



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

KANDÓ KÁLMÁN VILLAMOSMÉRNÖKI KAR
ELEKTRONIKAI ÉS KOMMUNIKÁCIÓS RENDSZEREK INTÉZET
MŰSZERTECHNIKAI ÉS AUTOMATIZÁLÁSI TANSZÉK



SZAKDOLGOZAT



OE-KVK
2023.

Tallián Richárd
T009478/F112904/K



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Óbudai Egyetem
Kandó Kálmán Villamosmérnöki Kar
Műszertechnikai és Automatizálási Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve: Tallián Richárd

Szkdolgozat száma: SZD2309040923381100

Törzskönyvi száma: T009478/F112904/K

Neptun kódja:

Szak: villamosmérnöki (magyar nyelven)

Specializáció: Műszer-automatika specializáció

A dolgozat címe: Alkatrész beültető gép tervezése

A dolgozat címe angolul: Designing a Pick and Place machine

A feladat részletezése: A számítógépen futó Python program feldolgozza a pick and place fájl sorait, ami egy ASCII kódolású szöveges fájl, és ezek alapján a parancsokat elküldi CAN buszon a mikrokontrollernek ami vezérli a motorokat, szelepeket, és üzeneteket küld vissza a gép állapotáról (aktuális pozíció, hibák, stb.). A pontos alkatrész elhelyezéshez kamerás képfeldolgozást használ.

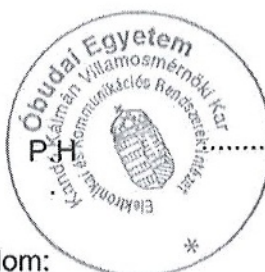
Intézményi konzulens neve: Sándor Tamás

A kiadott téma elévülési határideje: 2025. 12. 31.

Beadási határidő: 2023. 12. 15.

A szkdolgozat: Nem titkos.

Kiadva: Budapest, 2023. 10. 24.



Intézetigazgató

A dolgozatot beadásra alkalmasnak találom:

belső konzulens

külső konzulens



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

KANDÓ KÁLMÁN VILLAMOSMÉRNÖKI KAR
ELEKTRONIKAI ÉS KOMMUNIKÁCIÓS RENDSZEREK INTÉZETI
MŰSZERTECHNIKAI ÉS AUTOMATIZÁLÁSI TANSZÉK I



HALLGATÓI NYILATKOZAT

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

Budapest, 2023. 12. 11.



.....
hallgató aláírása

Tartalomjegyzék

1. Bevezetés.....	6
2. Specifikáció.....	7
2.1. A beültető fej és részei.....	8
2.2. Az alkatrész adagoló.....	11
2.3. Motorok és motorvezérlők, tengelyek hajtása.....	13
2.4. Alkatrész kamera.....	14
2.5. Pozícionáló és ellenőrző kamera.....	15
2.6. Vezérlő számítógép.....	15
3. Logikai rendszerterv.....	16
3.1. Mechanika.....	16
3.2. Elektronika.....	17
3.2.1. Léptető motor vezérlő.....	17
3.2.2. Vákuum és nyomásmérő.....	18
3.2.3. Vezérlő számítógép.....	18
3.2.4. Feeder.....	18
3.3. Szoftver.....	19
4. Fizikai rendszerterv.....	20
4.1. Mechanika.....	20
4.2. Elektronika.....	22
4.2.1. Léptető motor vezérlő.....	22
4.2.2. Vákuum és nyomásmérő.....	25
4.2.3. Vezérlő számítógép.....	26
4.2.4. Feeder.....	27
4.3. Szoftver.....	28
4.3.1. Kamera kezelés és képfeldolgozás.....	29
4.3.2. Feeder vezérlés és státusz olvasás.....	30
4.3.3. Tengely mozgás vezérlése.....	30
4.3.4. Vákuum és nyomásmérés.....	30
4.3.5. Grafikus felhasználói felület.....	31
5. Modulok leírása.....	32
6. Integráció.....	33
6.1. Mechanika.....	33

Tallián Richárd: Alkatrész beültető gép tervezése	5
6.2. Elektronika.....	35
6.3. Szoftver.....	38
7. Tesztelés.....	39
8. Továbbfejlesztési lehetőségek.....	40
9. Összefoglalás.....	41
10. Conclusion.....	42

1. Bevezetés

Az integrált áramkörök lehetővé tették komplex elektronikák építését alacsony áron, ami robbanásszerű terjedést tett lehetővé. A gyors fejlődés és egyre nagyobb integráció eredményeképpen ma már több ezer alkatrészt tartalmaznak a nyomtatott áramköri lapok. Ezek gyors és költséghatékony gyártására szolgálnak a beültető gépek. Az iparban használatos gépek nagy sebességgel mozognak, de emellett precízek, gyors képfeldolgozást használnak és erre speciális hardveres gyorsítást használnak. A dolgozat témája egy ilyen gép tervezése kis darabszámú gyártáshoz, motorok kiválasztása és azok vezérlése, CAN kommunikáció és magának a gépnek mechanikai és vezérlési kérdései és a választott megoldások kiértékelése. A cél egy olcsó és megbízható gép tervezése, a nagy sebesség nem fő szempont. Egy ilyen gép értéke a gyors helyi prototipizálás lehetősége, illetve alkatrészhiányos időkben helyi raktárkészletből való gyártás.



1. Ábra: Opulo asztali beültető gép

2. Specifikáció

Egy beültető gép több funkciót megvalósító részre bontható:

1. beültetést végző vákuumos fej
2. alkatrészeket tartó és adagoló rész, lehet tálca vagy feeder
3. motorok és motorvezérlők, tengelyek hajtása
4. statikus felfelé néző kamera (alkatrész kamera)
5. mozgó fejen lévő lefelé néző kamera (pozícionáló és ellenőrző kamera)
6. ezeket összehangoló vezérlő számítógép
7. stabil mechanikai váz

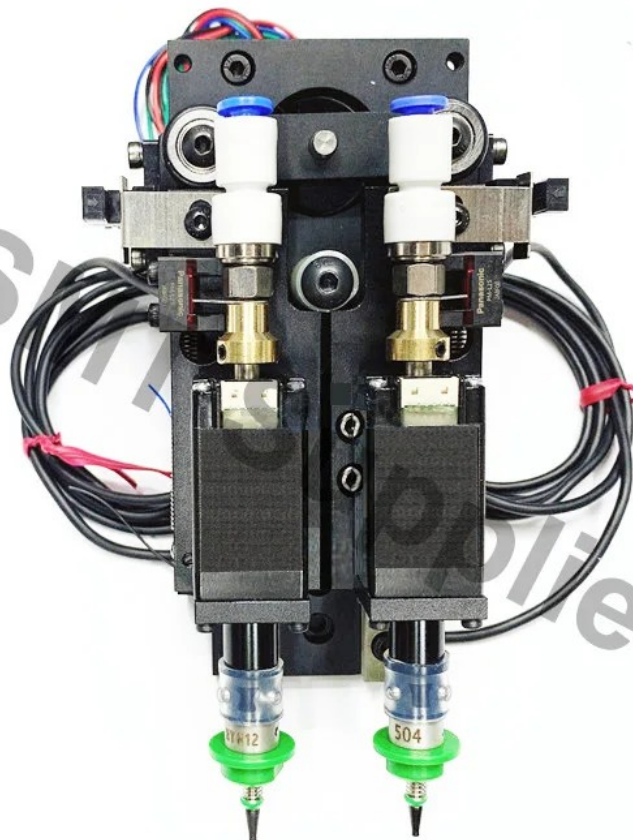
Ellentétben a 3D nyomtatókkal, ahol a motorok egymáshoz viszonyított szinkron mozgása elengedhetetlen a megfelelő működéshez, itt lehetőség van sokkal lazábban kezelni a mozgást.

Ez azt jelenti, hogy amíg egy nyomtatónál egy 45 fokos egyenes nyomtatásához a motoroknak egyszerre kell gyorsulniuk, lassulniuk és azonos sebességgel mozogniuk, itt elegendő eljutni a megadott koordinátára. Arra kell figyelni, hogy amíg a mozgás nem fejeződött be ne próbáljuk meg felvenni vagy lerakni az alkatrészt.

Hasonló igények előfordulnak a két gép között, például a pontos megismételhető mozgás. A két fogalmat meg kell különböztetni mivel lehet pontos egy mozgás, ha a parancsot kiadva a gép jó helyre megy, de ezt több ezerszer meg kell tudni tenni. Mechanikai túlterhelés hatására a gép elveszítheti a helyzetét és visszajelzés hiányában ezt nem lehet korrigálni. Enkóder segítségével megoldható a visszajelzés és egy PID szabályzóval pontos mozgás lehetséges, ám ezeknek az implementálása és beállítása nehéz feladat, amibe cégek sok időt fektetnek, hogy a lehető legelőnyösebb gépet készíthessék el.

2.1. A beültető fej és részei

A beültető fej feladata az alkatrészek felvétele és lerakása a megfelelő helyre, nagy pontossággal és megismételhetőséggel, ez ipari gépeknél általában 10um. Egy gépben több fej is lehet így vehet fel több egyforma alkatrészt vagy különböző fejekkel többféle alkatrészt. Lehetőség van automata fejcserést is végezni. Az általam tervezett gépben 4 fej van 0603, 0805, 1206 méretű passzív alkatrészekhez és IC ültetéshez. Minden fej függetlenül tud fel és le mozogni (Z1, Z2, Z3, Z4), forogni 360 fokban (R1, R2, R3, R4) és saját vákuum szivattyúval és nyomásmérővel rendelkeznek, amivel a rendellenes működés, mint sikertelen alkatrész felvétel vagy lerakás észlelhető. A fejek középpontja szoftverben tárolva van és pozicionálásnál felhasználásra kerül.



Type E

2. Ábra: Két fúvókás beültető fej

A fej JUKI 500-as szériából való fúvókákat használ. Ezek könnyen beszerezhetőek és az automatikus csere is megoldható egy egyszerű tartóval, ami a fúvóka vállát megfogva kihúzza azt a helyéről.

Product information			
NO.	500	501	502
Outer diameter	1.0*0.5mm	0.7*0.4mm	Φ0.7mm
Inner diameter	2*0.4mm	Φ0.25mm	Φ0.4mm
Appearance			
NO.	503	504	505
Outer diameter	Φ1.0mm	Φ1.5mm	Φ3.5mm
Inner diameter	Φ0.6mm	Φ1.0mm	Φ1.7mm
Appearance			
NO.	506	507	508
Outer diameter	Φ5.0mm	Φ8.5mm	Φ9.5mm
Inner diameter	Φ3.2mm	Φ5.0mm	Φ8.0mm
Appearance			

3. Ábra: JUKI fúvóka sorozat

#500	0402
#501	0201
#502	0402, 0603
#503	0603, 0805
#504	1206, SOT23
#505	SOP8, $\leq 5 \times 5 \text{mm}$
#506	$\leq 7 \times 7 \text{mm}$
#507	$\leq 10 \times 10 \text{mm}$
#508	$> 10 \times 10 \text{mm}$

1. Táblázat: Fúvókák és alkatrészek általános összerendelése

Nem kell pontosan így hozzárendelni a fúvókákat az alkatrészekhez. Lehetnek nagyobb tömegű alkatrészek, főleg tekercsek, amik igényelhetnek más méretű fúvókát. Nagyon nagy tömegű alkatrészeknél szükség lehet arra is, hogy lassabban mozogjon a fej, mivel az alkatrész tömegközéppontja elég messze lehet a fúvóka végétől.

2.2. Az alkatrész adagoló

Az SMT alkatrészek szabványos szélességű szalagokban (8, 12, 16, 24 mm) vagy tálcában kaphatóak, utóbbi csak integrált áramkörökre jellemző. A tálcák kezeléséhez a munkatérben való rögzítést kell megoldani, a szoftverbe bevitt tábla méretekből pedig megoldható az alkatrész felvétel. A szalagok kezeléséhez elegendő megoldani egy szalag darab rögzítését a munkatérben, mivel ezek szabványosak így elegendő az első alkatrész helyét megadni. Vannak alkatrészek, amikből sokkal több kellhet, mivel gyakran használatosak és akár egy pár tíz darabos gyártásnál is több száz kellhet belőlük, ezért érdemes lehet automata adagolókat alkalmazni hozzájuk. Ezek az ún. feeder-ek mindig egy adott pozícióba adagolnak alkatrészt, képesek egy teljes tekercset befogadni, ami 3000 – 5000 db passzív alkatrészt jelenthet.



4. Ábra: Siemens siplace feederek

Érdemes átgondolni, hogy hány fajta alkatrészhez kellhet feeder. A gép munkaterébe sok alkatrész fér, de ezeket elhelyezni és pozícionálni idő. Korábbi terveket átnézve és a leggyakrabban használt alkatrészeket kiválasztva összeállítható egy lista ezekről.

Alkatrész	Méret	Érték/Típus
Ellenállás	0603	10, 100, 1k, 4k7, 10k, 22k, 100k
Kondenzátor	0603	12p, 10n, 22n, 100n, 1u, 2u2, 10u, 22u
Kondenzátor	1206	1u, 2u2, 10u, 22u, 100u
Transzisztor	SOT23	2N7002, AO3400, AO3401

2. Táblázat: Gyakori alkatrészek

Ezen kívül szükség van még alkatrészekre, viszont azt ki kell értékelni, hogy mi legyen feeder-ben és mi tálcában vagy a géptérbe szimplán szalaggal rögzítve.

2.3. Motorok és motorvezérlők, tengelyek hajtása

A gép mozgására több megoldás közül lehet választani

- egyenáramú motor
- stepper motor
- szinkron motor
- pneumatikus aktuátor

A beültető fejen léptető motorok vannak, mivel ezek nagyon kis mozgást végeznek és ezt szinte terhelés nélkül, így pontosak maradnak és nem kell pozíció visszajelzés.

Olyan megoldást kerestem, ami egyszerűen megvalósítható, de lehetőség van későbbi továbbfejlesztésre. Ezen opciók közül csak a stepper motor képes open-loop vezérléssel viszonylag pontos pozícionálásra.

Egy fordulat 80 mm mozgást eredményez, a motor egy lépése $1,8^\circ$ tehát 200 lépés 1 fordulat. Továbbá a vezérlő képes ez lépést 256 részre bontani, tehát a felbontás

$$\frac{80000 \mu m}{200 * 256} = 1,5625 \mu m$$

Ez a valóságban csak terheletlenül elérhető el, egy gépben túlterhelés esetén visszaolvasás nélkül nincs lehetőség lépésvesztést detektálni így a pontosság nem garantálható, de megfelelő beállításokkal ez megközelíthető.

A stepper motorok vezérlésére céláramkörök léteznek, amik nagyban megkönnyítik a feladatot. Az általam választott IC SPI buszon pozíció parancsokkal képes a motort vezérelni.

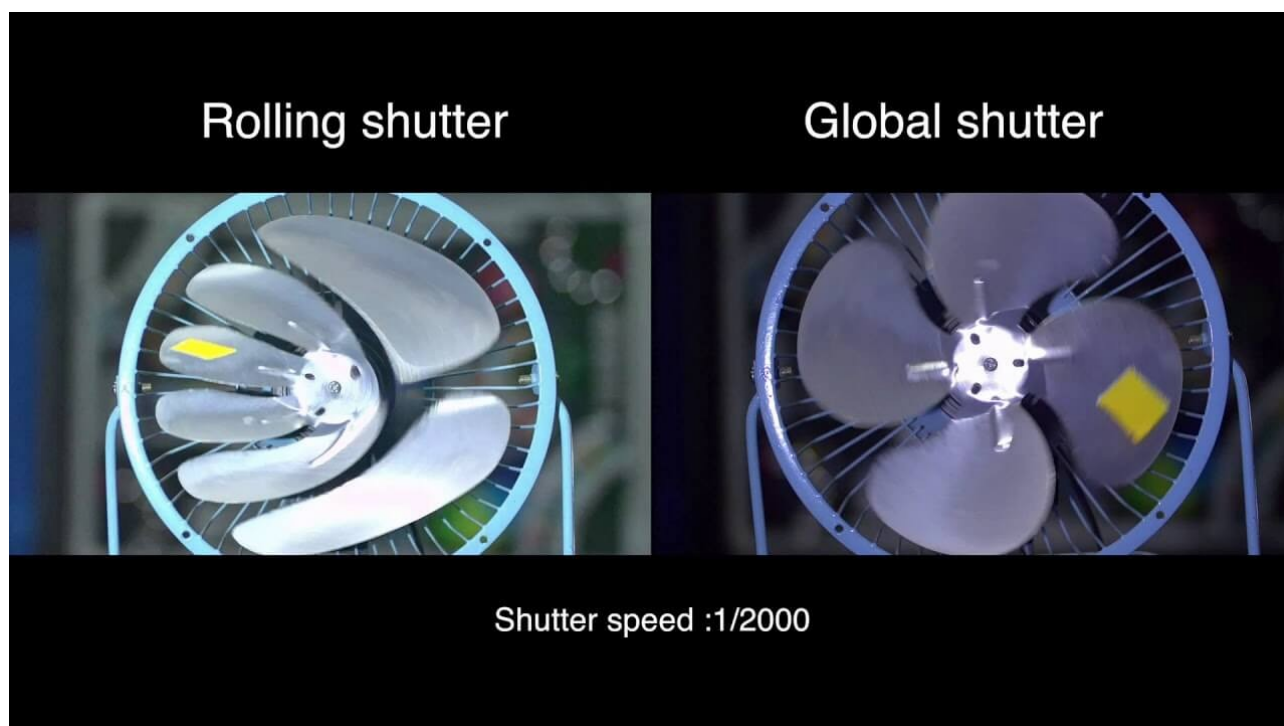
Az erőátvitelt 10 mm széles bordásszík biztosítja, a 3D nyomtatók elterjedésének köszönhetően motorok, szíjtárcsák és szíjak olcsón elérhetőek.

2.4. Alkatrész kamera

Az alkatrészt kivéve a szalagból vagy tálcából nem garantálható, hogy annak pontosan a közepét sikerült felvenni vagy nincs elfordulva. Ennek korrigálására szolgál egy ismert pozícióban levő kamera, ami fölé minden alkatrészt elvisz a fej. A szoftver a képből meghatározza az alkatrészt befoglaló legkisebb négyszöget, annak közepét és rotációját. Ezekből az adatokból elvégzi a korrekciót és a következő lépésben mér a korrigált értékekkel dolgozik a program.

$$\text{új cél pozíció} = \text{számított alkatrész pozíció} - \text{kamera pozíció} + \text{cél pozíció}$$

A beültető gépekben global shutter kamerákat használnak, ami azt jelenti, hogy a teljes szenzort egy adott pillanatban mintavételezi, ez lehetővé teszi, hogy a kamera felett ne kelljen megállnia a fejnek. Erre azért van szükség, mert egy átlagos kamera rolling shuttert használ, ami nem képes gyorsan mozgó tárgyról megfelelő képet készíteni. A rolling shutter soronként vesz mintát a szenzorból és így eltorzul a kép.



5. Ábra: A rolling shutter torzítása

A global shutter hátránya a magasabb ár azonos felbontás mellett. Természetesen alkalmazható normál kamera is, amennyiben figyelünk arra, hogy megálljunk a szenzor felett. Az utóbbi megoldást választottam, mivel az alacsonyabb ár mellett egyszerűbb szoftveres vezérlést tesz lehetővé.

2.5. Pozícionáló és ellenőrző kamera

A home pozíció megtalálásához nem csak a tengely végállás kapcsolók használhatóak. A munkatérbe helyezhető egy panel, ami fiduciális jeleket tartalmaz, ezt a szoftver megkeresi és az ismert pozíciót használja referenciának. Ez a kamera használható a beültetett panel ellenőrzéséhez is ami történhet manuálisan egy kezelő által vagy automatikusan mesterséges intelligenciával.

2.6. Vezérlő számítógép

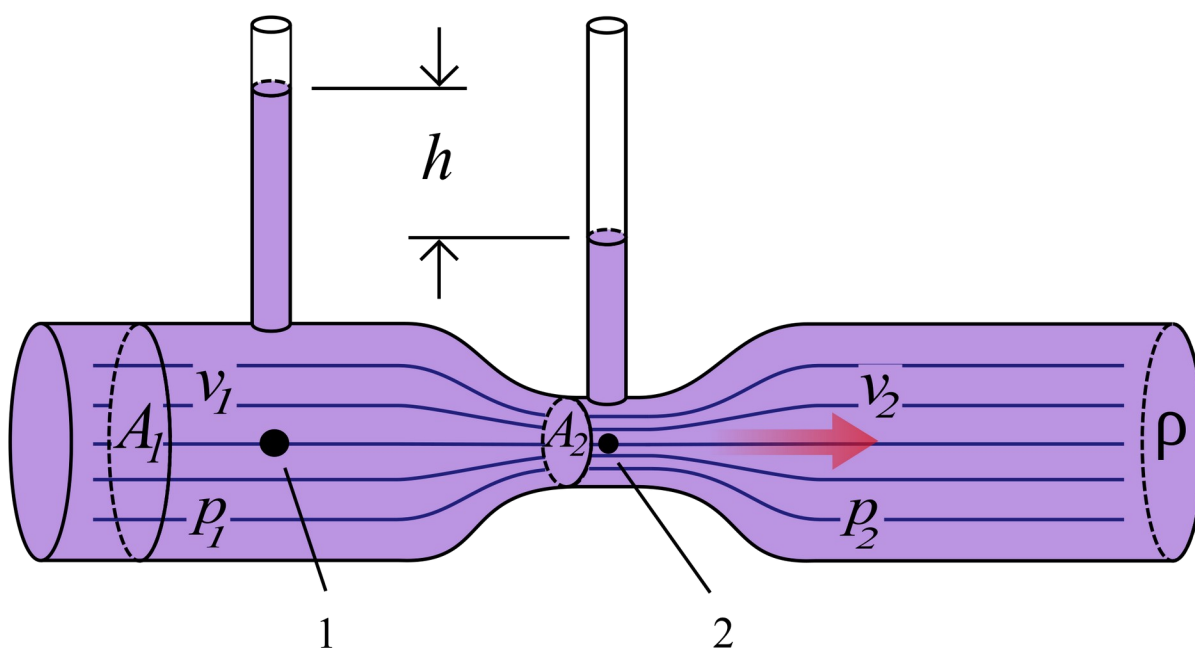
A gyors és egyszerű konfiguráláshoz érdemes egy grafikus felület megjelenítésére alkalmas számítógép. Egy modern általános számítógép erre alkalmas, nem kell speciális hardvert tervezni.

3. Logikai rendszerterv

3.1. Mechanika

Az iparban használatos gépek váza öntöttvasból szokott készülni a jó rezonancia csillapító hatása és tömege miatt. Mivel itt erre nincs lehetőség, így alumínium extrudált profilt választottam a gép vázának.

Az iparban a vákuumot sűrített levegő szolgáltatja, amit egy Venturi csövön átfújva állítják elő az alacsonyabb nyomást.



6. Ábra: A Venturi hatás szemléltetése

Ebben a megvalósításban nem áll rendelkezésre sűrített levegő, ezért más megoldást kell találni. Egy megoldás lehet olyan légszivattyút venni, aminek mindkét oldala ki van vezetve valamilyen porton. Ezt egy útválasztó szelepre kötve megoldható, hogy nyomás vagy vákuum legyen a kimeneten.

3.2. Elektronika

Az elektronika négy fő részre osztható:

1. Léptető motor vezérlő
2. Vákuum és nyomásmérő
3. Vezérlő számítógép
4. Feeder

A rendszer egy darab 24 voltos tápegységről kell üzemeljen. Minden egyéb feszültséget ebből kell előállítani. Teljesen felesleges 12 voltos és 5 voltos tápegységeket beszerezni ezzel növelve a gép árát és komplexitását. A teljesítményt úgy kell kiválasztani, hogy túlmelegedés nélkül ellássa a teljes gépet, lehetőleg passzív hűtéssel, hogy minél hosszabb élettartama legyen.

3.2.1. Léptető motor vezérlő

A legtöbb vezérlő egyszerűen egy STEP és egy DIR lábon keresztül teszi lehetővé a motor forgatását. A STEP lábon egy felfutó él egy lépést okoz a motoron. Az irányt a DIR láb szintje határozza meg. A motoráramot külső söntellenállás, az elfordulás szögét pedig lábak fel vagy lehúzásával lehet beállítani. Ezekkel a megoldásokkal nincs probléma viszont léteznek UART vagy SPI interfészen beállítható IC-k, amikkel nem kell a megfelelő STEP jelek kiadásával bajlódni. Ezeknél arra sem kell figyelni, hogy a tekercsáram a motoron szinuszos legyen, ami a halkabb működéshez előnyös. Ebben a megvalósításban egy ilyen vezérlőt érdemes használni.

3.2.2. Vákuum és nyomásmérő

A nyomás értékének mérésére nem feltétlen van szükség. Elegendő lenne egy előre beállított nyomáskapcsoló, ami jelzi, ha elértük a megfelelő értéket. Ez azonban csak akkor használható, ha pontosan tudjuk mekkora vákuum kell az alkatrész tartásához, valamint ezeknek a hiszterézisével is számolni kell, ezért valamilyen analóg vagy digitális MEMS technológiával készült szenzort érdemes használni. Szerencsére ezek a szenzorok elég elterjedtek, vérnyomásmérőkben gyakran alkalmazzák ezeket.

3.2.3. Vezérlő számítógép

Vezérléshez mindenképpen valamilyen Linux futtatására képes számítógépet érdemes használni. Itt nincs jelentős különbség egy beágyazott számítógép (SBC) és személyi számítógép (PC) között. A választás szimplán kényelmi szempont szerint zajlott, mivel a beágyazott számítógépeken általában egyszerűbb CAN és RS485 interfészeket használni, ezért erre esett a választás. Ezeken általában GPIO kivezetéseken elérhetőek UART és SPI vagy CAN interfészek, amik a megvalósításhoz kellenek.

3.2.4. Feeder

A feeder a rendszer legkomplexebb része. Sok apró mechanikus része van vékony kábelezéssel. A könnyű cserélhetőség miatt egy robosztus csatlakozó kell rá, ami valamilyen rugós kontaktfelület vagy gyorscsatlakozó. Felhasználható ipari gépekből származó használt feeder, amik általában még sokáig képesek működni vagy a számtalan nyílt forráskódú projekt közül valamelyiket. Az utóbbi opció előnye, hogy ezek gyakran egyszerűen beszerezhető vagy előállítható részeket használnak, ezért a karbantartás és fejlesztés könnyebben megoldható.

3.3. Szoftver

A projekt legösszetettebb része a szoftver. Meg kell oldani a kamerák képének feldolgozását, a PnP fájl beolvasását és feldolgozását, a CAN üzenetek küldését, fogadását, a grafikus interfész reszponzív működést, a feeder-ek vezérlését. Ezt egy monolitikus programmal nehéz megoldani, ezért érdemes több részre szétbontani.

A feladat felbontható a következő részekre.

1. Kamera kezelés és képfeldolgozás
2. Feeder vezérlés és státusz olvasás
3. Tengely mozgás vezérlése
4. Vákuum és nyomásmérés
5. Grafikus felhasználói felület

Mivel a feladatok végrehajtása teljesen szekvenciális, lehetőség van egyszerűsíteni a szoftvert. A kamerákat nem kell folyamatosan olvasni, hiszen semmi értelme egy üres képet elemezni. Amikor az alkatrészt felveszi a fej és megkapja a parancsot az alkatrész kamerához mozgásról, elég akkor egy képet csinálni, mikor odaért és utána elvégezni a korrekciót.

Természetesen szoftver nem csak a számítógépen fut. A mikrovezérlőn futó szoftver is segíthet a hatékony működésben. Nem kell a motorvezérlő paneltől folyamatosan lekérdezni a pozíciót és ezzel feleslegesen terhelni a buszt, ehelyett úgy kell megírni a kódot, hogy a mikrokontroller küldjön egy üzenetet, amikor odaért a pozícióra.

4. Fizikai rendszerterv

4.1. Mechanika

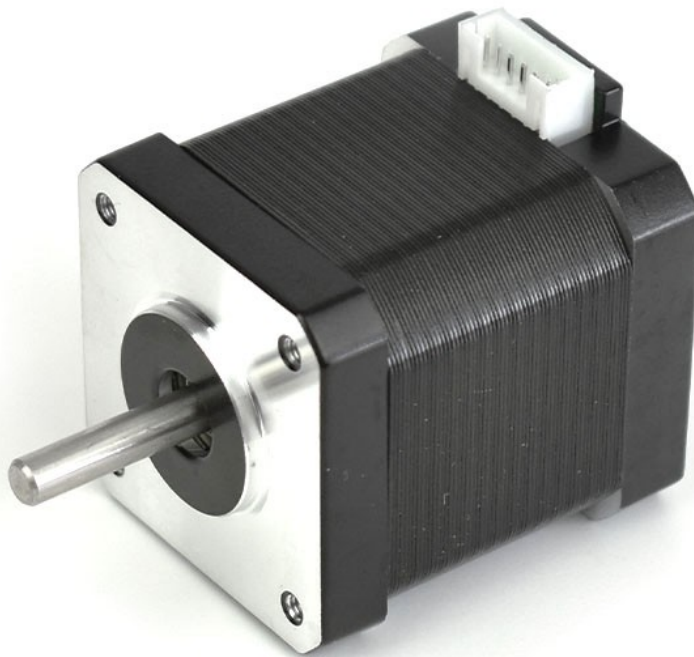
A gép váza 2020 alumínium extrudált profilból készült, amire lineáris vezetékek kerültek. A gép munkatérét ezeknek a hossza szabja meg ami ebben az esetben 750mm lett. Sokkal kisebbet nem érdemes tervezni mivel a beültetendő panelen kívül a tálcás alkatrészeknek is be kell férnie a munkatérbe.

A felvételhez vákuum, a lerakáshoz nyomás kell, így valószínűbb, hogy tiszta marad a fej. Ez megoldható 1 pumpával, aminek a mindkét oldala ki van vezetve. Egy 5/2-es útválasztó szelephez kötve elektromosan vezérelhető, hogy a beültető fejnél vákuum vagy nyomás van. A levegő bekötéshez 4mm átmérőjű átlátszó pneumatika csövet használok, amiben talán látható lesz a szennyeződés. Nyomás szabályzáshoz egy egyszerű fojtószelepet használok, hogy elkerüljem a nyomásmérő túlterhelését.



7. Ábra: 5/2-es útválasztó szelep

A tengelyek hajtását NEMA17 stepper motor végzi, ezeket gyakran használják 3D nyomtatókban ezért széles körben elérhetőek és olcsók.



8. Ábra: NEMA17 stepper motor

Ilyen standardizált elemekkel jó dolgozni, mivel sokféle rögzítő keret, csavar és kábel elérhető több forrásból.

4.2. Elektronika



9. Ábra: 24 voltos fémházas tápegység

Tápegységnek egy 200 wattos tápot választottam, ami több mint a szükséges, viszont ebben már van aktív PFC, ami csökkenti a hálózat terhelését és lehetővé teszi a továbbfejlesztést.

4.2.1. Léptető motor vezérlő

Motorvezérlőnek TMC5130-at[1] választottam az Analog Devices-től. Ennek az IC-nek kiváló paraméterei vannak erre a feladatra nézve. Rendelkezik végállás kapcsoló és enkóder interfésszel, a motoráram regiszterek segítségével szoftveresen állítható akár üzem közben is. Szabványos SPI interfésszel rendelkezik, amin minden paraméter állítható és státusz információ is kiolvasható.

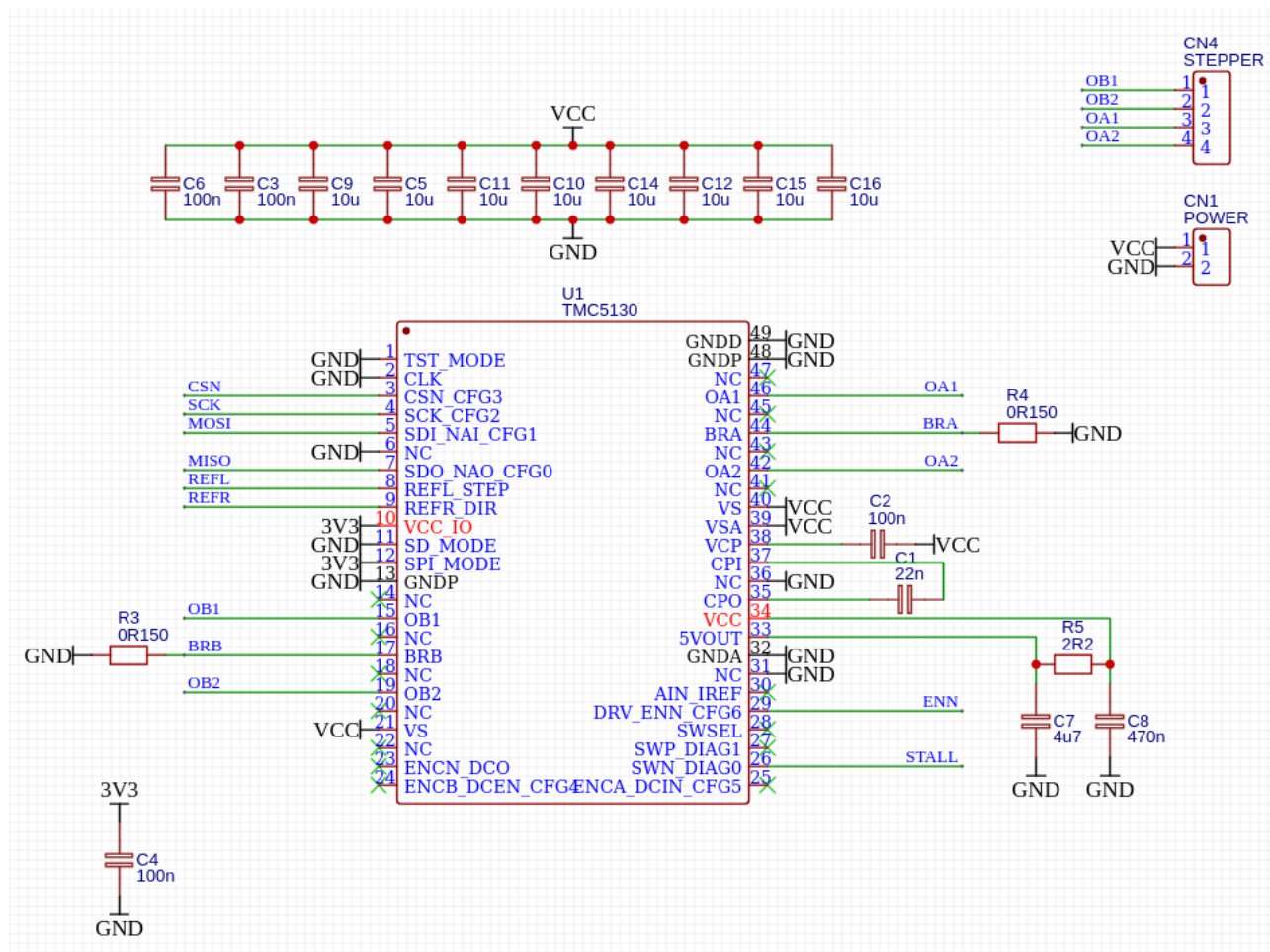
Mivel az IC tartalmazza a meghajtáshoz szükséges teljesítmény hídkapcsolást, a legfontosabb méretezési döntés az árammérő sönt kiválasztása volt.

CHOICE OF R_{SENSE} AND RESULTING MAX. MOTOR CURRENT		
R_{SENSE} [Ω]	<i>RMS current [A]</i> (CS=31, vsense=0)	<i>RMS current [A]</i> (CS=31, vsense=1)
1.00	0.23	0.12
0.82	0.27	0.15
0.75	0.30	0.17
0.68	0.33	0.18
0.50	0.44	0.24
0.47	0.47	0.26
0.33	0.66	0.36
0.27	0.79	0.44
0.22	0.96	0.53
0.15	1.35	0.75
0.12	1.64	0.91
0.10	1.92*)	1.06

*) Value exceeds upper current rating.

10. Ábra: Söntválasztás tekercsáram alapján

A motor egy szabványos NEMA17 stepper ami 1,2 amper maximális tekercsáramot bír, így az adatlapban található táblázatból a 0,15 ohmos ellenállásra esett a választás. Az áram finomhangolását az IRUN regiszter értékének beállításával lehet elvégezni. Mikrovezérlőnek RP2040-et[2] választottam a Raspberry Pi Foundation-től. Ennek fő okai a kiváló szoftveres támogatás és az IC elérhetősége. Minden szükséges interfésszel rendelkezik, ami a megvalósításhoz kell.

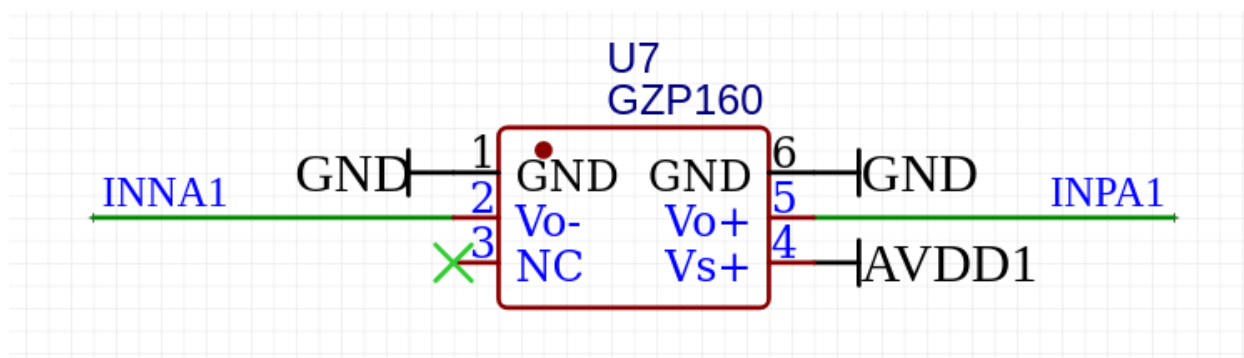


11. Ábra: A sztepper vezérlő bekötése

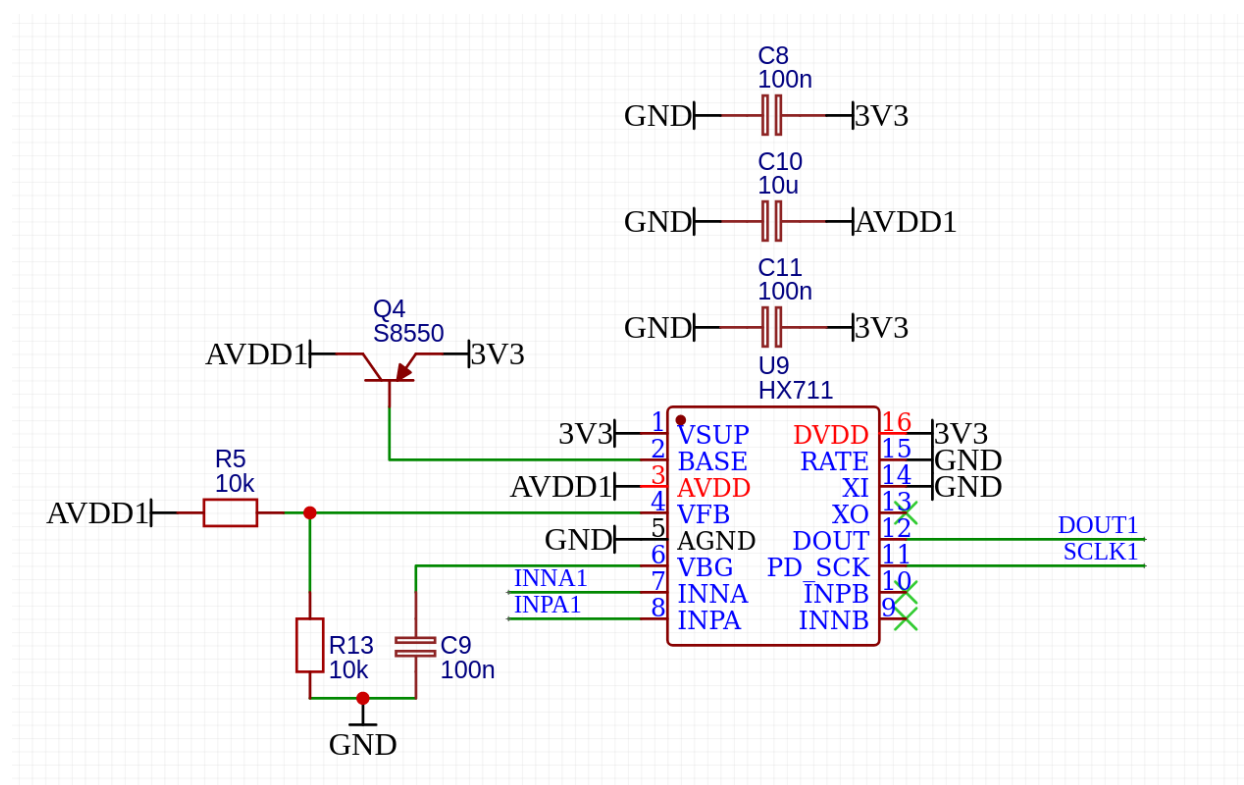
A motorvezérlőn bekötésre került az *ENN* engedélyező jel és a *STALL* mechanikai túlterhelés jelző jel is. Az *ENN* jel szükséges hibaüzenetek jelzőbitjeinek a törléséhez és a vezérlő engedélyezéséhez is. A *STALL* jel egy állítható érzékenységű mechanikai túlterhelést jelző kimenet, ami használható végállás kapcsoló nélküli home kereséshez vagy túlterhelés esetén vészleállításhoz.

4.2.2. Vákuum és nyomásmérő

Ez az áramkör felelős a szivattyú vezérlésért és a kialakult nyomás mérésért. A nyomást egy GZP160[3] abszolút nyomásmérő méri, amit egy HX711[4] 24 bites hídmérő IC illeszt SPI buszra. Ehhez a felhasználáshoz ez a megoldás megfelelően lineáris, így egy egyszerű kétpontos kalibrációval megkaphatjuk a nyomást pascal mértékegységben. A mikrokontroller itt is egy RP2040.



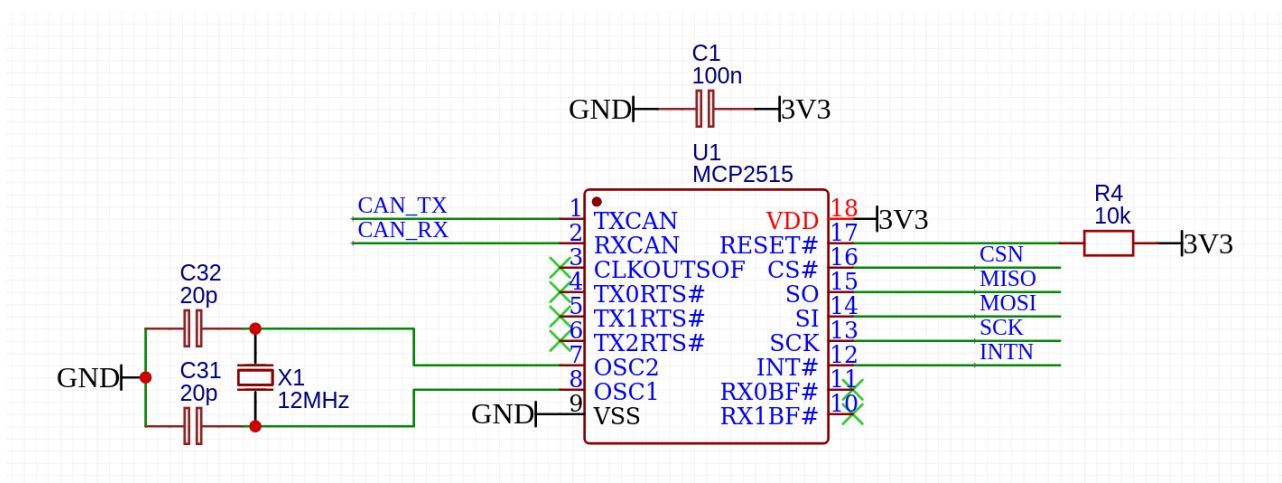
12. Ábra: GZP160 bekötése



13. Ábra: HX711 bekötése

4.2.3. Vezérlő számítógép

A szoftvert egy Raspberry Pi 4 futtatja. A rendszer többi részével egy HAT kompatibilis NYÁK biztosítja a kapcsolatot, ami 5 voltos tápfeszültséget is ad a Pi 4-nek. CAN kommunikációhoz egy MCP2515[5] CAN SPI bridge és SN65HVD232[6] biztosítja az interfészt. A feeder-ekkel történő kapcsolatot RS485-ön keresztül egy SP3485[7] adja. Ezek az IC-k 3,3 voltos tápfeszültségről működnek ellentétben az általánosabb társaikkal, amiknek 5 volt kell. Azért esett ezekre a választás mert így nem kell szint illeszteni a logikai jeleket.



14. Ábra: SPI CAN bridge bekötése

4.2.4. Feeder

Feeder-nek egy nyílt forrású projektet használtam. Az Opulo által fejlesztett feeder-ek 8mm széles szalaghoz készültek, nyílt forrású protokollal kommunikálnak és alapos teszteknek vetették alá így bátran használható ilyen projektben.

Az Y tengelyre merőlegesen elhelyezett alumínium profilokon OneWire EEPROM-ok vannak egy kis NYÁK-ön, erre csatlakoznak a feeder-ek így automatikusan felvevődnek a szoftverben, ahol már csak az alkatrészt kell hozzá társítani.



15. Ábra: Opulo nyílt forrású feeder

4.3. Szoftver

A rendszer elemeinek összehangolásához szükség van egy programra, ami képes mindent vezérelni, hibákat észlelni és ezekre reagálni.

A programot több logikai blokkra lehet bontani feladat szerint:

1. Kamera kezelés és képfeldolgozás
2. Feeder vezérlés és státusz olvasás
3. Tengely mozgás vezérlése
4. Vákuum és nyomásmérés
5. Grafikus felhasználói felület

A beültetés folyamata is jól meghatározott lépésekből áll

1. Home megkeresése
2. Alkatrész pozícióra mozgás
3. Alkatrész felvétele – Z tengely le, vákuum be, nyomás ellenőrzés, Z tengely fel
4. Ha feeder-ből vette ki új kiadagolása, ha tálcából hely ürítése
5. Alkatrész kamera fölé mozgás – Rotáció és helykorrekció, R tengely forgatás
6. Korrigált ültetési pozícióra mozgás
7. Alkatrész lerakása – Z tengely le, vákuum ki, nyomás be, nyomás ellenőrzés, Z tengely fel
8. Vissza a 2. ponthoz

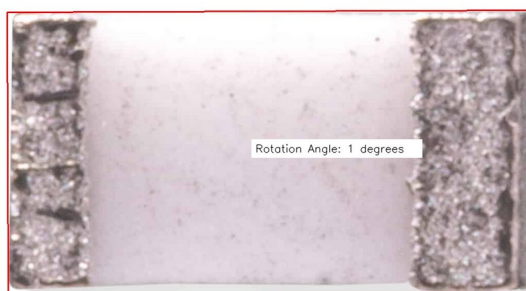
Ez a legegyszerűbb folyamat egy alkatrésszel, de fel kell készülni sikertelen felvételre vagy lerakásra, amit okozhat eltömődött cső, hibás szelep, pumpa. Ha több beültető fej van, több alkatrészt is felvehet, akár ugyanabból a feeder-ből.

4.3.1. Kamera kezelés és képfeldolgozás

A képfeldolgozáshoz OpenCV[8]-t használtam, ami rendelkezik egy függvénnyel, ami tökéletesen fedí az igényeit a feladatnak.

A `cv2.minAreaRect` visszatérési értéke [*center(x, y), (width, height), angle of rotation*]

A center koordináták használhatóak a pozíció korrekcióhoz, a rotation a beültetési szöghöz. A szélesség és magasság paraméterekkel ellenőrizhető az alkatrész helyes mérete, mivel a kamera fókusztáv állandó és minden alkatrész ugyanabban a magasságban lesz.



16. Ábra: Képfeldolgozás tesztelése

4.3.2. Feeder vezérlés és státusz olvasás

A feeder-ek a vázon elhelyezett EEPROM-okból tudják melyik címen vannak, amit kiolvasva belőlük a szoftver pozícióhoz társít. A felhasználói interfészen megadható, hogy melyik feeder-ben milyen alkatrész van, így az ültetési fájlból egyből hozzárendeli a sorokhoz. A feeder-en Photon[9] nyílt forrású firmware fut, a kommunikáció RS485-ön történik.

4.3.3. Tengely mozgás vezérlése

A számítógép a CAN buszon a motorvezérlő regisztereit címzi egyből, így elég egy egyszerű filter a CAN üzenetnek, nem kell a mikrovezérlőnek feldolgoznia a parancsot. Ez azt is lehetővé teszi, hogy más vezérlőt használjak és csak a számítógépen futó programot kell módosítani.

A motorvezérlő minden SPI válaszban elküld 8 státusz bitet, ami tartalmazza a végálláskapcsolók állapotát, így megvalósítható mozgás közben egy vészstop funkció.

A vezérlő IC-vel való kommunikációt a Trinamic által kiadott API[10] segítségével valósítottam meg.

4.3.4. Vákuum és nyomásmérés

A HX711 hídmérő IC 24 bites A/D konvertert használ, viszont ez a maximális sebességnél, 80 minta másodpercenként, csak 16,8 zajmentes bitet szolgáltat. Erre a feladatra ez elegendő, mivel nem pontos nyomásmérésre van szükség, csak határérték szintek átlépését kell figyelni. Az SPI busz az RP2040-en képes ennek a meghajtására.

PD_SCK Pulses	Input channel	Gain
25	A	128
26	B	32
27	A	64

17. Ábra: Bemenet és erősítés választás a HX711-en

4.3.5. Grafikus felhasználói felület

A felhasználói felületen lehet megnyitni a pnp fájlt, ami az alkatrészek tokozását, típusát, helyzetét, rotációját, darabszámát tartalmazza. Itt lehet beállítani, hogy melyik feeder melyik alkatrészt tartalmazza, hol van a tálca és milyen alkatrész van benne. Lehet manuálisan mozgatni a gépet és általános paramétereket állítani, mint gyorsulás, sebesség és home-olást indítani. A kamerák képét is kijelzi, hogy látható legyen mit csinál a program. A beültetési folyamat bármikor leállítható vagy szüneteltethető, ami hasznos lehet a kifogyott alkatrészek újratöltéséhez.

5. Modulok leírása

A rendszer szoftveresen és hardveresen is több modulra bontható.

Hardver modulok:

1. Motor vezérlő
2. Nyomásmérő és szelepvezérlő
3. Feeder
4. Vezérlő számítógép és interfészei
5. Kamera

Minden modul saját mikrovezérlőt és programot tartalmaz, ami felügyeli a működését és biztosítja a kommunikációs interfészt. Minden egységen biztosítva van egy könnyen elérhető interfész frissítéshez.

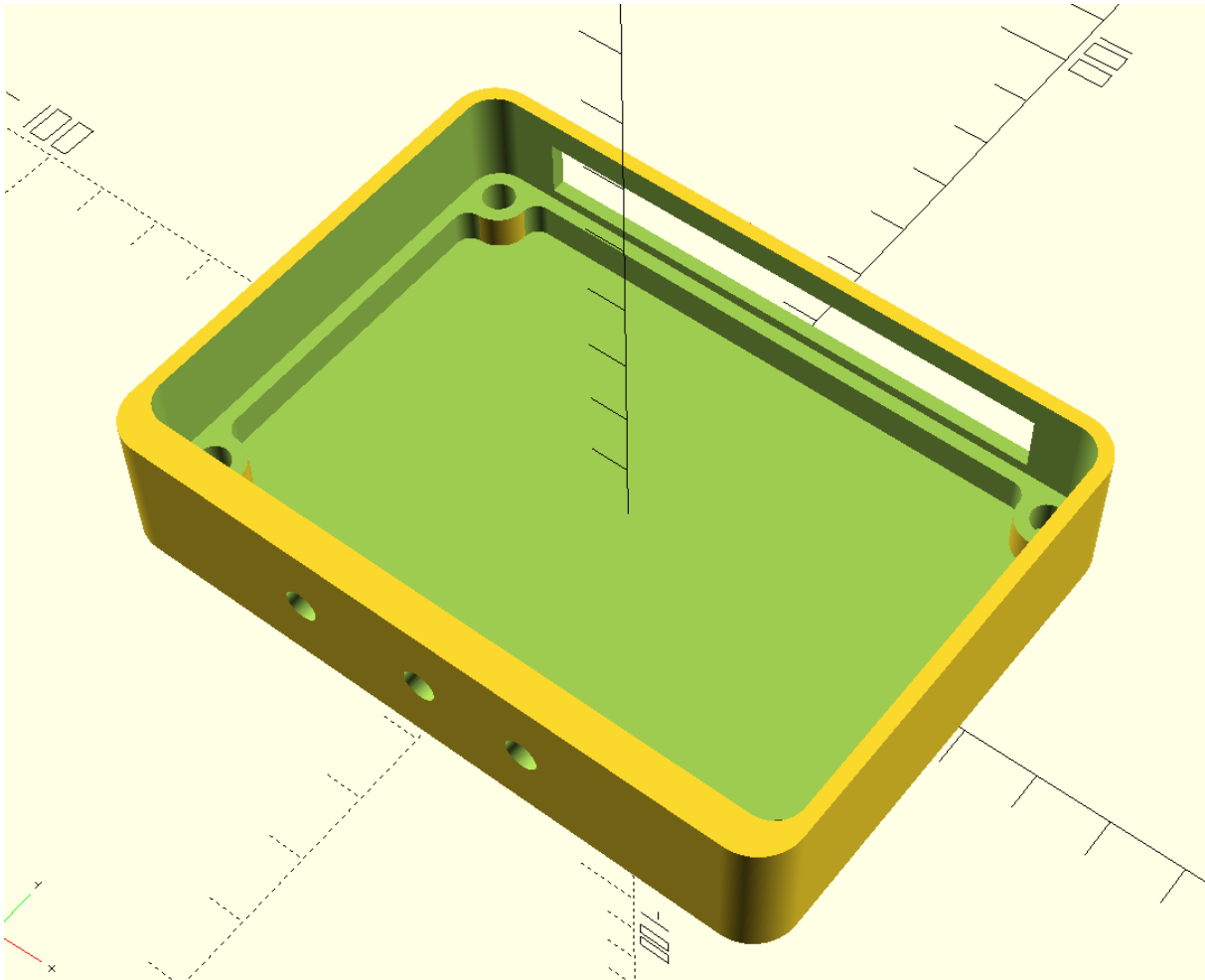
Szoftver modulok:

1. Képfeldolgozó
2. Ültetési fájl beolvasó
3. Feeder vezérlő
4. CAN kezelő (motor és vákuum)
5. Felhasználói felület

Minden egység önálló programrész, ami a saját működését felügyeli és hibával tér vissza, ha nem megfelelően üzemel.

6. Integráció

6.1. Mechanika



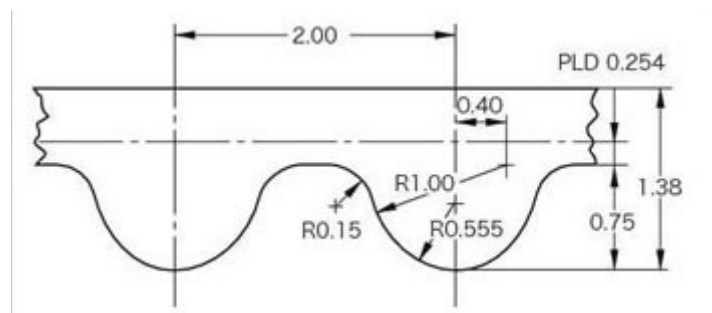
18. Ábra: A vezérlőknek tervezett doboz

A kész paneleket egyforma méretűre terveztem, így könnyen rögzíthetők dobozba. A dobozt kalapsínre rögzítve egymás mellé sorolva lehet rakni, mivel a csatlakozók egy oldalon vannak a kábelezés is egyszerű. A modell OpenSCAD szoftverben készült teljesen parametrikusan.



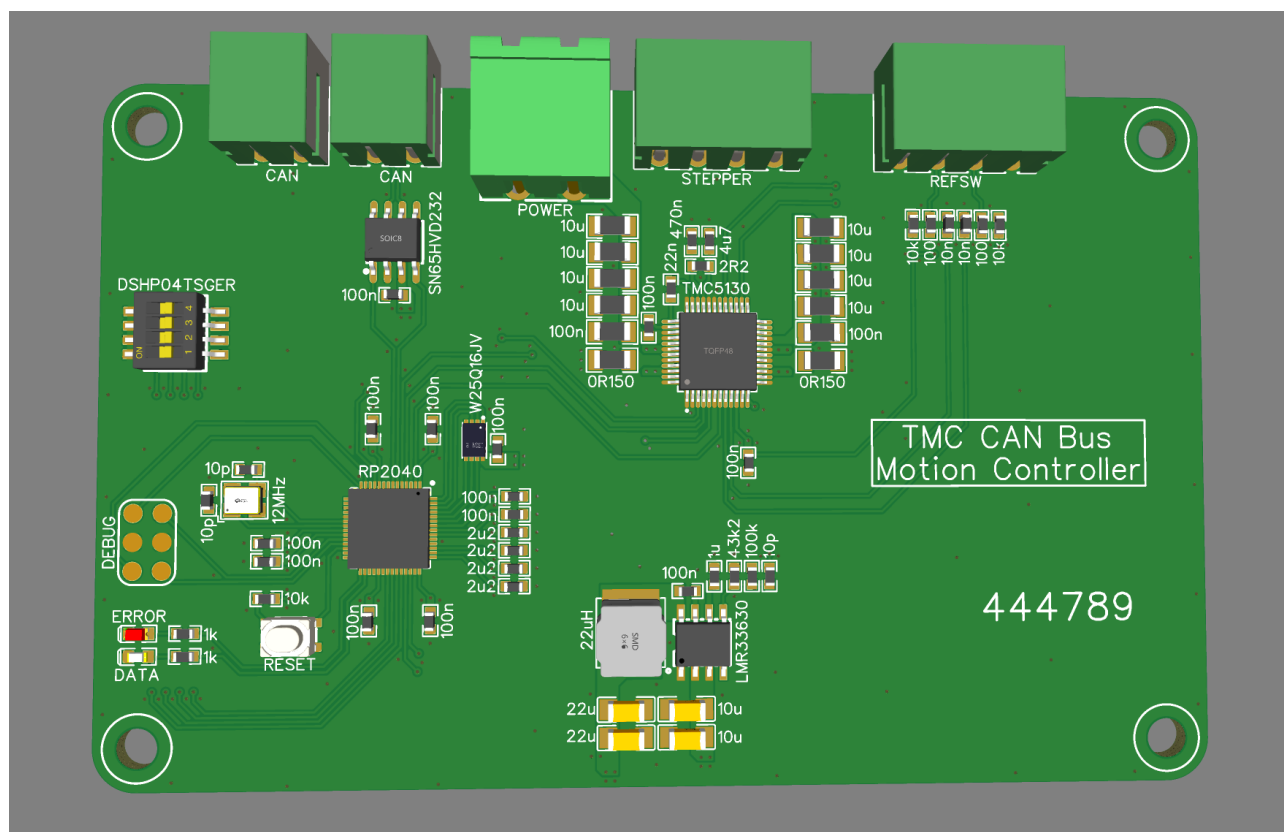
19. Ábra: Fogaskerék GT2 szíjhoz

Az erőátvitelt bordásszíz végzi, szintén a 3D nyomtatókon elterjedt GT2 szabványos szíjjal.



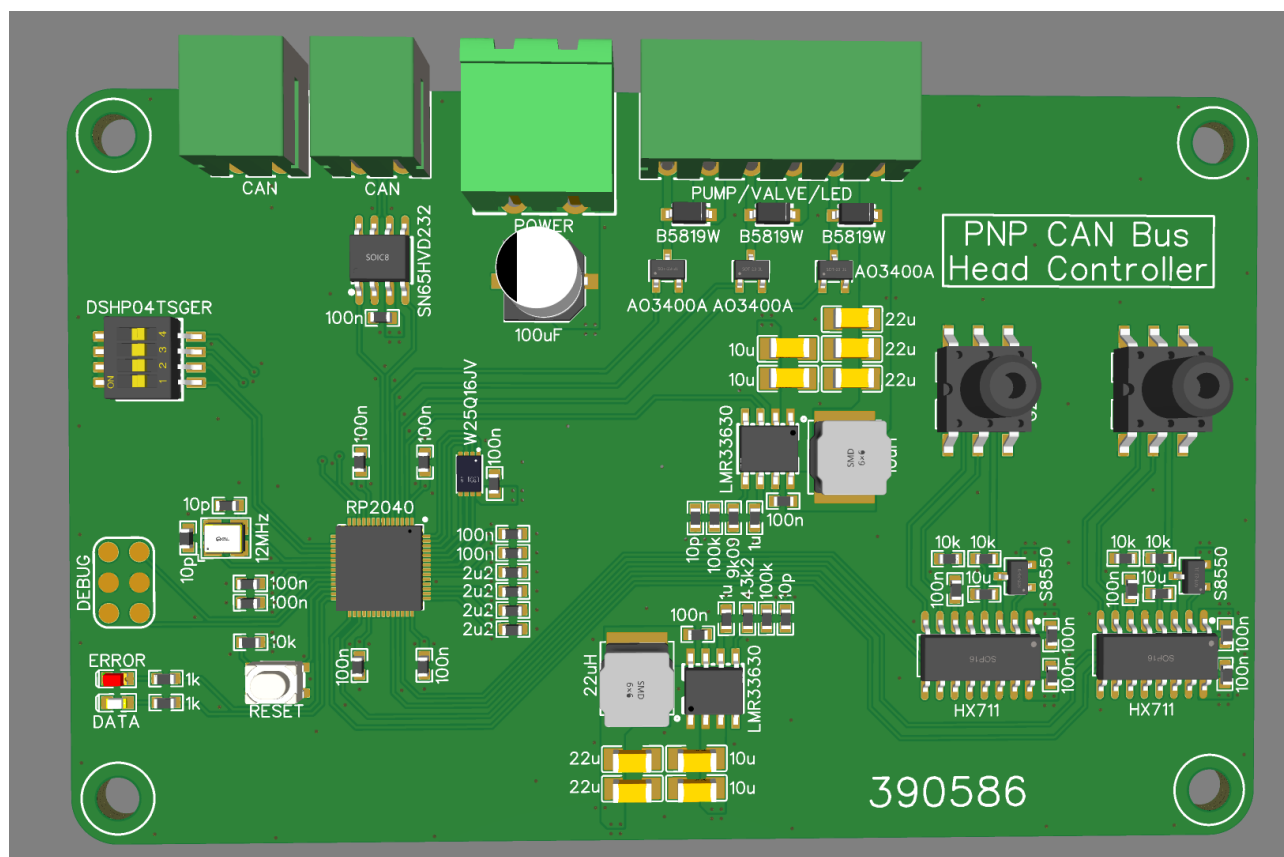
20. Ábra: GT2 szíz profilja

A jó minőségű szíjak valamilyen anyaggal meg vannak erősítve (üvegszál, kevlár, acélszál), hogy minimálisan nyúljon meg a használatból.



22. Ábra: Motorvezérlő panel

A motorvezérlőn minden könnyen bontható csatlakozót kapott. A CAN könnyű összeköthetősége érdekében két külön csatlakozót használok azonos bekötéssel. A tápcsatlakozó mérete és polaritása megegyezik minden panelen. A referencia kapcsoló bemenet alapból nyitott kapcsolóra számít, de ez szoftveresen állítható. A NYÁK 4 rétegű a disszipáció miatt, így teljesen passzívan működik. A panel 30 perc folyamatos üzem közben nem melegedett 70 °C fölé az IC alatt. A szélein, ahol a nyomtatott dobozzal érintkezik, 40 °C alatt maradt. A doboz anyaga PETG, ami 80 °C körül kezd el lágyulni.



23. Ábra: Pumpa és szelepvezérlő panel

A panel két fűvókás fejhez lett tervezve, így egyszerűbb a vezetékezés és a bővítés. A kimeneteket egy DC-DC konverter látja el 12 volttal, így a panel bemenetén a tápfeszültség változásra érzéketlen a működés. A tápbemenet és CAN itt is összhangban van a többi panellel. A CAN buszra felfűzött eszközökön nincs lezáró ellenállás, kivétel a Pi4-re kötött panel, ezért az összekötés után az utolsó NYÁK egyik CAN csatlakozójába kell kötni a lezárást. A nyomásmérő miatt kell egy derékszögű csődarab, hogy beleférjen a dobozba. Ennél a panelnél nem kell a melegedéssel számolni, ezért mindegy milyen anyagból készül a doboza.

6.3. Szoftver

A mikrokontrolleren futó szoftver C nyelven készült a Raspberry Pi Pico C/C++ SDK[11] felhasználásával. A program a CAN üzenet ID-je alapján eldönti, hogy a motorvezérlőnek kell-e továbbküldeni SPI-on vagy a mikrovezérlő dolgozza-e fel. A paneleken található címző kapcsolók állítják be, hogy milyen ID-re szűrjön a program.

A számítógépen Python nyelven írt szoftver fut. A főprogram kezeli a felhasználói interfész futtatását és az alprogramok szekvenciális meghívását. A programrészek tartalmaznak egyszer hibakezelést és időtúllépés figyelést, ha több próbálkozás után sem jön válasz hibaüzenetet ad vissza a program, amit egy hibaablakban jelez a program.

7. Tesztelés

A motorok meghajtásához használt 24 voltos tápfeszültség a vártnál kisebb gyorsulást és maximális sebességet tett lehetővé. A feszültség 36 voltig emelhető, amit a választott tápegység IC limitál. A nehéz beültető fej miatt, nagyobb motoráram kellett a hajtáshoz, ami miatt a motor magasabb hőfokon üzemelt.

A kamerakép megbízható eredményt szolgáltató működése sok kalibrációt igényelt, mind hardveresen a rögzítés igazításával, mind a szoftverben átlagolással.

A feeder-ek helyének pontos bevitele manuálisan történt. A fejet az alkatrész fölé mozgatva a szoftverből és az alkatrészt felvéve a kiírt aktuális pozíciót használtam feeder koordinátának.

8. Továbbfejlesztési lehetőségek

Elsősorban egy lineáris enkóder lehet a leghasznosabb fejlesztés. A tengelyekre rögzített enkóder fej a pontos elmozdulást adja, mivel nem a motor elfordulását figyeli, hanem direktben az adott tengelyét, sokkal kevésbé számítana a szíjak feszessége.

High Precision Magnetic Sensor



24. Ábra: 5 μ m felbontású mágneses enkóder

Ez az enkóder elektromosan olyan jelet szolgáltat, amit a motorvezérlő kezelni tud. Az ABN lábakra bekötve és a megfelelő regisztereket beállítva, μ m mértékegységben adná vissza a pozíciót.

9. Összefoglalás

Egy beültető gép kifejezetten komplex eszköz, sok rész együttműködését igényli a gyors és pontos működéshez. Egy iparban használt gép asztali szintre egyszerűsítése alapos megfontolást igényel és minden tervezési döntés számít. A modern elektronikai alkatrészek lehetővé tették, hogy egyszerűen megtervezett áramkörökkel megvalósítható legyen egy ilyen gép, de ez nem elég egy beültető géphez. Mechanikailag kell talán a legtöbb kompromisszumot kötni. Az öntöttvas szerkezetet nehéz helyettesíteni, ami hasonlóan jó élettartamot tud hozni. Az alumínium extrudált profilok könnyen szerelhetők, de idővel a keményebb acél elemek a rezgések miatt elgyengítik. Az acél zártszelvény jó megoldás, de ahhoz hegesztés, csiszolás és sok más kell, mindemellett nehéz is, ami a mozgatót problémássá teheti. A projekt legnagyobb része, a sok mechanikai fejtörés mellett, a szoftveres problémák voltak. A megvalósítás minden részéhez kell valamilyen program, ami kommunikál vagy vezérel valamit. Ezeknek az összehangolt működésére van szükség jól definiált interfészekre, az nem elegendő, hogy minden működik önállóan valahogy. A modern gépeknek a célja az egyre gyorsabb mozgás és képfeldolgozás stabil működés mellett. Ennek a feladatnak köszönhetően tapasztaltam, hogy ennyi rész összehangolása stabil működéssel önmagában elég nehéz, így érthető, hogy egy ipari gép fejlesztéséhez szükséges idő és erőforrásigény miért olyan nagy.

10. Conclusion

A Pick and Place machine is a highly complex device that requires the collaboration of many parts for fast and accurate operation. Simplifying a machine used in an industry to a desktop level requires careful consideration, and every design decision matters. Modern electronic components have made it possible to implement such a machine with easily designed circuits, but this alone is not sufficient for such a machine. Most of the compromises need to be made in the mechanical design. It is challenging to substitute the cast iron structure for something that can provide a similarly good lifespan. Aluminum extruded profiles are easy to assemble, but over time, the harder steel elements weaken it due to vibrations. Steel closed sections are a good solution, but welding, grinding, and much more are needed, and it is also heavy, which can make moving it problematic. The majority of the project, alongside many mechanical challenges, were software problems. Each part of the implementation requires some program that communicates with or controls something. Coordinated operation of these programs requires well-defined interfaces, it is not enough for everything to work independently. The goal of modern machines is faster movement and faster image processing with stable operation. While designing this machine, I have experienced that coordinating so many parts for stable operation is inherently difficult. Therefore, it is understandable why the time and resource requirements for the development of an industrial machine are so significant.

Felhasznált irodalom

- 1: TRINAMIC Motion Control GmbH, TMC5130A-TA DATASHEET, 2023. 08. 15.
- 2: Raspberry Pi Ltd., RP2040 Datasheet, 2022. 12. 15.
- 3: Wuxi Sencoch Semiconductor Co., LTD, GZP160 Datasheet, 2023. 02. 25.
- 4: AVIA Semiconductor, 24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales, 2022. 11. 11.
- 5: Microchip Technology Inc., MCP2515 Datasheet, 2021. 07. 22.
- 6: Texas Instruments, SN65HVD23x 3.3-V CAN Bus Transceivers, 2019. 06. 01.
- 7: MaxLinear, SP3485 Datasheet, 2021. 05. 08.
- 8: OpenCV, Contour Features, 2023. 10. 07.
- 9: Stephen Hawes, Justin Nesselrotte, Photon firmware, 2023. 03. 14.
- 10: TRINAMIC Motion Control GmbH, TMC-API, 2023. 02. 28.
- 11: Raspberry Pi Ltd., Raspberry Pi Pico C/C++ SDK, 2022. 09. 14.