



INTÉZMÉNYI TUDOMÁNYOS DIÁKKÖRI KONFERENCIA

2022. november 23.

Robotporszívó tervezése

Design of a Robot Vacuum Cleaner

Készítette: **Farkas Benjámín**
QXNNW4
Mechatronikai mérnöki, alapképzés
Nappali
Szent István Campus

Témavezető: Erdélyi Viktor Ferenc, Egyetemi tanársegéd, Szent István Campus, Műszaki Intézet, Mechatronika tanszék

Lehetséges OTDK szekció: „Műszaki Tudományi Szekció”

**Gödöllő
2022.**

Tartalom

1.	Bevezetés.....	4
1.1.	Téma jelentősége	4
2.	Célkitűzés	5
3.	Szakirodalom feldolgozása	6
3.1.	Egykártyás számítógépek.....	6
3.1.1.	Raspberry Pi	6
3.1.2.	Arduino.....	7
3.1.3.	CNC shield	7
3.2.	Akkumulátorok, akkumulátor töltésvezérlők (BMS)	8
3.2.1.	Akkumulátorok.....	8
3.2.2.	Akkumulátor töltésvezérlők (BMS – Battery Management System).....	9
3.3.	Léptetőmotorok.....	9
3.4.	Motorvezérlők.....	10
3.5.	Robotok.....	11
3.5.1.	Autonóm robotok.....	13
3.5.2.	Differenciál-meghajtású robotok.....	14
3.6.	3D nyomtatás	15
4.	Anyag és módszer	17
4.1.	A robotporszívó tervezése.....	17
4.1.1.	Tervezés Solid Edge-ben	17
4.1.2.	A robot elektromos áramkörének megtervezése	23
4.1.3.	A robot alkatrészeinek nyomtatása	24
4.2.	A robotporszívó összeszerelése	26
4.2.1.	Mechanikai összeszerelés	26
4.2.2.	Hardveres összeszerelés	29
4.3.	A robot szoftveres fejlesztése	35
4.3.1.	Arduino Uno programkódok (C nyelvű programozás).....	35

4.3.2.	Python alapú programozás.....	38
5.	Összefoglalás.....	41
6.	Summary	42
7.	Irodalomjegyzék.....	43

1. Bevezetés

„Nem tudom definiálni a robotot, de felismerem, ha látok egyet.” (Joseph Engelberger)

Az utóbbi években rohamosan fejlődött, és napjainkban is folyamatos az innováció az ipar minden egyes területén. A legnagyobb előrelépéseket a robotika területén figyelhetjük meg, amely már évtizedek óta jelen van az iparban, ezzel szemben sokan azt hiszik, hogy a robotika csupán csak pár éve jelent meg.

1.1. Téma jelentősége

Az első ipari robot prototípusát 1961-ben, New Jersey-ben fejlesztették ki. Később, 1974-ben pedig napvilágot látott az IRB 6, amelyet az ABB Björn Weichbrodt által vezetett munkacsoport alkotott meg. Az IRB 6-os modell abban különbözött elődjétől, hogy a vezérlését már egy mikroprocesszor végezte, ezáltal a ma is használatos ipari robotok erre a mintára épültek. [1]

TDK munkám témájából adandóan a robotporszívók történetét is érdemesnek tartom ismertetni. A robotporszívó első ötlete 1959-ben született egy amerikai kiállításon (American National Exhibition) a Szovjetunióban. Az ötletet bár megvalósították, de csak egy prototípus legyártásáig jutottak el, amelyről keveset lehet tudni.

A robotporszívó ötlete 1996-ban tűnt fel újra, amit a híres Electrolux svéd cég meg is valósított. Az elkészült robotporszívó az Electrolux Trilobite nevet kapta, működése hasonlított a mai is használatos robotporszívókéhoz. (Ütközés érzékelő lökhárító, Ultrahangos érzékelő). Magas ára miatt nem aratott akkora sikert, mint ahogy azt az Electrolux várta. Az első verziót 2001-ben indították el a piacon, amelynek ára 1800 dollár volt.

Szintén 2001-ben a brit eredetű Dyson cég is előállt egy saját fejlesztésű robotporszívóval, amely a DC06 nevet kapta, azonban ez a modell soha nem került kiadásra. 2002-ben viszont megtörtént az áttörés, az amerikai eredetű iRobot nevű vállalat bemutatta az első Roombát. A megjelenését követően az első évben megközelítőleg 50.000 darabot sikerült eladniuk. Az újnál újabb modellek, fejlesztések miatt pedig ez a szám egyre csak növekszik. [2]

2. Célkitűzés

TDK munkám célja egy olyan robotporszívó megalkotása a 3D nyomtatás technológiájával, valamint azzal a tudással, amelyet az egyetemen töltött éveim alatt szereztem, amely mind műszakilag, mind pedig gazdaságilag megfelel olyan területek takarítására, amelyekre egy átlagos, háztartásban alkalmazható robotporszívó nem képes. Az alkalmatlanság adódhat a méretezésből, ugyanakkor adódhat az akkumulátor kapacitásából. Én ezekre a problémákra kerestem megoldást munkám készítése során.

3. Szakirodalom feldolgozása

Ebben a fejezetben tárgyalásra kerülnek azon technológiák, amelyeket a robotporszívó készítése közben alkalmaztam, többnyire csak általánosságban írok róluk.

3.1. Egykártyás számítógépek

Az egykártyás számítógép (SBC – Single-Board Computer) egy olyan számítógép, amelyben egyetlen vékony merev lap található, amelyen elektromos áramkör helyezkedik el, más néven áramköri kártya. Ez az áramköri kártya tartalmaz memóriát, bemeneti és kimeneti opciókat, valamint mikroprocesszort és minden más működéshez szükséges komponenst. Az egylapos számítógépeket a szabványos asztali számítógépektől eltérően tervezték, mivel teljesen önállóak. Az egykártyás számítógépek tulajdonságaiból adódóan a hagyományos asztali számítógépekkel szemben előnyt élveznek, ezek az előnyök a méretükből, a kevesebb hőtermelésükből, valamint energiahatékonyságukból adódik, valamint megbízhatóbbak, mivel kevesebb alkatrészük van, amely meghibásodhat. Ipari alkalmazásokban az egylapos számítógépeket általában beágyazott rendszerekben használják, amelyeket úgy terveztek, hogy egy nagyobb rendszeren belül egy adott funkciót hajtsanak végre. Például egyetlen fedélzeti számítógép használható egy ipari vezérlőben, amely felügyeli és vezérli a gyártósort. Emellett folyamatvezérlési alkalmazásokban is használatosak, például összetett robotrendszerekben és processzorigényes alkalmazásokban. [3]

3.1.1. *Raspberry Pi*

A Raspberry Pi a Raspberry Pi Foundation, az Egyesült Királyság jótékonyági szervezete által készített SBC neve, amelynek célja, hogy az embereket elősegítse a számítástechnika tanulásában, és egyaránt megkönnyítse a számítástechnika oktatását. Az emberek a Raspberry Pi-t általában programozási készségük fejlesztésére, projektek megvalósítására, valamint egyszerű otthoni automatizálásra használják, nem utolsósorban pedig az iparban is fellelhetők. A Raspberry Pi egy viszonylag olcsóbbnak mondható számítógép, amely Linux operációs rendszert használ, de GPIO (általános célú bemeneti/kimeneti) pinek is találhatóak rajta, amelyek lehetővé teszik az elektronikus komponensek vezérlését és az Internet of Things (IoT) felfedezését. [4]

Számos ipari területen is felhasználják a Raspberry Pi számítógépeket, ezek közül néhány [5]:

- Távközlés
- Ipari automatizálás
- Kerékpár gyártás
- Vendéglátás

3.1.2. *Arduino*

Az Arduino egy nyílt forráskódú elektronikai platform, amely mind hardveresen mind pedig szoftveresen könnyen használható. Az Arduino képes beolvasni a bemeneti jeleket: egy érzékelő jeleit, egy gomb lenyomását. Ez után pedig a jeleket képes átalakítani kimenetűvé: például motort tud aktiválni, LED-et tud bekapcsolni. Az Arduino-t vezérelni is tudjuk, ha például elküldünk neki egy utasítást. Ehhez az Arduino programozási nyelvet, valamint az Arduino szoftvert (IDE) kell használni, amely C programozási nyelvet alkalmaz. Az elmúlt évek során az Arduino több ezernyi projekt alapjaként szolgált, a mindennapi alkalmazásban lévő eszközöktől a bonyolult tudományos műszerekig. [6] [7]

Az Arduino-t is megtalálhatjuk az ipar különböző területein, ezek például [8]:

- Egészségügy
- Űrtechnika
- Autóipar

3.1.3. *CNC shield*

A CNC shield egy nyílt forrású hardver a léptetőmotorok vezérlésére. Lehetővé teszi négy motor egyidejű vezérlését. Kivehető motorvezérlőket használ, amelyekről a 2.3-as alfejezetben olvashatunk. A léptetőmotorok csatlakoztatása négy tűs csatlakozókon keresztül történik. Tápfeszültsége 12-36 V-ig terjed, általában Arduino Uno R3-as egykártyás számítógéppel csatlakoztatják össze, és a két eszköz szimbiózisát hasznosítják. [9]

3.2. Akkumulátorok, akkumulátor töltésvezérlők (BMS)

Ebben az alfejezetben bemutatásra kerülnek az akkumulátorok, és a töltésvezérlő rendszerek.

3.2.1. Akkumulátorok

Az akkumulátor egy energiaforrás, amely egy vagy több elektrokémiai cellából és mindkét végén anódból (-) és katódból (+) álló kivezetésekből áll. Az elektrokémiai cellák a kémiai energiát elektromos energiává alakítják. Az akkumulátor belsejében egy elektrolit található, amely gyakran oldható sókból vagy savakból áll, vezető közegként szolgál, lehetővé téve az elektromos töltés áthaladását az akkumulátoron. Feszültségétől és terhelésétől függően egyetlen akkumulátor bármit képes táplálni, például az autó motorját, vagy a számítógépet, akár a mobiltelefont is. [10]

A leggyakrabban előforduló akkumulátor fajták [11]:

- Zselés akkumulátor
- Nikkel-kadmium
- Nikkel-metál-hidrid
- Lítium-ion
- Lítium-polimer

A házi hobbi projektek terén a legelterjedtebb akkumulátor fajta az ún. 18650-es akkumulátor, amely egy újratölthető lítium-ion akkumulátor. Általában 3,6 V névleges feszültséggel rendelkezik, kapacitása pedig 1800 mAh (milliamperóra) és 3600 mAh között terjed. A mAh az elektromos teljesítmény mérésére szolgáló mértékegység. Elnevezésében a 18-as az akkumulátor 18 mm-es szélességére vonatkozik, a 65 pedig a 65 mm-es hosszúságára. Alkalmazása a zseblámpáktól és az elektromos kéziszerszámoktól az elektromos cigarettáig, kameráig és laptopokig bármiben használják.

3.2.2. Akkumulátor töltésvezérlők (BMS – Battery Management System)

A BMS egy olyan elektronika, amely felügyeli és kezeli az akkumulátor teljes teljesítményét, valamint a legfontosabb, hogy megakadályozza az akkumulátor működését a biztonsági határokon kívül. A BMS kritikus fontosságú az akkumulátor biztonságos működése, általános teljesítménye és hosszú élettartama szempontjából. Sőt, védi azt az tárgyat is, amibe a lítium akkumulátor be van szerelve és az azt használó személyeket is egyaránt. A BMS elsődleges funkciója, hogy megvédje az akkumulátorcellákat a túltöltés vagy túlmelegedés okozta sérülésektől. Mindezek mellett kiegyensúlyozzák a töltést a cellák között, hogy minden cella maximális kapacitáson működjön. Ezenkívül léteznek olyan fajta BMS-ek, amelyek kiszámítják a fennmaradó töltést, figyelik az akkumulátor hőmérsékletét, az akkumulátor épségét. [12]

3.3. Léptetőmotorok

A léptetőmotorok egyenáramú motorok, amelyek ún. lépésekkel mozognak. Több tekercsük van, amelyeket „fázisoknak” elnevezett csoportokba vannak rendezve. A motort a fázisok egymás utáni áram alá helyezésével lehet lépésenkénti mozgásra bírni. Abban az esetben, ha nagyon precíz pozicionálást és- vagy sebességszabályozást szeretnénk, akkor a motort számítógéppel kell vezérelnünk. Ennek okán a legtöbb precizitást igénylő mozgásvezérlési feladathoz általában léptetőmotorokat alkalmaznak. [13]

A léptetőmotorok sok különböző típusra bonthatók, ebből két lényegesebb típust emelnék ki:

- Bipoláris léptetőmotorok
- Hibrid léptetőmotorok

A bipoláris léptetőmotorok olyan váltakozó áramú (AC) léptetőmotorok, amelyek feszültség alá helyezett elektromágneses tekercseket használnak arra, hogy az adott tengelyt forgatni tudják. Ezzel a struktúrával a tengely precíz mozdulatokkal forgatható, negatív és pozitív irányban egyaránt. Az unipoláris léptetőmotorokkal ellentétben a bipoláris motor lehetővé teszi az elektromos áram áramlását mindkét irányban, így csökkentve a feladatainak elvégzéséhez szükséges tekercsek számát. A bipoláris léptetőmotorok működése viszonylag egyszerű. A léptetőmotor-meghajtó elektromos áramimpulzust küld a motor tekercsén keresztül. A tekercs feszültség alá kerül, majd elektromágnessé válik. Ez azt eredményezi, hogy a motor forgórészét a tekercs felé húzza. A meghajtó ezután feszültség alá helyezi a sorban következő tekercset, és így a folyamat egymás után ismétlődik. [14]

A hibrid léptetőmotorok az állandó mágneses és a változó reluktanciájú léptetőmotorok metódusait egyaránt kombinálják. Az állandó mágneses motorokhoz hasonlóan a rotor fogáiban állandó mágnes található. Két fogsor, úgynevezett „csésze” gyűrűzik körbe a rotoron. Az egyik gyűrű az összes déli pólust foglalja magába, a másik gyűrű az összes északi pólust. A változó reluktanciájú motorokhoz hasonlóan a hibrid léptetőmotorok is tartalmaznak állórészpólust. [15]

A léptetőmotorok széles körben alkalmazhatók számos iparágban és tudományágban, amelyek közül párat meg is említenék [16]:

- Robotika
- Nyomtatás és szkennelés, beleértve a 3D nyomtatókat is
- Folyamatautomatizálás és csomagológépek
- Precíziós pozicionáló berendezések

3.4. Motorvezérlők

A motorvezérlő egy integrált áramkörrel rendelkező chip, amelyet autonóm robotokban és beágyazott áramkörökben motorvezérlő eszközként használnak. A motorvezérlők interfészként működnek a motorok és a vezérlésért felelős áramkörök között. A motor nagy áramot igényel, míg a vezérlő áramkör alacsony áramerősséggel rendelkező jelekkel működik. A motorvezérlők feladata az, hogy a kapott gyengeáramú vezérlő jelet egy nagyobb áramú jellé alakítsa, amely képes lesz meghajtani a motort. Leggyakrabban kisebb méretű robotokban, 3D nyomtatókban használják őket, de előfordulhatnak akár elektromos járművekben is. [17] [18]

A legelterjedtebb motorvezérlők:

- L293D
- ULN2003
- A4988
- DRV8825
- TMC2208

3.5. Robotok

A robotika a tudomány, a mérnöki munka és a technológia metszéspontja, amely olyan gépeket, úgynevezett robotokat állít elő, amelyek megismétlik vagy helyettesítik az emberi tevékenységeket.

A technológia fejlődésével a robotikának tekintett hatókör is bővül. A 2000-es évek elején majdnem az összes robotot az autóiparban használták összeszerelési feladatok elvégzésére. Ezek a robotok főként mechanikus karokból állnak, amelyek feladata az autó bizonyos részeinek hegesztése vagy csavarozása.

Ma a robotika egy továbbfejlesztett és kibővített változatát látjuk, amely magában foglalja a robotok fejlesztését, létrehozását és használatát, és olyan feladatokat hajtanak végre, mint a bolygó legzordabb körülményeinek feltárása, a bűnüldözés segítése, a sebészeti eljárások egyszerűsítése és a mentőakciók lebonyolítása.

Miközben a robotika általános világa bővül, a robotnak van néhány következetes jellemzője:

- A robotok valamilyen mechanikus szerkezetből állnak. A robot mechanikai aspektusa segít abban, hogy a feladatokat abban a környezetben végezze el, amelyre tervezték.
- A robotoknak elektromos alkatrészekre van szükségük, amelyek irányítják és táplálják őket. Lényegében elektromos áramra, például akkumulátorra van szükség a robotok többségének működtetéséhez.
- A robotoknak tartalmazniuk kell valamilyen szintű programozást is. Anélkül, hogy az adott kód, amit betápláltak a robot „agyába” megmondaná, hogy mit kell tennie, egy robot csak egy újabb egyszerű gépezet lenne. A program implementálása lehetővé teszi, hogy a robot tudja, mikor és hogyan kell az adott feladatot végrehajtani. [19]

A robotika iparág tele van a fejlődés csodálatra méltó ígéreteivel. Óceánjaink legmélyebb mélységeitől a világűr több ezer mérföldjéig a robotok olyan feladatokat hajtanak végre, amelyekről az emberek álmodni sem mertek volna. [19]

A robotok főbb típusai:

- Előre programozott robotok:
 - Az előre programozott robotok ellenőrzött környezetben működnek, ahol egyszerű, monoton feladatokat látnak el. Példa az előre programozott robotra egy mechanikus kar egy autóipari összeszerelő soron. A kar egy funkciót lát el és az a feladata, hogy ezt a feladatot hosszabb ideig, gyorsabban és hatékonyabban végezze, mint egy ember.
- Humanoid robotok
 - A humanoid robotok olyan robotok, amelyek emberi viselkedésre hasonlítanak vagy utánozzák azt. Ezek a robotok általában emberszerű tevékenységeket végeznek, például futnak, ugrálnak és tárgyakat cipelnek, és néha úgy tervezték őket, hogy úgy nézzenek ki, mint mi, még emberi arcokkal és kifejezésekkel is felruháztak belőlük pár darabot.
- Teleoperált, más néven távműködtetett robotok
 - A távműködtetett robotok félig autonóm robotok, amelyek vezetékek nélküli hálózatot használnak annak érdekében, hogy az emberi irányítás biztonságos távolságból történjen. Ezek a robotok általában szélsőséges időjárási és földrajzi körülmények között működnek. A távműködtetett robotok példái közé tartoznak a drónok, amelyeket a taposóaknak észlelésére használnak. [19]

A robotok főbb alkotóelemei:

- Vezérlőrendszer: a processzor, amely a robot feladatának ellátásáért felel.
- Érzékelők: olyan alkatrész, amely elektromos jeleket ad, hogy lehetővé tegye a robot számára, hogy kölcsönhatásba lépjen a környezetével.
- Aktuátorok: olyan eszközöket takar, amelyek gépet, vagy más eszközöket működtetnek. Jelen esetben ez azokat a motorrészeket jelenti, amelyek a robot mozgásáért felelősek.
- Tápegység: az akkumulátor, amely árammal látja el a robotot. [19]

A robotok főbb felhasználási területei [20]:

- Gyártás
- Logisztika
- Otthoni használat
- Közlekedés
- Egészségügy

3.5.1. Autonóm robotok

A robotok főbb típusaihoz hozzátartoznak még az autonóm robotok, amelyeket külön szeretnék megemlíteni, mivel a munkám során elkészített robot is ebbe a kategóriába tartozik.

Az autonóm robotok olyan intelligens gépek, amelyek önállóan, emberi irányítás és beavatkozás nélkül képesek feladatokat ellátni. Az autonóm robotok gyakran rendelkeznek olyan kifinomult funkciókkal, amelyek segíthetik megérteni fizikai környezetüket, és automatizálni a karbantartásuk és az irányításuk némely részeit, amelyek korábban emberi beavatkozást igényeltek. [21]

Az autonóm robotok főbb alkalmazási területei [22]:

- Logisztika
- Raktározás
- Egészségügy
- Biotechnológia

Az autonóm cselekvés három kulcsfogalma közé tartozik a(z):

- Észlelés:
 - Az embereknél az érzékelést négy érzékszervünk végzi. Szem, fül, bőr, orr. Egy robot számára az érzékelés szenzorokat jelent. Lézerszkennereket, kamerákat, ultrahangos szenzorokat (szem), ütközésérzékelőket (bőr), erőnyomaték érzékelőket (izomzat), sőt spektrométereket (szaglás) használnak bemeneti eszközként a robot számára, hogy segítsék őket tájékozódni a környezetükben.
- Döntés:
 - Az embereknél az agyunk hozza meg a legtöbb döntést, vagy bizonyos esetekben az idegrendszerünk. Az autonóm robotok hasonló döntéshozatali

struktúrával rendelkeznek. A robot „agya” általában egy számítógép, és az alapján hoz döntéseket, hogy mi a feladata, és milyen információkat kap működése során. A robotok olyan képességgel is rendelkeznek, mint az ember neurológiai rendszere, ahol a biztonsági rendszereik gyorsabban és az agy engedélye nélkül működnek, valójában a robotoknál az agy a biztonsági rendszer engedélyével működik. Egy autonóm robotban ezt a „neurológiai” rendszert beágyazott rendszernek nevezzük, gyorsabban és magasabb szintű jogosultsággal működik, mint a feladatot végrehajtó és az adatokat elemző számítógép.

- Cselekvés:
 - Az embereknek izmoknak nevezett működtetői vannak. A robotoknak is mindenféle működtető szerkezetük lehet, ez általában valamilyen motor formájában valósul meg. Legyen szó kerékről, lineáris működtetőről vagy hidraulikus munkahengerről, mindig van egy motor, amely az energiát mozgássá alakítja. [22]

Az autonóm mobil robotok teljes megértéséhez segít, ha működés közben láthatjuk őket. A mai piacon az egyik legismertebb és vitathatatlanul legkelendőbb, valóban autonóm robot a Roomba. Míg a Roomba fogyasztói termék, képességei és funkcionalitása a raktári és ipari területekre is kiterjed, és az Autonóm Mobil Robotokat (AMR) sokkal szélesebb körben hozzáférhető technológiává teszi. A Roomba a környezetében észlelt dolgok alapján tud dönteni és cselekedni. Egy sor érzékelő teszi lehetővé a Roomba számára, hogy érzékelje a környezetét, döntsön az észlelések alapján, majd elvégezze a megfelelő műveletet. Hasonló koncepció alapján működnek a raktári robotok is, ha egy AMR akadályba ütközik, miközben meghatározott feladatokat hajt végre a raktárban, akkor probléma nélkül navigál, és emberi beavatkozás nélkül folytatja a munkáját. [22] [21]

3.5.2. Differenciál-meghajtású robotok

A differenciál-meghajtású robot egy olyan robot, amelynek mozgása a testének két oldalán elhelyezkedő, két külön motor által hajtott keréken alapul. Így a kerekei forgási sebességének változtatásával megváltoztathatja képes megváltoztatni mozgásának irányát. Ez azt jelenti, hogy egy kétdimenziós térben a kívánt pozíció több manővert és potenciálisan bonyolultabb útvonaltervezést igényel. Ezen a működési elven alapul például a már előbb említett Roomba robotporszívó. [23]

3.6. 3D nyomtatás

Habár a 3D nyomtatás egy vadonatúj technológiának tűnhet, ha hisszük, ha nem, nem az. A legkorábbi 3D nyomtatási technológiák az 1980-as években jelentek meg. Ezeket Rapid Prototyping (RP) technológiáknak nevezték el, mert először gyorsabb és költséghatékonyabb eszköznek szánták prototípusok létrehozására az iparágon belüli termékfejlesztéshez.

Az RP technológia első szabadalmi kérelmét 1980-ban nyújtotta be Dr. Hideo Kodama, a japán Nagoya Városi Ipari Kutatóintézet munkatársa. Egy fotopolimereket használó gyors prototípus-készítő rendszert írt le. Egy tömör, nyomtatott modell rétegekben épült fel, minden réteg a modell egy keresztmetszeti szeletének felelt meg. Dr. Kodama sajnálatos módon elmulasztotta szabadalmi kérelmét, és soha nem hozta kereskedelmi forgalomba a nyomtatási eljárását. [24] [25]

Később, 1984-ben Charles Hull beírta magát a 3D nyomtatás történelmébe, amikor feltalálta a sztereolitográfiát. A sztereolitográfia egy háromdimenziós objektumok létrehozására szolgáló technika vagy eljárás, amelynek során egy számítógép által vezérelt mozgó lézersugárral rétegről rétegre felépítik a kívánt szerkezetet folyékony polimerből, amely lézerfényvel érintkezve megkeményedik. Ez az eljárás lehetővé tette a tervezőmérnökök számára, hogy digitális adatokat felhasználva 3D-s modelleket alkossanak, amelyeket utána fizikai tárgyként is megvalósítanak. A sztereolitográfia „titkos összetevője” egy akril alapú anyag, amely nem más, mint a fotopolimer. Az anyag egy tartályban van, folyékony halmazállapotban, amelyet UV lézersugárnak tesznek ki, majd a fénynek kitett rész azonnal szilárd műanyagdarabbá válik, ezt pedig a 3D-s modell formájára formálják. Ez nagyon jó hír volt a tudósok és feltalálók számára, mert nem kellett hatalmas költségekbe kezdeniük, ha a projektjeiket akarták tesztelni, hiszen ezzel a technológiával meg tudták azokat valósítani. [24] [25]

1986-ban Charles Hull megalkotta az első sztereolitográfiai készüléket, az SLA-t. Ez a 3D nyomtatási technika lehetővé tette összetett részek rétegenkénti nyomtatását a megszokottnál rövidebb idő alatt. Charles Hull ezután megalapította a 3D Systems Corporationt, amely bemutatta a világ első kereskedelmi 3D nyomtatási rendszerét, az SLA-1-et. A későbbi években több tudós és mérnök fejlesztette a 3D nyomtatás technológiáját. Az egyik legnagyobb áttörést a svéd Cellink vállalat érte el 2015-ben, az első kereskedelmi forgalomba hozott biotintával. Nanocellulóz alginátból, és tengeri hínárból származó anyagból készült, amely alkalmasnak bizonyult szöveti porcok nyomtatására. [24] [25]

Napjainkban már könnyebben hozzá lehet férni a 3D nyomtatás technológiájához, hiszen egy kezdő szintű nyomtatót 100.000 forint alatt is be tudunk szerezni. Az effajta nyomtatók működése elég egyszerű: a kiválasztott speciális filamentet, amely az ilyen típusú nyomtatóknál egy hőre lágyuló alapanyagot takar befűzzük a nyomtató adagolójába, amelynek szabályozásáért egy szoftveresen vezérelt léptetőmotor felel. Aztán az anyagot egy felhevített fűvóka megolvasztja, és az olvasztás közben a nyomtató már dolgozni is kezd, természetesen csak ha megadtuk neki, hogy mit szeretnénk, hogy nyomtasson. Ilyen nyomtatók például a Creality által forgalmazott Ender nyomtatók.

A 3D nyomtatás technológiájának főbb alkalmazási területei [26]:

- Gépészet
- Repülés és űrtechnológia
- Egészségügy
- Autóipar

Felvetődhet bennünk az a kérdés is, amely sok emberben megfogalmazódik, ha szóba kerül a 3D nyomtatás: Gazdaságilag kifizetődő-e? A válasz az esetek többségében az lesz, hogy igen, mivel a 3D nyomtatás technológiáját alkalmazva sok olyan alkatrészt, vagy akár berendezéseket, valamint eszközöket is készíthetünk, amelyek már nem elérhetőek a piacon, vagy éppen nem tudjuk azokat megfizetni. Számos nagyobb presztízzsel rendelkező cég alkalmaz 3D nyomtatást a termékei megvalósításához, gyártásához. Például az autógyártásban a BMW, az orvoslási szférában az amerikai Allevi, az űrtechnikában pedig a NASA is alkalmaz 3D nyomtatott elemeket [27] [26].

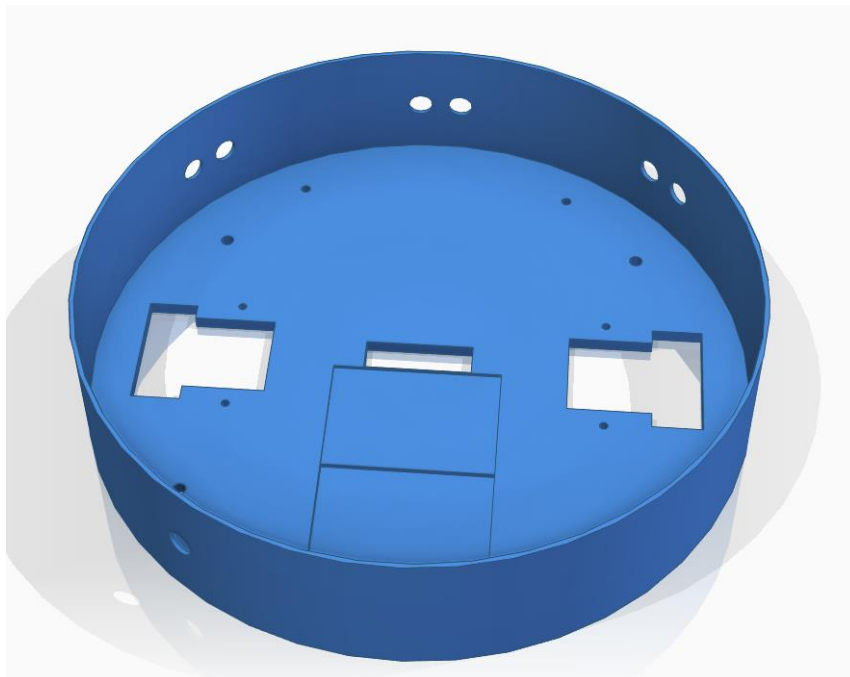
4. Anyag és módszer

Ebben a fejezetben részletezni fogom a TDK munkám elkészítésének módszereit, a felhasznált alkatrészeket, és technológiákat, valamint a programkódot folyamatábra formájában.

4.1. A robotporszívó tervezése

4.1.1. Tervezés Solid Edge-ben

A robot tervezésének kezdete előtt tanulmányoztam a piacon fellelhető robotporszívókat, mind méreti tulajdonságaik, mind pedig műszaki tartalmuk szerint. A tervezési fázis első lépéseként összegyűjtöttem, hogy milyen alkatrészekre lesz szükségem a robot megépítése során. Miután összegyűjtöttem a megfelelő komponenseket, vizsgálni kezdtem azokat a technológiákat, amelyekkel a robot vázát, valamint különböző alkatrészeit le lehet gyártani. A megvalósításhoz a legopcionálisabb eljárás a 3D nyomtatás volt, mivel az egyetemen elsajátíthattam azt a tudást, amely lehetővé tette a CAD (Számítógéppel segített tervezés) programok használatát, ezáltal a robotporszívót meg tudtam tervezni, és képes voltam a legyártására is.

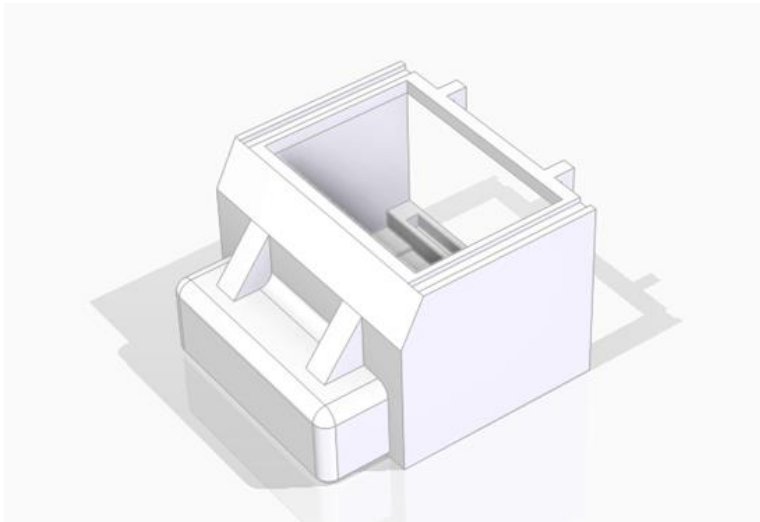


1. Ábra: A robot váza a szükséges kivágásokkal
Forrás: saját kép

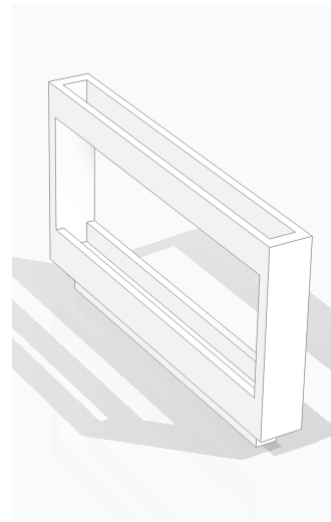
A robotporszívó tervezése a Solid Edge programmal készült, amely lehetővé tette, hogy 3 dimenziós környezetben tervezzem meg a robotot. A tanulmányozott robotporszívókból kiindulva, egy 400 mm átmérőjű, 8 mm vastagságú kört terveztem meg, amit utána úgy vágtam

ki, hogy körben egy 5 mm vastagságú falat képezzek ki. Ez az alkatrész a robot alapjaként, valamint az elektronikai és más alkatrészeknek ad otthont.

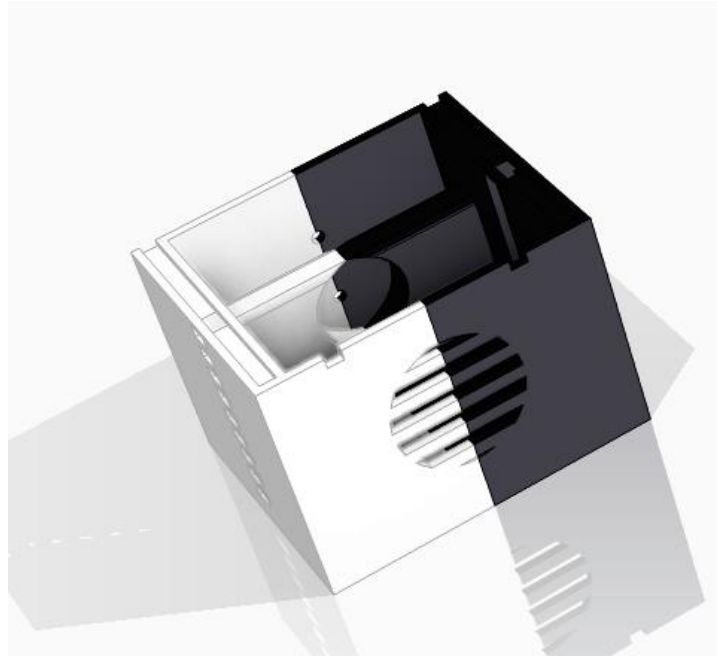
Következő lépésként beszereztem egy morzsaporszívót, amelyből aztán kisereltem a motort, amely a vákuum előállításáért felelt. Ez után megmértem a már kiserelt motor befoglaló méreteit tolómérő segítségével, majd méretei alapján megterveztem a házat, amelyben a későbbiekben fog operálni. Mivel a porszívóknak szükségük van egy porgyűjtő tartályra is, így megvizsgáltam a rendelkezésre álló helyet a robot vázában, majd mérések alapján megterveztem a tartályt is. A motor biztonsága érdekében kigondoltam egy olyan megoldást, amit már sok más porszívóban láttam, ami jelen esetben egy szivacsot (filter) jelent. A filternek a porgyűjtő tartályon belül alakítottam ki helyet, ahol egy keretbe fog illeszkedni.



2. Ábra: A porgyűjtő tartály 3 dimenziós ábrázolása.
Forrás: saját kép



3. Ábra: A filter 3 dimenziós ábrázolása.
Forrás: saját kép

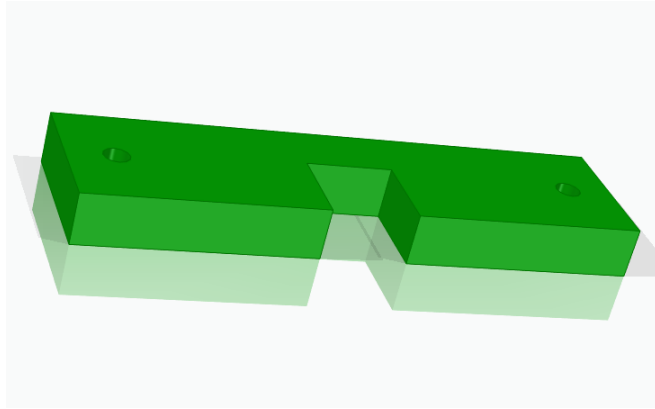


*4. Ábra: A motorház 3 dimenziós ábrázolása.
Forrás: saját kép*

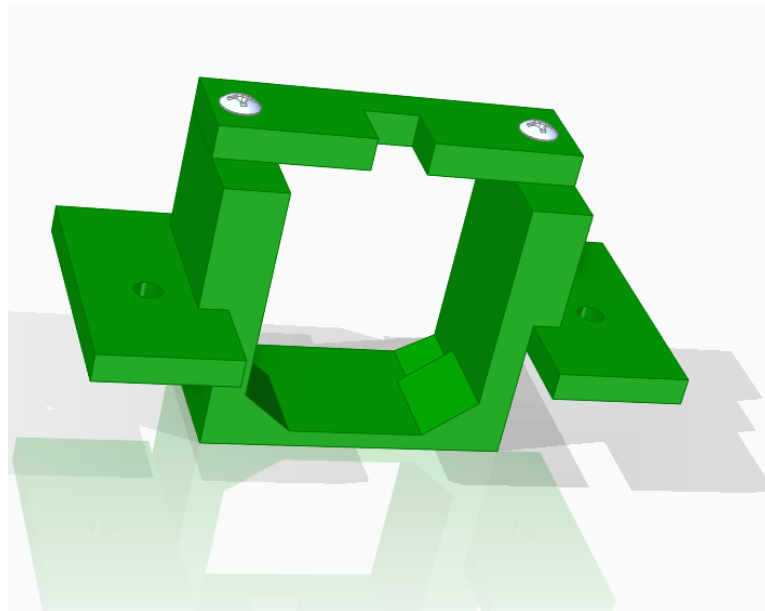
A motorház és a tartály megtervezése után tanulmányoztam, hogy mely motorok lennének a leghatékonyabbak a robot mozgatására. Pár napos kutatás után a léptetőmotorokra esett a választásom, mivel könnyebben vezérelhetők, mint a DC motorok, energiahatékonyabbak, illetve sokkal precízebben lehet velük dolgozni. Mivel a robot belsejében nem tudtam olyan helyet kialakítani a motorok számára, ahová bármilyen ház nélkül beférnének, így tartóelemeket terveztem a léptetőmotorok részére. Ezt követően tolómérőt használva megmértem a léptetőmotorok befoglaló méreteit, és azok alapján megterveztem a motorokat tartó alkatrészeket. A tartóelemeket úgy terveztem meg, hogy a motorok felfelé ható erőhatás - a léptetőmotorokon elhelyezkedő kerek - ellenére se tudjanak szabadon elmozdulni. A motortartók egy házból - amibe a motor beilleszkedik -, valamint egy lefogató elemből - amely meggátolja a motort az esetleges kimozdulásokban - állnak.



5. Ábra: A motortartó 3 dimenziós ábrázolása.
Forrás: saját kép



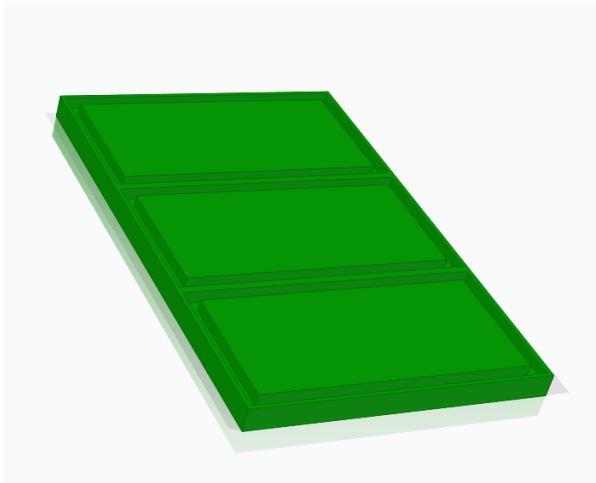
6. Ábra: A motor lefogató elem 3 dimenziós ábrázolása.
Forrás: saját kép



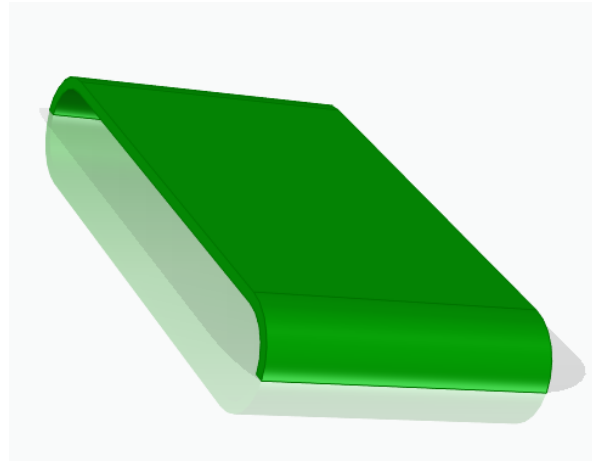
7. Ábra: A motortartó és a motort lefogató elem összeállításának 3 dimenziós ábrázolása.
Forrás: saját kép

A motortartó elempár megtervezése után az akkumulátorokat kezdtem el tanulmányozni, olyan szempontok szerint, mint például élettartam, kapacitás, töltési idő, és méret. A választás viszonylag egyszerű volt, mivel a 18650-es Li-ion akkumulátorok mindegyik kritériumnak megfeleltek. A kiválasztás után elkezdtem megtervezni az akkumulátorok házát, amely hat darab cellának kell otthont adjon. Ehhez beszereztem olyan elemtartókat, számszerűsítve három darabot, amelybe a 18650-es egységek könnyen illeszkednek, így csak egy talpat kellett terveznem az elemtartók számára, ehhez szükségem volt a konzolok méreteire. Az elemtokok

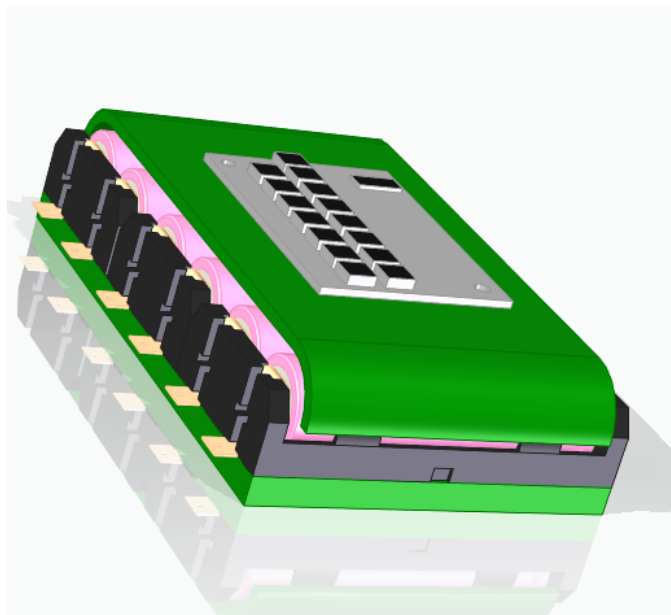
lemérése után megterveztem a talpként szolgáló elemet, amelybe az elemtokok alján található bepattanó elem illeszkedik. Ez után pedig terveztem egy íves takaróelemet, amely az akkumulátor telep tetejeként szolgál, valamint erre a későbbiekben erre lesz elhelyezve a töltésvezérlő egység is.



8. Ábra: A talpként szolgáló elem 3 dimenziós ábrázolása.
Forrás: saját kép



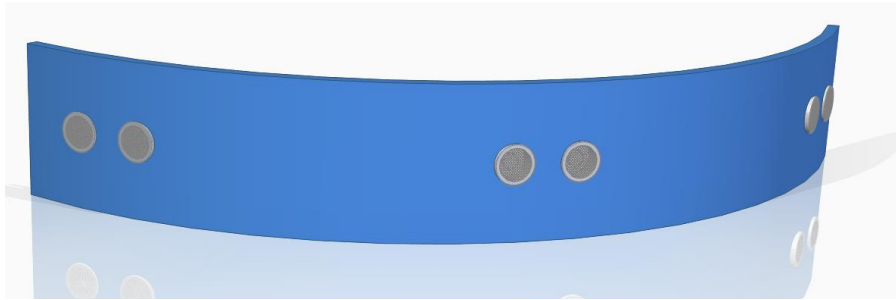
9. Ábra: Az akkumulátor telep tetejének 3 dimenziós ábrázolása.
Forrás: saját kép



10. Ábra: Az akkumulátor telep kész összeállítása 3 dimenziós ábrázolásban.
Forrás: saját kép

Következő lépésként megvizsgáltam, hogy milyen eszközök lennének alkalmasak arra a célra, hogy koordinálják a robotot. Számos eszköz közül választhattam, mint például lézeres távolságmérő szenzor, LiDAR, vagy a legegyszerűbb távolságmérésre alkalmas szenzor: az ultrahang szenzor. Mivel nem akartam túlbonyolítani a robotporszívó koordinációért felelős

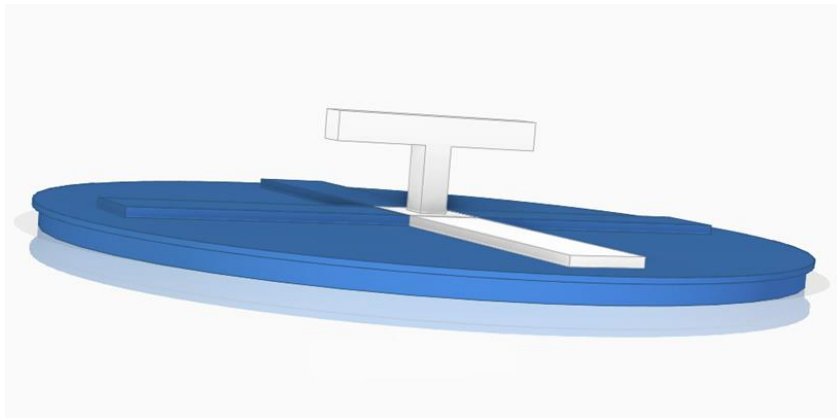
egységét, valamint törekedtem arra, hogy minden eszközöm kompatibilis legyen egymással, és viszonylag ne legyen drága az adott berendezés, így az ultrahangos távolságmérő szenzorok mellett döntöttem. Ezek után megterveztem, hogy hol fognak elhelyezkedni a szenzorok, és készítettem nekik egy hajlított elemet, amely hasonlít egy robotporszívó lökhárítójára. Az elülső szenzor középen helyezkedik el, azaz 0° -ot zár, míg a két oldalsó szenzor egyenlően 45° -os eltolással van beültetve a lökhárítóba.



11. Ábra: A lökhárító elem 3 dimenziós ábrázolása szenzorokkal.

Forrás: saját kép

A robotporszívó utolsó eleme, amelyet terveztem a váz teteje volt. A tetőn található egy fogantyú, ami megkönnyíti a tető levételét, valamint négy összefogató elem, amelyek fontosságáról a későbbiekben fogok szót ejteni.



12. Ábra: A robotporszívó vázának teteje 3 dimenziós ábrázolásban.

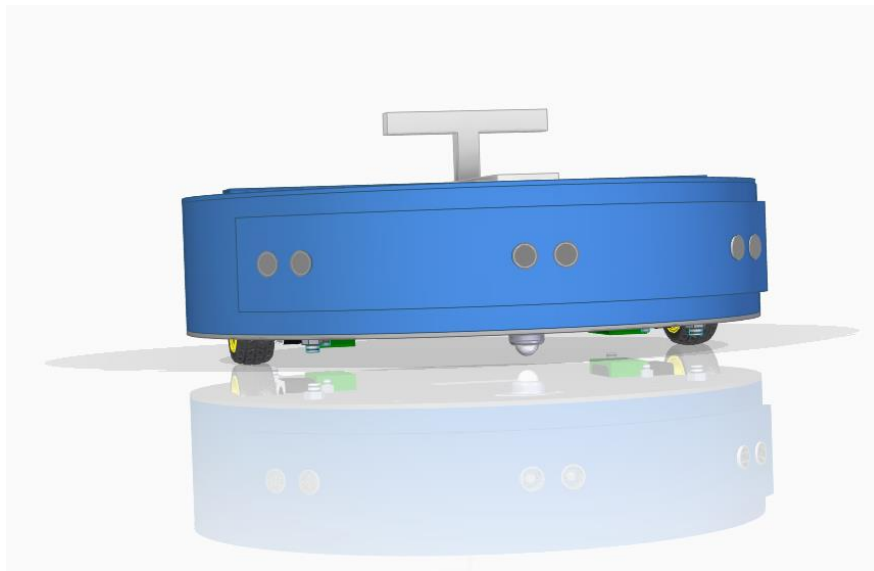
Forrás: saját kép

Utolsó lépésként a megtervezett elemeket, valamint az elektronikus alkatrészeket 3D-s környezetben elhelyeztem a kialakított helyükön a robot vázában, valamint a robot vázán egy összeállítási fájl keretein belül. A végeredményt az alábbi ábrák szemléltetik:



13. Ábra: A robotporszívó összeállított ábrázolása (tető elem nélkül) 3 dimenziós ábrázolásban.

Forrás: saját kép

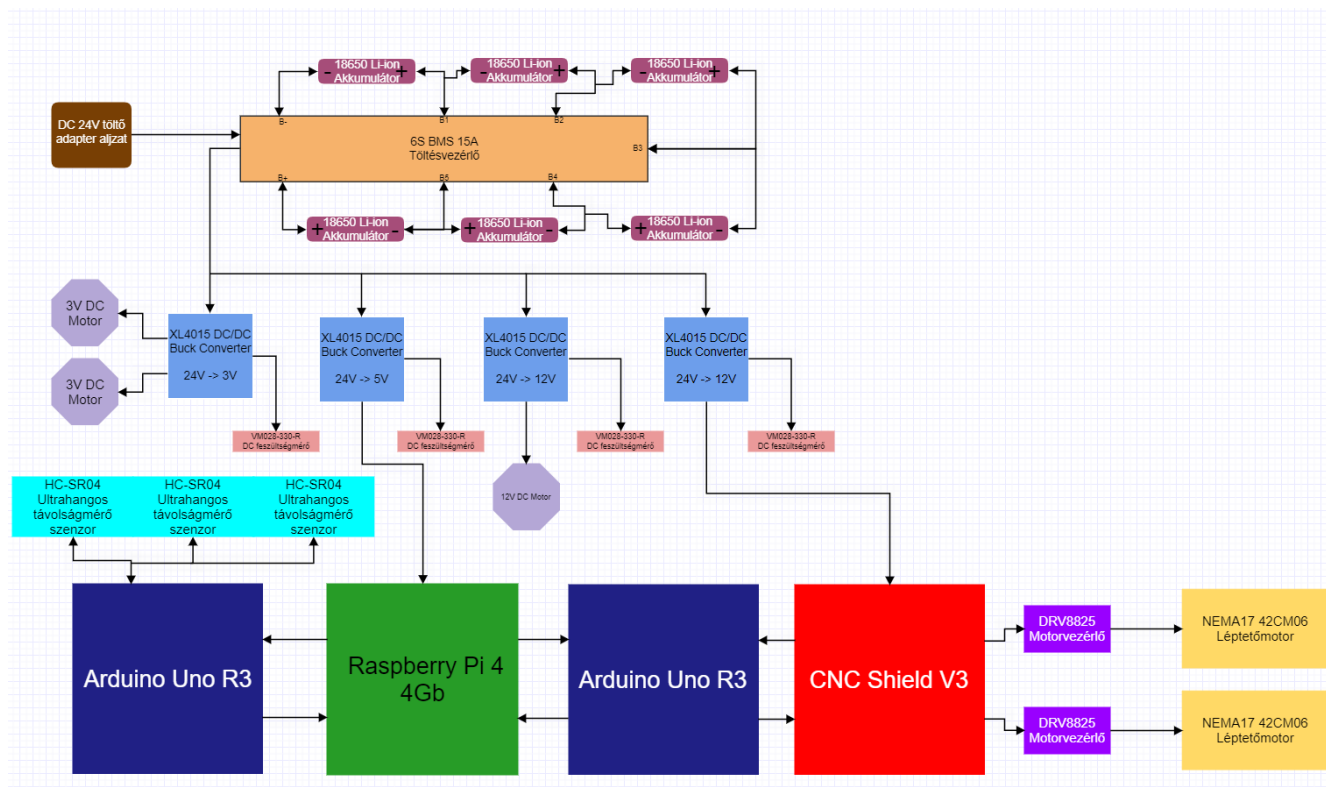


14. Ábra: A robotporszívó összeállított ábrázolása 3 dimenziós ábrázolásban.

Forrás: saját kép

4.1.2. A robot elektromos áramkörének megtervezése

A 3 dimenziós tervezés után az elektronikus alkatrészek kapcsolási rajzát terveztem meg. A rajz készítéséhez a Gliffy nevű weboldalt használtam, ami lehetővé teszi, az egyszerű, blokkdiagrammos ábrázolását az elektromos kapcsolásnak.

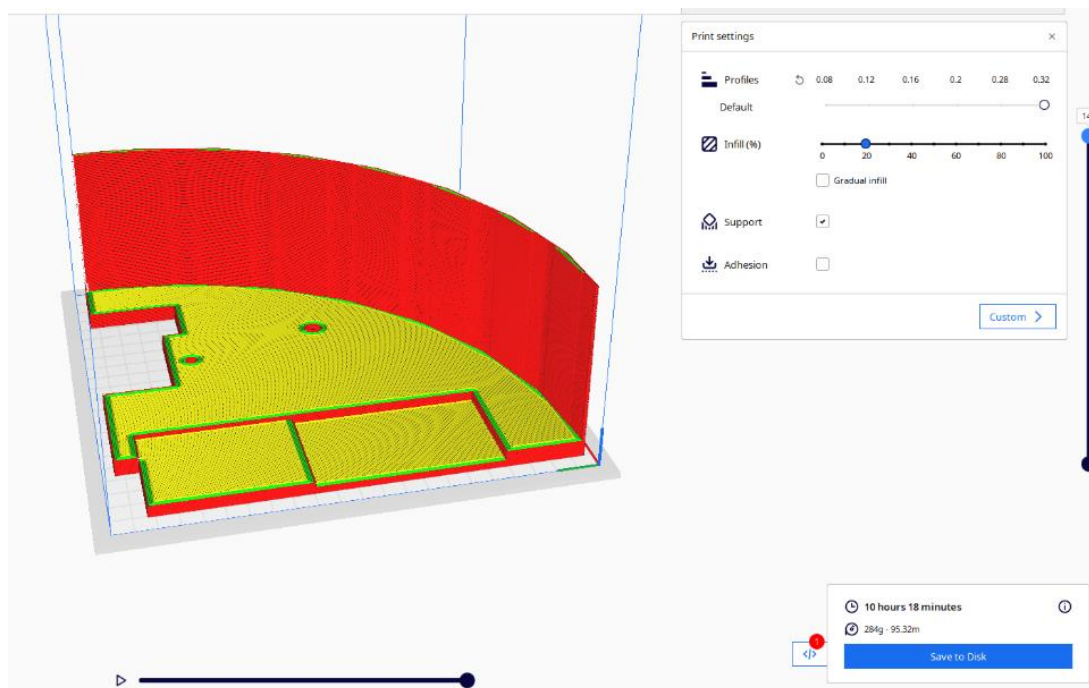


15. Ábra: A kapcsolási rajz blokkdiagrammos ábrázolásban
Forrás: saját kép

4.1.3. A robot alkatrészeinek nyomtatása

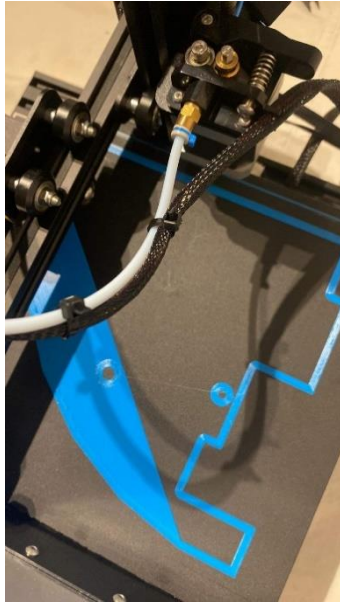
Az alkatrészek, valamint a kapcsolási rajz megtervezése után következett a nyomtatási fázis. A nyomtató beüzemelése után a Cura nevű programot használtam. A Cura egy felhasználóbarát program, amelyben testre szabhatjuk a nyomtató beállításait, mint például, hogy milyen gyorsan dolgozzon a nyomtató, mekkora legyen a tálca és a fűvóka hőmérséklete, ezek mellett sok más beállítást is konfigurálhatunk. A Solid Edge-ben készített fájlokat „OBJ” formátumban érdemes elmentenünk ahhoz, hogy a Cura méretpontosan tudja beolvasni a 3 dimenziós testeket. Miután beimportáltuk a fájlt, és elvégeztük a szükséges beállításokat, a Cura legenerálja a G-kódot, amivel a nyomtató már tud is dolgozni.

A nagyobb alkatrészeket szét kellett vágnom, mivel a nyomtatóm tálcájának mérete miatt (220x220x250 mm) nem tudok nagyobb alkatrészeket gyártani. Erre a problémára a Netfabb nevű program adott választ, amely lehetővé teszi a 3 dimenziós testek szétvágását X, Y és Z tengelyirányban.

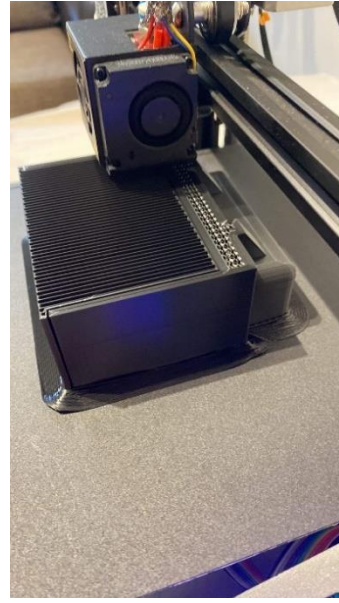


16. Ábra: A robot vázának egyik része, a Cura programba importálva.
Forrás: saját kép

Az alkatrészek PLA-ból készültek, ami egy hőre lágyuló monomer, általában cukornádból vagy kukoricakeményítőből készítik. Azért választottam ezt az anyagot, mert szilárd halmazállapotban viszonylag nehezen törik, gyorsan lehet vele nyomtatni (a nyomtatás sebessége az anyagtól is függ), valamint lebomlási ideje meglehetősen hosszú (körülbelül 80 év), egyetlen nagyobb hátránya az alacsony hőtűrési tulajdonsága. A nyomtatási beállítások javarészt minden alkatrésznél megegyeztek. A fűvóka hőmérséklete 215 °C, a tálca hőfoka 60 °C volt. A kitöltésük 20%-os, gyroid alakú volt, az alkatrészek falvastagsága általában négy szál volt. A nagyobb alkatrészek legyártásához 0.8mm-es fűvókát használtam, a kisebb, precíziót jobban igénylő alkatrészeknél pedig 0.4mm-es fejet használtam.



17. Ábra: A robot aljának egyik cikkeje nyomtatás közben.
Forrás: saját kép



18. Ábra: A porgyűjtő tartály első prototípusának nyomtatása.
Forrás: saját kép



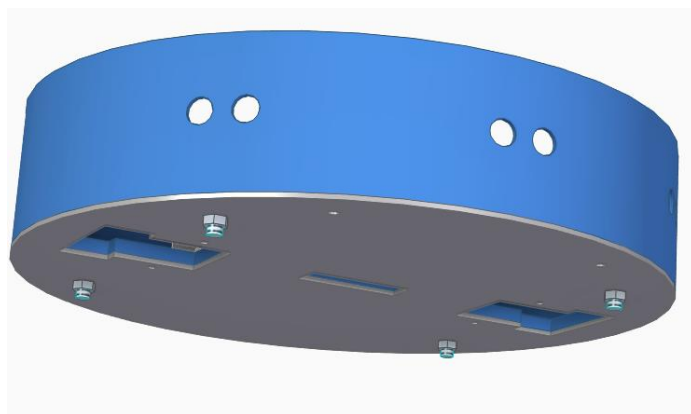
19. Ábra: A robot váza, még össze nem szerelt állapotban.
Forrás: saját kép

4.2. A robotporszívó összeszerelése

Ebben a fejezetben bemutatásra kerül a robotporszívó összeszerelése, valamint az összeszerelés közben használt módszerek és technológiák

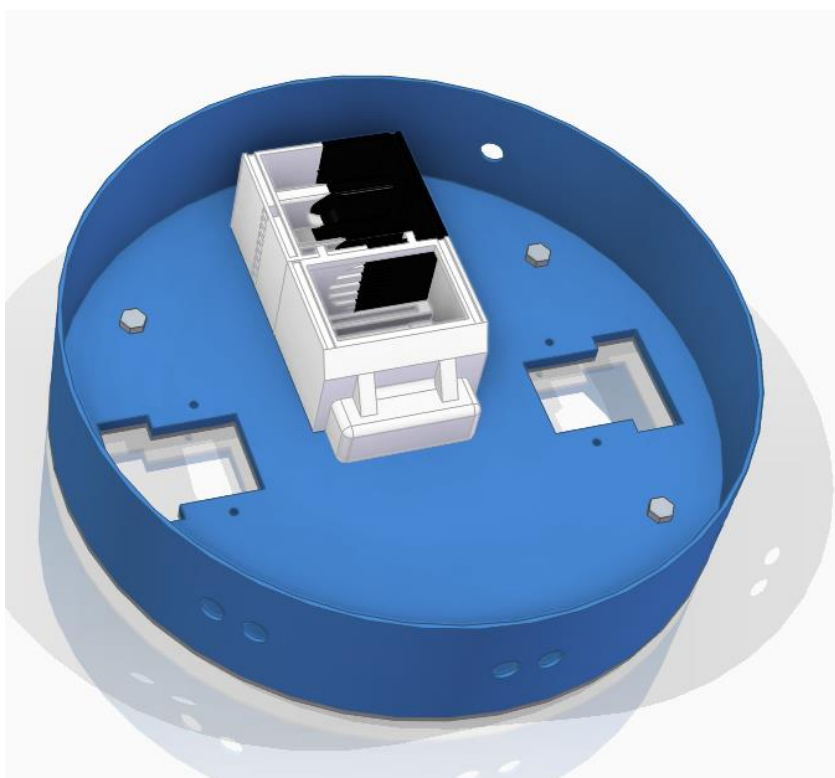
4.2.1. Mechanikai összeszerelés

Elsőként a robot vázát kezdtem el összeszerelni, amely a nyomtatás során négy részre lett szétvágtva. A váz stabilitása, és összeszerelésének megkönnyítése érdekében felkerestem egy műhelyt, ahol foglalkoznak lézervágással, és készítettem egy méretpontos, 3 mm vastag fémlemezt, amelyre a váz kétkomponensű epoxy ragasztóval, valamint négy darab M8-as hatlapfejű csavarral lett rögzítve.



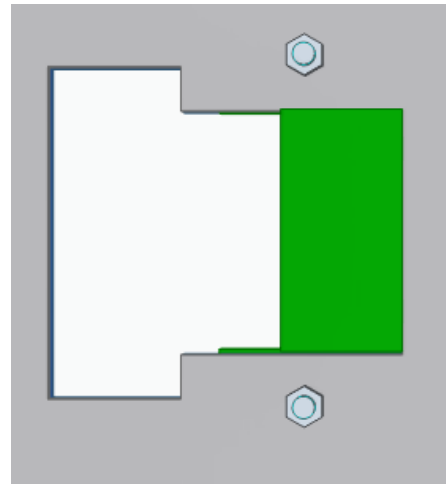
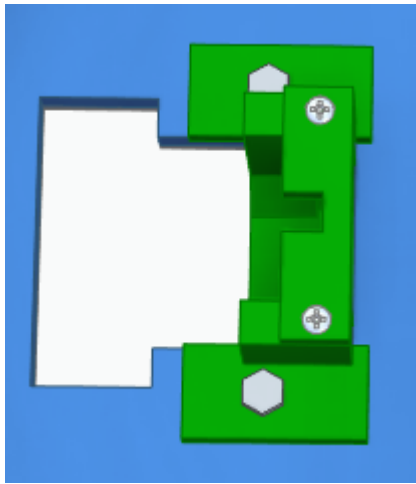
*20. Ábra: A robot váza összeszerelt állapotban.
Forrás: saját kép*

Következő lépésként a porszívó motor házát, valamint a porgyűjtő tartályt illesztettem a helyére, amelyek rögzítése ragasztással történt.



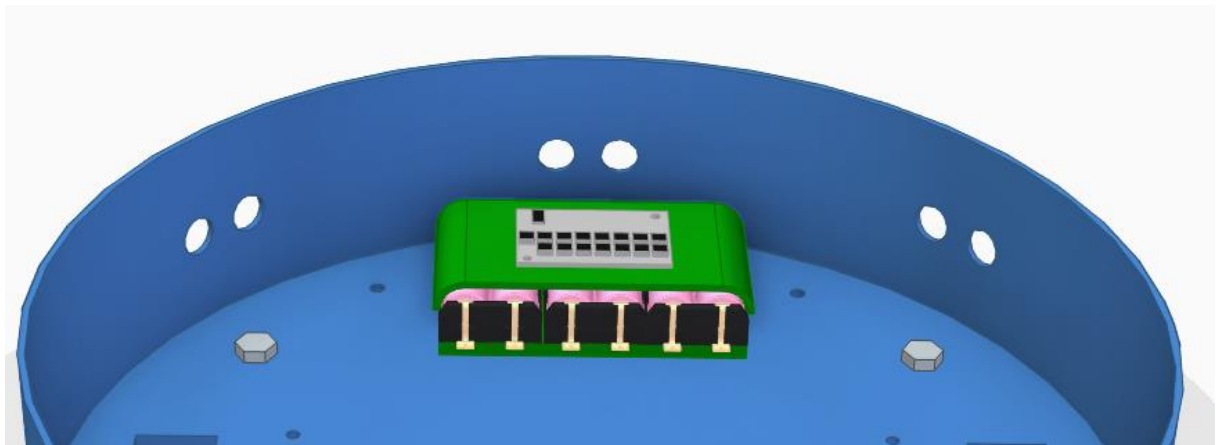
*21. Ábra: A motorház, és a porgyűjtő tartály behelyezve a kialakított helyükre.
Forrás: saját kép*

Ez után a léptetőmotorokat tartó elempárokat illesztettem egymáshoz és helyükre mindekét esetben csavarkötést alkalmazva.



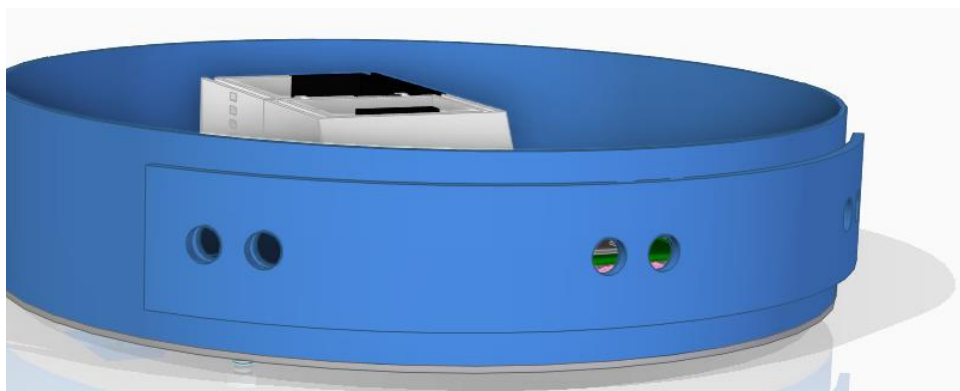
22. Ábra: A motortartó elempárok helyükre illesztett állapotban.
 Forrás: saját kép

A vázon belül már csak egy összeszerelést kellett elvégezni, amely az akkumulátor telep elhelyezését jelentette. A telep talp részét a már előzőekben említett ragasztóval rögzítettem, majd a cellákat a helyükre illesztettem, majd a takaró elemet kétoldalú ragasztóval illesztettem az akkumulátorok tetejére, végül pedig a töltésvezérlő elektronikát négy sarkán szintén kétoldalú ragasztóval rögzítettem, azért, hogyha a későbbiekben meghibásodna, könnyen ki lehessen cserélni.



23. Ábra: Az akkumulátor telep behelyezve.
 Forrás: saját kép

Utolsó elemként a lökhárítót illesztettem a váz elejére, két oldalú ragasztó használatával, ha esetleg eltörne egy nagyobb erő hatására (meghibásodnak a szenzorok, ezáltal ütközik) akkor könnyen leszerelhető legyen.



24. Ábra: A lökhárító a helyére rögzített állapotban.
Forrás: saját kép

A következő alfejezetben az elektronikus összeszerelés kerül tárgyalásra.

4.2.2. Hardveres összeszerelés

Elsőként az akkumulátor telepet állítottam össze, amely hat darab 18650-es Li-ion akkumulátorból áll, valamint a töltésük vezérléséért egy 6S BMS felel. Az elemtartókon található akkumulátor kimenetekre kábeleket forrasztottam, és így kötöttem össze a cellákat a töltésvezérlő egységgel.

1. táblázat: Az akkumulátorok fontosabb adatai táblázatos formában.

Névleges feszültség:	3.7V
Kapacitás:	2600 mAh
Maximális kisütőáram:	2.6A

2. táblázat: A BMS fontosabb adatai táblázatos formában.

Töltési feszültség:	25.5V
Töltőáram (maximum):	15A
Hőmérséklet tartomány:	-30 és +80 °C

Azért döntöttem a hat darab akkumulátoros telep megépítése mellett, mert nagyobb kapacitást szerettem volna elérni, így a robot több ideig tud aktívan dolgozni, valamint a motorok is erősebben tudnak működni nagyobb feszültségről. Az akkumulátor telep névleges maximális

feszültsége 23.7V. Természetesen találhatóak olyan alkatrészek a robotban, amelyek nem igényelnek ekkora feszültséget, vagy akár meg is sérülnének, ha ekkora elektromos töltést kapnának. Erre a problémára a buck konverterek adtak választ. A buck konverter egy olyan egyenáram teljesítményének átalakítására szolgáló eszköz, amely a magasabb bemeneti feszültséget alacsonyabb kimeneti feszültségre konvertálja át.

A robotban négy darab ilyen konverter található, amelyekre kábeleket forrasztottam. Ezeken a vezetékeken keresztül táplálja őket az akkumulátor telep. Két konverter 23.7V-ból 12V-ot állít elő, ezek táplálják a szívmotort, a léptetőmotorokat, és az aktív hűtést generáló ventilátorokat. A harmadik konverter 5V-ot állít elő, amely a robot „agyát”, a Raspberry Pi-t táplálja, emellett maga a Raspberry Pi is táplál két másik fontos eszközt, a két Arduino Uno R3-as egykártyás számítógépet. Az utolsó feszültség konverter 3V-ot állít elő, amelyet a két seprűt hajtó DC motorok vesznek igénybe. A feszültség szétosztásához párhuzamosítókat használtam, minden konverter ágon legalább két ilyen alkatrész található.



25. Ábra: Párhuzamosító, más néven WAGO
Forrás: www.soselectronic.hu

Következő lépésként a hajtást valósítottam meg. Először két bipoláris, 0.4 Nm-es nyomatékú NEMA17-es léptetőmotorral kezdtem el dolgozni. Azért választottam ezeket a motorokat, mert vezérlésük egyszerű, nagyon precízek, és nagy erő kifejtésére is alkalmasak, amely számomra nagyon fontos szempont volt, mivel a robot viszonylag nagy súllyal rendelkezik (6kg). Sajnos a tesztelések során a motorok gyengének bizonyultak, nem tudták elmozdítani a robotot.

A probléma megoldására erőhatékonyabb motorokat szereztem be, amelyek képesek 0.6 Nm nyomaték kifejtésére is. Pár tesztelés után megbizonyosodtam, hogy ezek a motorok már képesek lesznek mozgatni a robotot.

3. táblázat: A léptetőmotorok lényegesebb adatai táblázatos formában.

Nyomaték [Nm]:	0.6
Fázis ellenállás: [Ohm]	0.9
Maximum fázisáram [A]:	2.5
Hossz [mm]:	47
Tengelyvég [mm]:	5



26. Ábra: NEMA17 léptetőmotor
Forrás: www.klavio.hu

A motorok vezérléséért egy Arduino Uno R3-as egykártyás számítógép felel, amelyre egy CNC Shield V3 is csatlakozik. Azért választottam ezeket az eszközöket, mert munkám készítése előtt különböző irodalmakból, cikkekből, valamint videó anyagokból kiderült számomra, hogy ezt az eszközpárosítást a legegyszerűbb használni a motorok, illetve motorvezérlők menedzselésére.

TDK munkám készítése során a motorvezérlők használata volt az egyik legnagyobb kihívás, mivel háromfajta motorvezérlő közül tudtam választani, amelyek a feladatnak, valamint követelményimnek megfelelték, ezek a következők:

- TMC2208
- DRV8825
- A4988

Először olyan típusú A4988-as motorvezérlőket telepítettem a CNC Shield-re, amelyeken az áramkezelő ellenállások 200 Ohmosak voltak, több tesztelés után arra a következtetésre jutottam, hogy ez a fajta motorvezérlő nem megfelelő arra a célra, amire én használnám, mivel a motorok nem tudtak akkora erőt kifejtetni, hogy megmozdítsák a robotot.

Ezek után a CNC Shieldre DRV8825-ös motorvezérlőket telepítettem, amelyek 8.2V-tól 45V-ig képesek működni, maximum áramfelvételük 2.5A. Ezáltal a motorok nagyobb erőt tudtak kifejteni, így a robot már képes volt a mozgásra. Az összeépítés során nagyobb hűtőbordákat szereltem fel a két motorvezérlőre, mert a gyári hűtőbordákkal túlságosan nagy hőt generáltak, ami olyan problémákat szült, mint például a motorvezérlőkre csatlakoztatott kábelek megolvadása.

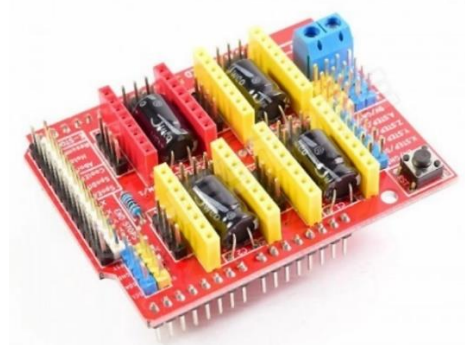
A motorvezérlőkön állítható az áram erőssége, ami befolyásolja a léptetőmotorok erőfejlesztését is. A DRV8825-ös motorvezérlőkhöz az alábbi képletet kell használni ahhoz, hogy a megfelelő áramerősséget kapjuk végeredményül:

$$V_{REF} = \frac{I_{max}}{2}$$

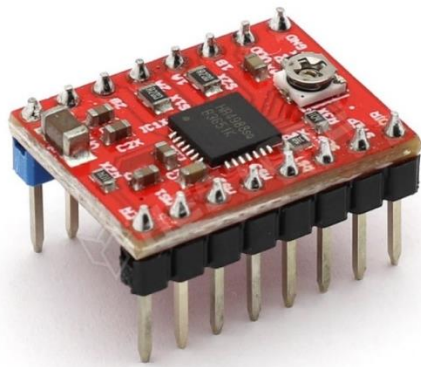
A képlet alapján a motorvezérlőim 2 ampernyi áramot engednek át a motoroknak egyenként, ami a robot stabil mozgását eredményezi.



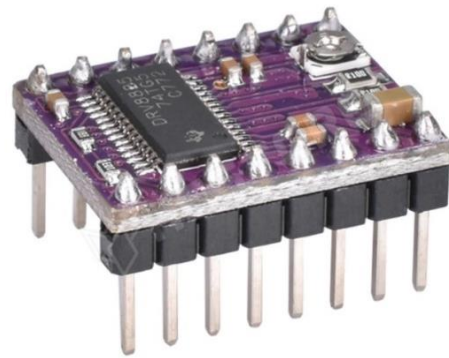
27. Ábra: Arduino Uno R3
Forrás: www.hestore.hu



28. Ábra: CNC Shield V3
Forrás: www.hestore.hu



29. Ábra: A4988-as motorvezérlő
Forrás: www.hestore.hu



30. Ábra: DRV8825-ös motorvezérlő
Forrás: www.hestore.hu

A hajtás összeszerelése után a már helyén lévő szívmotorként funkcionáló DC motor kábelezését alakítottam át, mivel eredetileg szivargyújtó töltős kialakítással rendelkezett. A szivargyújtós kábelvéget lecsíptem, majd blankoló fogó segítségével megcsupaszítottam a vezetékeket, és bevontam őket ónnal, majd az ón bevonatú kábeleket becsatlakoztattam a párhuzamosítók egyikébe.

Következő lépésként a Raspberry Pi tápellátását oldottam meg. A feszültséget az 5V-ot szolgáltató Buck konverter ágáról veszi igénybe, azon belül is egy C típusú USB kábel juttatja el a Raspberry számára az energiát. Az USB kábelt műanyag bevonatát blankoló fogó segítségével eltávolítottam, majd a vezetékek végeit ónnal vontam be, és egy párhuzamosító egységben elhelyeztem a kábelvégeket.

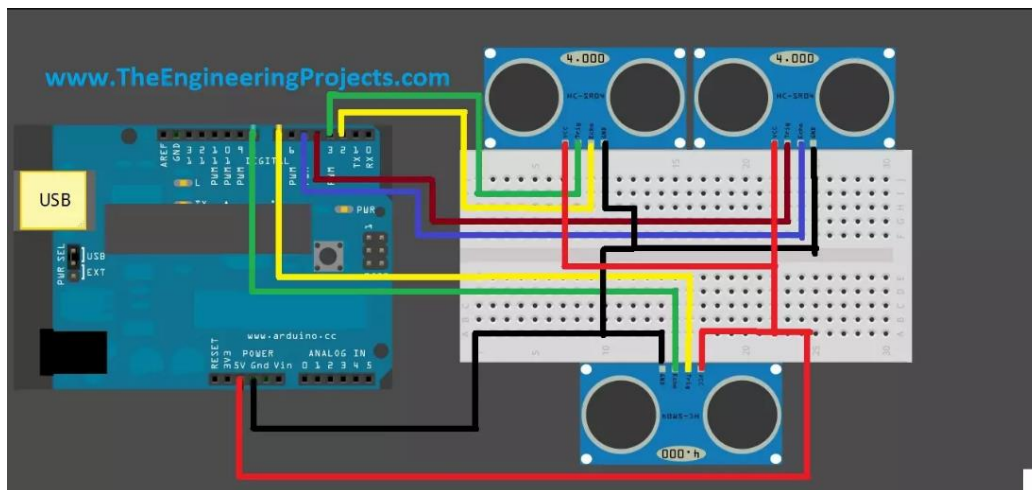


31. Ábra: Raspberry Pi 4
Forrás: www.waveshare.com

A Raspberry tápellátása után elkezdtem vizsgálni, hogy a két Arduino Uno vezérlőkártyának hogyan tudnék energiaforrást biztosítani, és arra jutottam, hogy a Raspberry Pi-on található B

típusú USB portok megfelelnek erre a célra, mivel 5V-nyi feszültséget tudnak szolgáltatni, ami a két Arduino tápellátási kritériumának eleget tesz.

Miután az Arduino vezérlőkártyáknak is biztosítottam tápellátást, így azok is képesek voltak tápfeszültséget előállítani. Ezt az energiaforrást az ultrahangos távolságérzékelő szenzorok használják. A távolságérzékelők 5V-nyi feszültséget igényelnek, amit szalagkábeleken, másnéven jumper kábeleken keresztül kapnak meg, valamint a kommunikáció az Arduinoval is ilyen kábeleken keresztül történik.



32. Ábra: Az ultrahangos távolságmérő szenzorok bekötési rajza
Forrás: www.theengineeringprojects.com

Ezek után a seprűket működtető 3V-os DC motorok energiaellátását oldottam meg. Feszültségforrásukat egy Buck konverter biztosítja, amely 3V-nyi feszültséget generál a kimenő ágán. A DC motorok kivezetéseire kábeleket forrasztottam, amelyek egy-egy párhuzamosítóba vannak bekötve. Annak érdekében, hogy a robotporszívó valóban képes legyen a seprési művelet elvégzésére, így az egyik DC motorra fordított polaritással forrasztottam kábeleket. Leegyszerűsítve, a forrasztás során felcseréltem a pozitív és negatív oldalt, hogy az egyik motor az óramutató járásának ellentétesen forogjon, ezáltal a seprűk a szívónyílás irányába seprik a szennyeződések.

Utolsó lépésként a CNC Shield-et láttam el tápfeszültséggel. A CNC Shield 12V-nyi feszültséget igényel, ezt a feladatot az előzőekben is ismertetett feszültségkonverterek egyike látta el. A CNC Shield esetében csak két szimpla kábelre volt szükségem, amelyeknek egyik vége a CNC Shield-en található oldható kötésű aljzatba, a másik kábelvégek pedig egy párhuzamosítóba csatlakoznak.

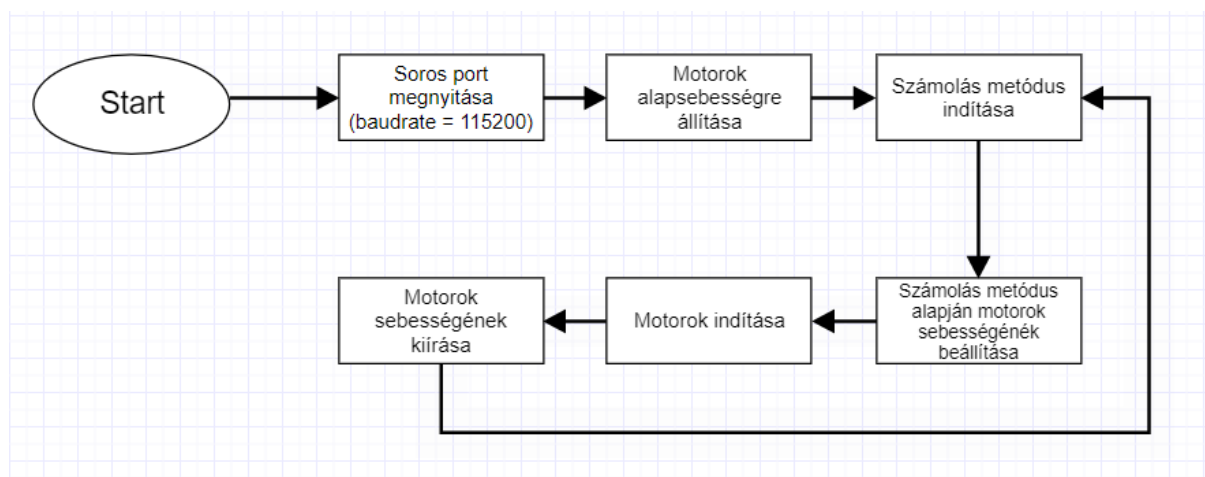
Ezzel az elektronikus összeszerelés fejezet végéhez érkeztünk. A következő fejezetben ismertetésre kerül a robotporszívó programkódja, és annak működése

4.3. A robot szoftveres fejlesztése

Ebben a fejezetben bemutatásra kerülnek a szoftverek, amelyek a robotot működtető egykártyás számítógépeken futnak.

4.3.1. Arduino Uno programkódok (C nyelvű programozás)

Első lépésként azt az Arduino szoftvert valósítottam meg, amely a motorok mozgásáért felel.

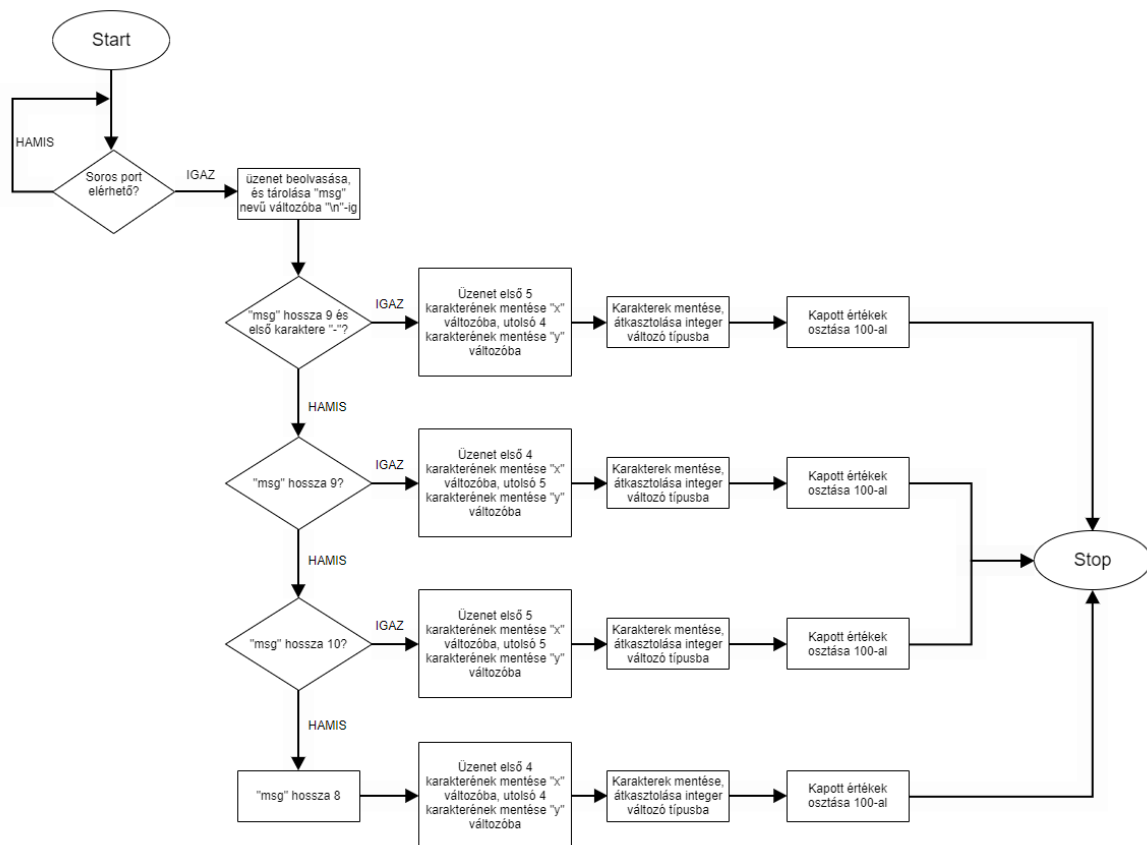


33. Ábra: Az léptetőmotorok mozgatásáért felelős Arduino kód első részének folyamatábrája.
Forrás: saját kép

Először megvizsgáltam, hogy milyen könyvtárakra lesz szükségem Arduino programozásom során. A könyvtárak különböző fájlokból állnak össze, amelyek plusz funkcionalitást tesznek hozzá a programhoz. Vizsgálatom során egyetlen könyvtárat találtam, amely elősegíti a motorjaim vezérlésének megoldását. Ez a könyvtár az „AccelStepper” nevet viseli.

A kód elején létrehozásra kerültek különböző változók, amelyek közreműködnek a motorok pontos sebességének beállításában. Ez után a kód belép az ún. „setup” függvénybe, amely a program indítása során csak egyszer fut le. Ez a szekció tartalmazza a kommunikáció

megnyitását a soros porton, amely jelen esetben egy AB típusú USB kábeles összeköttetést jelent az Arduino és a Raspberry Pi között. Következő lépésként a kód alaphelyeztetre állítja a motorokat, azaz nulla értéket küld a motorok sebességváltozójába. A program ez után belép a „loop” függvénybe, ami nevéből adódóan ismétlődve fut a program futása alatt. A loop szekció első lépésként a „szamolas” függvényt hívja meg, amelyet a következő bekezdésben fogok ismertetni.

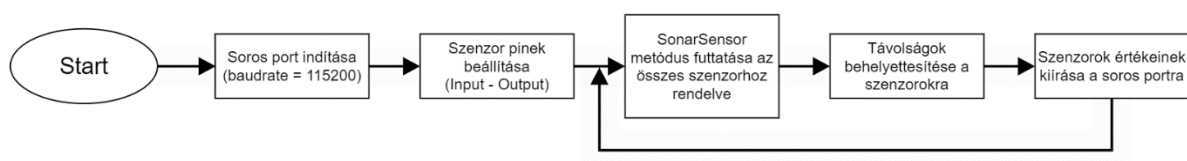


34. Ábra: A motorok mozgatásáért felelős Arduino programkód „szamolas” függvényének folyamatábrája
Forrás: saját kép

A „szamolas” függvény megvizsgálja, hogy a soros port elérhető-e, ha igen akkor a kommunikáció során beérkezett üzenetet beolvassa és eltárolja egy változóban. Az eltárolt üzenet ez után több vizsgálaton esik át, amelyekhez „if” (ha) elágazásokat használtam. Ezek lekérdezik, hogy az üzenet mennyi karaktert tartalmaz, illetve található-e benne mínusz előjel (utóbbi csak egy esetben kerül lekérdezésre). Miután a függvény megvizsgálta az üzenetet, szétbontva eltárolja két változóban, amelyeket utána „integer” (egész szám) típusú változókra konvertálok át. Ennek az az oka, hogy a léptetőmotorok vezérlésére használt csomag csak egész szám típusú változókat tud kezelni sebesség értéként. Utolsó lépésként a már integer típusú változókat elosztom százszal, mert a kapott értékek a Python kódban futó sikeres adatküldésért felelős átalakítás során elvesztik tizedes elválasztó jelüket.

Miután a számolásért felelős függvény lefutott a loop-ban, beállításra kerülnek a motorok sebességei, ez után a motorok elindításra kerülnek, majd ellenőrzés okán ezeket az adatokat az Arduino visszaküldi a Raspberry Pi-on futó Python kódnak.

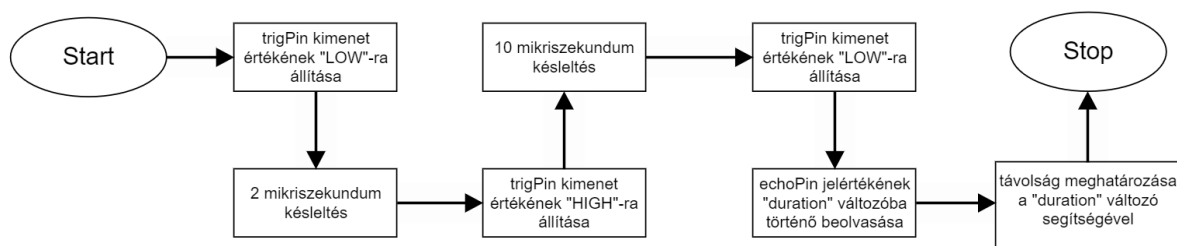
A következő Arduino program az ultrahangos szenzorok által mért távolságok beolvasásáért és elküldéséért felel.



*35. Ábra: Az ultrahangos szenzorokat kezelő Arduino programkód első részének folyamatábrája.
Forrás: saját kép*

A program első parancssorai az ultrahang szenzorok kommunikációs csatlakozásait definiálják, majd különböző változók kerülnek létrehozásra, amelyek az ultrahangos szenzorok által mért adatokat tárolják, illetve továbbítják. A setup függvényben megnyitásra kerül a soros portos kommunikáció a Raspberry Pi-al, majd a szenzorok csatlakozói beállításra kerülnek, a „Trig” pinek outputként, az „Echo” pinek inputként kerülnek beállításra. A program ez után belép a loop függvénybe, amely kiolvassa a „SonarSensor” függvényben kapott távolságok értékeit, majd továbbítja azokat a Raspberry Pi felé, szenzoronként kötőjellel elválasztva.

A SonarSensor függvény kiküld egy impulzust, majd megvárja amíg a jel visszaér, az impulzus kiküldése és visszaérkezési közti időt megméri. Ezen időegység alatt a jel eljutott az akadályig, majd az akadálytól vissza az érzékelőhöz. Mivel a jel az akadály és a szenzor közötti távolságot kétszer tette meg, ezért a beérkezett jel értékét kettővel osztottam, majd az így kapott milliszekundumban értelmezett időegységet továbbosztottam 29.1milliszekundummal, ami az az időegység, amely alatt a hang 1 centimétert tesz meg. Így a számított érték az akadály szenzortól való távolságát jelenti centiméterben.



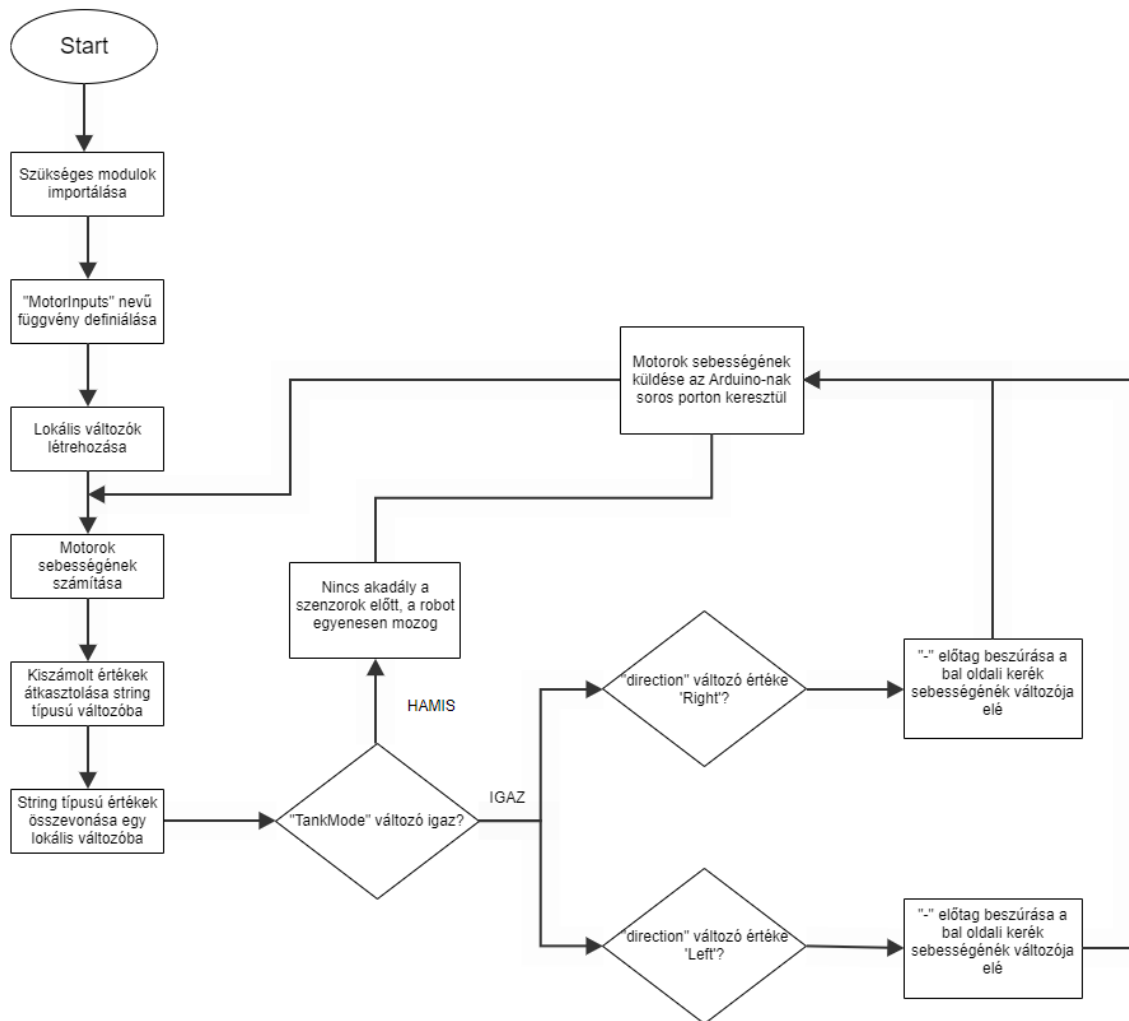
36. Ábra: A SonarSensor függvényhez tartozó magyarázó folyamatábra.
Forrás: saját kép

4.3.2. Python alapú programozás

Az Arduino kódok elkészítése után a Raspberry Pi-on futó Python programozási nyelvű kódokat készítettem el. Azért döntöttem a Python használata mellett, mert az egyetemi tanulmányaim alatt volt szerencsém részt venni a „Robotok alkalmazása és programozásuk” című kurzuson. A tanórák keretein belül én, és szaktársaim megismerkedtünk a Robot Operating System (ROS) rendszerrel, ami főként C++ és Python alapú programokat használ. Ennek hatására tanulmányozni kezdtem a Python programozási nyelvet, és annak sajátosságait. Habár a robotporszívómon nem ROS alapú rendszer fut, a jövőben, mint fejlesztési opció szeretném, ha ez a rendszer irányítaná a robotomat. Ezen indok alapján használtam a Python programozási nyelvet. Jelen állapotban tehát egyszerű, Python alapú programkódok felelnek a robotporszívó irányításáért, valamint környezetének érzékeléséért. Elsőként egy olyan programot készítettem el, amely két adott értéket számol ki. A két számított érték először tört számként kerül eltárolásra, amelyet utána a program kerekített értékre alakít át, azaz egész számmá, majd végül „String” (szöveg) típusú változóba kerülnek átkasztozásra. Ez után a Raspberry Pi elküldi az Arduino vezérlőkártyának az értékeket, amely ezeket az adatokat állítja be a motorok sebességeként.

A kód tartalmaz egy „TankMode” névre hallgató függvényt is, amely akkor kerül meghívásra, ha az ultrahangos távolságmérő szenzorok akadályt érzékelnek maguk előtt. A függvényen

belül található két elágazás is, amelyek azt vizsgálják, hogy a szenzorok bal, vagy jobb oldalt érzékelnek akadályt maguk előtt, majd a kapott információ alapján döntést hoz, hogy mely irányba forgassa a robotot az akadály kikerülésének érdekében. Ennek a műveletnek az elvégzése nagyon egyszerű, a kód szimplán csak annyit csinál, hogy vagy a bal, vagy a jobb oldali motor sebességének értéke elé beimportál egy mínusz előtagot, attól függően, hogy melyik oldalon észlelte a szenzorok akadályt, így a robot képes az egyhelyben történő forgásra.

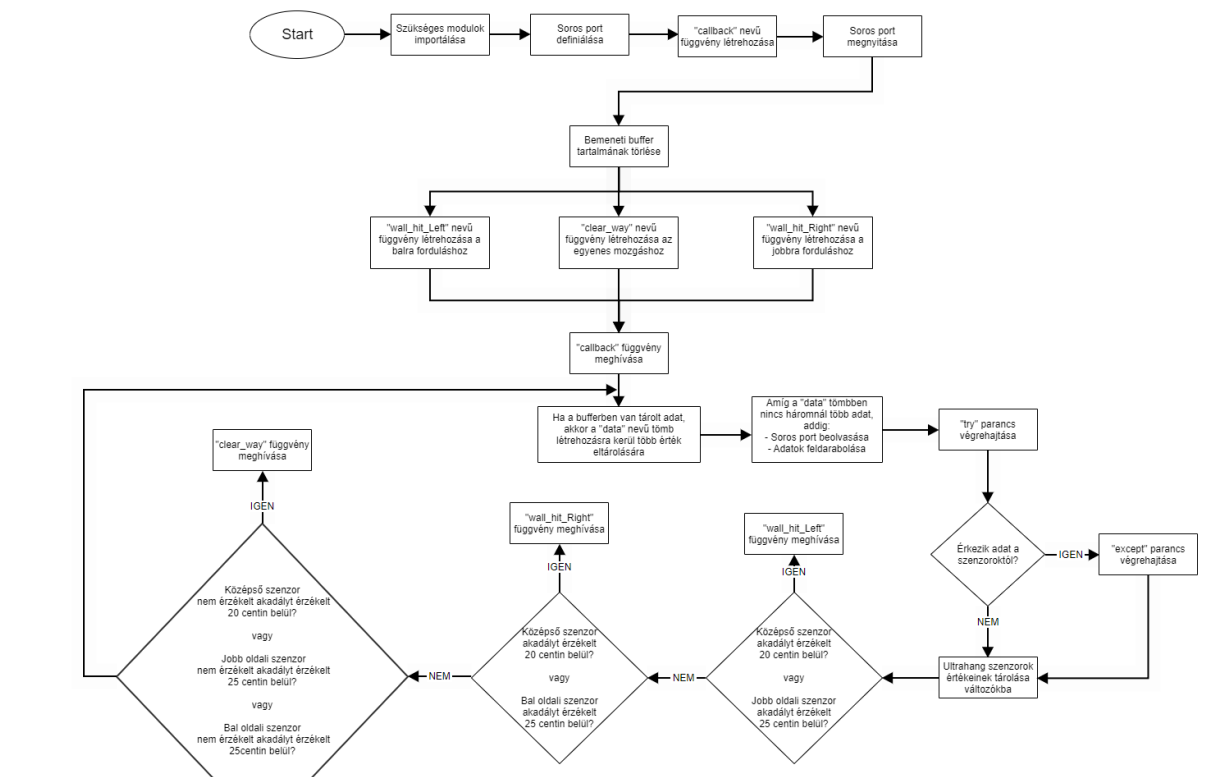


37. Ábra: A motorok sebességének beállításáért felelős program folyamatábrája.

Forrás: saját kép

A következő Python alapú programkód az Arduino-tól kapott ultrahangos szenzorértékeket kezeli. Ennek leegyszerűsítésére létrehoztam négy függvényt. A négy függvényből három a robot mozgásának koordinálásáért felel. Ezek, a szenzorok jelei alapján kerülnek meghívásra, például, ha a szenzorok bal oldalt érzékelnek akadály, akkor a „wall_hit_Left” névre hallgató függvény aktiválódik, ami ilyenkor az előzőekben említett TankMode függvénybe importálja

az akadály kikerüléséhez szükséges adatokat. Ellenkező esetben, tehát ha jobb oldalt érzékelhető az akadály veszélye, akkor a „wall_hit_Right” függvény aktiválódik, ha pedig a robot előtt nem található akadály, akkor a „clear_way” metódus kerül aktív állapotba.



38. Ábra: Az ultrahangos szenzorok által küldött jeleket kezelő Python alapú programkód folyamatábrája.
Forrás: saját kép

5. Összefoglalás

TDK munkám témája egy olyan robotporszívó elkészítését követelte meg, amely alkalmas nagy területek takarítására. A robotporszívó elkészítéséhez számos szakirodalmat, kutatást, illetve oktató videót tanulmányoztam, ezek mellett piackutatást is végeztem, hogy az ehhez hasonló feladatok ellátását elvégző robotporszívók milyen alkotóelemekből tevődnek össze.

A robot megtervezése előtt összegyűjtöttem a szükséges elektronikai alkatrészeket. A robotporszívó mozgatásához léptetőmotorokat alkalmaztam. A környezeti érzékelésért, illetve koordinációért három ultrahangos távolságérzékelő szenzor felel. Mindezen eszközök vezérlésére egy Raspberry Pi 4, illetve két Arduino Uno típusú egykártyás számítógépet alkalmaztam.

Ez után elkészítettem a robotporszívó 3 dimenziós terveit a Solid Edge program segítségével, minden komponenst úgy terveztem meg, hogy figyelembe vettem az elektronikai alkatrészek méreteit. Ezek után a robot alkotóelemeinek elkészítéséhez a 3D nyomtatás technológiáját alkalmaztam, amely egyszerűbbé tette a robotporszívó vázának és kisebb alkatrészeinek legyártását.

A robot váza, illetve kisebb alkatrészei elkészültét követően összeszereltem a robotporszívó 3D nyomtatott elemeit, amelyeket csavarkötéssel és ragasztással illesztettem egymáshoz.

Az összeszerelést követően az elektronikai alkatrészeket illesztettem helyükre, illetve kötöttem össze egymással. A robotban négy áramkör található, amelyek létrehozásához buck konvertereket alkalmaztam.

A hardveres összeszerelés után elkészítettem a robotporszívót működtető programokat. A programok két „C” nyelvű programkódból, illetve három Python nyelvű programból tevődnek össze.

A robotporszívó a célkitűzésben foglaltak alapján feladatát megfelelően teljesíti. Emellett számos fejlesztési opció érhető el. Amennyiben a robotporszívót precízebben szeretném, hogy közlekedjen, illetve érzékelje környezetét, a ROS rendszer alkalmazása lenne a legmegfelelőbb választás. Ennek alkalmazásával elkészíthető lenne a térképezési funkció, illetve számos, precizitást segítő funkció, amellyel a piacon fellelhető robotporszívók többsége már rendelkezik.

6. Summary

The topic of my thesis required the creation of a robotic vacuum cleaner that is suitable for cleaning large areas. In order to create the robot vacuum cleaner, I studied a literatures, researches, and educational videos. In addition, I also conducted market research on similar robot vacuums so that I can understand what components are used inside these cleaner robots.

Before designing the robot, I collected the necessary electronic components. I used stepper motors to move it. Three ultrasonic distance sensors are responsible for environmental perception and coordination. I used a Raspberry Pi 4 and two Arduino Uno single-board computers to control all these devices.

After that, I prepared the 3-dimensional plans of the robot vacuum cleaner with the help of the Solid Edge program, I designed each component taking into account the dimensions of the electronic components. After that, I used 3D printing technology to make the components of the robot, which made it easier to manufacture the robot vacuum cleaner's frame and smaller parts.

Next, when the robot's frame and its smaller parts were completed, I assembled the 3D printed elements of the robot vacuum cleaner, which I attached to each other by screwing and gluing.

Following the assembly, I put the electronic components in place and connected them together. There are four circuits in the robot, which I used buck converters to create.

Subsequently the hardware assembly, I prepared the programs that operate the robot vacuum cleaner. The programs consist of two "C" language program codes and three Python language programs.

The robot vacuum cleaner fulfills its task properly based on the objectives. In addition, many development options are available. If I want the robot vacuum cleaner to move and sense its environment more precisely, the application of the ROS system would be the most appropriate choice. By using this, the mapping function could be created, as well as a number of precision-aiding functions, which the majority of robot vacuum cleaners on the market already have.

7. Irodalomjegyzék

- [1] D. T. Attila, „BANKWEB,” 26. július 2017.. [Online]. Available: <https://www.bankweb.hu>. [Hozzáférés dátuma: 27. szeptember 2022.].
- [2] ribarszki, „Logout.hu,” 30 07 2017. [Online]. Available: <https://logout.hu>. [Hozzáférés dátuma: 27 09 2022].
- [3] M. Taufer, „Commodity single board computer clusters and their applications,” *Future Generation Computer Systems*, pp. 201-212, 2018.
- [4] S. W. Matt Richardson, *Getting Started with Raspberry Pi*, Sebastopol, CA: Maker Media Inc., 2013.
- [5] A. Javaid, „linuxhint,” 03 10 2022. [Online]. Available: <https://linuxhint.com>. [Hozzáférés dátuma: 08 10 2022].
- [6] A. A. Galadima, „Arduino as a learning tool,” Institute of Electrical and Electronics Engineers, Abuja, Nigéria, 2014.
- [7] Y. A. Badamasi, „The working principle of an Arduino,” Institute of Electrical and Electronics Engineers, Abuja, Nigéria, 2014.
- [8] S. P. S. S. Kuldeep Singh Kaswan, „Role Of Arduino In Real World Applications,” *International Journal of Scientific & Technology Research*, pp. 1-2, 2020.
- [9] V. Tibor, „ResearchGate,” *Design and realization of a feeder part of the automated production line*, p. 54, 2020.
- [10] K. Chakraborty, „Techopedia,” 25. február 2020. [Online]. Available: <https://www.techopedia.com>. [Hozzáférés dátuma: 01. október 2022.].
- [11] „Powergom,” 04. április 2017. [Online]. Available: <https://www.powergom.hu>. [Hozzáférés dátuma: 01. október 2022.].
- [12] H. A. Gabbar, „Review of Battery Management Systems (BMS) Development,” University of Ontario, Oshawa, Canada, 2021.

- [13] R. Crowder, in *Electric Drives and Electromechanical Systems*, Elsevier Ltd, 2019, pp. 209-226.
- [14] L. R. Palmer, „About Mechanics,” 03 09 2022. [Online]. Available: <https://www.aboutmechanics.com>. [Hozzáférés dátuma: 03 10 2022].
- [15] A. Morega, „Design and modeling of a hybrid stepper motor,” in *Institute of Electrical and Electronics Engineers*, Bukarest, 2017.
- [16] UTMEL, „UTMEL Electronic,” 02. március 2021. [Online]. Available: <https://www.utmel.com>. [Hozzáférés dátuma: 07. október 2022.].
- [17] Anita, „Easybom,” 22. szeptember 2021. [Online]. Available: <https://www.easybom.com>. [Hozzáférés dátuma: 04. október 2022.].
- [18] „Components-101,” 13 november 2019. [Online]. Available: <https://components101.com>. [Hozzáférés dátuma: 04 október 2022].
- [19] S. Daley, „BuiltIn,” 18. augusztus 2022. [Online]. Available: <https://builtin.com>. [Hozzáférés dátuma: 05. október 2022.].
- [20] D. Partida, „Rehack,” 07 12 2020. [Online]. Available: <https://rehack.com>. [Hozzáférés dátuma: 05 október 2022].
- [21] F. Fahimi, *Autonomous Robots*, New York, NY: Springer New York, NY, 2009.
- [22] J. Walker, „Locus Robotics,” [Online]. Available: <https://locusrobotics.com>. [Hozzáférés dátuma: 05. október 2022.].
- [23] J. Snape, „Smooth and Collision-Free Navigation for Multiple Robots Under Differential-Drive Constraints,” *Institute of Electrical and Electronics Engineers*, Taipei, Taiwan, 2010.
- [24] „BCN3D,” 15. május 2020. [Online]. Available: <https://www.bcn3d.com>. [Hozzáférés dátuma: 30. szeptember 2022.].
- [25] J. Horvath, in *Mastering 3D Printing*, Apress Berkeley, CA, 2014, pp. 3-10.
- [26] N. Shahrubudin, „An Overview on 3D Printing Technology: Technological, Materials, and Applications,” *Procedia Manufacturing*, pp. 1286-1296, 2019.

[27] J. Bell, „NS Medical Devices,” 19. feburár 2021. [Online]. Available: <https://www.nsmedicaldevices.com>. [Hozzáférés dátuma: 30. szeptember 2022.].