



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Alba Regia Műszaki Kar

Mérnöki Intézet

SZAKDOLGOZAT

Mobil robot platform fejlesztése

OE-AMK

2023

Hallgató neve: Gáti Krisztián

Hallgató törzskönyvi száma: T000987/FI12904/A-M



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Óbudai Egyetem
Alba Regia Műszaki Kar
Mérnöki Intézet

SAKADOLGOZAT FELADATLAP

Hallgató neve: Gáti Krisztián

Szakdolgozat száma: SZD2307061212032203WKSGHF

Törzskönyvi száma: T000987/FI12904/A-M

Neptun kódja: WKSGHF

Szak: gépészmérnöki

Specializáció: Folyamattechnika

A dolgozat címe: Mobil robot platform fejlesztése

A dolgozat címe angolul: Development of a Mobile Robot Platform

A feladat részletezése:

A hallgató feladata, hogy készítsen el az open source TurtleBot4-hez hasonló mobil robot platformot, mely egyedi ROS vezérlésen keresztül kapott utasításokat legyen képes végrehajtani.

Intézményi konzulens neve: Dr. Széll Károly

A kiadott téma elévülési határideje: 2026. 02. 10.

Beadási határidő: 2023. 12. 15.

A szakdolgozat: Nem titkos.

Kiadva: Székesfehérvár, 2023. 10. 26. P.H

Intézetigazgató



A dolgozatot beadásra alkalmasnak találom: 2023. 12. 15.

belső konzulens

külső konzulens



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Alba Regia Műszaki Kar
Mérnöki Intézet

HALLGATÓI NYILATKOZAT

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

Kelt: Mór, 2023.11.28.

Gati Kristina
.....
hallgató aláírása

Tartalomjegyzék

1.	Bevezetés	6
2.	A Mobil robotika áttekintése	8
2.1	TurtleBot 4 ismertetése	10
3.	Projektben felhasznált eszközök	12
3.1.1	3D nyomtató	12
3.1.2	3D nyomtatásban használt besüthető anyák	12
3.1.3	Laboratóriumi tápegység	13
4.	Kiválasztott elektronikai komponensek bemutatása.....	14
4.1	Raspberry Pi 4	14
4.2	Arduino UNO és Arduino Zero.....	15
4.3	Léptetőmotor	16
4.4	Arduino CNC motor shield V3	17
4.5	DRV8825 léptetőmotor vezérlő	18
4.6	DFR0205 feszültségátalakító	21
4.7	Áramforrás	22
4.7.1	Lítium akkumulátor	22
4.7.2	Akkumulátor töltő.....	23
4.8	Lítium akkumulátor kapacitás visszajelző kijelző	24
4.9	RPLidar A1	24
5.	Mechanikai elemek megtervezése	26
5.1	Omni wheel	26
5.2	Léptetőmotor tengely	28
5.3	Többszintes váz.....	29

6.	Elektromos bekötés.....	32
7.	Programkörnyezet kialakítása Robot Operating System-hez	35
7.1	Ubuntu Mate telepítése a Raspberry Pi 4-re	35
7.2	Ubuntu 22.04 telepítése a fejlesztői számítógépre	36
7.3	Robot Operating System (ROS) telepítése és testre szabása	37
8.	A robotot irányító programkódok	41
8.1	ROS programkörnyezet rövid bemutatása	41
8.1.1	Package	41
8.1.2	Topic	42
8.1.3	Publisher és Subscriber	42
8.1.4	Service	42
8.1.5	Action.....	43
8.2	Mozgás szerver.....	43
9.	Projekt kiértékelése.....	45
9.1	Továbbfejlesztési lehetőségek.....	45
9.1.1	IPS kijelző.....	45
9.1.2	Felhasználói felület.....	45
9.1.3	Automatikus navigáció a valós térben	45
9.2	Összegzés	46
10.	Summary	47
11.	Köszönetnyilvánítás.....	48
12.	Irodalomjegyzék	49
13.	Ábrajegyzék	51
14.	Mellékletek	52

1. BEVEZETÉS

A mobil robotikának az ipari környezetben ma már elengedhetetlen helye van, ami régen közel sem volt így. Ha csak visszatekintünk az 1986-os évre a csernobili katasztrófát követően egy hatalmas áttörést jelentett, hogy az ember egészségére kártékony környezetben egy robotot tudtak távolról vezérelni, annak a célnak az elérése érdekében, hogy megtisztítsa a környezetet. Az azóta eltelt közel 40 év alatt az akkor csúskategóriájú robottechnológiát minden megközelítésből fejlesztettük, tökéletesítettük és egyszerűsítettük. A mai technológia és az információ hozzáférhetősége miatt ma már lelkes egyéni szakemberek is képesek lehetnek saját robotot építeni, ha rendelkeznek a robotika témaköréhez tartozó szakágak iránti lelkesedéssel és szenvedéllyel, ami szükséges a sikeres végeredményhez.



1-1. ábra A megvalósított mobil robot

Az elérendő célom a képen megvalósításra került olcsó, saját hobbi mobil robot létrehozása, melyhez a ROS (Robot Operating System) nyílt forráskódú szoftver környezetet alkalmazom informatikai háttérként. Az inspirációt a TurtleBot 4, az interneten elérhető nagy közönségnek örvendő robot specifikációi szolgáltatták, amelyből igyekeztem minél

többet megvalósítani anélkül, hogy használatkész eszközök beszerzésére kényszerültem volna. A TurtleBot 4 remek példa arra, hogyan lehet a modern robottechnológiát széles körben elérhetővé és megvalósíthatóvá tenni.

A projekt során célom egy funkcionális, költséghatékony robot létrehozása volt, amely képessé tehető alapvető navigációs és interaktív feladatok végrehajtására, és bemutatja a mobil robotika jelenlegi lehetőségeit és kihívásait. A projekt azt is célozza, hogy inspiráljon másokat a mobil robotika területén való kísérletezésre, és megmutassa, hogy ma már mennyire elérhető és megvalósítható ez a technológia a szélesebb közönség számára is.

A dolgozat részletesen ismerteti a tervezési és építési folyamatot, beleértve az alkatrészek kiválasztását, a hardver konfigurációját és a szoftverintegrációt. Kiemelt figyelmet szenteltem a ROS alkalmazásának és különböző vezérlőelemek integrálásának bemutatására a robotban.

2. A MOBIL ROBOTIKA ÁTTEKINTÉSE

A mobil robotika nevéből adódóan egy kulcs aspektusban tér el a hagyományos robotikától, ami pedig a képesség a helyváltoztatásra. Egy ipari robotkar fejlesztése során az alábbi szempontok merülnek fel:

- Önsúly
- Hasznos teher nagysága
- Elérési távolság
- Visszaállási pontosság
- Ismétlési pontosság
- Maximális sebesség
- A mozgató egységek hatásfoka

A fentebb felsorolt szempontok egy része a mobil robotokra is jellemző, sőt a legtöbb nélkülözhetetlen tulajdonság, mely nélkül használhatatlanná válna a robot, kivéve egy: az elérési távolság, mint adat nem létezik a mobil robotikában, mivel a konstrukció nincs a talapzathoz szegezve. Ebből következik, hogy az elérési távolság nagyságát a konstrukcióba beépülő áramforrás határozza meg. A mobilitásban nyert szabadság megannyi felhasználási lehetőséget teremtett, melyek elősegítették az ipari folyamatok, valamint az anyagmozgatás további automatizálását.

A mobil robotika a robotikának egy hatalmas csoportja, mely témakör a sajátos mozgási szabadság kihasználása érdekében eddig még ismeretlen problémaköröket eredményezett. A felhasználási területe a robotnak nagyban meghatározza a mozgásformát a végső konstrukcióban. Az alábbi felsorolásban megtalálhatóak a hobbi mobil robotikában leggyakrabban felhasznált mozgásformák:

- Lábbal lépő
- Kerekeken guruló
- Repülő

Az első robotkar kialakításhoz az inspirációt a mérnököknek az emberi kéz jelentette válltól a csuklóig. A mobil robotok esetén is a természetből próbáltak mozgásformákat átvenni a valós térben való közlekedéshez. Mivel az ember a saját lábával képes volt bejárni az egész világot, ezért a változó minőségű talajon történő mozgás biztosításához lábbal rendelkező mobil robotokkal kezdtek kísérletezni. Viszont nehézséget okoz a magas szabadságfok igénye és a bonyolult matematikai számítások szükséglete.

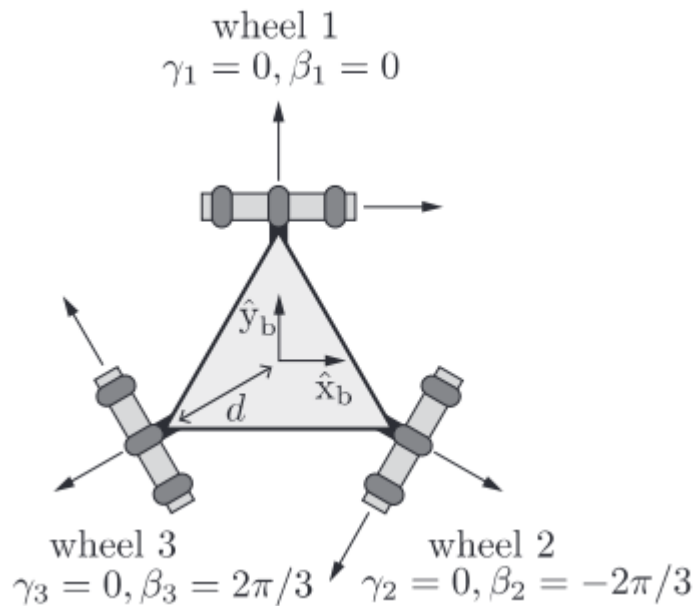
A repülésre alkalmas megoldások biztosítják a legmagasabb fokú szabadságot és tartogatják a legtöbb lehetőséget a jövőre nézve, de hobbi szinten a tervezés és a matematikai háttér ebben az esetben is a kereken guruló kialakításhoz képest sokkal összetettebb, így hatványozottan több időt emészt fel a megtervezése, ezért egy kereken guruló megoldás lett megvalósítva.

A hobbi robotikában ötfajta kerék terjedt el, melyek különböző szabadságfokkal rendelkeznek, ami így kihat az elérhető mozgáspályára. A következő táblázat bemutatja a szabadságfok számát:

2-1. táblázat Hobbi robotikában használt kerekek szabadságfoka

	Standard	Görgős	Svéd	Golyós vagy gömb	Omni
Szabadság- fok száma	2	2	3	3	3

A robot tervezése során a Svéd és az Omni jött számításba az egyszerű motorhoz való csatlakoztatásának köszönhetően. A 3D nyomtathatóság szempontjából a Svéd kerék a 45 fokban eldöntött görgői miatt nehézségeket okozott volna a nyomtatási minőség biztosításában, ezért a választás az Omni kerékre esett, melynél az előbbi probléma nem állt fenn, illetve egymástól jól elhatárolható részekre is könnyen szét lehetett bontani.



2-1. ábra Omni kerekes mobil robot [1]

A kerék fajta kiválasztása mellett még fontos a kerekek kiosztása a robot vázán, melyet a választott kerék nagyságban meghatároz a legmagasabb kihasználtság elérése érdekében. A maximális mozgásszabadság a 2.1-es ábrán látható kiosztás segítségével érhető el. [1] [2]

2.1 TurtleBot 4 ismertetése

A referenciának használt mobil robothoz kétféle modellben lehet hozzájutni, amely modellek közül a standardnak a funkciói jó kiindulási alapot biztosítottak az elérendő cél meghatározásában. Ezek közül a legfontosabb az átlagos terhelés mellett 2,5-4 órás üzemeltetési idő volt. Vezérlő egységnek egy Raspberry Pi 4-et használnak, ami alkalmas a ROS környezet futtatására, amely biztosítja a kommunikációs csatornát a további komponensek között. A robot fizikai környezetéről egy OAK-D-PRO sztereó kamera segítségével tudunk képet kapni, ami képes bonyolult képfelismerős algoritmusok futtatására. A szobában lévő távolságok valós idejű felméréséhez pedig egy RPLIDAR-A1 lett felszerelve, ami lehetővé teszi 360 fokban felmérni a robothoz képest a tárgyak helyzetét, ami elősegíti a nem kívánatos ütközések elkerülését. A beépített motorjainak köszönhetően képes a saját – közel 4 kilogrammos – súlyán kívül, akár még kilenc kilót mozgásra bírni, szükség esetén 0,3 méter per másodperc sebességgel. A felhasználói élmény javítása miatt felszerelték egy OLED kijelzővel, mely alkalmas diagnosztikai adatok vagy egyedi

program által saját adatok megjelenítésére. További tartozéka még egy töltőállomás, melyhez automatikusan visszatér, ha az akkumulátor töltöttsége alacsony szintre csökken.

A robot megvásárlása esetén használatra kész állapotban érkezik meg, számos a megértését elősegítő dokumentáció, szimulációs model és példakódok kíséretében. Összegezve: a TurtleBot 4 nem véletlenül lett a világ legnépszerűbb mobil robotika oktató csomagja, mely egy robosztus alapot biztosít, mégis teret hagyva az egyedi felhasználáshoz való kiépítéséhez mind mechanikailag, mind szoftveresen. [3]

3. PROJEKT BEN FELHASZNÁLT ESZKÖZÖK

3.1.1 3D nyomtató

A ház kialakításához, melynek a feladata, hogy összefogja a komponenseket egy Ender 3 S1 és egy Ender 3 V2 3D nyomtató lett használva. A választást az indokolta, hogy kedvező árúak miatt nagyon elterjedtek, illetve rendelkeznek a legtöbb hobbi nyomtató specifikációival, ami lehetővé teszi a kinyomtatott modellek reprodukálását a legtöbb forgalomban kapható 3D nyomtató segítségével. A legfontosabb specifikáció a két nyomtató esetén a rendelkezésre álló nyomtatási terület, mely a V2-nél 220x220x250 mm az S1-nél 220x220x270 mm.

3.1.2 3D nyomtatásban használt besüthető anyák

A komponensek összeillesztésénél a ruthex 3D nyomtatáshoz használt műanyagba besüthető anyák kerültek felhasználásra.



3-1. ábra Besüthető anyák

A használatuk mellett több érv is felmerült, melyek hosszú távon előnyt jelentettek a hagyományos anyákhoz képest. Az összeszerelés szempontjából a leginkább kézenfekvő fejlesztés, hogy a hagyományos anyához képest a besütést követően fix helyzetbe kerül,

ezért a csavar meghúzása és lazítása egyszerűbbé vált. A használatuk mellőzése esetén, a nyomtatásra kerülő modelleknél kitöréseket kellett volna alkalmazni, ami negatívan hatott volna a részegységek mechanikai szilárdságára. Több model esetén tapasztalható, hogy az illesztéseknél csökkenthető az anyag vastagsága, amivel spórolni lehet a nyomtatáshoz használt filament-tel. Ami azt eredményezte, hogy így a saját árát képes volt visszatermelni, valamint csökkent a nyomtatási idő.

3.1.3 Laboratóriumi tápegység

Az elektronikai bekötések teszteléséhez szükséges egy labortáp, melynek segítségével biztonságosan elvégezhető az elektromos bekötés. Az eszköz néhány esetben rámutatott a felmerülő hibákra, az esetleges károk kialakulása nélkül. A kalkulált áramfelvétel és a szükséges feszültségtartomány miatt egy NANKADF márkájú labortápra esett a választás, mely képes 0-30 Volt-ot és 0-10 Amper egyenáramot létrehozni, valamint fel van szerelve túláram és rövidzár védelemmel.



3-2. ábra NANKADF laboratóriumi tápegység

4. KIVÁLASZTOTT ELEKTRONIKAI KOMPONENSEK BEMUTATÁSA

A komponensek kiválasztásánál fontos szerepet játszott a kompatibilitási problémák elkerülése, valamint, hogy a saját kategóriájában olcsó és könnyen beszerezhető alkatrészek kerüljenek beépítésre, melyek nagy része Magyarországon is elérhető. A szakdolgozat 1-es számú mellékletében megtalálható egy lista az összes komponensről a beszerzés helyével. A mellékletben még feljegyzésre került az egyes komponensek tömege, melyből könnyen kiszámolható a teljes konstrukció súlya.

4.1 Raspberry Pi 4

A vezérlőnek egy Raspberry Pi 4-es (későbbiekben Raspberry) modell került kiválasztásra, a robotikában széleskörű felhasználhatóságának köszönhetően, amely képes csatlakozni a wifi hálózatra, de ha szükség van rá fel van szerelve egy Ethernet csatlakozóval, valamint egy Bluetooth 5.0 modullal. Találhatunk rajta még két darab USB 2.0-át, valamint 2 darab USB 3.0-át melyen keresztül adatátvitel valósítható meg soros kommunikációra képes eszközzel. A rajta található két darab micro-HDMI segítségével csatlakoztatni lehet a legtöbb forgalomban kapható kijelzőhöz. A primitívebb elektronikai komponensek csatlakoztatásához pedig 40 darab úgynevezett GPIO-val látták el, mely segítségével a következő tevékenységeket képes ellátni:

- Kommunikálni mikrokontrollerrel
- Magas vagy alacsony jel küldésére vagy fogadására alkalmas
- 5 Volt-al más komponenseket tud üzemeltetni
- A beépített USB C csatlakozó kiváltására alkalmas a megfelelő pinek használata

Előszeretettel használják a kis áramfelvétele és helyigénye miatt, illetve, hogy futtatni képes saját operációs rendszerén kívül a legtöbb Linux disztribúciót, amivel szinte végtelenségig testre szabható.

A 4 gigabyte-os modell biztosítja a párhuzamosan futó programok zökkenőmentes futtatását. Eredetileg a processzor egy hűtőbordával volt felszerelve, de a felgyülemlett hő a

teljesítmény visszafogását eredményezte a vezérlőben. A maximális teljesítmény és a biztonságos futás érdekében egy dupla ventilátoros megoldás lehetővé tette a stabil üzemi hőmérséklet tartását. A Raspberry beüzemelése a 6.2 fejezetben olvasható.

4.2 Arduino UNO és Arduino Zero

A motorok mozgatásához egy Arduino UNO (későbbiekben UNO) került kiválasztásra, ami a világ egyik legismertebb mikrokontrollere a nyílt és egyszerű fejlesztőkörnyezetének köszönhetően. Oktatásban és a hobbi elektronikában előszeretettel használják a hozzáférhető ára és a jó dokumentációja miatt, nem is beszélve az interneten található közösségekről, melyek keretében már nagyon sok lélegzetelállító projekt látott napvilágot. Az ismertségének köszönhetően nagyon sok gyártó készít külön az UNO-hoz kiegészítőket, amivel sok esetben számtalan kábeleztést lehet megspórolni, amely javítja a végtermék átláthatóságát. Nem is beszélve az elérhető funkciókról, ami ma már magába foglal GPS, Wifi, Bluetooth, motorvezérlő és még sok más modult, amivel az alap áramkörnek a képességet maximálisan ki tudjuk használni.

Az UNO-n található USB csatlakozó segítségével csatlakoztatható a Raspberry-hez, mely biztosította az áramellátását és a soros kommunikációt. Végül viszont a ROS-ban rejlő lehetőségek kihasználásában hátráltató tényezőként jelentek meg a hiányosságai. Az üzemeltetés közben érezhető késés figyelhető meg a két eszköz kommunikációja közben, így a kommunikációs csatorna gyorsabb válaszideje miatt váltásra volt szükség. A mögöttes meghúzó közösségi háttér és annak fejlesztési lehetőségeit viszont hiba lett volna veszni hagyni, ezért a mikro ROS oldalán található kompatibilis eszközök listáján is szereplő Arduino Zero-ra (későbbiekben Zero) esett a választás.

A Zero az UNO ATmega328P chipje helyett egy ARM Cortex-M0+ 32 bites processzoron alapuló mikrokontroller található, melynek 256 Kilobyte memóriájának köszönhetően nagyobb tárhelyet biztosít, alkalmassá téve teljesítmény igényű projektekben való felhasználását.

A Zero programozása többségében az UNO-hoz hasonlóan az Arduino IDE segítségével történik. A környezet támogatja a C/C++ nyelveket. A fejlesztők számára ez biztosítja az

egyszerű átmenetet UNO-ról, mivel az IDE használata és a programozási nyelv nagy része megegyezik. Telepíthető hozzá a mikro ROS programok írásához szükséges könyvtár, amely a már rendelkezésre álló ROS kommunikációs típusok segítségével egyszerűen tud adatot átadni, a mikrokontroller és a Raspberry-n futó bármilyen kód között, melynek listája a következő forrásban megtekinthető. [4]

A listában szereplő kommunikációs fajták a legtöbb alkalmazásban, ami a ROS-ban fellelhető Publisher, Subscriber és Service-ek esetén elégnak bizonyult. Viszont a robot mozgatásánál ebben az esetben is korlátozást jelentett, hogy nem volt képes saját információs csatornát létrehozni, csak a lista elemeire hagyatkozni, ami így negatívan hatott volna a jövőbeli fejlesztésekre, így ismét más megoldást kellett keresni. Végül a mikrokontrollerek használatát elvetve maradt a Raspberry, ami így olcsóbbá tette a teljes konstrukciót. A motorvezérlők irányításához pedig a Raspberry-n található GPIO tűskék használhatók.

4.3 Léptetőmotor

A projektben három darab NEMA 17-es léptetőmotor került beépítésre a robot mozgatására. A széles körben alkalmazható léptetőmotortípus elkerülhetetlen a pontosságot megkövetelő CNC és 3D nyomtatók esetén, valamint egyéb automatizált szerkezetek körében. A „NEMA 17” szókapcsolat a National Electrical Manufacturers Association (NEMA) által előírt motor méret szabványra utal, ahol a „17” szám a motor tengely oldali négyszög alakú, 1,7 hüvelyk x 1,7 hüvelyk (kb. 43,18 mm x 43,18 mm) méretű fixálási felületére utal.

Ezek a motorfajták rendelkeznek a „léptetési szög” sajátossággal, ami meghatározza, hogy hány darab lépés szükséges a 360 fok megtételéhez. A léptetési szög ennél a modellnél 1,8 fok (ami azt jelenti, hogy 200 ciklus szükséges 1 körbeforduláshoz), de eltérő léptetési szögekkel is megvásárolható.

A NEMA léptetőmotorok többségében két fázissal vannak felszerelve, amely magával hozza azt, hogy két tekercsegység van megkülönböztetve a kivezetésein. Ezek a motorok kétpólusúak, mely növeli a teljesítményt és pontosabb vezérlést eredményez, ellentétben

az egypólusú motorokkal. A szükséges vezérlő egység, többek közt a A4988 vagy a DRV8825 vezérlő modulokkal lehet szabályozni a motor:

- Irányát
- Sebességét
- Lépési finomságát
- Áramfelvételét

Az előnye, hogy könnyen integrálhatóak a legtöbb meglévő eszközbe, mivel szabvány határozza meg méretüket és csatlakoztatási pontjait. Ezek a motorok megfelelőek terhelés mellett is pontos pozicionálásra. Másik előnyük az aránylag alacsony ráfordítási igényük, amely biztosítja a felhasználó számára, hogy költséghatékonyan építsenek vagy fejlesszenek gépeket és berendezéseket. Ez a motortípus széles körben elérhető és könnyen beszerezhető a legtöbb elektronikával vagy robotikával foglalkozó üzletből.

Az általam választott gyártónál a NEMA 17 léptetőmotor 42 Ncm nyomatékkal rendelkezik, mely elegendő a 3,5 kilogrammos robot mozgatásához. Névleges áramerőssége pedig 1,5 Amper, melynek beállítását a motorvezérlőn kell elvégezni. A motorvezérlő beállítását a 4.5. fejezetben mutatom be.

Összefoglalva: a NEMA 17 léptetőmotor egy megbízható, pontos és költséghatékony megoldás, amely ideális számos különböző alkalmazáshoz, különösen ott, ahol a pontosság és a megbízható teljesítmény kulcsfontosságú. Ezek a motorok jól illeszkednek a modern automatizálási és gyártástechnikai kihívásokhoz, így kiváló választás a mérnökök, hobbi felhasználók és oktatási intézmények számára egyaránt. [5]

4.4 Arduino CNC motor shield V3

Az átláthatóság elősegítése érdekében fontos egy olyan motor shield használata, amely rendelkezik legalább három motorvezérlő foglalattal, amire a három léptetőmotor csatlakoztatása biztosítható.

A shield rendelkezik a szükséges négy motorvezérlő aljzattal, melyek a népszerű A4988 vagy DRV8825 motorvezérlő befogadására alkalmasak. A CNC-re jellemző koordináta

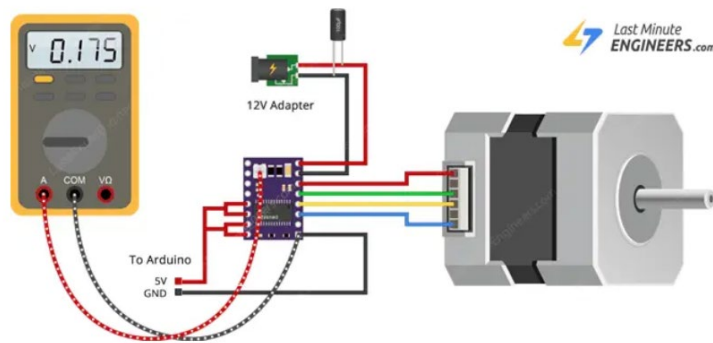
rendszer miatt három vezérlőt X, Y, és Z jelzéssel látták el, ha a felhasználás igényli, akkor egy negyedik csatlakoztatására is képes. Kivezetéseire továbbá csatlakoztathatunk véghelyezetszenzorokat. Üzemeltetése 12-36 V egyenfeszültségig terjed, melyből a 36 V-ot csak a DRV8825 képes meghibásodás nélkül hasznosítani. Az A4988 esetében maximum 24 V-ot használhatunk a problémamentes működés érdekében. Az akkumulátor kialakítása miatt 12 V állt rendelkezésre, ami lehetővé tette volna mindkét vezérlő használatát.

A shield támogatja a mikro léptetést, ha jó vezérlővel párosítjuk, amely finomabb lépések között eredményez. Az alapértelmezett léptetési szög használatával a motorok nagy rezonanciával működtek. A rezonancia következtében nagy mértékben rázkódott az egész szerkezet, ami hangossá tette a robotot működés közben. A finom léptetésnek köszönhetően viszont sokkal egyenletesebb tehető a motor körbefordulása. A tesztelés 1600 lépés/körbefordulás értékel zajlott le, ami megfelelőnek bizonyult.

Összegezve: az Arduino CNC shield V3 egy sok célra felhasználható eszköz, amely lehetővé teszi a felhasználók számára, hogy saját, több léptetőmotort igénylő eszközt építsenek vagy a rendelkezőt egyszerűsítsék.

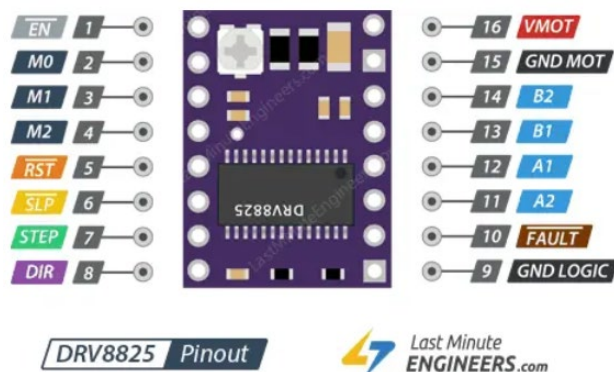
4.5 DRV8825 léptetőmotor vezérlő

Motorvezérlőnek a már korábbiakban említett DRV8825-re (későbbiekben motor vezérlő) esett a választás. Legnagyobb előnye, hogy képes akár 2,5 Amper-ig terhelni a léptetőmotorokat. A rendelkezésre álló motor névleges áramerőssége ennél alacsonyabb, ezért a vezérlőn található potenciométer segítségével limitálni tudjuk a kívánt erősségre, ami jelen esetben 1,5 Amper.



4-1. ábra DRV8825-ös vezérlő beállítása [6]

A fenti kép jól szemlélteti a referencia feszültség megméréséhez szükséges csatlakoztatási pontok helyzetét. Még mielőtt a mérést elvégezhetnénk meg kell győződnünk, hogy a vezérlőn a 200 lépés/körbefordulás van beállítva, hiszen hamis mérési eredményt kaphatunk. Ha az alább található ábrán a 2.-3.-4. csatlakozási pontot szabadon hagyjuk azáltal biztosra vehetjük, hogy a lépésköz a kívánt mennyiségen áll.



4-2. ábra DRV8825-ös vezérlő lábkiosztása [6]

A különböző multiméterek kialakítása miatt az eszközök üzemeltetése némiképp eltérhet. A saját multiméterem esetén egyenfeszültségre kellett állítani az eszközt; a méréshatár meghatározását a műszer automatikusan végezte el. A feszültségmérő negatív érintkezőjét krokodil csipesszel láttam el, hogy a műszerész csavarhúzó fém részére erősíthessem, így biztosítva a valós idejű mérést. A kívánt áramerősség beállításához az 1,5 Ampert el kell osztanunk kettővel és így kapjuk meg azt a referencia feszültséget, amit a vezérlő két ábrán látható két pontja között elengedhetetlen beállítanunk. A multiméter soros

csatlakoztatására a nagyon érzékeny potenciométer miatt volt szükség a pontos végeredmény biztosításához.

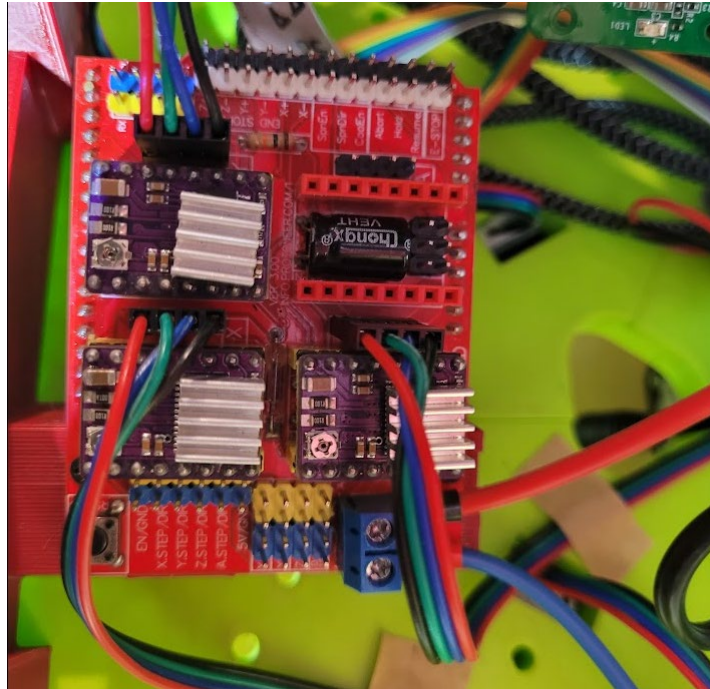
Az egyik fontos jellemző a finomléptetési képesség, amely szolgáltatja a motorok finomabb, pontosabb mozgását. Az DRV8825 akár 1/32 finomléptetési felbontást is támogat, ami precíz pozíció felvételt garantál. A lépések között a 4.6-2. ábra bal felső sarkánál található M0-M1-M2 jelölésű kivezetésekre csatlakoztatott magas, alacsony jel párosításával tudjuk a kívánt lépésközt beállítani, ami az alábbi táblázatból egyszerűen kiolvasható.

4-1. táblázat DRV8825 mikro lépésköz szabályozó jelszintek [6]

M0	M1	M2	Mikro léptetési felbontás
Alacsony	Alacsony	Alacsony	teljes lépés
Magas	Alacsony	Alacsony	1/2 lépés
Alacsony	Magas	Alacsony	1/4 lépés
Magas	Magas	Alacsony	1/8 lépés
Alacsony	Alacsony	Magas	1/16 lépés
Magas	Alacsony	Magas	1/32 lépés
Alacsony	Magas	Magas	1/32 lépés
Magas	Magas	Magas	1/32 lépés

A stabil üzemi hőmérséklet megtartásához 1-1,5 Amper áramerősség között javasolt hűtőbordát használni, felette viszont kötelező!

A CNC motor shield foglalataiba a vezérlők beépítése egyszerű, de kétféleképpen is megtehetjük, viszont csak az egyik a helyes. Az alábbi kép szemlélteti a motorvezérlők helyes orientációját a bővítő panelben.



4-3. ábra Arduino CNC motor shield-be behelyezett DRV8825-ös vezérlők

Összegezve: a DRV8825 motorvezérlő egy sokoldalú, biztos alapokon nyugvó és hatékony megoldás a léptetőmotorok üzemeltetésére. Alkalmas számtalan alkalmazásra, különösen olyan felhasználásokban, ahol a pontos motorvezérlés kritikus. Mikroléptetési adottsága, 2,5 Amperes terhelhetősége és széles tápfeszültségtartománya lehetőséget biztosít profi és amatőr felhasználók számára, hogy precízen és magas hatásfokon vezéreljék a léptetőmotorjaikat.

4.6 DFR0205 feszültségátalakító

A DFR0205 (későbbiekben feszültségátalakító) egy kis méretű, hatékony feszültségátalakító modul, melyet gyakran alkalmaznak hobbi elektronikai, valamint robotikai projektekben. Ez a komponens külföldről gyakran "buck converter", illetve "step-down converter" néven terjedt el, arra lett tervezve, hogy az ellenállásoknál magasabb hatásfokon, minimális hőveszteség segítségével legyen képes magasabb bemeneti feszültségből

alacsonyabb kimeneti feszültséget előállítani. Legnagyobb előnye az akkumulátor üzemeltette rendszerekben van, ahol a legkisebb energia megtakarítás is megbecsülendő.

A modult felszerelték egy kapcsolóval, ami előre meghatározott 5 Volt vagy potenciométer használatát engedélyezi. Ezzel biztosíthatjuk, hogy az alacsonyabb feszültségtartományt a működtetni kívánt eszközök bemeneti feszültségére tudjuk finom hangolni. Ez a választási lehetőség megfelelő volt, mert a Raspberry és a Zero is az 5 Volt-os feszültségi ágot használja. Bemenetére egy 12 Volt-os áramforrás került beépítésre.

Biztonságtechnikában se volt rá panasz, mert ellátták túlmelegedés és a kimenetein rövidzár védelemmel, mely zsúfolt elektronika áramköröknél magasabb árkategóriájú komponenseinket mentheti meg.

Mindezek mellett a kis mérete kedvez a legtöbb projektnek, különösen azoknak, ahol limitált hely áll rendelkezésre. Mint például a hordozható komponensek témaköre, nem is beszélve az önjáró mobil robotokról.

Összefoglalva: a DFR0205 feszültségátalakító egység egy többfelhasználású, kiváló hatásfokú eszköz, amely a hobby elektronikában, hordozható eszközökben és robotikai felhasználásban fordul elő leggyakrabban.

4.7 Áramforrás

A robotnál a kezdeti teszteléseket a korábban említett labortáppal végeztem el a komponensek védelme érdekében, amely a kapcsolási rajz stressz nélküli elkészítésében segített. A mobil robot a nevéből adódóan nem függhetett a rendelkezésre álló kábel hosszától, ezért be kellett fektetnem egy akkumulátorba.

4.7.1 Lítium akkumulátor

Az iparban a szabadon alakítható forma hatalmas előnyt jelent a LiPo áramforrások számára, amely lehetővé teszi a végtermék maximális helykihasználtságát. Ez biztosítja, hogy a robotokat minél kisebb házzal láthassuk el ezzel is hatékonyabbá téve. A LiPo akkumulátorok gyors töltési lehetőséggel rendelkeznek, ami előnyös olyan alkalmazásokban, ahol kritikus a maximális üzemidő kihasználás.

A legnagyobb hátrányuk viszont a biztonsági kockázat. A nem megfelelő vagy nem rendeltetésszerű használat hatalmas veszélyeket rejt a felhasználóra vagy eszközre, amibe be van építve. Elengedhetetlen a megfelelő töltési protokoll, valamint védelmi intézkedés, hogy elkerülhető legyen a tűz, illetve robbanásveszély. Kellő mélységű gyakorlati elektromikai ismeretek hiánya miatt egyedi akkupakk megépítésére nem került sor, helyette egy előre elkészített versenyautóba szánt modell oldotta meg ezt a problémát, aminek a következménye az lett, hogy meg kellett a robot váz méretét növelni, hogy képes legyen befogadni a nem kedvező formájú akkumulátort.

A robotban használt akkumulátort a 3S1P jelöléssel látták el, ami az angol (3 serial -1 parallel) rövidítése, ami magyarra fordítva azt az információt jelenti számunkra, hogy három cellát tartalmaz, amit sorba kötöttek össze. Mellette található, hogy a 11,1 Volt-tal rendelkezik, amit a cellák számával elosztva 3,7 Volt-ot kapunk, ami a szabvány által előírt legelterjedtebb lítium cella feszültség szintje. A következő érték 88,8 WH, mely értékből lehet meghatározni, hogy az áramkör összes teljesítményfelvétele hány óra üzemidőt biztosít. A mérések alapján 20-22 Watt körüli érték figyelhető meg melyet, ha osztóként használva a feltüntetett értékkel, eredményül kapjuk, hogy egy töltéssel 4-4,5 órán keresztül képes üzemelni. Az akkumulátort 50C értékkel látták el, mely kifejezés a kisütési sebességére utal. Ebből a jelölésből lehet meghatározni a maximális áramleadási intenzitást, amit még károsodás nélkül képes rövid ideig elviselni. A C szám és a kapacitás szorzatával tudjuk ezt az értéket meghatározni. Ennek a jelölésnek a használatát leggyakrabban olyan alkalmazásokban használják, ahol nagy áramot kell felszabadítani nagyon rövid időn belül, például drónokban, RC járművekben vagy nagy teljesítményigényű elektromos eszközökben.

4.7.2 Akkumulátor töltő

Az előző fejezetben felvázolt veszélyek miatt az LiPo akkumulátor töltését is csak a gyártó saját kész megoldása biztosította, elősegítve a hosszú távú felhasználhatóságát. Mindezek mellett a töltési idő minimalizálását is megerősítette, amivel elérhetővé vált a közel 60 perces töltési ciklus, amely a gyakori felhasználást igénylő területeknél elengedhetetlen.

4.8 Lítium akkumulátor kapacitás visszajelző kijelző

Az akkumulátor üzemeltetésél három eset kialakulását kell elkerülni, melyek a következők:

- Túltöltés
- Túlmerítés
- Rövidzárlat

A túltöltés problémáját az akkumulátor saját töltőjének használata orvosolta, mert a töltési ciklus közben valós időben ellenőrzi vissza a feszültségszintet, így biztosítva a szekvencia időbeni megállítást.

A rövidzár védelmet két lépésben sikerült biztosítani. Az első a megfelelő feszültségátalakító használata, mely rendelkezik rövidzár elleni védelemmel. Az 5 Voltos ágra csatlakoztatott eszközök védelme így biztosított. A 12 Voltos ágra pedig egy sorba kötött olvadó biztosíték beépítésével lett megoldva a teljes áramkör védelme.

Az egyedüli veszély, ami még maradt a túlmerítés kockázata, mely esetén egy programozható kapacitás visszajelző jelentette a megoldást, mely párhuzamosan csatlakoztatva a LiPo cellákkal valós időben jelzi a töltöttségüket százalékosan. Sőt, beállítható minimális feszültségszint, melyet, ha az akkumulátor átlép éles sípolásba kezd, jelezve a kísérlet vagy oktatásban történő felhasználásának szüneteltetését.

4.9 RPLidar A1

Az RPLIDAR A1 (későbbiekben LIDAR) egy kedvező árkategóriájú, 360 fokos forgó lézeres radarszkennerű (LIDAR) egység, melynek a leggyakoribb felhasználási területe kifejezetten a robotika, illetve automatizálást elősegítő alkalmazások számára készült. Kompakt kiterjedése, valamint könnyű súlya és csekély energiafogyasztása által ideális választás az akkumulátort igénylő mobil robotikához és más, térképezést igénylő felhasználásokhoz.

Főbb technikai jellemzői közé tartozik a 360 fokos valós idejű látómező, mely biztosítja a környezet pontos feltérképezését. Technológiájából adódóan pontos távolságmérést hajt

végre, ami elkerülhetetlen fontosságú a navigáció és térképezés során. Fontos megjegyezni, hogy ennek a modellnek a választása biztosítja a könnyű integrálhatóságot a legtöbb modern robotvezérlő rendszerbe, beleértve a népszerű Robot Operating System (ROS) környezetet.

A robot vezérlőjéhez két módon is csatlakoztatható, melyből az egyszerűbb a micro USB kábel segítségével a Raspberry egyik kimenetéhez történő összekötés. A másik csatlakoztatási mód a Raspberry GPIO kimeneteinek felhasználása, de a kábelezés átláthatóságának elősegítése érdekében az első módszer lett megvalósítva.

Összegezve: az RPLIDAR A1 egy széles körű, megbízható és költséghatékony eszköz a modern robotikában. Képessége a környezet pontos érzékelésére és a robotok navigációs képességének javítására teszi a rendszert ideális választássá a robotikai kutatásokhoz, fejlesztésekhez és kereskedelmi alkalmazásokhoz egyaránt.

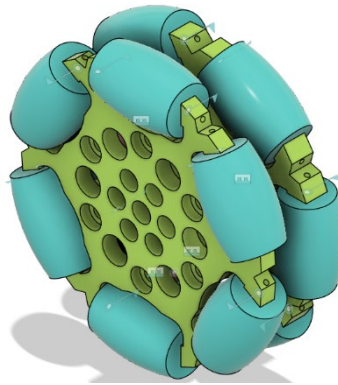
5. MECHANIKAI ELEMOK MEGTERVEZÉSE

Ebben a fejezetben a legyártásra kerülő mechanikai komponensek tervezésének folyamata kerül bemutatásra. A fejezet magába foglalja a vázszerkezet, a kerék és egyéb elemek megtervezését, ami meghatározza a robot megjelenését. Az összes elkészített CAD model tömörítve megtalálható a 2-es számú mellékletben.

5.1 Omni wheel

A mozgásforma meghatározásának kiemelkedő darabja mobil robotok esetén a kerék. A típus kiválasztása a 2. fejezetben már részletezve lett, így itt magának a kerék legyártásának ismertetése kerül bemutatásra.

Az Omni elnevezés az angol „omnidirectional” szóból ered melynek jelentése „minden irányú”, ezzel utalva a standard kerekektől eltérő haladási irányokra. A kereket kerülete mentén görgőkkel látták el, melyek a legkisebb tengely irányú erő következtében elfordulnak. A görgők számát a kerék átmérő és a kerekre ható súly befolyásolja, mely így általában négytől tetszőleges általában páros számig terjed. Terheléstől függően több síkon helyezhetők el a görgőket összefogó lemezek, melyek kiosztásának meg kell egyeznie. Biztosítani kell, hogy minden időpillanatban érintkezzen legalább egy görgő a talajjal, melyet a lemezek szögben történő elforgatásával lehet biztosítani.



5-1. ábra Omni wheel

Az Omni wheel egy magasabb árkategóriát képvisel a hagyományoshoz képest, mely nagyban kihatott volna a végtermék árára, ezért egy saját verzió előállítása tűnt

célszerűnek. A megtervezett kerék két oldalból áll össze, melyek kerülete mentén hat-hat darab görgő található az 5.1 ábra alapján.

Az összeállítás négy különböző nyomtatandó egységet foglal magába, melyek elhatárolásával egyszerűsödött a kerék terve, aminek köszönhetően a nyomtatási idő rövidült. A négy előállítandó összetevőt a következő táblázat tartalmazza a hozzá szükségesek filament megnevezésével.

5-1. Táblázat Filament fajták

Elnevezés	Anyag neve	Anyag jellege
csavar oldali váz	PLA	Műanyag
motor oldali váz	PLA	Műanyag
toldó	PLA	Műanyag
görgők	TPE	Gumi

A PLA-t igénylő elemek nyomtatása nem okozott nehézséget, viszont a TPE-vel nehézségek merültek fel. Egy szaktársam Zólmay Zénó segített a görgők legyártásában. Az általam elkészített CAD model segítségével legyártott számomra három szettre elegendő mintadarabot, melyek minősége megfelelő volt, így beépítésre kerültek a végső konstrukcióba.

A görgők rögzítése tengely felhasználásával került kivitelezésre, melyet kezdetben egy kiegyenesített gémkapocs töltött be a kedvező ára miatt. Tesztek során nagy súrlódási ellenállást merült fel, ami negatívan hatott a kerék élettartalmára, ezért egy kedvezőbb súrlódási tulajdonságú anyag felderítésére volt szükség. A rézre esett a választás a jóval kedvezőbb tulajdonságai miatt. A megfelelő átmérőjű réz rudat elektródák formájában használják az iparban, melyek még rendelkeznek alacsony fokú merevséggel is. Az új tengely már sokkal simábban futott az elődjéhez képest, így az utolsó verzióban a réz elektródák kerültek felhasználásra.

Nehézségeket okozott még a váz szerkezetén található tengely befogadására kialakított furatok átmérőinek meghatározása. A kiválasztott rézrúdon kotyogás mentesen fordultak

el a korábban kinyomtatott görgők, így a rögzítésük a vázon lett megvalósítva. A maximális súrlódás elérése volt a cél, ami még biztosította az összeszerelést. A 3D nyomtatók, különösen a hobbi nyomtatók tizedes pontossággal képesek anyagot elhelyezni a tárgyasztalon, mely minden esetben némileg eltért a számítógépen meghatározott értéktől. A folyó műanyag megszilárdulásának sebessége nagyban kihat a végtermék méreteire. A kedvező anyagköltségnek és a rövid nyomtatási időnek köszönhetően hét próbadarab került legyártásra, Ø 1-1,3 mm nagyságban öt százados érték különbséggel, melyből az Ø 1,2 mm-es rendelkezett a megfelelő összeszerelhetőséggel a kívánt súrlódás mellett.

Egy teljes kerék összeállításához végül szükség volt:

- 20 gramm PLA filamentre
- 30 gramm TPE filamentre
- 420 mm elektródára
- 4 darab M3-as csavarra
- 4 darab M3-as besüthető anyára

Ezek összértéke egy kereskedőtől vásárolt darabnak a 10 %-a. Teherbírása és élettartama ugyan elmarad a készen vásárolhatókéhoz képest, de az ár-érték arányt figyelembe véve egy működő alternatívát képvisel.

5.2 Léptetőmotor tengely

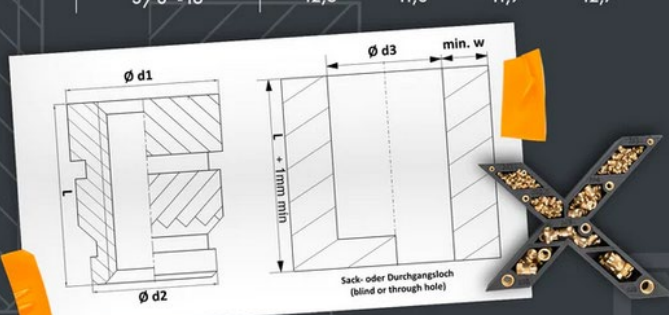
A léptetőmotorok esetén a tehermentesítés érdekében csapágy elhelyezésére volt szükség. A léptetőmotor és a csapágy túrései annyira finomak voltak, hogy a szerelhetőséget lehetlenné tette a szoros kötés, ezért egy áthidaló tengely tervezésére volt szükség. Az első prototípus darabot műanyagból kellett kinyomtatni, de az Ø 5-ös belső átmérőjű csapágy behatárolta a tengely méreteit. A kinyomtatott csapágy már szemrevételezve is alkalmatlannak bizonyult a feladat ellátására. A tapasztalat azt mutatta, hogy ilyen kicsi átmérő esetén a műanyag hosszú távon nem bírhatja ki a rá nehezedő terhelést. A műanyagot elvetve a fém lett az egyetlen alternatíva. A Grundfos Hungary Kft-nél, ahol duális gyakorlaton veszek részt, a középiskolásoknak forgácsolást oktató tanár segítségével

legyártásra került a 3-as számú mellékletben található tervek szerint három darab tengely. Az elkészült tengelyek már alkalmasnak bizonyultak a robot önsúlyának megtartásához.

5.3 Többszintes váz

A váz kinyomtatásánál problémát okozott a megfelelő befoglaló méretek kialakítása. Mivel a fókusz az átláthatóságon volt, a kerekeket leszámítva a váz külső átmérője közel 250 mm-re növekedett. A rendelkezésre álló nyomtatók nem bizonyultak alkalmasnak az egy egységben történő kinyomtatásra, ezért több részre törésre volt szükség. A motorok effektív működéséhez 120 fokos eltolásra volt szükség, így a három részre osztás volt célszerű. A furatokat fix helyen lehetett kialakítani a körcikkek felezővonalától mért távolsággal. A kialakult alakzat oldalain túlnyúlások lettek elhelyezve, melyek egymásba csúsztak. A vastagabb túlnyúlás magassága a besüthető anyára előírt minimális „L” hossz alapján került meghatározásra, mely az alábbi ábrán látható táblázat sorából leolvasható a használatra kiválasztott M3-as menet esetén:

Metrische ISO-Gewinde	Zöllische UNC-Gewinde	Ø d1	Ø d2	Ø d3	L	W
M2	#2-56	3,6	3,1	3,2	4,0	1,3
M2,5		4,6	3,9	4,0	5,7	1,6
M3 Short		4,6	3,9	4,0	4,0	1,6
M3x5x4 Voron		5,0	4,25	4,4	4,0	1,3
M3	#4-40	4,6	3,9	4,0	5,7	1,6
M4 Short		6,3	5,5	5,6	4,0	2,1
M4	#8-32	6,3	5,5	5,6	8,1	2,1
M5 Short		7,1	6,3	6,4	5,8	2,6
M5	#10-24	7,1	6,3	6,4	9,5	2,6
M6	¼"-20	8,7	7,9	8,0	12,7	3,3
M8		10,1	9,5	9,6	12,7	4,5
	3/8"-16	12,6	11,8	11,9	12,7	6,0



The technical drawing shows a cross-section of the frame part with dimensions: Ø d1, Ø d2, Ø d3, L, and min. w. A 3D model of the part is shown to the right, illustrating the multi-layered structure and the internal holes. The text 'Sack- oder Durchgangsloch (blind or through hole)' is visible at the bottom of the drawing.

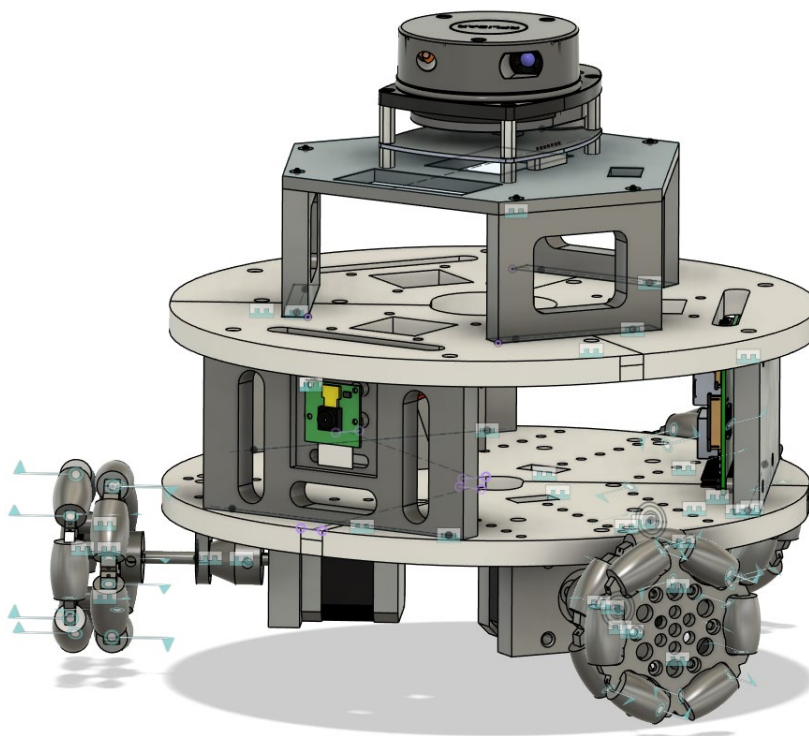
5-2. ábra Ruthex méret táblázat [7]

A leolvasott értéket felfelé kerekítve kapott egész szám lett a vastagabb túlnyúlás magassága. A bázis teljes magassága 10 mm-ben került meghatározásra a csavarok szorítóerejének megfékezésének biztosítása érdekében. Így a vékonyabb túlnyúlás magassága 4 mm lett. Ugyanezen körcikk tetején a motorok rögzítéséhez szükséges furatok kaptak helyet, míg az alján a felsőbb szinteket tartó falaké. Az alap összeszerelése három daraból sikeresnek bizonyult, nem igényeltek egyedi módosításokat, így javítva az újra felhasználhatóságot. A körcikken komponensek elhelyezésére alkalmas falak kerültek megtervezésre, melyek a következők:

- A shield-et tartó fal
- A Raspberry-t tartó fal
- Kamerát tartó fal

A falak végső kialakítása a furatok és a csatlakozók helyzetének figyelembevételével lett megtervezve. A nehézséget a shield jelentette, melyet eredetileg arra terveztek, hogy egy UNO-ba beillesztve kerüljön rögzítésre. Mivel a mikrokontrollerek használata el lett vetve, így a rajta helyet kapott két furat segítségével volt esély a rögzítésre. A csavarok helyének kialakítása ebben az esetben viszont nem volt elég a shield alján található egyenetlen felület miatt. A NYÁK lap készítése során a forrasztási pontjain kitüremkedések alakultak ki, viszont síknak tekinthető „szigetek” maradtak, melyeket fel lehetett használni. Ezekből a szigetekből képzett lenyomat segítségével biztosíthatóvá vált a lap biztonságos rögzítése. A falak tetejét lezáró szint hasonlóan került kialakításra, mint az alap, viszont a Raspberry-t tartó fal következtében változtatásokat kellett végezni. A Raspberry felhelyezése a mikro-HDMI-k mennyezet felé nézésével lettek kialakítva a könnyű hozzáférhetőség miatt. A mennyezetet a portok helyénél viszont szabadon kellett hagyni, a kábel egyszerű csatlakoztatásának biztosítása érdekében. A kikönnnyítés viszont egy helyre esett volna a rögzítést jelentő vállakkal, mely negatívan hatott volna a csavarok helyzetére, ezért 30 fokkal el lett tolva a felső lemez. A módosításokat követően az alaphoz hasonlóan három egyforma darab szükséges az összeszereléshez. A kapcsoló, a töltöttség visszajelző és az RPLidar elhelyezéséhez szükség volt egy harmadik szintre a második szint egyszerű kialakításának megtartásához. A felső szinten helyet kaptak, a

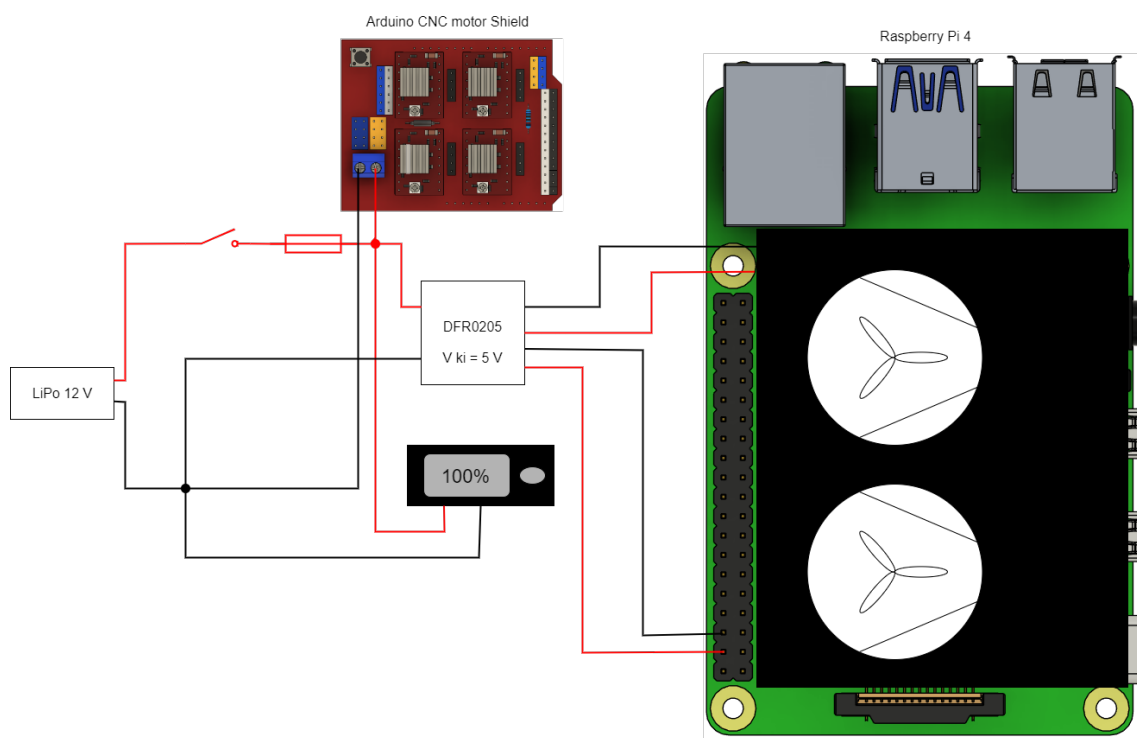
rögzítéshez szükséges csavarhelyek, a RPLidar rögzítését biztosító csavarhelyek, illetve a téglalap alakú vájatok melyekbe beilleszthető a kijelző és a kapcsoló. A model-ek összeállítása a következő ábrán látható:



5-3. ábra Robot összeállítás

6. ELEKTROMOS BEKÖTÉS

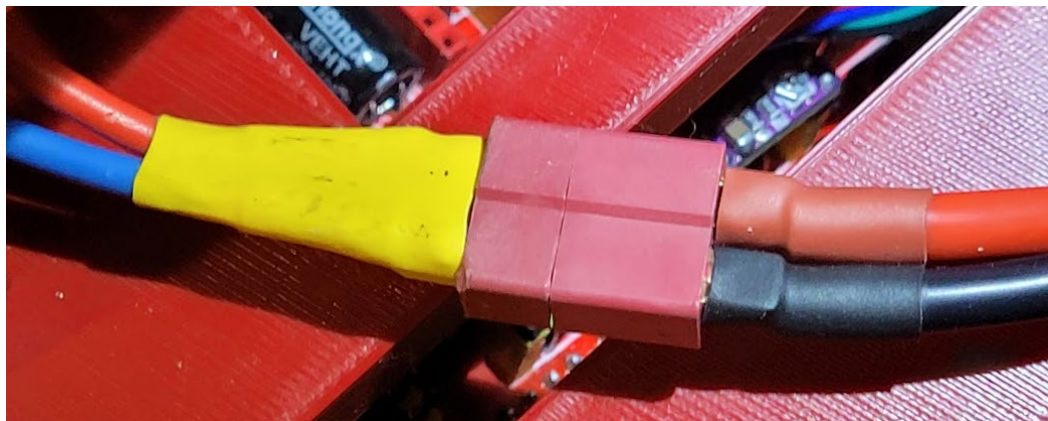
Az elektronikai bekötésnél fontos figyelmet kapott a biztonság és az átláthatóság. Kezdetben a vezetékek csatlakoztatása sorkapocs segítségével történt, mely egyszerűvé tette a módosítások véghezvitelét. A végleges kapcsolást a következő ábra szemlélteti:



6-1. ábra Teljes robot kapcsolási rajza [8] [9]

A sorkapocs használata a fent látható ábra kialakulásában hatalmas szerepet töltött be, rávilágított egyszerűsítésekre, melyek átláthatóbbá tették az áramkört. Nagy helyigénye miatt viszont a továbbiakban szükségessé vált cseréje. A vezetékek összeillesztéséhez a képen látható csomópontoknál Wago-k kerültek felhasználásra, melyek segítségével továbbra is bonthatók maradtak az erek. A forrasztással való kötés kialakításának mellőzése volt a rövid távú cél az egyszerű reprodukció biztosítása érdekében, egy helyen viszont szükségesnek bizonyult. Az akkumulátorhoz való csatlakoztatáshoz egyedi kábel elkészítésére volt szükség, hiszen rendelkezésre csak a T csatlakozó állt.

A következő képen látható az elkészült kábel, mely zsugorcsővel lett ellátva az emberi biztonság védelmének érdekében:

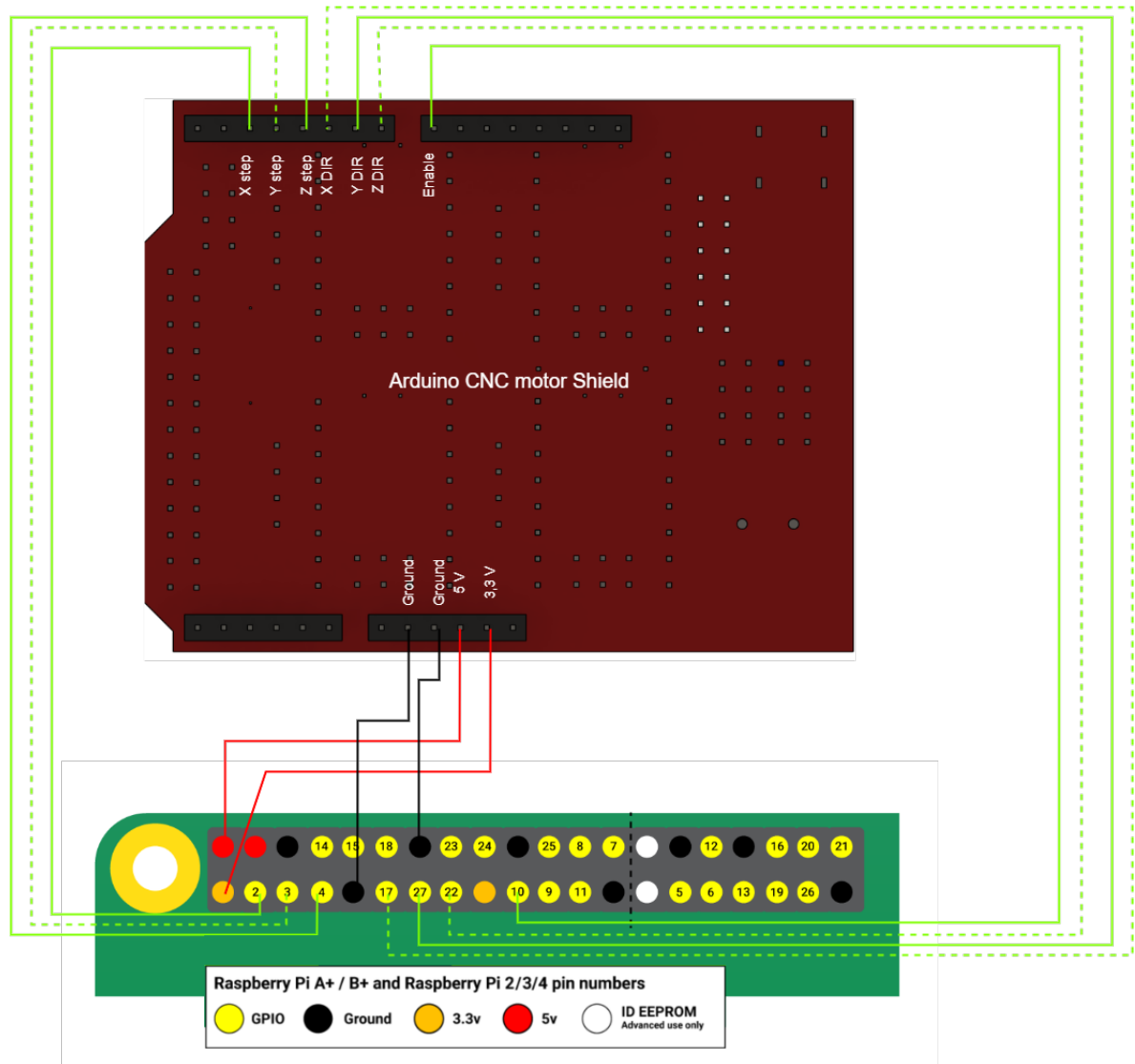


6-2. ábra Akkumulátort csatlakoztató kábel

A shield és feszültségátalakító kimenetein lemezes sorkapcsot helyeztek el a gyártók, így azokhoz való csatlakoztatásához lapos érvéghüvelyek lettek felhasználva, biztosításuk az eszközökön található csavarok meghúzásával történt meg. Az áramkörben mért maximális áramerősség értéke 3,5 Amperre tehető. A biztosíték foglalat autókban használt kivitel befogadására tervezték, melybe egy 5 Amper értékű védelem lett elhelyezve. A Raspberry és a rajta található ventilátorok csatlakoztatásához „jumper cable” került felhasználásra a rajtuk található csatlakoztatási formának megfelelően. Színkódolt kábelek lettek elhelyezve biztosítva a fázis (piros) és föld (fekete) könnyű megkülönböztetését.

A léptetőmotor működtetését megfigyelve a shield-el bebizonyosodott, hogy tekercseinek kivezetései nem egyeznek meg a motor vezérlőjével. A felcserélt kábelek egy tiltott állapotot idéztek elő, mely az óramutató járásával megegyező és ellentétes irányú elfordulásának egyszerre biztosított magas jelet, mely hosszú távon a léptetőmotor károsodásához vezet. A hiba megszüntetését a kábel módosításával lehetett megszüntetni a két középső vezeték felcserélésével. A shield alján található kivezetéseket eredetileg UNO-hoz való csatlakoztatásra fejlesztették ki, viszont a szalagkábel anya típusú kivezetése is kompatibilisnek bizonyult.

A Raspberry és a shield összekapcsolása az alábbi ábrán látható módon került kialakításra:



6-3. ábra Raspberry Pi és CNC shield kapcsolata [8] [10]

A kivezetések közel fele nem került felhasználása, mert véghelyzet jelzőknek, illetve soros kommunikáció használatának biztosítására lettek felszerelve, melyre ebben az esetben nem volt szükség. Az RPLidar üzemeltetésére USB-n keresztül került sor. A robot elején található kamera a saját szalagkábelével csatlakozik a Raspberry-n külön erre kialakított csatlakozójához. A bontható kialakításnak köszönhetően a jövőben további eszközök elhelyezésére is van lehetőség, mind 12 V, mind az 5 V-os ágon.

7. PROGRAMKÖRNYEZET KIALAKÍTÁSA ROBOT OPERATING SYSTEM-HEZ

Ebben a fejezetben a fejlesztő környezet kialakítása kerül bemutatására, mely lehetővé teszi, hogy problémamentesen tudjuk a ROS környezet adta lehetőségeket hasznosítani a robot programozásához. A ROS környezetnek a legújabb hosszú távon támogatott verziója a ROS2 Humble Hawksbill (innenről Humble). Ennek a disztribúciónak (verziónak) a teljes lehetőségeit ki tudjuk használni elengedhetetlen a linux környezetek megismerése. A Humble fejlesztői a fejlesztőkörnyezet futtatását Ubuntu 22.04.03 LTS verziójával ajánlják, ami az egyik legelterjedtebb Linux disztribúció a könnyen átlátható kezelő felülete és gyakori biztonsági frissítéseinek köszönhetően. A ROS környezet telepítését meg kell tennünk legalább a Raspberry-n, ahol a robot mozgásának teszteléséhez elengedhetetlen, hogy telepítve legyen. Opcionálisan pedig a Raspberry-nél erősebb számítógépen is felállíthatunk egy fejlesztőkörnyezetet, ahol nem kell a hardware számítási limitációi miatt kompromisszumokat kötnünk.

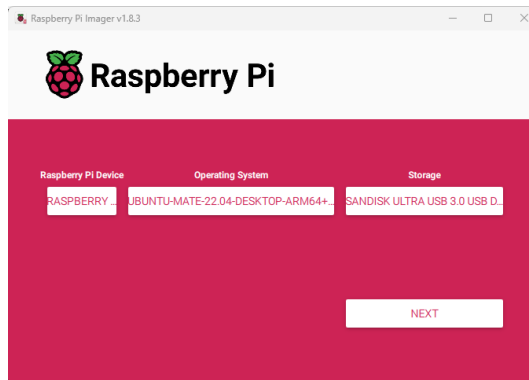
7.1 Ubuntu Mate telepítése a Raspberry Pi 4-re

A Raspberry-ben található processzor nem elég erős a teljes verziós Ubuntu 22.04.03 LTS futtatására, ezért ebben az esetben a maximális lehetőségek kihasználásához a maximális kompatibilitást biztosító Ubuntu Mate telepítése javasolt, mely rendelkezik a Humble futtatásához szükséges csomagokkal, de a teljesítményt visszafogó grafikus eszközökkel pedig nem.

A szükséges telepítő képfájlokat az Ubuntu Mate weboldaláról tudjuk elérni, ahol a letöltés fülről ki tudjuk választani a Raspberry-vel kompatibilis verziót, amiből a 64 bites verzióval kompatibilis az általam választott ROS disztribúció.

A letöltést követően a fájlt a Raspberry saját SD kártyák írására fejlesztett szoftverével tudjuk telepíteni az eszközünkre. Csatlakoztassuk eszközünkhöz a fejlesztéshez használni kívánt SD kártyát (minimum 4 Gigabyte), ajánlott egy 32 Gigabyte-os tárhely a későbbi

fejlesztések miatt. A csatlakoztatást követően indítsuk el a Raspberry Pi Imagert, melyben a telepítési paraméterek kitöltésével testre tudjuk szabni.



7-1. ábra Raspberry Pi Imager

Ha mindent jól állítottunk be, akkor a fenti kép fogad minket, mely esetén a NEXT gombbal elindíthatjuk az írási folyamatot.

A telepítés befejezése után az adathordozót behelyezhetjük a Raspberry-be, ami a telepítési folyamat végét jelenti. A Raspberry indítása után, már csak a felhasználónév és jelszó konfigurálása vár ránk, amit követően már használatra kész az eszköz.

7.2 Ubuntu 22.04 telepítése a fejlesztői számítógépre

A fejlesztés megkönnyítése miatt a saját eszközre is szükség volt az Ubuntu megfelelő verzióját telepíteni, melynek különböző módjai lehetségesek, a három legismertebb:

- Standard telepítés
- Virtuális gép használata
- Dual boot telepítés

A standard telepítési opció elvetésre került, mert akkor elérhetetlen lett volna a Microsoft Windows-al kompatibilis alkalmazások melyekből például a Fusion 360 használata elengedhetetlen a vázszerkezet megtervezéséhez.

A virtuális gép ma egy nagyon elterjedt megoldás, ami ebben az esetben is működőképes, de a kommunikációs protokollokkal, valamint a távoli elérés nemkívánt nehézségekkel

járhat, ezért ez az opció is elvetésre került. (Akik tapasztaltabbak a virtuális gépek üzemeltetésében azoknak ez az opció is járható.)

Végül maradt a dual boot opció, amely a nevéből adódóan két részre osztja háttértárunkat, így bármelyik operációs rendszer indítása mellett dönthetünk. Ennek a megoldásnak nincsenek negatívumai, leszámítva, hogy ez a megoldás foglalja a legtöbb tárhelyet, de cserébe így használhatjuk ki a legtöbb számítási kapacitását a rendszerünkben.

Ezek voltak a főbb módszerek az Ubuntu 22.04 LTS telepítésére. Mindegyikük más-más igényt képes kielégíteni, illetve más technikai tudást igényel. Fontos, hogy minden esetben a hivatalos források weboldaláról töltsük le az operációs rendszer telepítéséhez szükséges képfájlokat, és aprólékosan kövessük a telepítési útmutatókat a nem várt hibák elkerülése érdekében.

7.3 Robot Operating System (ROS) telepítése és testre szabása

A ROS telepítéséhez a Linux operációs rendszerben Windows-al ellentétben nem egy telepítő fájl letöltésével valósul meg amit letöltöttünk az internetről, hanem az összes linux disztribúcióban megtalálható terminál segítségével tudunk alkalmazásokat telepíteni, köztük a ROS-t is. A telepítés megkezdése előtt fontos megbizonyosodni arról, hogy az operációs rendszer naprakész, amit a következő paranccsal tudunk megtenni.

```
sudo apt update
```

7-2. ábra Frissítések keresése az Ubuntu terminál segítségével [11]

A sudo-val kezdődő parancsok minden esetben adminisztrációs jogosultsággal fogja végrehajtani az őt követő parancsokat, amely jelszómegadási kötelezettséggel jár. A sudo parancsoknak a saját biztonságunk érdekében érdemes minden esetben utánanézni.

Ha a rendszer talál frissítési lehetőségeket, ami előfordul egy frissen telepített operációs rendszer esetén is, a következő parancs használatával tudjuk telepíteni. Ez eltarthat percekig is, valamint újraindítási kötelezettséggel is járhat, ami után tanácsos ezt a két parancsot ismételten használni.

```
sudo apt upgrade
```

7-3. ábra Frissítések telepítése az Ubuntu terminál segítségével [11]

Megbizonyosodva a rendszerünk naprakész állapotáról a következő paranccsal kiadásával megbizonyosodhatunk, hogy a rendszerünk alkalmasnak bizonyul-e a ROS telepítésére.

```
locale # check for UTF-8

sudo apt update && sudo apt install locales
sudo locale-gen en_US en_US.UTF-8
sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8
export LANG=en_US.UTF-8

locale # verify settings
```

7-4. ábra ROS előfeltétel vizsgálata [11]

Ha hibát kapunk, akkor a ROS közösségi platformjain tudunk a hibaüzenethez megoldást találni, de ha az operációs rendszer telepítésénél pontosan követtük a telepítési lépéseket, akkor ilyen hiba nem léphet fel.

Ezt követően hozzá kell adnunk a következő forrásokat a rendszerünkhöz. A későbbiekben, ha frissítést hajtunk végre, akkor a ROS-hoz köthető adatcsomagokat is ellenőrzi a rendszer, amit következő parancsokkal tudunk megtenni:

```
sudo apt install software-properties-common
sudo add-apt-repository universe
```

7-5. ábra Ubuntu Universe engedélyezése [11]

```
sudo apt update && sudo apt install curl -y
sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o
/usr/share/keyrings/ros-archive-keyring.gpg
```

7-6. ábra ROS szerver hozzáférési kulcs hozzáadása [11]

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-
archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(. /etc/os-release &&
echo $UBUNTU_CODENAME) main" | sudo tee /etc/apt/sources.list.d/ros2.list >
/dev/null
```

7-7. ábra ROS repository hozzáadása a forrásokhoz [11]

Az így hozzáadott forrásokból származó adatcsomagokat a ábránkon használt parancsokkal tudjuk letölteni, melyek elengedhetetlenek a ROS környezet telepítéséhez. A telepítést viszont a következő paranccsal tudjuk elindítani, mely tartalmaz általános demókat, példa gyűjteményeket és a 3D szimulációhoz elengedhetetlen RViz programot.

```
sudo apt install ros-humble-desktop
```

7-8. ábra ROS telepítése [11]

A ROS specifikus parancsok használatához a környezeti változókat még be kell állítani melyeket a következő paranccsal tudunk megtenni:

```
source /opt/ros/humble/setup.bash
```

7-9. ábra ROS környezeti változók hozzáadása a terminálhoz [11]

Ezek után már csak létre kell hozni a számítógépünkön egy munkatér, melybe bekerül az összes program, model, szövegfájl és még sok más, amit a következő parancsokkal tudunk megtenni:

```
mkdir ros2_ws
cd ros2_ws
mkdir src
colcon build
```

7-10. ábra ROS munkatér létrehozása

Ezek után már csak egy beállítás maradt, – ami nem kötelező – de hosszú távon sok időt lehet megtakarítani vele. A Linux terminálban, ha egy parancsot elkezdünk írni, akkor a tabulátor segítségével a rendszer automatikusan befejezi nekünk, ha az addig beírt szöveg alapján csak egyetlen lehetséges parancsot talál. A ROS parancsok esetén ez még nincs engedélyezve, de kézzel hozzá tudjuk adni a terminál indításával egy időben, amit a következő paranccsal tudunk megtenni:

```
gedit ~/.bashrc
```

7-11. ábra Terminál indítási változók megtekintése

Ez megnyit nekünk egy szöveges fájlt, aminek a végére a következő két sort kell bemásolnunk vagy begépelnünk.

```
source /opt/ros/humble/setup.bash
source /usr/share/colcon_argcomplete/hook/colcon-argcomplete.bash
```

7-12. ábra Bashrc file módosítása

A ROS munkatér felállítása ezzel befejeződött, alkalmas a mobil robot platformhoz tartozó kódok fogadására és futtatására. A felvázolt lépések egységesek a Raspberry és a fejlesztői gép között, nem igényelnek plusz lépéseket.

8. A ROBOTOT IRÁNYÍTÓ PROGRAMKÓDOK

A fizikailag és elektromos tekintetben kész konstrukció birtokában elkezdődhetett a programkód írása, melyek biztosítani hivatottak, hogy az egymástól eltérő elemek egy egészet alkossanak. Az általam írt programok elérhetők a 4-es számú mellékletben található publikus GitHub oldalamon, hogy bárki meg tudja nézni és képes legyen tanulni belőle.

8.1 ROS programkörnyezet rövid bemutatása

A GitHub oldalamon található programok megértéséhez elengedhetetlen a ROS környezet alapvető ismerete. A neten fellelhető dokumentációban a következő kifejezések gyakran felmerülnek, melyek magyarázatra szorulnak [11]:

- Package
- Topic
- Publisher
- Subscriber
- Action

8.1.1 Package

Package-ek hasonló feladatot látnak el, mint a mappák a számítógépünkön. Azt a célt szolgálják, hogy megszerezni tudjuk a kódokat felhasználási terület alapján. A GitHub oldalamon található mappák is mind-mind egy külön Package-t jelölnek. Az elnevezések nem kötöttek, de fellelhetőek névadási normák, melyeket követve projekt átadás esetén szükségtelenné válik dokumentációban körülírni a másik szakembernek a név mögött álló tartalmat, hiszen ő is ezeket a normákat használja. A norma azt írja elő, hogy „<robot neve_felhasználási kör>”. A GitHub-on is látható, hogy a robot a Humblebot nevet kapta, a ROS distro Humble és a robot „bot” szó összepárosításával.

8.1.2 Topic

A Topic egy futó programhoz társított név, mely segítségével be tudjuk azonosítani az aktuális pillanatban munkát végző egységeket. A ROS-nak erre egy parancsa is van, ami kilistázza az összes futó alkalmazás nevét. Erre azért is van szükség, mert a név alapján információhoz juthatunk a Topic mögött álló program tartalmára anélkül, hogy azt megnéztük volna. A legfontosabb viszont, hogy csatornaként is működik a tévéhez hasonlóan. A csatorna nevének ismeretében megtekinthető az adott időpillanatban átfolyó adatok, mely magába foglalja a legtöbb programozási nyelvben ismert típusokat, melyekből akár többet is felhasználhatunk egy csatornán belül. A kerék mozgatásához három darab Float32 változó került összefűzésre egy csatornába a három különböző kerékre.

8.1.3 Publisher és Subscriber

A Publisher és a Subscriber a csatorna (Topic) két végét testesíti meg. Az emberi kommunikációval ellentétben itt a szerepek nem cserélődnek fel. Az oda-vissza kommunikáció két különböző csatornán kerül lebonyolításra. Erre nagyon jó példa a rádióadó működése. A jelet biztosító torony küldi a felhasználó felé (Publish-eli) az információt, amibe egy zsebrádióon keresztül bele tud hallgatni (Subscriber-elni), ha viszont zenét akar kérni egy másik csatornán, azt telefonon keresztül teheti meg.

8.1.4 Service

A Service a Topic-hoz hasonlóan itt is két oldala van a kommunikációnak. Van egy szerver oldal, amely folyamatosan készenlétben fut, egy előre meghatározott adattípust várva a klientsől. Az adat beérkezését követően a szerver feldolgozza az algoritmus alapján és a kapott eredményt továbbítja a kliensnek. Hasonlóan működik, mint az interneten található időjárás előrejelző oldalak. A felhasználó (kliens) megadja a weboldalnak (szervernek) a település nevét, amelyről jelentést szeretne lekérni, a szerver pedig tudja a hozzá tartozó időjárás adatokat, amit megjelenít a lekérdezőnek. A roboton az RPLidar elindítása és megállítása történik Service-el. A szerver ebben az esetben egy „stop/ start” jelet fogad, aminek függvényében megállítja vagy elindítja a robot tetején található RPLidar forgását.

8.1.5 Action

Az Action funkciók szempontjából nagyon hasonlít a Service-hez. További három tevékenységi körrel egészül ki:

- Bemeneti adat ellenőrzése
- Valós idejű érték visszajelzés
- Valós idejű megszakítás

A bemeneti adat ellenőrzése magában foglalja a bemeneti adatok értékének vizsgálatát, beállítható, hogy mit fogadjon vagy utasítson el. Túl alacsony vagy magas értékeket ezzel a módszerrel ki lehet szűrni még mielőtt hibához vezetne. Ha elfogadta az értéket, valós időben visszajelzés küldésére képes a kliens számára a feladat alakulásáról. Utolsóként pedig a szerver futása megszakítható, ha korábban éri el a meghatározott célt, vagy a felhasználó közbeavatkozott.

8.2 Mozgás szerver

A robotom mozgatására egy Action szerveret működtetnek a 8.1.5. fejezetben felsorolt előnyök miatt. A szerver bemenetként három Float32-es értéket kap, amelyek a következők:

- X irányú elmozdulás [m/s]
- Y irányú elmozdulás [m/s]
- Z tengely körüli elfordulás [rad/s]

A beérkező adatokat fel kell dolgozni olyan formába mely egy pozitív vagy negatív irányú elmozdulásban jelenik meg a kerekeken, ami pedig a következő képlettel számítható ki:

$$V_x \cdot \sin \gamma + V_y \cdot \cos \gamma + L \cdot \omega \quad (1)$$

, ahol

V_x : az X irányú elmozdulás

V_y : az Y irányú elmozdulás

γ : a kerék szögben való elhelyezkedése a vázon (0, 120, 240 fok) radiánban

ω : a Z tengely körüli elfordulás

L: a robot z tengelye és a kerék középpontja közötti távolság méterben

A 0 foknál elhelyezkedő kerékre jutó sebesség 1,0-ás értékű elmozdulását a következőképpen helyettesíthetjük be:

$$1,0 \cdot \sin 0 + 0,0 \cdot \cos 0 + 0,160 \cdot 0,0 = 0 \quad (2)$$

A léptetőmotorok esetén a lépések számát kell meghatározni, amihez ki kell számolni a kerék területét és elosztani az egy teljes körbeforduláshoz szükséges lépések számával. Az így kapott értéket a fent látható képlet segítségével meghatározott sebesség mm/s-os értékével tovább kell osztani. A képletek segítségével így az egy méterre elküldött robot a valóságban képes volt 0,98 cm megtételére.

Már csak egy probléma merült fel. A kerekek mozgásánál biztosítani kellett, hogy a mozgást egy időben kezdjék el és fejezzék be, még abban az esetben is, ha a két kerék lépésszámai között 2000 különbség van. A probléma megoldása, hogy külön változók kerültek bevezetésre az összes motor esetén, melyekben tárolásra kerül a már megtett lépések száma. A keréken aktuálisan megtett lépésszámot el kell osztani a korábban kiszámolt értékkel, majd be kell szorozni százszal, ami egy százalékos értéket határoz meg. A kapott százalékos értéket össze kell hasonlítani a maximális lépéssel rendelkező motor esetén kapott eredménnyel és ha az adott kerék esetén alacsonyabb, mint a legmagasabbnál, akkor léptetés történik. A teljes programkód a GitHub oldalon a következő elérési úttal tekinthető meg: `humblebot_nodes/humblebot_nodes/stepper_vel_action_server.py`

9. PROJEKT KIÉRTÉKELÉSE

9.1 Továbbfejlesztési lehetőségek

Projekt szűkös határideje következtében a fejlesztést a korábban taglalt pontig sikerült elvégezni. A továbblépési lehetőségek száma viszont végtelen. A következőkben olyan ötletek kerülnek ismertetésre, melyek brainstorming során felmerültek, viszont a jelenlegi állapotban nem valósultak meg.

9.1.1 IPS kijelző

A ROS alapú mobil robot esetén egy beépített kijelző nagyban megkönnyítheti az egyszerűbb feladatok elvégzését, ami felmerülhet az üzemeltetés közben, például az eszköz csatlakoztatása egy másik wifi hálózatra vagy Bluetooth-os kontroller párosítása.

A kijelző a robot állapotára utaló fontos adatok megjelenítését teszi lehetővé, többek között az akkumulátor töltöttségi szintjének, hálózati kapcsolat jelerősségének, valamint pillanatnyi feladatainak megjelenítését. Így átláthatóvá tehető a felhasználóknak, akik teljesebb képet kapnak a robot működéséről. Hibák esetén pedig könnyebben lehet a hibaüzenetekhez vagy a naplófájlokhoz hozzájutni, esetleg instrukciókat adni a felhasználó számára.

9.1.2 Felhasználói felület

A kijelzővel együtt egy felhasználói felület kialakítása is nagyban növelte volna a felhasználói élményt. A linux terminál használata a Windows-hoz szokott személyek körében gyakran vált ki nehézséget. Az egyedi felület lehetőséget biztosíthat a már meglévő programok paraméterezésére, láncba fűzésére, az összetettebb feladatok megoldásához.

9.1.3 Automatikus navigáció a valós térben

A ROS2 beépített Nav2 rendszer segítségével bármelyik mobil robot képessé tehető környezete feltérképezésére. Az így elkészült térképen meghatározott útvonalakon ütközésmentesen képes áthaladni.

9.2 Összegzés

A szakdolgozat elején támasztott célt részben sikerült megvalósítani, miszerint csak hobbi vagy könnyen hozzáférhető eszközök megengedettek a projekt sikerre vitelében. A projekt nagy részében e cél tarthatónak bizonyult, kivéve a kereket tartó tengelyek esetében, ahol elkerülhetlenné vált a forgácsoló gépek használata. A tengelyeket leszámítva a cél szem előtt tartása maximálisan teljesíthetőnek bizonyult, ami így inspirálóan hathat másnak a saját robotja megalkotásához.

A beépülő elemek esetében is az elsődleges szempont a hozzáférhetőség biztosítása volt, többségében olyan elemek kerültek beépítésre, melyek a legtöbb nagyvárosban található elektronikai szaküzletből beszerezhetőek. A használt szerszámok esetén sem volt ez másképp a 3D nyomtató és a labortáp kivételével, melyek viszont a legtöbb mérnöki intézményben rendelkezésre állnak. Az 1-es számú mellékletben található Excel táblából látható, hogy ár szempontjából már távol sem abba a kategóriába esik, mint a felvezetőben használt robot 1986-ból.

A dolgozat szerkesztésénél fontos szempont volt a felhasznált komponensek és szoftverek általános bemutatása is, melyek segítségével szolgálhatnak a témában teljesen újak számára is.

Továbbá a 2-es számú mellékletekben megtalálhatók a 3D nyomtatáshoz szükséges modellek STL és Fusion 360 formátumban, mely könnyen reprodukálható és módosítható.

A projekt a jelenlegi eredménnyel biztos alapot jelent a mobil robotika iránt érdeklődők számára, mert már egy olyan szinten áll, mely segítségével az online dokumentációkkal karöltve hozzájuthat az alapokhoz, de emellett teret is hagy a fejlesztéseknek, melyekből csupán néhány lett felsorolva.

10.SUMMARY

The initial goal of the thesis was partially achieved, which was to only allow hobby or easily accessible tools for the project's success. This goal proved feasible for most of the project, except for the wheel axles, where the use of machining tools became inevitable. Excluding the axles, adhering to this goal was entirely achievable, providing inspiration for others in creating their own robots.

In terms of components, accessibility was a primary consideration, with most elements sourced from electronics stores found in most major cities. The same principle applied to the tools used, with the exception of a 3D printer and a laboratory power supply, which are generally available in most engineering institutions. The attached Excel sheet shows that, in terms of cost, this does not fall into the same category as the robot used in the introduction from 1986.

Editing the thesis also emphasized a general introduction to the used components and software, which can be helpful even for those new to the subject. Additional attachments include models for 3D printing in STL and Fusion 360 formats, which are easy to reproduce and modify.

The project, with its current results, provides a solid foundation for those interested in mobile robotics, as it has reached a level where, in conjunction with online documentation, it offers access to the basics while leaving room for further developments, some of which have been listed.

11.KÖSZÖNETNYILVÁNÍTÁS

Köszönöm Szél Károly Tanár Úrnak, hogy ismeretségi körét bevetve támogatta a projekt sikerét tanácsaival, illetve az eszközökkel.

A projekt megvalósításában köszönöm Zólogy Zénó segítségét, hogy a 3D nyomtatott Omni Wheel megvalósításánál a saját projektje mellett időt szánt, hogy segítsen a gumi görgők legyártásában.

A köszönöm a Grundfos Hungary Kft-től Nagy Norbertnek és az ott szakmai gyakorlatot teljesítő középiskolás diákoknak, hogy segítettek az esztergált tengelyeket legyártani.

12.IRODALOMJEGYZÉK

- [1] M. L. Kevin és C. P. Frank, Modern Robotics, Cambridge University Press, 2017.
- [2] I. R. N. I. R. N. Roland Siegwart, Introduction to Autonomous Mobile Robots (Intelligent Robotics and Autonomous Agents) 2nd Edition, London, England: The MIT Press, 2011.
- [3] „TurtleBot 4 robotics learning platform,” Clearpath Robotics, Rockwell Automation, [Online]. Available: <https://clearpathrobotics.com/turtlebot-4/>. [Hozzáférés dátuma: 15 11 2023].
- [4] P. Garrido, „available_ros2_types,” [Online]. Available: https://github.com/micro-ROS/micro_ros_arduino/blob/iron/available_ros2_types. [Hozzáférés dátuma: 27 11 2023].
- [5] N. E. M. Association, „Motion/Position Control Motors, Controls and Feedback Devices”. Rosslyn, Virginia 22209 Szabadalom száma: 100162, 06 10 2004.
- [6] LastMinuteEngineers.com, „In-Depth: Interface DRV8825 Stepper Motor Driver Module with Arduino,” LastMinuteEngineers.com, [Online]. Available: <https://lastminuteengineers.com/drv8825-stepper-motor-driver-arduino-tutorial/>. [Hozzáférés dátuma: 29 11 2023].
- [7] ruthex, „ruthex M2/M3/M4/M5 Gewindeeinsatz Sortimentskasten für den 3D Druck,” ruthex, [Online]. Available: <https://www.ruthex.de/collections/gewindeeinsatze/products/ruthex-gewindeeinsatz-m2-m3-m4-m5-sortimentskasten-70-100-50-50-stuck>. [Hozzáférés dátuma: 15 11 2023].

- [8] A. Pednekar, „Arduino Uno CNC Shield | 3D CAD Model Library | GrabCAD,” Stratasys Inc., [Online]. Available: <https://grabcad.com/library/arduino-uno-cnc-shield-1>. [Hozzáférés dátuma: 29 11 2023].
- [9] H. Shuja, „Raspberry Pi 4 Model B | 3D CAD Model Library | GrabCAD,” Stratasys Inc., [Online]. Available: <https://grabcad.com/library/raspberry-pi-4-model-b-1>. [Hozzáférés dátuma: 29 11 2023].
- [10] R. P. Ltd, „Raspberry Pi Documentation - Raspberry Pi hardware,” Raspberry Pi Ltd, [Online]. Available: <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>. [Hozzáférés dátuma: 29 11 2023].
- [11] O. Robotics, „Installation — ROS 2 Documentation: Humble documentation,” Open Robotics, 2023. [Online]. Available: <https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debian.html>. [Hozzáférés dátuma: 18 11 2023].

13.ÁBRAJEGYZÉK

1-1. ábra A megvalósított mobil robot.....	6
2-1. ábra Omni kerekés mobil robot [1]	10
3-1. ábra Besüthető anyák.....	12
3-2. ábra NANKADF laboratóriumi tápegység.....	13
4-1. ábra DRV8825-ös vezérlő beállítása [6]	19
4-2. ábra DRV8825-ös vezérlő lábkiosztása [6]	19
4-3. ábra Arduino CNC motor shield-be behelyezett DRV8825-ös vezérlők	21
5-1. ábra Omni wheel.....	26
5-2. ábra Ruthex méret táblázat [7]	29
5-3. ábra Robot összeállítás	31
6-1. ábra Teljes robot kapcsolási rajza [8] [9]	32
6-2. ábra Akkumulátort csatlakoztató kábel	33
6-3. ábra Raspberry Pi és CNC shield kapcsolata [8] [10]	34
7-1. ábra Raspberry Pi Imager	36
7-2. ábra Frissítések keresése az Ubuntu terminál segítségével [11]	37
7-3. ábra Frissítések telepítése az Ubuntu terminál segítségével [11].....	38
7-4. ábra ROS előfeltétel vizsgálata [11].....	38
7-5. ábra Ubuntu Universe engedélyezése [11]	39
7-6. ábra ROS szerver hozzáférési kulcs hozzáadása [11]	39
7-7. ábra ROS repository hozzáadása a forrásokhoz [11]	39
7-8. ábra ROS telepítése [11].....	39
7-9. ábra ROS környezeti változók hozzáadása a terminálhoz [11]	39
7-10. ábra ROS munkatér létrehozása	40
7-11. ábra Terminál indítási változók megtekintése.....	40
7-12. ábra Bashrc file módosítása.....	40

14.MELLÉKLETEK

A mellékletek a diplomamunka portálon linkelve megtekinthetők:

1-es számú melléklet: Mobil robot alkatrészek.xlsx

2-es számú melléklet: Robot modellek mappa

3-as számú melléklet: Motor tengely.pdf

4-es számú melléklet: <https://github.com/Kikiman2/HumbleBot>