelenleg rendkívül lassan képes.

Amiket *nem valósítottam meg*, azok a **betaníthatóság**ot érintő funkciók, a **sakkozása** való kialakítás és képesség. Ennek okai a sakkozásra tervezett megfogó egység eltolt TCP adataival történő számítás nehézségei, az érzékelőkkel ellátott sakktábla elkészítésének hiánya, valamint az ezért felelős szoftveres megvalósítás elmaradása volt.

A sakkjáték szabályainak matematikai leírására sem került sor, de az alfa következő verzió is valószínűleg csak a lépések szabályrendszerével lett volna tisztában és véletlenszerű lépéseket tudott volna csak elvégezni. Az implementálás legnagyobb akadályát az ún. *állapotkiértékelés* és *optimális lépési döntés* meghozatala jelentette volna, mely messze túlmutat a diplomamunka keretein.

Semmiilyen előben hasznosítható vagy akár a kezelő felületen megjeleníthető **vizualizáció** nem került megvalósításra.

A [7.3-as fejezetben](#) kifejtett hibákra való tekintettel általánosságban levonható, hogy a tagok közötti flexibilis nyomtatványok **holtjátéka** minimalizálandó és hogy kellő

figyelmet szükséges fordítani az **ismételhetőség** követelményének betartására, ugyanis nehéz alkalmazni egy olyan robotkart, ami azonos koordináták esetén nem ugyanoda tér vissza. Amennyiben a polimer hullámhajtómű lecserélése mellett döntenék, az az **árak** emelkedését okozná, ugyanis egy a tervezett áttelekkel rendelkező fém hajómű szállítás nélkül ~ 19500 HUF (~56\$) vételártól kezdődően kapható. A költségkalkulációt áttekintve ezen robotkar több éves fejlesztés után - azt feltételezve, hogy a felhasználó rendelkezik 3D nyomtatával - **megérné**, azonban önmagában a prototipizálás összegét kifejezetten drágának találom.

Véleményem szerint a 3D nyomtatásnak - a mérnöki rajongás keltette hiedelmekkel ellentétben - megvannak a természetes gazdasági határai. Ennek többek között a gyakorlatban a **munkaerő** és annak költsége szab határt, ugyanis hiába állítható elő gyakorlatilag ma már szinte minden 3D nyomtatás segítségével, hogyha a prototipizálás költségei az időrőlfordítással és munkaerőigénnyel arányosan megnövekednek.

Végül egyes alkatrészek esetében okozott némi fennakadást a **chiphiány** és a **hosszú disztribúciós idők** okozta csúszás.

Az, hogy a háztartásokban mennyire használható vagy egyáltalán szükséges lehet-e egy ilyen eszköz, szubjektíven megítélhető, akár megkérdőjelezhető. Ezt azonban a piackutatási és marketing kérdések válaszolták volna meg, amiket nem feszegettem.

A terveket az abszolvált feladatok szerint készítettem el, melyeket ezen **dokumentumban** - talán a terjedelmi határokat is feszegetve - igyekeztem bemutatni. A diplomamunkába végül főleg a tervezési folyamatok és főbb műszaki megoldások gondolatisága fért, míg a függelékbe a marginálisabb, specifikus adatokat tartalmazó műszaki dokumentációra hajazó csatolmányokat helyeztem el.

Tanulság, hogy nagyfokú robottervezési rutin híján a robotkar tervezését - a dolgozat felépítésével ellentétben - érdemes lehet először az elektronikai és szoftveres funkciók kialakításával kezdeni, amit a hardveres és gépészeti oldalon célszerű folytatni.

A munka befejeztével elmondhatom, hogy rengeteget tanultam, főleg a *KiCad*-ban történő tervezés során a törpefeszültségű elektronikai elemekről és nyáktervezésről. Számos alkalmazott komponens és fogalom számomra új volt.

## **11 Köszönetnyilvánítás**

Elsősorban szeretném megköszönni a biztosított eszközöket és diplomamunkám koordinálását, felügyeletét a konzulensemnek, **dr. Stumpf Péternek** [51].

A minden nap érzelmi és szellemi támogatást pedig páromnak, **Kapin Lillának** köszönöm!

Továbbá hálás vagyok azoknak a tanároknak, akiknek hírével és munkásságával már jóval a Műegyetemi tanulmányaimat megelőzően találkoztam. Ilyenek voltak **dr. Czigány Tibor Pál**, a legendás **dr. Stépán Gábor**, valamint a roppant motiváló **dr. Kiss Rita Mária** Tanárő.

Számottevő jelenlétéükért pedig köszönöm azoknak, akikhez a teljes képzés alatt kivétel nélkül lehetett tanácsért fordulni; az önszerveződő baráti csapatunknak, a "Gérvágók" egyesületének. Ők név szerint: **Dénes Péter**, **Pethő Attila Gábor**, **Szima Zsolt**, **Földes Levente Bence**, **Geréb Ábris**, **Reuter Richárd**, **Bolemányi Balázs**, valamint **Somodi Mihály András**.

Ezen felül **Torma Máté** és **Serfőző Máté** fiatal szakembereknek, akik őszinte barátságukkal egyengették tanulmányaimat a munkával övezett úton.

Továbbá köszönet jár **Gosztonyi Csenge Lucának**, aki "golyaként" sokunknak önszántából és önerejéből segített beilleszkedni a Műegyetem közösségebe.

Végül pedig szerető támogatásuk nyomán vitathatatlan érdeme van édesapámnak és édesanyámnak is.

## 12 Irodalomjegyzék

- [1] M. A. Maczik: „**Ipari Robotok** - Békéscsaba,” [Online]. Elérhetőség: [https://www.nemesgszi.hu/components/docs/MM\\_IpariRobotok.pdf](https://www.nemesgszi.hu/components/docs/MM_IpariRobotok.pdf). [Hozzáférés dátuma: 2022. május 29.]
- [2] Szabó Zsolt: *Mechanikai rendszerek és robotok dinamikája - saját jegyzet*. [előadás]. BME - Műszaki Mechanikai Tanszék, 2021.
- [3] „**ROBOTTECHNIKA I.**,” [Online]. Elérhetőség: [https://oszkdk.oszk.hu/storage/00/00/60/46/dd/1/Kulcsar\\_Robottechnika\\_1.pdf](https://oszkdk.oszk.hu/storage/00/00/60/46/dd/1/Kulcsar_Robottechnika_1.pdf). [Hozzáférés dátuma: 2022. május 29.]
- [4] D. Ivor: **Beágyazott rendszerek\_2016 - elektronikus jegyzet**, Budapest: BME - Méréstechnika és Információs Rendszerek Tanszék, 2021.
- [5] B. C. D. K. L. é. D. L. G. Dr. S. Zsolt.: "ROBOTMECHANIZMUSOK", TÁMOP. 2014. ISBN 978-963-313-170-1, Budapest: BME MOGI, 2014 február.
- [6] H. H. L. Luigi Pagliarini: „**The future of Robotics Technology**,” *Networking and Artificial Life*, 2017 március.
- [7] K. N. J. N. P. Martin Hägele: „**Industrial Robotics**,” *ResearchGate*, 2007.12.31.
- [8] S. Zoltán: **IPARI ROBOTIKA KÉZIKÖNYV**, ISBN 978-615-00-0747-2, Tatabánya, 2018.
- [9] W. D. K. G. W. H. H. I. G. K. M. S. Adolf Frischherz: **Fémtechnológiai táblázatok**, B+V Kiadó, 2013.
- [10] S. J. G. N. C. C. Kiss Ferenc: **AZ EMBER ANATÓMIÁJÁNAK ATLASZA I.**, ISBN: 9632426444, Budapest: MEDICINA Könyvkiadó, 1967.
- [11] „EnjoyRobotics,” [Online]. Elérhetőség: <https://www.enjoyrobotics.com/kommunikacios>. [Hozzáférés dátuma: 2022. augusztus 12.]
- [12] M. L. I. K. S. D. Mahdi Azizian: „**THE DA VINCI SURGICAL SYSTEM**”.
- [13] M. J. B. 1. G. C. 2. J. G. A. P. 1. a. W. G. B. Emily K. C. Bullock: „**Robotics in Total Hip Arthroplasty: Current Concepts**” *Journal of Clinical Medicine*, 2022. novemeber 10.
- [14] E. Robotics, „ElephantRobotics,” [Online]. Elérhetőség: <https://www.elephantrobotics.com/en/mycobot-280-pi-2023-en>.
- [15] F.-h. Hsu: „**IBM's Deep Blue Chess Grandmaster Chips**”, <https://web.archive.org/web/20041228122752/http://www.csis.pace.edu/~ctappe/rt/dps/pdf/ai-chess-deep.pdf>, 28 december 2004.

- [16] 7Bot: „YouTube,” 30. 08. 2015. [Online]. Elérhetőség: <https://www.youtube.com/watch?v=OHazT3y0WpI>. [Hozzáférés dátuma: 2023. május 13.].
- [17] K. Schumacher: „YouTube,” 06. 04. 2021. [Online]. Elérhetőség: <https://www.youtube.com/watch?v=2t4XxeOtrnI>. [Hozzáférés dátuma: 2023. május 13.].
- [18] A. H. M. Firas Abdullah Thweny Al-Saedi: „**Design and Implementation of Chess-Playing Robotic System**”, ISSN: 2231-0711, 2015 május.
- [19] B. M. R. A. M. P. D. L. B. C. M. K. L. L. J. R. S. D. F. Cynthia Matuszek: „**A Robust Chess-Playing Robotic System**” *International Conference on Robotics and Automation*, 2978-1-61284-385-8, 2011 augusztus 18.
- [20] BCN3D, „GitHub,” [Online] Elérhetőség: <https://github.com/BCN3D/BCN3D-Moveo>.
- [21] W. B. J. F. K. G. Pahl: **Engineering Design – A Systematic Approach**, Springer, 2007.
- [22] D. S. Zsolt: **HOLONOM RENDSZEREK**, Budapesti Műszaki És Gazdaságtudományi Egyetem Gépész-mérnöki Kar Műszaki Mechanikai Tanszék, 2016.
- [23] Dejan, „HowToMechatronics,” 18. 06. 2020. [Online]. Elérhetőség: <https://howtomechatronics.com/how-it-works/what-is-strain-wave-gear-harmonic-drive-a-perfect-gear-set-for-robotics-applications/>. [Hozzáférés dátuma: szeptember 2022].
- [24] „igus.com,” [Online]. Elérhetőség: <https://www.igus.hu/info/3d-printing-overview>. [Hozzáférés dátuma: december 2022.].
- [25] „medical.evonik.com,” VESTAKEEP, 1 március 2022. [Online]. Elérhetőség: <https://medical.evonik.com/en/new-osteocompatible-vestakeep-fusion-peek-filament-for-3d-printed-implants-170049.html>.
- [26] T. D. szakdolgozat, **Design and build of a multi-DOF robotic arm**.
- [27] D. S. István: **Gépelemek**, 9789632750767, Budapest: Tankönyvmester Kft., 2010.
- [28] „www.frankebearings.co.uk,” 06 április 2016. [Online]. Elérhetőség: <https://www.frankebearings.co.uk/3d-printed-bearings-interview/>. [Hozzáférés dátuma: november 2022].
- [29] J. G. Zhiyuan Yu: „**Introducing Kinematic Fundamentals of Strain Wave Gear for Robotic Arm Joint**” március 2022.
- [30] D. B. István: **Gépszerkezettan I.-II. - saját jegyzet**, Budapest: Óbudai Egyetem - Bánki Donát Gépész és Biztonságtechnikai Mérnöki Kar, 2019.

- [31] V. Z. Fodor Attila: **BEÁGYAZOTT RENDSZEREK ÉS PROGRAMOZHATÓ LOGIKAI ESZKÖZÖK** - Egyetemi tananyag, ISBN: 978-963-279-500-3, Szegedi Tudományegyetem, 2011.
- [32] „Arduino.cc,” [Online]. Elérhetőség: <https://store.arduino.cc/products/arduino-mega-2560-rev3>. [Hozzáférés dátuma: 2023. április 10.].
- [33] „malnapc.hu,” [Online]. Elérhetőség: <https://malnapc.hu/yis/raspberry-pi>. [Hozzáférés dátuma: 2022 október 28.].
- [34] „TME.eu/hu,” [Online]. Elérhetőség: <https://www.tme.eu/hu/news/library-articles/page/41861/Leptetmotorok-tipusok-es-alkalmazasok/>. [Hozzáférés dátuma: 2021. szeptember 08.].
- [35] TechFactory: „YouTube - oktató videó,” 4 április 2019. [Online]. Elérhetőség: <https://www.youtube.com/watch?v=QRpirhXcOwE>. [Hozzáférés dátuma: 2022 szeptember].
- [36] K. J. A. G. Bogdan Bednarski: „**Influence of Microstepping Signal Shape on Shaft MovementPrecision and Torque Variation of the Stepper Motor**” "Energies", 2021. szeptember 25.
- [37] J. Hrisko, „Makersportal,” 25 január 2021. [Online]. Elérhetőség: <https://makersportal.com/blog/raspberry-pi-stepper-motor-control-with-nema-17>. [Hozzáférés dátuma: 2023 január].
- [38] „Wikipédia,” [Online]. Elérhetőség: [https://hu.frfwiki.wiki/wiki/Circuit\\_imprimé](https://hu.frfwiki.wiki/wiki/Circuit_imprimé). [Hozzáférés dátuma: 2023. február 15.].
- [39] „UniPCB,” [Online]. Elérhetőség: <https://www.unipcb.hu/info.html>. [Hozzáférés dátuma: 2023. április 23.].
- [40] A.-T. Barbara: „**Kalibrációs módszer kidolgozása egy hat szabadságfokú UR3 ipari robot inverz kinematikai modelljének pontosítására külső mérőrendszer felhasználásával**” Budapest.
- [41] L. A. S. B. I. Z. A. Berényi Dániel: „**DENAVIT-HARTENBERG KONVENCIÓ ALKALMAZÁSA TÉRBELI 3DoF NYÍLT KINEMATIKAI LÁNCÚ HENGERKOORDINÁTÁS ÉS GÖMBI KOORDINÁTÁS ROBOTOKRA**” 2019 február 13.
- [42] M. Luqman, „Udemy,” [Online]. Elérhetőség: <https://www.udemy.com/course/robotics-with-ros-build-robotic-arm-in-gazebo-and-moveit/>. [Hozzáférés dátuma: 2022 december].
- [43] B. I. K. D. O. I. S. Z. Dr. Tevesz Gábor: „**Robotirányítás rendszerteknikája**” december 2022. [Online].

Elérhetőség: <https://www.aut.bme.hu/Course/VIAUMA10>.  
[Hozzáférés dátuma: 2023 május].

- [44] C. István, 2013. [Online]. Elérhetőség: [https://megtestesules.info/hobbielektronika/mbed/pwm.html#Az\\_ímpulzus-szélesség\\_moduláció\\_alapjai](https://megtestesules.info/hobbielektronika/mbed/pwm.html#Az_ímpulzus-szélesség_moduláció_alapjai).  
[Hozzáférés dátuma: 2022 október].
- [45] R. Blum: **PHP, MySQL, JavaScript & HTML**, ISBN: 9786155186721, T. Könyvek, Szerk., 2020.
- [46] A. G.-J. Ofir Mulla: „**Using the Intel RealSenseTM Depth cameras L515 in Multi-CameraConfigurations**” [Online].  
Elérhetőség: [https://www.intelrealsense.com/wp-content/uploads/2020/08/L515\\_Multiple\\_Camera\\_WhitePaper\\_rev1.0.pdf](https://www.intelrealsense.com/wp-content/uploads/2020/08/L515_Multiple_Camera_WhitePaper_rev1.0.pdf).  
[Hozzáférés dátuma: 2023. február 3.].
- [47] P. M: szakdolgozat, *Robotkar- modell beágyazott vezérlőegységének fejlesztése*.
- [48] „Programcreek.com,” [Online].  
Elérhetőség: <https://www.programcreek.com/python/index/9145/RPi.GPIO>.  
[Hozzáférés dátuma: 2023. január 18.].
- [49] e. f. A feszültségstabilizátorok, „Tudasbázis,” [Online].  
Elérhetőség: [https://tudasbazis.sulinet.hu/hu/0d0cc85d-f7b5-41fb-aec0-d1b8362c7ebf\\_e90c4562-46d5-4b3a-a5ed-d640f67b512a\\_102e5c57-b50f-4053-b4a3-c1a7c5aa564f\\_f940cea3-2aaf-4f85-9723-d0c1666de03a\\_95b02fda-724b-4a5f-933e-f121897cb1d2\\_6ee8402f-e935-40e1-ab46-465559d77f02](https://tudasbazis.sulinet.hu/hu/0d0cc85d-f7b5-41fb-aec0-d1b8362c7ebf_e90c4562-46d5-4b3a-a5ed-d640f67b512a_102e5c57-b50f-4053-b4a3-c1a7c5aa564f_f940cea3-2aaf-4f85-9723-d0c1666de03a_95b02fda-724b-4a5f-933e-f121897cb1d2_6ee8402f-e935-40e1-ab46-465559d77f02).  
[Hozzáférés dátuma: 2023. május 4.].
- [50] D. L. M. A. Márton Tamás Birosz: „**Effect of FDM infill patterns on mechanical properties**”, TKP2021-NVA-29, 1. szám107654, 2022.
- [51] „BME.hu,” [Online]. Elérhetőség: [https://www.bme.hu/hirek/20161227/A\\_BME\\_VIK\\_aranyfokozatu\\_Kivalo\\_Oktatoja\\_is\\_Bolyai-dijas\\_lelt\\_iden](https://www.bme.hu/hirek/20161227/A_BME_VIK_aranyfokozatu_Kivalo_Oktatoja_is_Bolyai-dijas_lelt_iden).  
[Hozzáférés dátuma: 2016. 12. 27.].

## 13 Függelék

Ezen utolsó fejezet rendre a következő **.pdf** mellékleteket tartalmazza:

1. követelményjegyzék (*kovetelmenyjegyzek.pdf*),
2. RASPBERRY PI 3 B interfész-adatlap (*rp\_pi\_interfaces.pdf*),
3. DSI érintőképernyő leírása (*display.pdf*),
4. kontroller GPIO portjának bemutatása (*gpio\_40pin.pdf*),
5. központi vezérlőpanel huzalozási terv (*kapcsolas.pdf*),
6. központi vezérlőpanel beültetési terv (*nyakterv.pdf*),
7. aktuális motor-specifikációk (*specifikaciok.pdf*).

A dolgozatba történő befűzésre nem alkalmas részletek pedig az alábbi **.zip csatolmányok** formájában állnak rendelkezésre.

1. 3D modell (*modell.zip* → *.sldprt*, *.sldasm*, *.step*),
2. nyomtatott áramkör gyártási fájlok (*aramkor.zip* → *.gbr*),
3. szoftver forráskód (*kod.zip* → *mapparendszer*),
4. a robotkarról készült digitális képgaléria (*galeria.zip* → *galeria.pdf*),
5. DRV8825-ös vezérlő adatlapja (*drv8825.zip* → *vezerlo\_adatlap.pdf*).

Google Drive szolgáltatás segítségével az alábbi [link](#) birtokában a felsoroltak szabadon hozzáférhetők:

[https://drive.google.com/drive/folders/1bacq2nzB0w21XxJmhpG7ZI5QuvpyqOGj?usp=drive\\_link](https://drive.google.com/drive/folders/1bacq2nzB0w21XxJmhpG7ZI5QuvpyqOGj?usp=drive_link)

| Követelményjegyzék                       |   |  |      |   |   |
|--|---|--|------|---|---|
| Többszabadságfokú csuklókaros berendezés |   |  |      |   | 2022.09.10.   |
| Nr.                                      | Követelmény   | Érték  | Súly | Külső forrás / hivatkozás                         | Megjegyzés  |
| 1.                                       | <b>Alapzat stabilitása</b>                              | Nem borulhat fel (2. és 7. értékek szorzata) | A    | BCN3D robotmodell                                 | Tengelytávolságok a forrásmodellből lettek importálva             |
| 2.                                       | <b>Hasznos terhelhetőség</b>                            | ~ 1000 g                                     | S    | [22], munkaközi tesztek                           | Maximális kinyúlásnál   |
| 3.                                       | <b>Sebesség</b>   | Legalább $5 \frac{mm}{s}$                    | O    | -   | Racionális becslés  |
| 4.                                       | <b>TCP megfelelő mértékű pontossága</b>                 | x, y, z $\pm 5$ mm                           | A    | 5.4-es alfejezet                                  | A globális-krdsz. -ben értelmezendő                               |
| 5.                                       | <b>Ismétlési pontosság</b>                              | tagonként $\pm 5^\circ$                      | S    | -   | Egy tag ismétlődő és ellentétes értelmű szögelfordulásainál       |
| 6.                                       | <b>Alkalmazkodó képesség</b>                            | Passzív kompenzálás                          | S    | [1], 3.4-es fejezet                               | Feszültségeket a csukló nyeli el                                  |
| 7.                                       | <b>Maximális karkinyúlás</b>                            | 598.62 mm                                    | A    | 5.1-es alfejezet                                  | Horizontálisan értelmezendő                                       |
| 8.                                       | <b>Egyidejűleg működtetett motorok száma</b>            | min.1 és max. 4 db                           | S    | Tapasztalatok: [25], és hasonló projektek alapján | <a href="https://stackoverflow.com">stackoverflow.com</a> - fórum |
| 9.                                       | <b><math>q_1</math> ízületi tartomány</b>               | 210°   | S    | 5.1-es alfejezet                                  | -   |
| 10.                                      | <b><math>q_2</math> ízületi tartomány</b>               | 212°   | S    | 5.1-es alfejezet                                  | -   |
| 11.                                      | <b><math>q_3</math> ízületi tartomány</b>               | 244°   | S    | 5.1-es alfejezet                                  | -   |
| 12.                                      | <b><math>q_4</math> ízületi tartomány</b>               | 180°   | S    | 5.1-es alfejezet                                  | -   |
| 13.                                      | <b><math>q_5</math> ízületi tartomány</b>               | 220°   | S    | 5.1-es alfejezet                                  | -   |
| 14.                                      | <b>Megfogó párhuzamossága</b>                           | ~ 4 mm                                       | A    | MSZ ISO 1101 [8]                                  | Párhuzamosság tűrés   |
| 15.                                      | <b>Megfogható objektum minimális mérete [vastagság]</b> | ~ 2.5 mm                                     | O    | 3.5-ös alfejezet                                  | Egyéb geometriájú objektumot nehezen kezel                        |
| 16.                                      | <b>Megfogható objektum maximális mérete [vastagság]</b> | 78 mm  | O    | 3.5-ös alfejezet                                  | Egyéb geometriájú objektumot nehezen kezel                        |

|     |   |   |   |                            |   |
|-----|---|---|---|----------------------------|---|
| 17. | Végállás detektálás                       | 100/99<br>[-]                                   | A | Empirikus elvárás          | [Sikerességi ráta]                            |
| 18. | Szomszédos tagok közötti legnagyobb játék | Maximum 2 mm                                    | A | 3D modell alapján          | $z_i$ forgatási tengelyek mentén mérendő      |
| 19. | Hullámhajtás hüvelyelemének ciklusszáma   | Minimum 800 [fordulat]                          | S | Empirikus elvárás          | Hullámgenerátor fordulataiból számolom        |
| 20. | Delamináció nélküli átlagos üzemiidő      | Legalább 1 óra szerelés nélküli üzem            | S | 3.4.2-es alfejezet         | Nyomtatási paraméterek függvénye              |
| 21. | Munkaterén kívüli pontok figyelembevétele | -   | O | 3.2-es alfejezet           | -   |
| 22. | Pályavezérlés megvalósítása               | -   | O | 5.1-es alfejezet           | Pont-pont vezérlés helyett                    |
| 23. | Önsúlya nem befolyásolhatja mozgását      | (Különböző terhelési sorok összevetése alapján) | O | -                          | Nehezen hitelesíthető                         |
| 24. | A léptetőmotorok pontossága               | Elegendő a $\pm 1,8^\circ$                      | A | Nema17-17HS40-1204 adatlap | A hajtómű a hibát az áttétel arányában osztja |
| 25. | Berendezés modellhű működtetése           | <u>helyes/helytelen</u> szoftveres leképzés     | A | 5. fejezet                 | Demonstráció kimenetele határozza meg         |

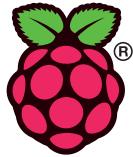
A dolgozat elkészítése során nem teljesített követelmények sorszámát vörös színű háttérrel jelöltem, melyek részletesebb átgondolása a 7. fejezetben található.

#### Jelöléstechnika:

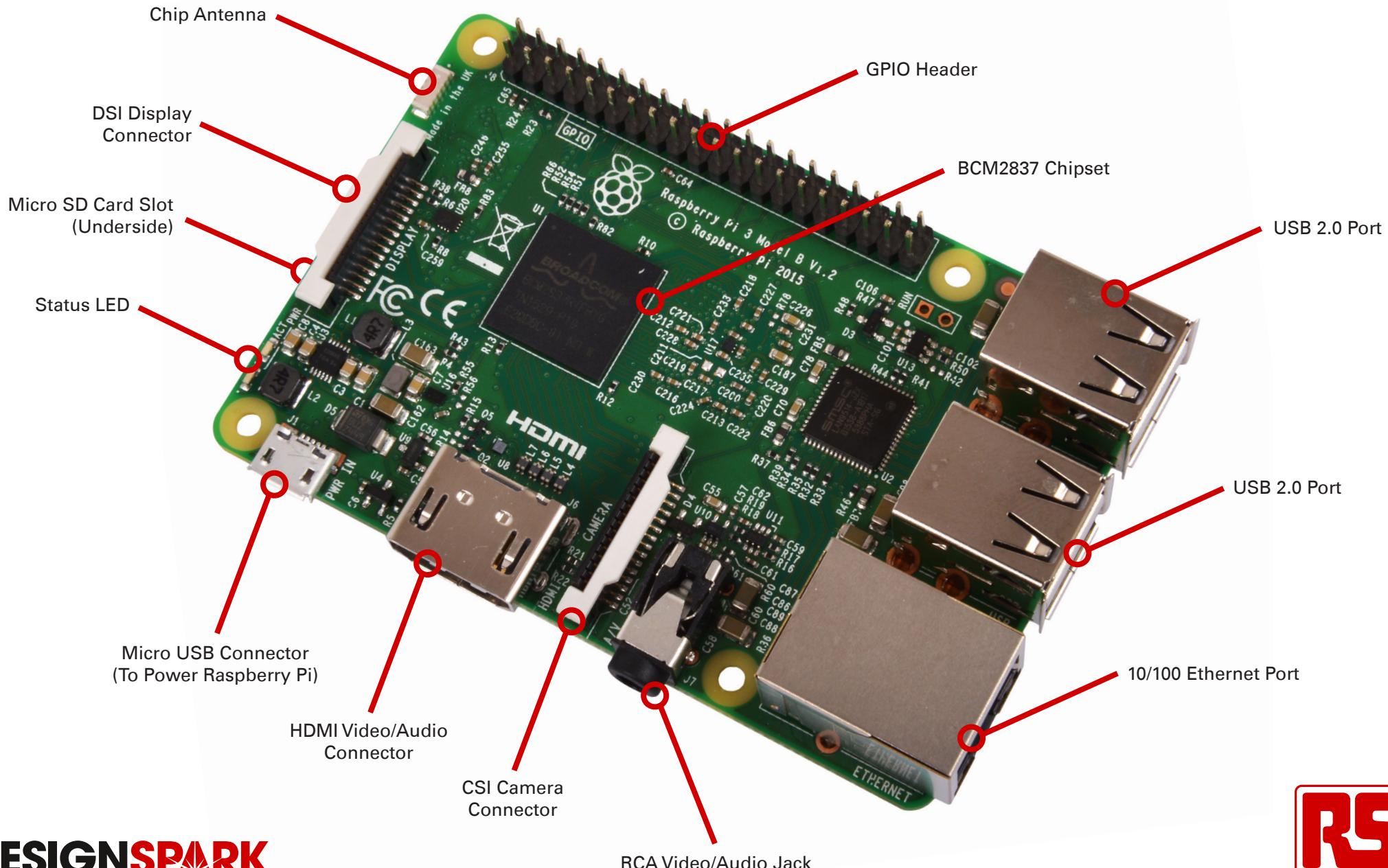
**Alapkötetelmény:** Az a követelmény, ami berendezés működőképességéhez nélkülözhetetlen. Jelölése: **A**

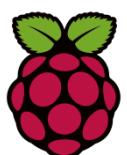
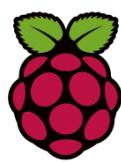
**Szintkötetelmény:** Az a követelmény, amit a megfelelő végcélhoz és funkcióhoz teljesíteni szükséges. Jelölése: **S**

**Óhaj:** Olyan kívánlalom, ami nem kritikus a terméket illetően, csupán a felhasználói élményt javítja. Jelölése: **O**



# Raspberry Pi 3 Model B





# Raspberry Pi 7" Touch Screen Display

## Fact Sheet

### Description

The 7" Touch screen Monitor for Raspberry Pi gives users the ability to create all-in-one, integrated projects such as tablets, infotainment systems and embedded projects. The 800 x 480 display connects via an adapter board which handles power and signal conversion. Only two connections to the Pi are required; power from the Pi's GPIO port and a ribbon cable that connects to the DSI port present on all Raspberry Pi's. Touch screen drivers with support for 10-finger touch and an on-screen keyboard will be integrated into the latest Raspbian OS for full functionality without a physical keyboard or mouse.



### Technical Specification

- 7" Touch screen Display.
- Screen Dimensions: 194mm x 110mm x 20mm (including standoffs)
- Viewable screen size: 155mm x 86mm
- Screen Resolution 800 x 480 pixels
- 10 finger capacitive touch.
- Connects to the Raspberry Pi board using a ribbon cable connected to the DSI port.
- Adapter board is used to power the display and convert the parallel signals from the display to the serial (DSI) port on the Raspberry Pi.
- Will require the latest version of Raspbian OS to operate correctly.

### Kit Contents

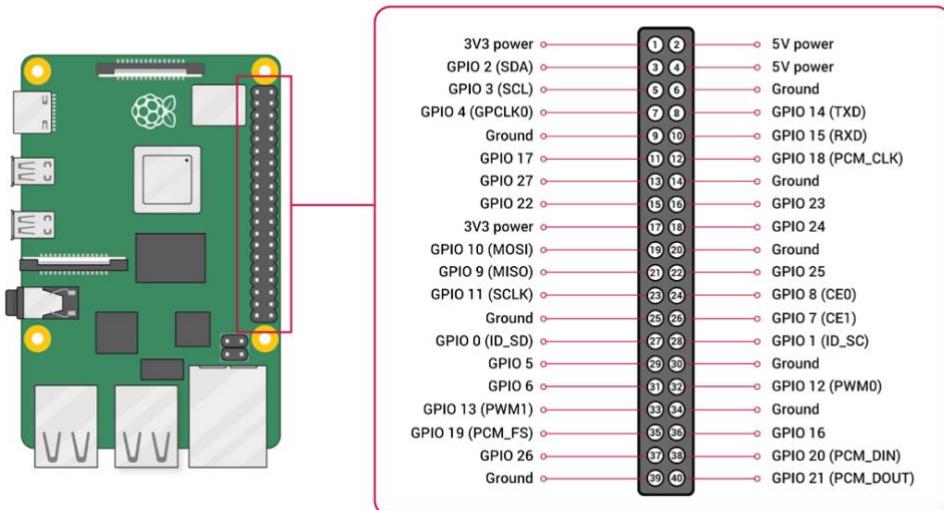
- 7" Touch screen Display
- Adapter Board
- DSI Ribbon cable
- 4 x stand-offs and screws (used to mount the adapter board and Raspberry Pi board to the back of the display)
- 4 x jumper wires (used to connect the power from the Adapter Board and the GPIO pins on the Pi so the 2Amp power is shared across both units)

JUHÁSZ ÁKOS ISTVÁN - YCLS37

- |   |   |
|---|---|
| 1. 3,3V - Az alaplapi feszültségszint                             | 21. GPIO9 (MISO) - A SPI Master bemeneti adatok érintkezője   |
| 2. 5V - A teljesítménybemenet külső áramforrás csatlakoztatásához | 22. GPIO25 - A GPIO25-ös érintkező digitális vagy PWM kimenet |
| 3. GPIO2 (SDA) - Az I2C adatjel                                   | 23. GPIO11 (SCLK) - Az SPI Master órajel érintkezője          |
| 4. 5V - A teljesítménybemenet                                     | 24. GPIO8 (CE0) - Az SPI Master Chip Select érintkezője       |
| 5. GPIO3 (SCL) - Az I2C órajel                                    | 25. Föld - Az alaplapi föld                                   |
| 6. Föld - Az alaplapi föld  | 26. GPIO7 (CE1) - Az SPI Master Chip Select érintkezője       |
| 7. GPIO4 - A GPIO4-es érintkező digitális vagy PWM kimenet        | 27. ID_SD - A HAT EEPROM I2C adatjel                          |
| 8. GPIO14 (TXD) - Az UART soros adatkimenet                       | 28. ID_SC - A HAT EEPROM I2C órajel                           |
| 9. Föld - Az alaplapi föld  | 29. GPIO5 - A GPIO5-ös érintkező digitális vagy PWM kimenet   |
| 10. GPIO15 (RXD) - Az UART soros adatbevitelhez                   | 30. Föld - Az alaplapi föld                                   |
| 11. GPIO17 - A GPIO17-es érintkező digitális vagy PWM kimenet     | 31. GPIO6 - A GPIO6-os érintkező digitális vagy PWM kimenet   |
| 12. GPIO18 - A GPIO18-as érintkező digitális vagy PWM kimenet     | 32. GPIO12 - A GPIO12-es érintkező digitális vagy PWM kimenet |
| 13. GPIO27 - A GPIO27-es érintkező digitális vagy PWM kimenet     | 33. Föld - Az alaplapi föld                                   |
| 14. Föld - Az alaplapi föld                                       | 34. GPIO13 - A GPIO13-as érintkező digitális vagy PWM kimenet |
| 15. GPIO22 - A GPIO22-es érintkező digitális vagy PWM kimenet     | 35. GPIO19 - A GPIO19-es érintkező digitális vagy PWM kimenet |
| 16. GPIO23 - A GPIO23-as érintkező digitális vagy PWM kimenet     | 36. GPIO16 - A GPIO16-os érintkező digitális vagy PWM kimenet |
| 17. 3,3V - Az alaplapi feszültségszint                            | 37. GPIO26 - A GPIO26-os érintkező digitális vagy PWM kimenet |
| 18. GPIO24 - A GPIO24-es érintkező digitális vagy PWM kimenet     | 38. GPIO20 - A GPIO20-as érintkező digitális vagy PWM kimenet |
| 19. GPIO10 (MOSI) - A SPI Master kimeneti adatok érintkezője      | 39. Föld - Az alaplapi föld                                   |
| 20. Föld - Az alaplapi föld                                       | 40. GPIO21 - A GPIO21-es érintkező digitális vagy PWM kimenet |

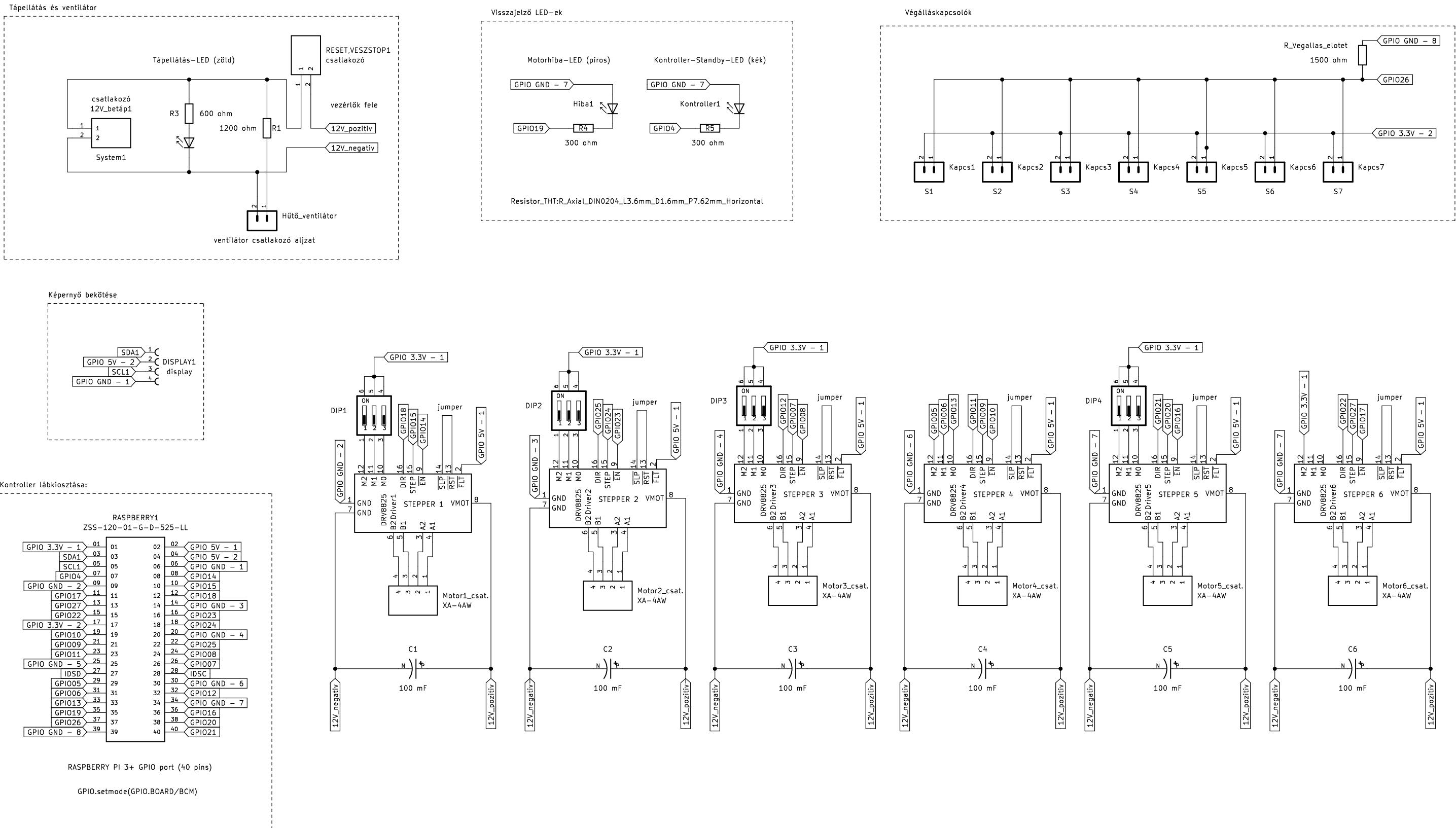
#### 4. számú melléklet - Kontroller GPIO portjának bemutatása - *gpio\_40pin.pdf*

JUHÁSZ ÁKOS ISTVÁN - YCLS37



#### RASPBERRY PI 3 Model B és a PIN-kiosztás térképe

[<https://community.microcenter.com> - fórum]



Megnevezés: Integráló áramkör huzalozási terv (diplomamunka) 2023  
Fájl: kapcsolas.pdf  
Konzulens: Dr. Stumpf Péter Pál  
NK: YCLS37

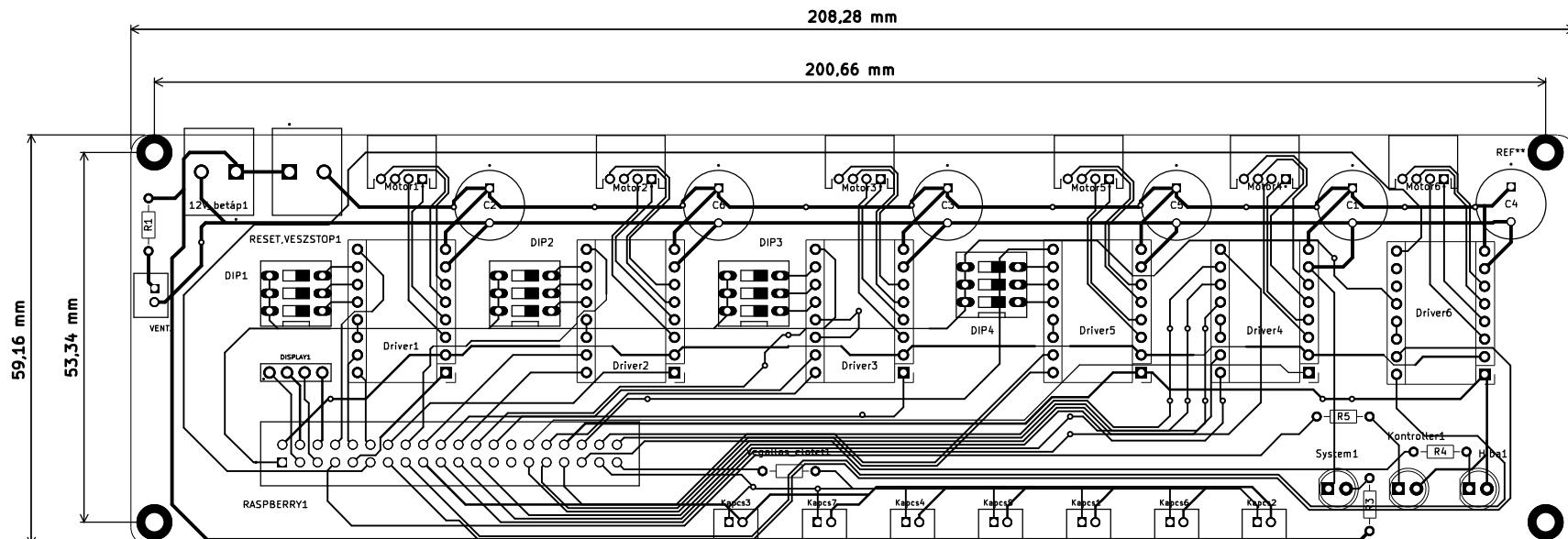
Készítette: Juhász Ákos István

Sheet: /  
File: aramkor.kicad\_sch

**Title: 5. számú melléklet**

Size: A3 Date: 2022.11.29.  
KiCad E.D.A. kicad (6.0.7-1)-1

Rev: EU  
Id: 1/1



Megnevezés: Integráló áramkör beültetési terv (diplomamunka 2023)

Fájl: nyakterv.pdf

Konzulens: Dr. Stumpf Péter Pál

NK: YCLS37

**Készítette: Juhász Ákos István**

Sheet:

File: aramkor.kicad\_pcb

**Title: 6. számú melléklet**

Size: A4 Date: 2022.11.28.

KiCad E.D.A. kicad (6.0.7-1)-1

**Rev: EU**

Id: 1/1

## Aktuális motor-specifikációk

Forgásirány:

- (1) - óramutató járásával megegyező,
- (-1) - óramutató járásával ellentétes.

A motor forgásiranya a hajtómű kimeneti oldalán a motor csonk oldala felől tekintve értelmezendő.

A "mindig alacsony", "mindig magas" jelszintek szoftveresen nem aktivizálandók. Ezeket eredendően az áramkör határozza meg.

- Továbbá:
- Kontroller-LED (kék) - **07(BOARD), 04(BCM)**
  - Hiba-LED (piros) - **35(BOARD), 19(BCM)**
  - Végálláskapcsolók (az összes) - **37(BOARD) 26(BCM)**



RASPBERRY PI 3 Modell B "BCM" és "BOARD" (pirossal keretezett) kiosztása

|                                      | Motor sorszáma |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |
|--------------------------------------|----------------|-----|-------|-------|-----|-------|-------|-----|-------|-------|-----|-------|-------|-----|-------|-------|-----|-------|
|                                      | 1              |     |       | 2     |     |       | 3     |     |       | 4     |     |       | 5     |     |       | 6     |     |       |
|                                      | PROP.          | BCM | BOARD | PROP. | BCM | BOARD | PROP. | BCM | BOARD | PROP. | BCM | BOARD | PROP. | BCM | BOARD | PROP. | BCM | BOARD |
| <b>Irányváltás van-e:</b>            | 1              | --- | ---   | -1    | --- | ---   | -1    | --- | ---   | 1     | --- | ---   | 1     | --- | ---   | 1     | --- | ---   |
| <b>Áttétel:</b>                      | 24             | --- | ---   | 20    | --- | ---   | 20    | --- | ---   | 1     | --- | ---   | 1     | --- | ---   | 1     | --- | ---   |
| <b>ENABLE pin:</b>                   | ---            | 14  | 08    | ---   | 23  | 16    | ---   | 08  | 24    | ---   | 10  | 19    | ---   | 16  | 36    | ---   | 17  | 11    |
| <b>STEP pin:</b>                     | ---            | 15  | 10    | ---   | 24  | 18    | ---   | 07  | 26    | ---   | 09  | 21    | ---   | 20  | 38    | ---   | 27  | 13    |
| <b>DIR pin:</b>                      | ---            | 18  | 12    | ---   | 25  | 22    | ---   | 12  | 32    | ---   | 11  | 23    | ---   | 21  | 40    | ---   | 22  | 15    |
| <b>M0</b>                            | M              | --- | ---   | M     | --- | ---   | M     | --- | ---   | S     | 13  | 33    | M     | --- | ---   | L     | --- | ---   |
| <b>M1</b>                            | M              | --- | ---   | M     | --- | ---   | M     | --- | ---   | S     | 06  | 31    | M     | --- | ---   | L     | --- | ---   |
| <b>M2</b>                            | M              | --- | ---   | M     | --- | ---   | M     | --- | ---   | S     | 05  | 29    | M     | --- | ---   | H     | --- | ---   |
| <b>Nullázás forgási irány</b>        | -1             | --- | ---   | 1     | --- | ---   | 1     | --- | ---   | -1    | --- | ---   | 1     | --- | ---   | 1     | --- | ---   |
| <b>Nullázástól vett szélsőállás:</b> | 210°           | --- | ---   | 212°  | --- | ---   | 244°  | --- | ---   | 180°  | --- | ---   | 220°  | --- | ---   | 180°  | --- | ---   |

Jelölésteknika: "PROP": jellemzők, "M": manuálisan megadható,

"S": szoftveresen megadható, "L": minden alacsony, "H": minden magas.