Nyitrai Zoltán Patrik

Szakdolgozat

Budapesti Műszaki és Gazdaságtudományi Egyetem

Gépészmérnöki Kar

Mechatronika, Optika és Gépészeti Informatika Tanszék



Szakdolgozatok

Budapesti Műszaki és Gazdaságtudományi Egyetem

Gépészmérnöki Kar

Mechatronika, Optika és Gépészeti Informatika Tanszék

NYITARI ZOLTÁN PATRIK

Szakdolgozat

3D lézerszkenner tervezése

|  |  |
| --- | --- |
| Konzulens:  *Bojtos Attila*  Tanársegéd | Témavezető:  *Bojtos Attila*  Tanársegéd |

Budapest, 2019

Nyilatkozatok

*Beadhatósági nyilatkozat*

A jelen szakdolgozat az üzem/intézmény által elvárt szakmai színvonalnak mind tartalmilag, mind formailag megfelel, beadható.

Kelt,

Az üzem részéről:

*üzemi konzulens*

*Elfogadási nyilatkozat*

Ezen szakdolgozat a Budapesti Műszaki és Gazdaságtudományi Egyetem Gépészmérnöki Kara által a Diplomatervezési és Szakdolgozat feladatokra előírt valamennyi tartalmi és formai követelménynek, továbbá a feladatkiírásban előírtaknak maradéktalanul eleget tesz. E szakdolgozatot a nyilvános bírálatra és nyilvános előadásra alkalmasnak tartom.

A beadás időpontja:

*témavezető*

*Nyilatkozat az önálló munkáról*

Alulírott, *Nyitrai Zoltán Patrik* (AP4Z4I), a Budapesti Műszaki és Gazdaságtudományi Egyetem hallgatója, büntetőjogi és fegyelmi felelősségem tudatában kijelentem és sajátkezű aláírásommal igazolom, hogy ezt a szakdolgozat meg nem engedett segítség nélkül, saját magam készítettem, és dolgozatomban csak a megadott forrásokat használtam fel. Minden olyan részt, melyet szó szerint vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a hatályos előírásoknak megfelelően, a forrás megadásával megjelöltem.

Budapest, 20…

*szigorló hallgató*

Tartalom

[Előszó viii](#_Toc27644434)

[Jelölések jegyzéke ix](#_Toc27644435)

[1. Bevezetés 1](#_Toc27644436)

[1.1. Célkitűzések 1](#_Toc27644437)

[2. Szakirodalmi áttekintés ÉS Előzmények 2](#_Toc27644438)

[2.1. 3D szkennelési eljárások elemzése 2](#_Toc27644439)

[2.1.1. Fényforrás 2](#_Toc27644440)

[2.1.2. Kamera 5](#_Toc27644441)

[2.1.3. Mozgatás 6](#_Toc27644442)

[2.1.4. Kereskedelmi forgalomban lévő termékek 6](#_Toc27644443)

[2.2. Előzetes munka 8](#_Toc27644444)

[2.3. Szkennelés matematikai háttere 9](#_Toc27644445)

[2.3.1. Kamera matematikai modellje 9](#_Toc27644446)

[2.3.2. Kalibráció 12](#_Toc27644447)

[2.3.3. Szkennelés 14](#_Toc27644448)

[3. Feladatok, tervek 15](#_Toc27644449)

[3.1. Állvány 15](#_Toc27644450)

[3.2. Elektronika 15](#_Toc27644451)

[3.3. Szoftver 15](#_Toc27644452)

[4. prototípus tervezése, megvalósítása 16](#_Toc27644453)

[4.1. Mechanikus alkatrészek 16](#_Toc27644454)

[4.2. Elektronika 19](#_Toc27644455)

[4.2.1. Raspberry 19](#_Toc27644456)

[4.2.2. Asztal 22](#_Toc27644457)

[4.2.3. Firmware 24](#_Toc27644458)

[4.3. Állvány 29](#_Toc27644459)

[5. Képfeldolgozó szoftver 32](#_Toc27644460)

[5.1. Csatlakozás 32](#_Toc27644461)

[5.2. Kamerák 34](#_Toc27644462)

[5.3. Kalibráció 34](#_Toc27644463)

[5.4. Szkennelés 37](#_Toc27644464)

[5.5. Utómunka 39](#_Toc27644465)

[6. Eredmények 41](#_Toc27644466)

[6.1. Következtetések és javaslatok 44](#_Toc27644467)

[7. Felhasznált források 45](#_Toc27644468)

[8. Summary 46](#_Toc27644469)

# Előszó

Ez a dolgozat a

\* \* \*

Szeretném megköszönni Bojtos Attilának, hogy ellátott hasznos tanácsokkal munkám során és segédkezett az eszközök beszerzésében. Végtelenül hálás vagyok családomnak és barátaimnak biztatásukért és türelmükért.

Budapest, 2019.12.09

Nyitrai Zoltán Patrik

# Jelölések jegyzéke

|  |  |  |
| --- | --- | --- |
| Jelölés | Megnevezés, megjegyzés, érték |  |
| *A* | Kamera mátrix |  |
| *𝛼* | Pixel nyírása |  |
| *f* | Fókusztávolság |  |
| *T* | Transzlációs vektor |  |
| *R* | Rotációs vektor |  |
| *s* | Pixel skálafaktor |  |
| *H* | Homográfia |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
|  |  |

# Bevezetés

## Célkitűzések

Ez a dolgozat azért íródott, hogy bemutassa egy 3D lézerszkenner prototípustervezésének fázisait, az alapötletek mérnöki megvalósításának menetét és az eredmények kritikai elemzését.

Az iparban, egészségügyben és a művészet világában is találkozhatunk 3D szkennerekkel. Ezek különböző elven működnek attól függően, hogy milyen feladatot látnak el. Például egy épület digitalizálásához más berendezés szükséges, mint egy kisebb részletes alkatrész vagy az emberi test szkenneléséhez. Azonban alapvetően minden eszköz kamerából vagy kamerákból és legtöbbször valamilyen speciális fényforrásból áll. Kamerák által rögzíthető a tárgyról visszaverődő fény, ami hasznos információval rendelkezik annak alakjáról vagy megjelenéséről. Ez az információ a képekből képfeldolgozó szoftver segítségével nyerhető ki [1].

Egy 3D szkennerhez manapság viszonylag egyszerűen, elérhető áron hozzájuthatunk. Egy ilyen eszköz azonban összetettnek mondható tervezési szempontból, mivel (mint a legtöbb mai berendezés) működéséhez szükséges informatika, elektronika, valamint magának a szerkezetnek is el kell látnia a neki szánt feladatokat. A tervezett berendezés egy tavalyi mechatronika projekt nevű tárgy keretein belül elkészített szkenner folytatása. A projekt már több évfolyam kezén folyt keresztül és mire eljutott hozzánk használható és működőképes volt. Számunkra a mechatronika projekt elsődleges feladata a kétkamerás szkennelés megvalósítása volt. Emellett elvégeztünk néhány fejlesztést és kiküszöböltük néhány hiányosságát az eszköznek. A szkenner sajnos nem készült el olyan szinten, mint azt vártuk, ezért ennek a dolgozatnak a célja, hogy továbbfejlesszem és befejezzem a berendezést, valamint az eredmények elemzése az esetleges további munkálatok segítésére, lehetőségek és megoldások javaslása.

# Szakirodalmi áttekintés ÉS Előzmények

## 3D szkennelési eljárások elemzése

Első körben érdemes megvizsgálni, hogy az eddig használt szkennelési technika megfelelő-e, esetleg a már létező megoldások közül inspirálódni az eszköz fejlesztéséhez.

A piacon sok különböző 3D szkenner beszerezhető az asztali verziótól az egészen pontosnak mondható metrológiai rendszerekig. Így a termékek technikai adatait és tulajdonságait az határozza meg jelentősen, hogy milyen használatra szánják. A projekt szempontjából érdekes berendezések az asztali 3D szkennerek, amelyek közepes, nagyjából egy 300 mm magasságú és 200 mm átmérőjű henger munkatéren belül képesek nem túl kicsi tárgyakat szkennelni. Ezeket főleg hobbi célra szánják, de számos munkakörben hasznos lehet. Találkozhatunk ilyen eszközzel például dizájnerek, szobrászok asztalán, de többek között a fogtechnikában is nagy érdeklődést mutatnak az egyre pontosabb termékek iránt vagy akár reverse engineering céljára is használható a műszaki világban. Rengeteg lehetőség nyílik meg azokon a területeken, ahol szükség van egy test számítógépes modelljére, de a test bonyolult, CAD modellezési eljárással nehezen előállítható geometriákkal rendelkezik vagy időt szeretnénk megspórolni. Fontos, hogy az asztali verzió szorosan illeszkedik a 3D nyomtatás és gyors prototípus gyártás világába, ezért a nyomtató gyártók kínálják a legtöbb ilyen berendezést. Érdemes tehát a saját tervezésű eszköz által generált pontokat nem egyszerű vizualizációra használni, hanem olyan formátumra hozni, amivel a felhasználónak lehetősége van tovább dolgozni [1][2].

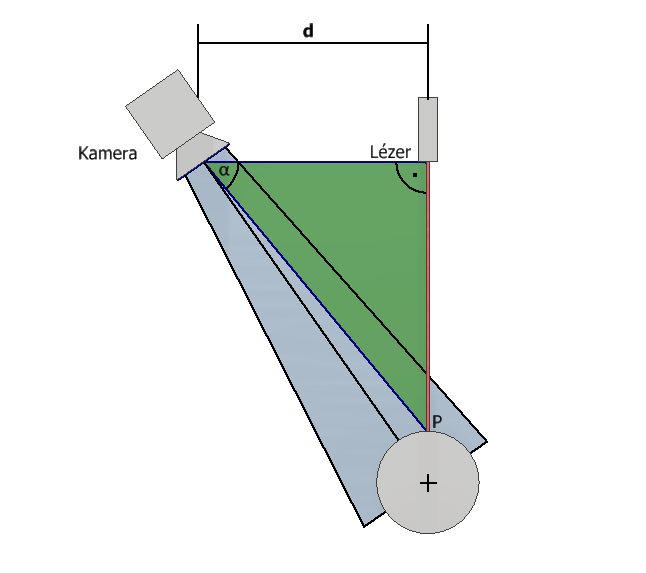
Minden asztali 3D szkenner alapeleme a kamera, a speciális fényforrás és legtöbbször van egy a mozgást megvalósító szerkezet. A különböző berendezések közötti különbségek ennek a három elemnek az eltéréséből adódnak, ezért ezeket célszerű külön-külön is elemezni. Érdemes megjegyezni, hogy létezik fotogrammetrián alapuló (3D szkennelésnek nem nevezhető) eljárás 3 dimenziós pontok meghatározására, ami a későbbiekben egy új irány lehet a saját tervezésű eszköznél is, mint extra funkció [4].

### Fényforrás

A leglényegesebb pont egy szkenner működésében a fényforrás típusa, mivel ez határozza meg, hogy milyen szkennelési eljárást alkalmaz az eszköz. Az asztali verziónál két típus terjedt el: a lézer és a strukturált (kódolt) fény.

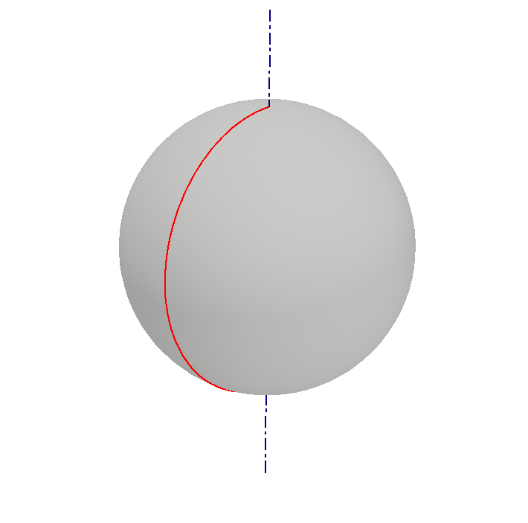
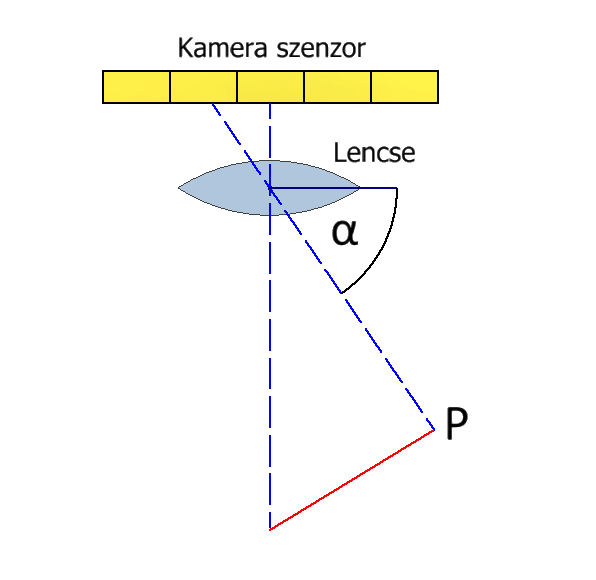
Az előbbi esetében egy vonal lézer vagy egy pásztázó pontlézer halad végig a test felületén és háromszögeléssel határozza meg a test felületének egyes pontjait.

2‑1. ábra: Lézerszkenner elvi vázlata



**e**

A háromszögelés elve az, hogy pontosan ismerjük a kamera és a lézer távolságát (amit az 2-1. ábrán **d** jelöl), valamint tudjuk, hogy a keresett **P** pont csak a lézer által kijelölt síkra eshet. Ezáltal a szkenner és a **P** pont távolsága kiszámítható, ha ismerjük az **e** egyenes egyenletét, amely az 2-1.ábrán 𝛂 szöggel tér el a kamerát és lézert összekötő egyenestől. Tehát a szkennelés közben ezt az **e** egyenest kell meghatározni a kamera segítségével minden felületi pont esetében, amely a vonallézer síkjának metszésében van. A kamerában található szenzor pixelei és az egyenes között egyértelmű kapcsolat van, ami a matematikai háttér kidolgozásánál kerül részletes elemzésre, de a 2-2. jobb ábra jól szemlélteti. A vonallézer szerepe tehát a meghatározandó pontok egyértelmű, a többi ponttól jól megkülönböztethető kijelölése a vetületi képen.



**Lézer síkja**

**e**

2‑2. ábra: balra egy gömbfelületre vetülő vonallézer, jobbra a kamera és a **P** pont kapcsolata

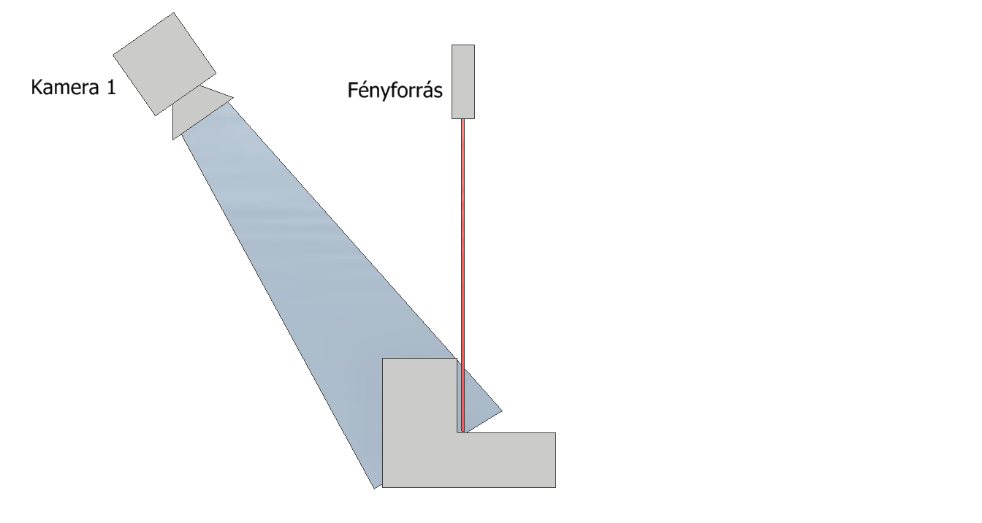
Strukturált fénnyel való szkennelés esetében is a háromszögelés módszerével dolgozik az eszköz, de a meghatározandó pontok kiválasztása a kamera képén nem lézerrel, hanem általában egy nagy felbontású projektorral kivetített minták segítségével történik. Ez a minta legtöbbször különböző frekvenciával váltakozó fekete fehér téglalapokból álló raszterháló (2-3.ábra) [3].

2‑3. ábra: Tárgyra vetülő raszterháló [3]

A kamera ennek a mintának a tárgyon való torzulását érzékeli és hasonlóan, mint a lézer esetében, az észlelt éleken található pontok távolságát meghatározza. Ezzel az eljárással pontosabb és gyorsabb szkennelés valósítható meg, mivel egyszerre több párhuzamos síkban jelöl ki pontot. Ez azt is eredményezi, hogy a képeket feldolgozó szoftver sokkal bonyolultabb, mint az előző esetben. Azt is érdemes szem előtt tartani, hogy egy nagy felbontású projektor általában egy vonallézer többszörösébe kerül és jóval nagyobb méretű [3].

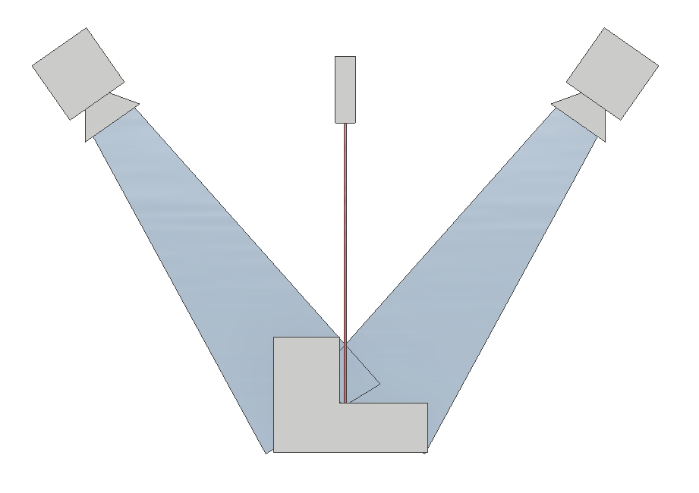
### Kamera

A kamera tekintetében a felbontás a legmeghatározóbb specifikáció, ami jellemzi a terméket, hiszen minél több pixelt képes érzékelni az adott pont közelében, annál kisebb hibával tudja közelíteni azt. Azonban nem érdemes feleslegesen túl nagy felbontású és drága kamerákat használni, mivel a legtöbb esetben a fényforrás pontatlansága vagy a felület minősége miatt a pontokat kijelölő fény és annak hiánya nem egy tökéletesen definiált határ.

 Az eddig ismertetett szkennelési módszereknek van egy hiányossága, ugyanis azokban az esetekben, amikor a tárgynak olyan alakja van, hogy a rávetülő lézert maga a tárgy kitakarja a kamera elől, a generált pontfelhőben lyukak keletkeznek.

2‑4. ábra: Kitakarás egy kamera és egy fényforrás esetén

A kitakarás könnyedén kiküszöbölhető két módon: ha a kamera két oldalára helyezzük a fényforrásokat vagy a fényforrás két oldalára helyezünk egy-egy kamerát.



2‑5. ábra: Kitakarás megszűnése két kamerával

Lézer esetében általában egy kamerát és két lézert helyeznek el. Projektornál fordítva gazdasági és helytakarékossági okokból. Korábban két kamerát és egy lézert kapott a projekt ezért ezt a megoldást érdemes használni.

### Mozgatás

Eddigiekben a szkenner a tárgy csak egy-egy szeletének körvonalát volt képes digitalizálni. Más helyzetekben is el kell végezni a szkennelést hogy egy teljes testfelületet kapjunk. Strukturált fénnyel való eljárásnál egy helyzetben képes a berendezés egy teljes felületszkennelést végrehajtani a fekete-fehér minta és annak frekvenciájának változtatásával. Ezeknél az eszközöknél sokszor elegendő kézzel más pozícióba helyezni a tárgyat, majd a különböző felületeket összeilleszteni a szoftver segítségével. A vonallézer ezzel ellentétben, mint azt a neve is mutatja nem képes 2 dimenziós minták kivetítésére, tehát finom eltérésekkel kell az egyes szeleteket szkennelésnek alávetni, majd ezeket összeilleszteni. Erre megoldásul szolgál ha a szkenner mozog a tárgyhoz képest, de a legpraktikusabb ha a tárgyat körbeforgatjuk. Ezt a forgóasztalos megoldást alkalmazza a legtöbb lézerszkenner és minél kisebb forgást hajt végre az asztal, annál részletesebb lesz a generált pontfelhő.

### Kereskedelmi forgalomban lévő termékek

A piacon jelenleg kapható szkennerek jelentős részét hobbi célra szánják ezzel ellentétben áruk magánszemélyeknek viszonylag magas. Azonban egy vállalkozás számára, ahol indokolt használatuk és nem szükséges a nagy pontosság, időt spórolhatnak meg és az eszközbe való befektetés hamar megtérülhet.

Néhány termék műszaki adatai:

Einscan SE from Shining 3D

|  |  |
| --- | --- |
| Típusa: | Strukturált fény |
| Pontosság: | 0,05 mm |
| Szkennelési idő: | kb. 1 perc |
| Szkennelési térfogat: | 570 x Ø210 mm |
| Ár: | 450-900E Ft |



EVIXMATIC EVATRONIX

|  |  |
| --- | --- |
| Típusa: | Strukturált fény |
| Pontosság: | 0.0130 mm |
| Szkennelési idő: | kb. 1,5 perc |
| Szkennelési térfogat: | 370 x 265 x 150 mm |
| Ár: | 10-50K $ |



Matter & Form V2 3D scanner

|  |  |
| --- | --- |
| Típusa: | Lézer |
| Pontosság: | 0.1mm |
| Szkennelési idő: | kb. 9 perc |
| Szkennelési térfogat: | 250 x Ø190 mm |
| Ár: | 450-700E Ft |



|  |  |
| --- | --- |
| Típusa: | Lézer |
| Pontosság: | 0.1mm |
| Szkennelési idő: | kb. 8 perc |
| Szkennelési térfogat: | 250 x Ø190 mm |
| Ár: | 600$ |

Scan Dimension SOL 3D scanner



Da Vinci 1.0 Pro 3-in-1 from XYZprinting

|  |  |
| --- | --- |
| Típusa: | Lézer |
| Pontosság: | 0,25 mm |
| Szkennelési idő: | kb. 8 perc |
| Szkennelési térfogat: | 150 x Ø150 mm |
| Ár: | 350-450E Ft |



## Előzetes munka

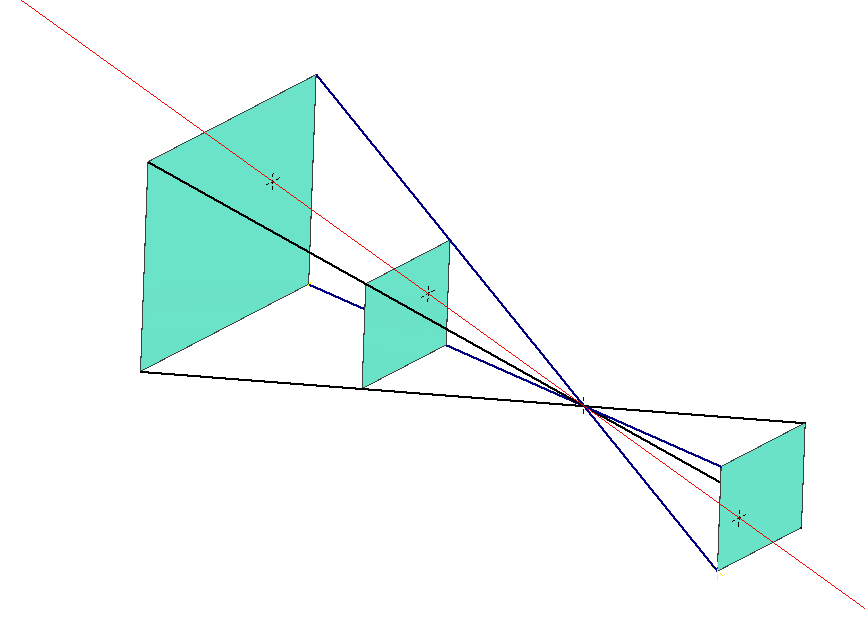
Ahogyan az már a bevezetésben említésre került ezen a projekten már mások is dolgoztak ez előtt, kiindulási anyagot biztosítva ezzel. A mechatronika projekt nevű csapatmunkát igénylő tárgy egyik feladata volt egy asztali 3D szkenner tervezése, amelyet a csapatunk kiválasztott. A feladatunk elsődlegesen az volt, hogy az előző projekteket folytatva készítsünk egy két kamerás lézerszkennert. Ehhez megkaptunk minden, addigra elkészített eszközt és programot. Így került hozzánk egy forgóasztal, egy fényképezőállvány, amelyen egy lézer és egy kamera volt rögzítve egy alumínium profilra, az elektronika vezérlésére használt arduino nano és a MATLAB kód. A forgóasztal egy léptetőmotorból, egy 3D nyomtatott asztallapból és egy fél léptetésre képes vezérlő áramkörből állt. Ez az összeállítás a dokumentációja szerint képes volt szkenneléshez ideális viszonyok között 1-2 mm pontossággal pontfelhőt generálni egy kamerával. Tehát kidolgoztak egy használható technikát a szkennelésre, ami transzformáláson alapuló technika. Az eljárás első lépéseként, egy képet készítettek egy ismert geometriájú kalibrációs mintát helyezve a vonallézer síkjába. Fontos, hogy a minta olyan legyen, amiről készült képen egy algoritmus könnyen megtalálja a jellegzetes pontokat. Nekik a kalibrációnál gyakran használt sakktábla mintára esett a választásuk, aminél az egyes négyzetek sarkait keresi meg a képen az erre készült algoritmus. Minden további a tárgy felületén meghatározandó pont távolságát ezekhez a jellegzetes pontokhoz viszonyítva állapították meg. A módszer nagy hátránya, hogy könnyedén lehet nagyon pontatlan a mérés, emiatt a viszonyítási pontok biztos, hogy számottevően eltérnek az ideálistól, valamint ezek a pontok csak egyszer lettek rögzítve, így az első körben rosszul rögzített minta miatt az egész szkennelés hibás lesz.

A projektfeladat alatt a csapatunk próbált javítani a módszer hibáin, ezért terveztünk egy mintatartó állványt, amit az asztal megfelelő furataiba lehet helyezni, majd ezt 3D nyomtatással el is készítettük. Továbbá az asztalt elláttuk egy műanyag burkolattal, amire rászereltünk egy hall szenzort annak érdekében, hogy az asztallapba helyezett mágnessel pontosan érzékelni lehessen egy a kalibráció során fontos pozíciót. Az asztalba egy ATTiny 2313 típusú mikrovezérlőt helyeztünk, hogy párhuzamosan tudjon működni az asztal forgatása, a hall szenzor érzékelése és a szkenner többi részének vezérlése. A már megírt kódot több helyen ezeknek megfelelően módosítottuk, egyes részeit újraírtuk, hogy a kalibráció automatikus legyen és két kamerával is alkalmazhatóvá váljon az eljárás. A két kamera által külön-külön rögzített pontfelhő összeillesztésének kidolgozására végül idő hiányában nem került sor.

## Szkennelés matematikai háttere

### Kamera matematikai modellje

A kamera modellezésének az a célja, hogy kapcsolatot teremtsen a 3 dimenziós pontok és az azoknak megfeleltetethető kép pontjai között. A kamera leírása első közelítésben a lyukkamera modell segítségével történik, ami alapján egy tárgy képe centrális vetítéssel keletkezik a kamera szenzorján fordított kép formájában. A kamera jellemezhető a virtuális képsíkkal, amelyen nem fordított kép keletkezik, de a szenzor minden egyéb meghatározó tulajdonságával rendelkezik.



Kamera szenzor

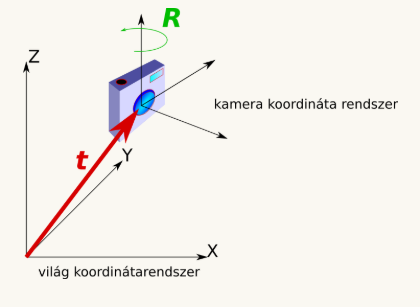
Virtuális képsík

Főpont

2‑6. ábra: Centrális vetítés

Elsődlegesen ismerni kell a kamera elhelyezkedését a térben, hiszen a tárgyról készült vetületi kép szempontjából lényeges, hogy a kamera milyen szögből, milyen távolról néz a tárgyra. A kamera lokális **(x,y,z)** koordinátarendszere és a világ **(X,Y,Z)** koordinátarendszer közötti transzformációt, egy **T** transzlációs vektor és egy **R** rotációs mátrix segítségével írhatjuk le, így minden térbeli **P** pontról megmondható, hogy hol van a kamerához képest, annak lokális koordináta rendszerben. Az eljárás szemléletesen az, hogy eltoljuk az egyik rendszer origóját a másikba, majd a tengelyeket forgatással fedésbe hozzuk, ezért ez egy 3D-ből 3D-be transzformálás.

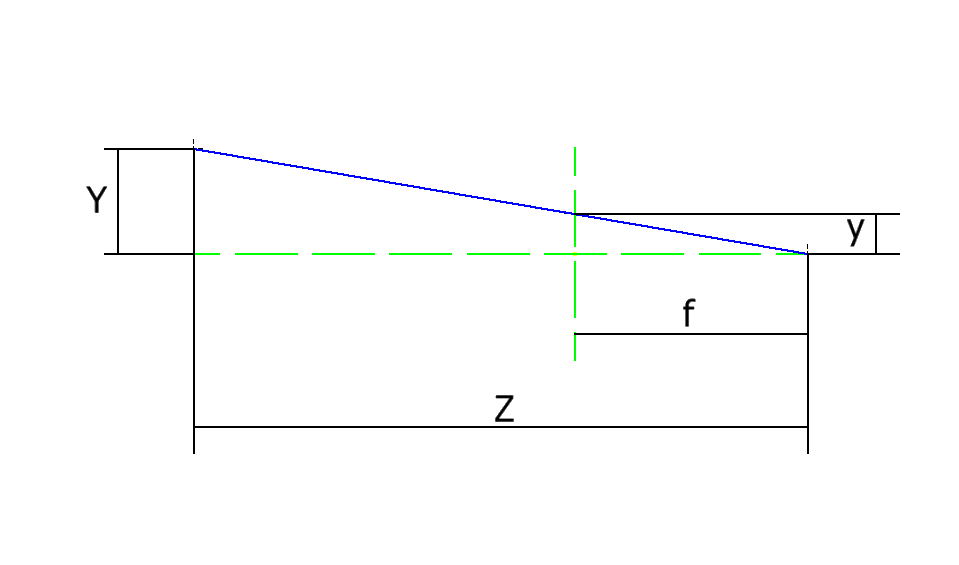
, ahol p a pont koordinátái a kamera rendszerében.



2‑7. ábra Külső paraméterek []

Ha ismerté váltak a külső paraméterek, akkor a vetítés tulajdonságait kell matematikailag kezelhetővé tenni, ez a kamera belső paramétereinek segítségével történik. Ebben az esetben a kamera által alkotott kép és az előzőleg említett lokális koordinátarendszer között szeretnénk kapcsolatot találni, ami egy 3D-ből 2D-be transzformálás. Első közelítésben a kamera lokális rendszerében a képsíkot kifeszítő két tengelyt a kép tengelyeivel páhuzamosra választjuk úgy, hogy az origók egybeesnek. Ebben az esetben a 2-8. ábrán látható, hogy egy 3 dimenziós pont Y koordinátája és a képen való vetületének y koordinátája között a hasonló háromszögek törvénye adja meg a kapcsolatot.

, ahol Z a pont képsíkra merőleges koordinátája és f a fókusztávolság. Ez az összefüggés felírható a pont X és a képen való vetületének x koordinátája között is.



Virtuális képsík

Főpont

3 dimenziós pont

2‑8. ábra: Vetített pont meghatározása

Az x és y a kép inhomogén koordinátái. Homogén koordinátákra áttérve felírható a leképezés mátrixos alakban, amivel könnyebben kezelhetőek a transzformációk.

A képek koordinátáinak () origója általában a kép egyik sarkában van, így ez az előző esethez képest egy eltolásként jelenik meg.

Valamint az egyes 3 dimenziós pontokat a kamerák csak diszkrét értékekre, pixelekre tudják leképezni, így a kép koordinátái nem lesznek folytonosak ez () skála faktorok segítségével írható le. Valamint a szenzorok tengelyei a pixelek nyírása miatt nem tejesen merőlegesek, ezeket az α nyírással lehet figyelembe venni.

Az előzőleg említett két transzformációt a vetítés mátrixával figyelembe véve megkaphatjuk az **A** kamera mátrixot, ami lyukkamera modell esetén leírja a 3 dimenziós helyvektorok transzformációját a kép síkjára.

Így a használt modell által megvalósított leképzés egylete:

### Kalibráció

„Az előzőekben láttuk, hogy egy kamera lényegében a mátrixával egyértelműen reprezentálható. Ha ismert a kamera mátrix, akkor egy tetszőleges pont képét elő tudjuk állítani, illetve bármely képpontot vissza is tudunk vetíteni. A gyakorlatban tehát fontos, hogy meg tudjuk határozni a kamera mátrixot.” [] A Kalibráció során meghatározandó a kamera paraméterei egy ismert geometriájú kalibrációs minta segítségével. A kamera képpontjai és a világ koordináták közötti kapcsolatot az eddig megismert transzformációk adják meg. Ha a választott világkoordináták XY tengelye feszíti ki a kalibrációs minta síkját a fenti mátrix egyenlet egyszerűsíthető.

Látható, hogy a képpontok és a minta pontjai között homográfia áll fenn. Így bevezethető a H homográfia mátrix:

, ahol egy tetszőleges skalár.

Így a leképezés egyenlete módosul.

, ahol az m a képpont koordináták a kép rendszerében és M a 3 dimenziós pont a világ rendszerében.

A forgómátrix ortonormáltsága miatt felírhatók a következők:

;

⊥→=0;

= 1 →=1;

Ebből következik, hogy:

A fentieket behelyettesítve:

Illetve;

Bevezetve a B mátrixot:

és elvégezve a műveletet:

=

Figyelembe véve, hogy B szimmetrikus felírható az alábbi egyenletrendszer:

, ahol []

b= []

A kalibrációs minta pozíciói mind más homográfiákat eredményeznek így minden minta két sorral bővíti a **v** mátrixot. Ezáltal a B elemeit meghatározva kifejezhetők az A mátrix elemei.

### Szkennelés

Az előzőleg bemutatott kamera mátrix meghatározása egyben alkalmas arra, hogy a lézer síkját is bekalibráljuk. Az első lépésben detektálásra kerülnek a kalibrációs pontok, amelyeket úgy állítunk be, hogy a megfelelő síkban legyenek, majd meg kell határozni a homográfiát. Ez egy olyan transzformáció lesz, amellyel később minden egyes kép a lézer síkjába projektálható és így mérhető a távolság.

# Feladatok, tervek

## Állvány

A kalibrálás és a szkennelés szempontjából is fontos részlet a végeredmény tekintetében, hogy az asztal és a lézer síkja egymásra merőleges legyen, valamint a lézer és a minta síkja egybeessen. Egy olyan állvány tervezése, ami biztosítja ezt növelheti a módszer pontosságát. Az állványnak emellett könnyen szétszedhetőnek kell lennie, hogy a szállítása és tárolása egyszerű legyen.

## Elektronika

A berendezés elektronikáját érdemes lenne kissé áttervezni és a csatlakozásokat egyszerűbben megoldani a jelenleginél. A szkenner kommunikációját a számítógéppel úgy kell átalakítani, hogy az lehetőleg csak egy kábel keresztül történjen.

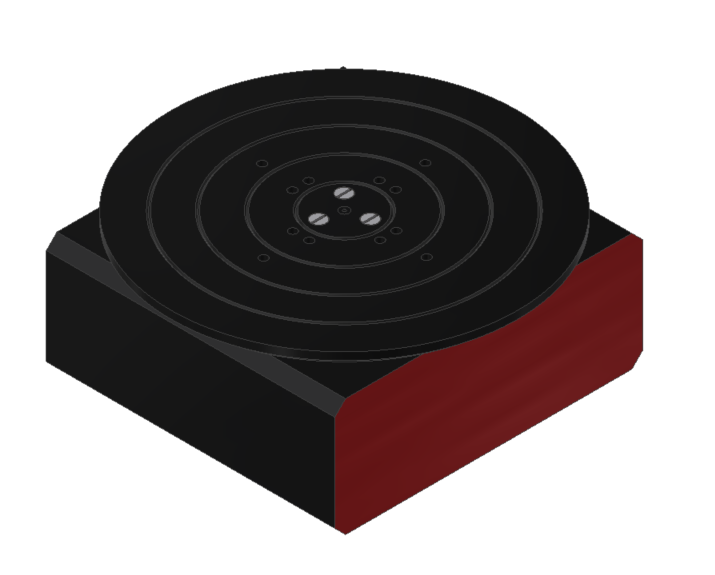
## Szoftver

A több kamera által generált pontfelhő összeillesztése fontos megoldandó feladat, hiszen a két kamera egyidejű szkenneléséből így lesz egy kitakarásmentes modell. Az egyszerűbb kezelhetőség érdekében hasznos volna egy ergonomikus felhasználói felülettel rendelkező számítógépes szoftver fejlesztése, első körben MATLAB nyelven. Ez a program később átíródhat olyan nyelvre, aminél nem szükséges a licenszköteles szoftver használata. Olyan kimenetet kell generálnia az eszköznek, amely más programok számára is használható, értelmezhető.

# prototípus tervezése, megvalósítása

Elsődlegesen az állványt és magát a szkennert érdemes elkészíteni úgy, hogy az előzőekben részletezett pontoknak megfeleljen. Ezek után a képfeldolgozó szoftvert a szkennerhez illeszteni.

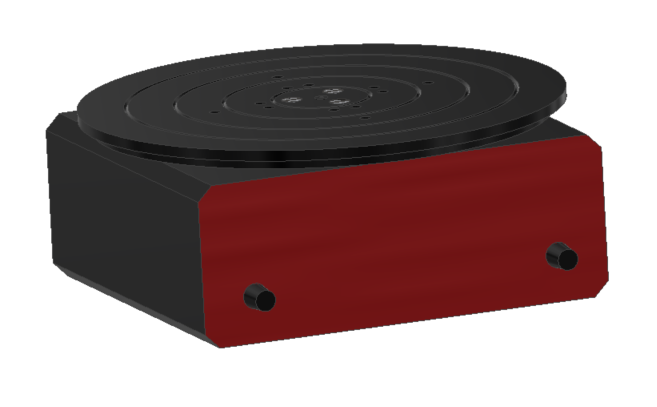
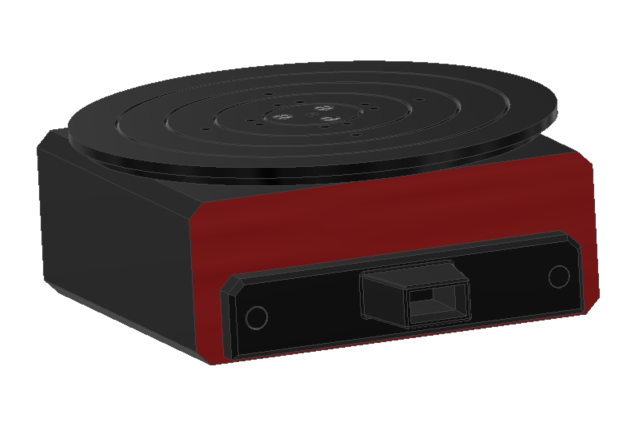
## Mechanikus alkatrészek

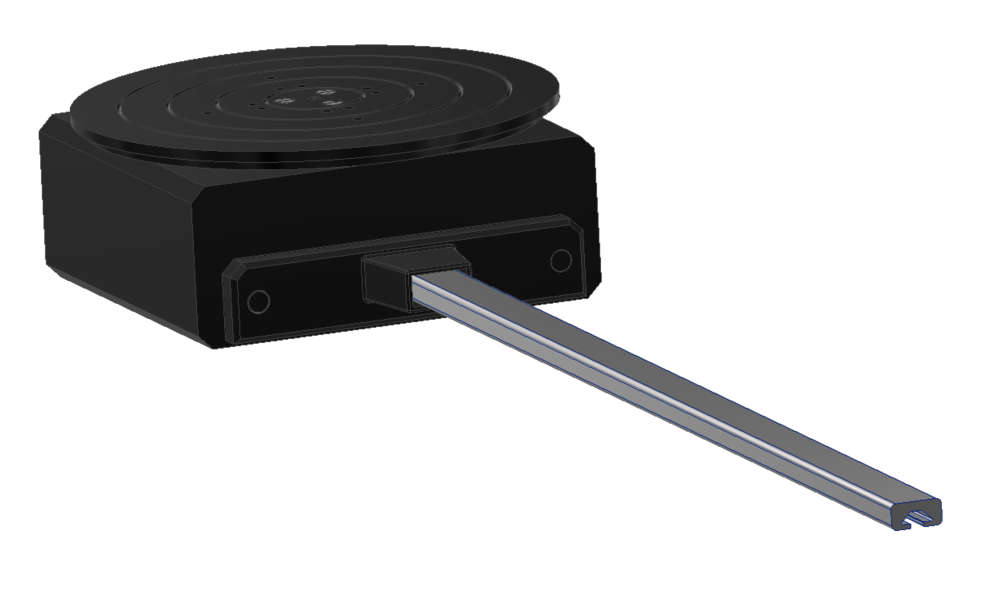
 A forgóasztal műanyag burkolata jó választás, hogy az állványt és az asztalt összekapcsolja. Ehhez elkészült a már meglévő burkolat és az asztallap CAD modellje, hogy későbbiekben az egész összeállítás látható legyen még a megvalósítás előtt. Az állvány funkcionalitását így előzetesen ellenőrizni lehet.

4‑1. ábra: Asztal CAD modellje

Az asztal 4.1 ábrán pirossal jelölt oldalához érdemes rögzíteni az állványt, mivel ez a legnagyobb síkfelület. Az állvány és az asztal egy a konzulenstől kapott 20x10-es alumínium profillal lesz összekapcsolva. Ezért a profil végére kell egy kapcsolatot, kötést tervezni, aminek prototípusról lévén szó könnyedén, egyedi szerszámok nélkül is megvalósíthatónak kell lennie. A prototípus tervezésnél nagy segítséget nyújtanak a 3D nyomtatással készült alkatrészek, mivel viszonylag gyorsan elfogadható méretpontossággal készülnek, így a valóságban is ellenőrizhetjük a tervek helyességét. Az alkatrészek nagyrésze ezért FDM típusú 3D nyomtatással kerül előállításra. Az állvány és az asztal kapcsolatánál figyelembe kell venni, hogy 3D nyomtatással túlságosan bonyolult geometriákat nem lehet létrehozni és a kötésnek könnyen oldhatónak kell lennie, hogy egyszerűen szétszedhető legyen a berendezés. Így egy-egy erősebb 1,24 T remanenciájú mágnes került az asztalba és a kapcsolatot megvalósító elembe, valamint az asztal felületén két az érintkező felületek elcsúszását meggátló központosító kúp került kialakításra, ezek a 4.2 ábrán láthatók.

‑. ábra: Asztalra merőlegességet biztosító elem

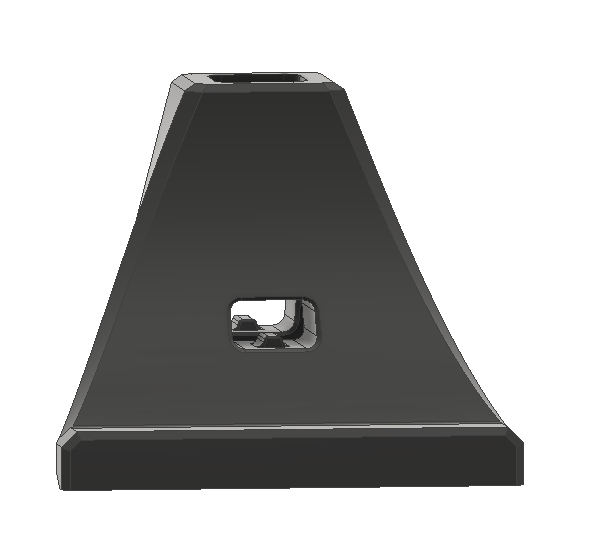
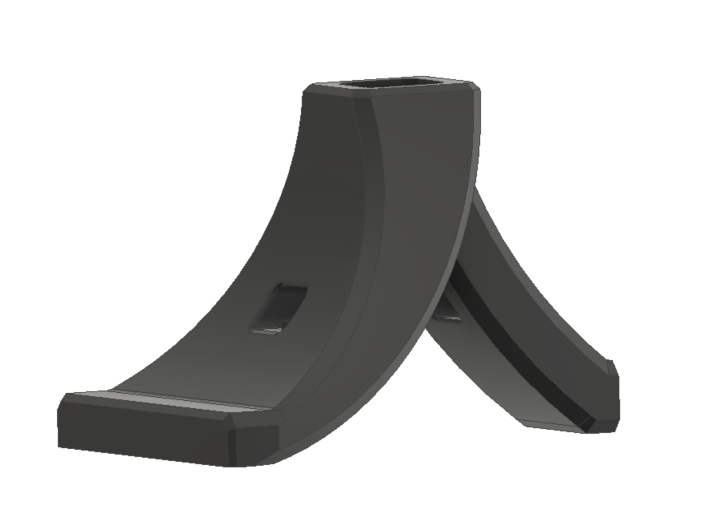


Ezek után az asztal oldallapjára való merőlegességet biztosító elembe rögzíthető a profil. Ez a profil formából adódóan sínként szolgálhat az állványhoz, ha esetleg módosítani kéne a szkenner és asztal távolságán.

4‑3. ábra: Állvány vízszintes tengelye

Az állvány függőleges tengelyét, tehát érdemes a profil mentén csúsztathatóvá tenni, valamint egy olyan talpat tervezni, ami stabilan tartja a kamerákat. Egy ilyen alkatrész modelljét ábrázolja a 4-4. ábra.

4‑4. ábra: Állvány talpa



Ez a talpazat összeköti az állvány két tengelyét. A függőleges tengely szerepét az eddig vízszintes helyzetben használt 20x20-as profil tölti be.

4‑5. ábra Állvány két tengelye

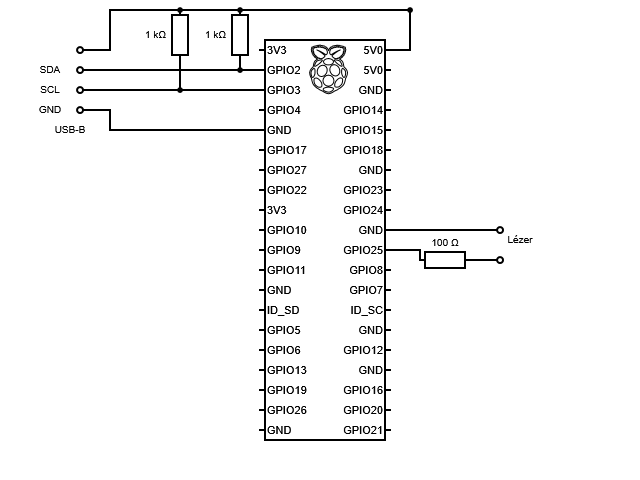
## Elektronika

A szkenner elektronikájának helyigényét és a csatlakozók pozícióját a szkenner házának tervezése előtt kell meghatározni. A berendezés vezérlése az előzetes munka során már el lett készítve, így az adott volt. Ennek a fő komponensei egy arduino nano, egy ATTiny2313 mikrovezérlő, a léptetőmotor vezérlőáramköre és egy hallszenzor. Az elektronika működésének elve, hogy a számítógép USB-n keresztül kommunikál az Arduino-val, ami kommunikációs protokollt használva bizonyos byte-okat küld a mikrovezérlőnek. Minden byte egy kódolt parancs, amit a mikrokontroller végrehajt, ha szükséges forgatja az asztalt, irányt vált vagy megkeresi a kitüntetett alappozíciót a hallszenzor jele alapján. A két webkamera közvetlenül a számítógéphez csatlakozott szintén USB-n keresztül, ezért a szkenner működtetéséhez 3 USB port volt szükséges, amivel elég kényelmetlen dolgozni főleg, ha a felhasználó 3 porttal rendelkező gépén más eszközt pl.: egeret is szeretne használni.

Kevesebb USB port felhasználását tenné lehetővé, ha az Arduino-t leváltaná egy a kamerákkal kommunikálni képes Linux rendszerű fejlesztőpanel. Ez végül egy a konzulenstől kapott Raspberry PI 3 B lett, valamint a tanácsára a fél léptetésre képes motorvezérlő is lecserélődött egy szintén tőle származó DRV8825 típusú mikroléptetésre alkalmas vezérlőre. Ezekkel az új alkatrészekkel újra kellett gondolni az elektronikát, a firmware-t és végeredményben a szkennert vezérlő szoftvert is.

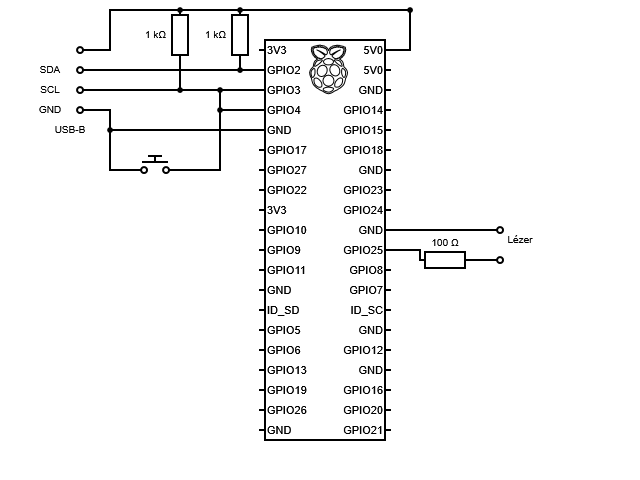
### Raspberry

Az elektronika megépítése az Raspberry-vel kezdődött. A Raspberry-nek van egy camera board elnevezésű a panelhez illeszthető 5 megapixeles kamerája, aminek a beszerzése szintén a konzulens segítségével történt meg. Ez könnyedén csatlakoztatható az alaplaphoz egy egyszerű 16 eres FPC kábel és a hozzá való alaplapon elhelyezett csatlakozó segítségével, így az egyik webkamera feleslegessé vált és felszabadult egy USB port. A másik kamera a Raspberry-n latálható egyik USB porthoz csatlakoztatható, tehát mindkét kamera a szkenner elektronikájának részévé vált. Ez a fejlesztőpanel az arduino-hoz hasonlóan rendelkezik GPIO lábakkal, amelyekre a megfelelő analóg vagy digitális jeleket küldhetünk vagy fogadhatunk. Ezeket használva kontrolálhatjuk a lézer ki és be kapcsolását és küldhetünk adatot az asztalnak. Az asztal es a szkenner továbbra is -vel kommunikál, mivel ehhez elegendő három összekötővezeték: a GND a feszültség szintek illesztéséhez, valamint az SCL és SDA, az órajel és adat küldéséhez. Az USB kábelben található 4 vezetékkel elegánsan meg lehet oldani a csatlakozást. Azonban míg az asztalban található mikrovezérlő 0-5 V-os feszültségszinttel dolgozik a Raspberry lábai max. 3,3V-ot adnak, ami a mikrovezérlőnek nem elég magas logikai feszültség az adatok fogadására ezért mindkét kommunikációban részvevő láb egy-egy 1kΩ-os ellenálláson keresztül az 5V-os lábhoz csatlakozik.

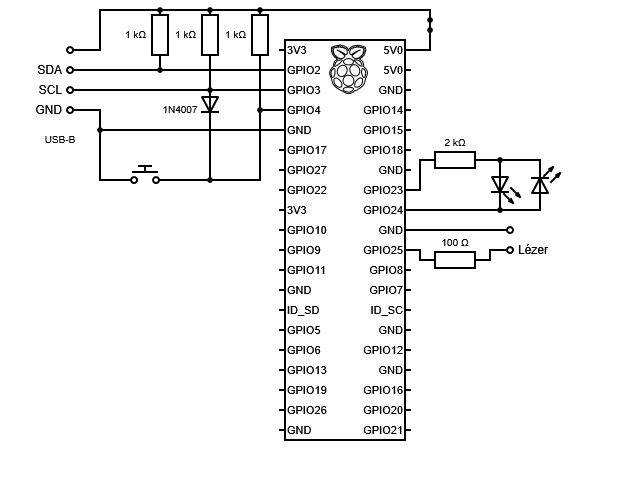
A kapott lézermodul egy LFL650-5-12(9x20)90-F250 típusú lézer, aminek az adatlapja szerint minimum üzemi feszültsége 3 V/DC és üzemi árama 20 mA ezért csatlakoztatható a Raspberry egyik kimenetéhez. A Raspberry ebben az összeállításban nem végez komoly számítási feladatokat és csak a lézerhez, a kamerákhoz és a kommunikációhoz nélkülözhetetlen energiát kell szolgáltatnia, ezért nem szükséges extra táplálás. Eddigieket figyelembe véve a Raspberry-hez illeszkedő áramkör rajza az ábrán látható.

‑. ábra: Áramkör

A Raspberry áramellátásáról saját tápja gondoskodik és mivel alapvetően egy fejlesztőpanel nem rendelkezik bekapcsológombbal, ezért a bekapcsolása a táp megszüntetésével és visszakapcsolásával lehetséges. A gomb beiktatása könnyedén megoldható, mert a SCL lábon alvó állapotában is 3,3V van ezt kis időre a föld potenciáljára lehúzva az eszköz érzékeli a feszültségesést és bekapcsolt állapotba kerül. Kikapcsolását szoftveresen lehet elindítani, de a gomb megnyomásával is el lehet érni, ha létrehozunk egy olyan python programot, ami minden SETUP után lefut és várja, hogy a bekapcsoláshoz hasonlóan az egyik kijelölt lábán (pl. 4, ahol szinten állandó 3,3V) van megjelenjen egy lefutó él.



4‑7. ábra: Gomb beépítése

Mivel az SCL és a 4. lába a panelnak azonos potenciálon van és adat küldés során lefutó élek keletkeznek a Raspberry feleslegesen kikapcsolna, ezért egy 1N4007 típusú dióda került a két láb közé. Így most az SCL lábon kb. 4,4V feszültség van a 4. lábon a dióda nyitása miatt szintén 4,4V van, ha adatot küldünk és az SCL lábon kb. 1,1V feszültség mérhető, akkor a dióda zár és a 4. lábon 3,3V lesz, ezt a feszültség esést érzékelve szintén kikapcsolna a berendezés. Megoldás, ha a 4. láb és az 5V-os láb közé is kerül egy 1kΩ-os ellenállás, így a dióda zárásakor is megmarad ezen a bementen az eredeti feszültség. A gomb rendelkezik két beépített LED-del, amelyek a feszültség polaritásától függően felváltva tudnak világítani. Ezzel a két színnel hasznos jelzések adhatók a felhasználónak pl.: az eszköz bekapcsolt állapotban van, vagy éppen dolgozik.

4‑8. ábra: Végleges áramkör

### Asztal

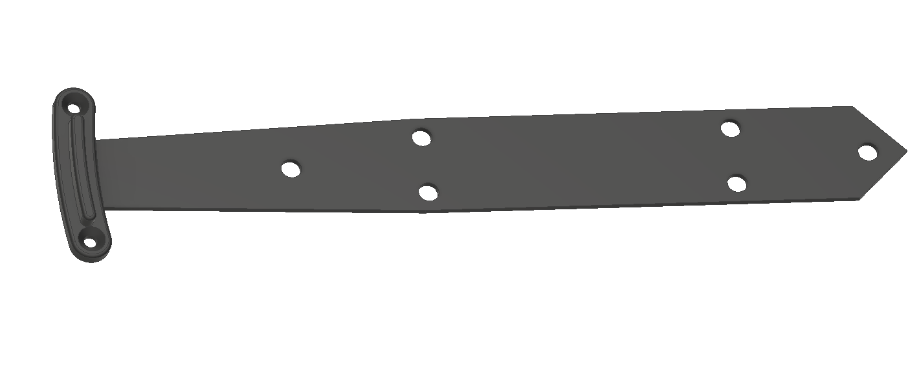
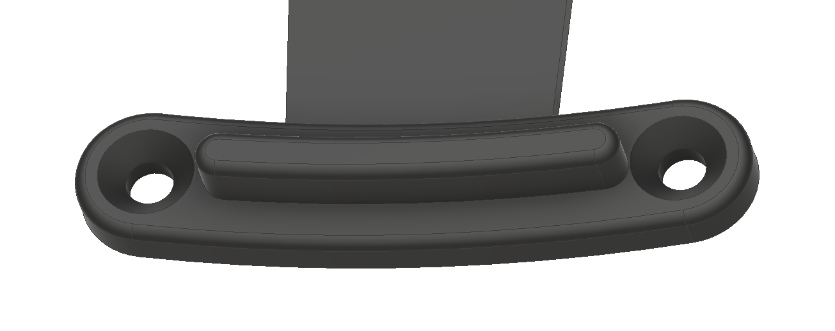
A kisebb felbontású léptetésnél érdemes változtathatóvá tenni a léptetőmotor fogásának frekvenciáját így elérhető, hogy motor léptetése a megfelelő frekvencián kevésbé tűnik osztottnak, darabosnak, valamint egy lassítás és gyorsítás is elérhető, hogy a szkennelendő tárgy ne mozduljon el az asztalon tehetetlensége miatt a hírtelen mozgatás folytán. Az asztalban jelenleg egy 2 kb flash memóriával rendelkező mikrovezérlő van, ami kevésnek bizonyulna egy finomabb mozgatást megvalósító kódnak, de mivel az arduino nano felszabadult így ez kerül az asztalba. A kapott léptető driver **stp** lába felfutó élre érzékeny, ekkor a motor tápfeszültségének a lépés felbontása szerint egy bizonyos hányadát az egyes tekercsekre adja. Egy pulzáló jellel a motor léptethető, forgás irányát a **dir** lábra adott logikai feszültség értéke szabja meg. Azt, hogy milyen felbontású mikroléptetést szeretnénk végrehajtani az **M0-M2** logikai értéke adja meg az adatlap alapján. A dokumentáció továbbá javasolja a motor táp +12V-os bemenete és a GND bemenete közötti 100 μF-os kondenzátor használatát.

Az asztal fejlesztése közben fény derült egy hibára, miszerint a mágnes által keltett mező olyan nagy, hogy több lépést átfedően érzékeli a hallszenzor ez azt eredményezi, hogy nem lesz egyértelmű az alaphelyzet. Több különböző mágnest és távolságot is próbálva megállapítható, hogy a mágnes az érzékelőtől való távolságában kicsi az a sáv, ahol egyáltalán érzékel és csak egy lépésnyi pozícióban érzékel a szenzor. Ezért egy másik típusú érzékelő került beszerzésre, ami egy Omron EE-SX1106 típusú miniatűr villás fénykapu [].

4‑9. ábra Használt fénykapu

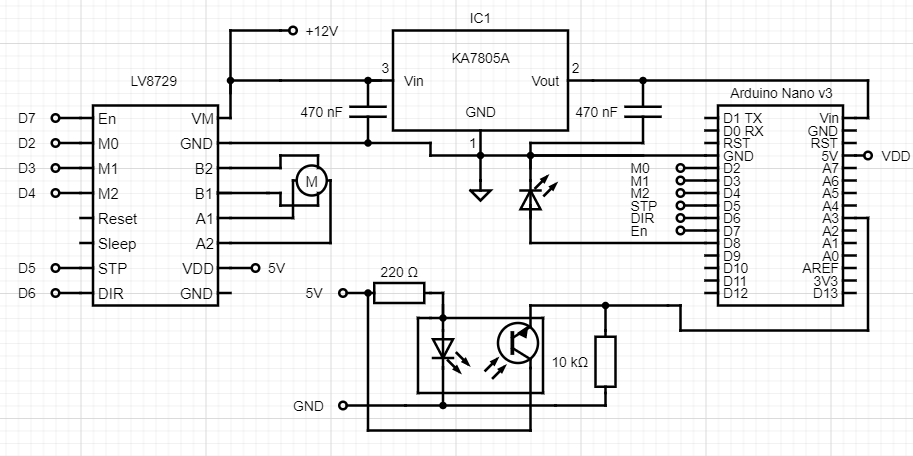
Ehhez 3D nyomtatással készült egy alkatrész, ami rögzíthető az asztallap aljára és a megfelelő pozícióban kitakarva, majd újra szabad utat adva a fénynek pontosabban detektálhatóvá teszi a számunkra fontos helyzetet. Az optocsatoló résmérete 0,4 mm, ez elég kicsi az összeszerelés során használt szerszámok pontosságához és az állvány hibáihoz mérten ezért ez a pontosság bőven elegendő. A kalibrációs minta tartója a megfelelő furatokhoz rögzíthető, ezért a szerelés pontosságának érdekében a nyomatott alkatrész egy a furatokhoz igazító sablonnal együtt készült.

4‑10. ábra: Fénykapu jelét kitakaró elem



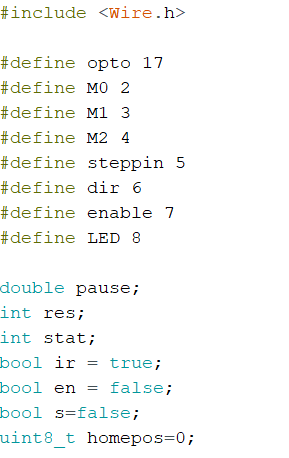
Az asztalban használt arduino-nak van egy beépített feszültség regulátora, amely képes egy max. 20V/DC feszültségű áram 5V-á alakítására, ha az áram erőssége nem nagyobb, mint 1A. Ezek a paraméterek megfelelnek a motor tápjának, így megoldható volna, hogy a motor és a vezérlése egy táplálásról működjön. A próbák alapján az arduino-n található kicsi felületű SMD feszültség regulátor nem képes elegendő hőt disszipálni, ezért az eszköz veszélyesen túlmelegedhet. Ezt a problémát orvosolja egy nagyobb, furatszerelt, KA7805A megnevezésű feszültségszabályzó beépítése. Az alkatrész Vin és Vout oldali kivezetéseit a földdel egy-egy 0,47μF-os kondenzátor köti össze.

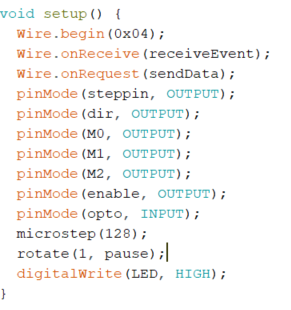
Az asztal megépítése utáni teszteken a motor minden forgási frekvencián hangosnak bizonyult, ezért a léptető driver lecserélődött egy MKS LV8729 léptető vezérlőre, amely „ultra quiet stepper motor driver” címszóval van hirdetve. A motor valóban jóval csendesebb lett és a bekötése is a DRV8825 bekötésével megegyező, hiszen ezeket az áramköröket 3D nyomtató léptetőmotorjainak vezérléséhez készítik elsődlegesen, ezért ezeknek csereszabatosnak kell lennie. Ami változott, hogy az új típusú áramkör már mikroléptetésre is alkalmas, így ezt implementálni kell az arduino programjába.

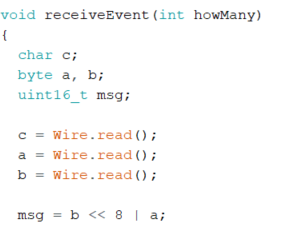


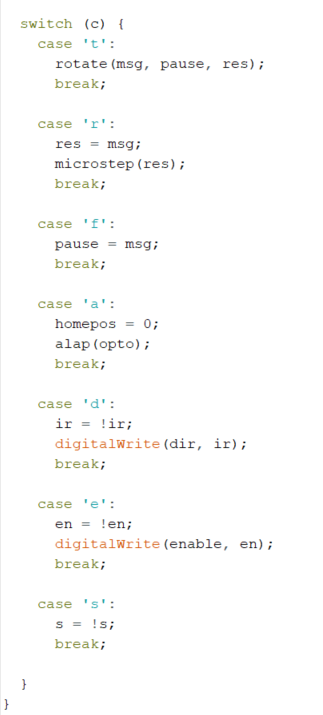
4‑11. ábra: Asztal áramköre

### Firmware

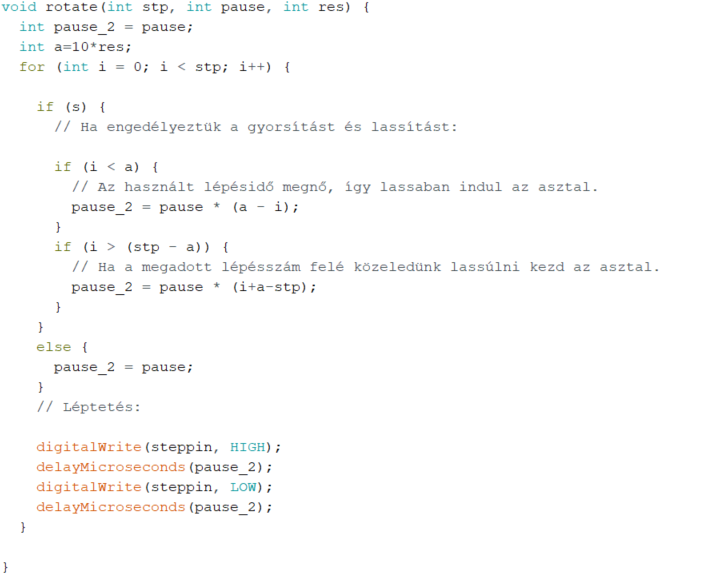
 Arduino IDE-ben (Arduino programozására szánt fejlesztőkörnyezetben) készült a program, mert sok, könnyen használható könyvtárával gyorsabbá teszi a prototípustervezést. Elsődlegesen betöltjük az kommunikációt kezelő Wire nevű header fájlt, majd minden használt ki és bemenetet, aminek értéke a program futása során nem változik define-nal létrehozzuk, ezek után azokat a változókat, amiket használ a program a megfelelő típusban inicializáljuk.

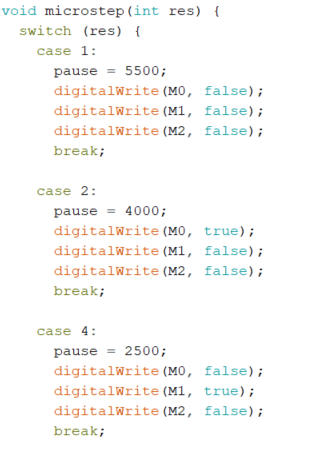
A következő lépésként lefut a setup nevű függvény, ebben a függvényben létrehozzuk a Slave-ként funkcionáló Arduino címét, amihez a Master Raspberry kapcsolódni tud. Létrehozunk továbbá egy-egy eseményt, ami akkor hívódik meg, ha mester eszköz adatot küld vagy ha adatot kíván olvasni a szolgaeszközről. Beállítjuk, hogy az egyes lábak ki vagy bemenetek legyenek és felvillantunk egy működést jelző LED-et. Minden, amit a vezérlő végrehajt valamelyik esemény meghívása következtében teszi. Adatok fogadásakor 3 byte adatmennyiség jön egyszere, ami egy parancsot kódol. Az első byte egy karakter a másik kettő egy-egy uint8 típusú szám, amelyek bitek eltolásával uint16 típusúvá alakítandó így kapjuk meg a msg változó értékét. A 128 mikroléptetéses üzemmódban egy egész fordulat 25600 lépésből áll, mivel egész lépésekben 200 lépés szükséges egy teljes fordulat megtételéhez. Uint16 típussal 65535-ig tudjuk tárolni a számokat, ezért alkalmas a kívánt lépésszám megadására, míg 1 byte kevés volna.



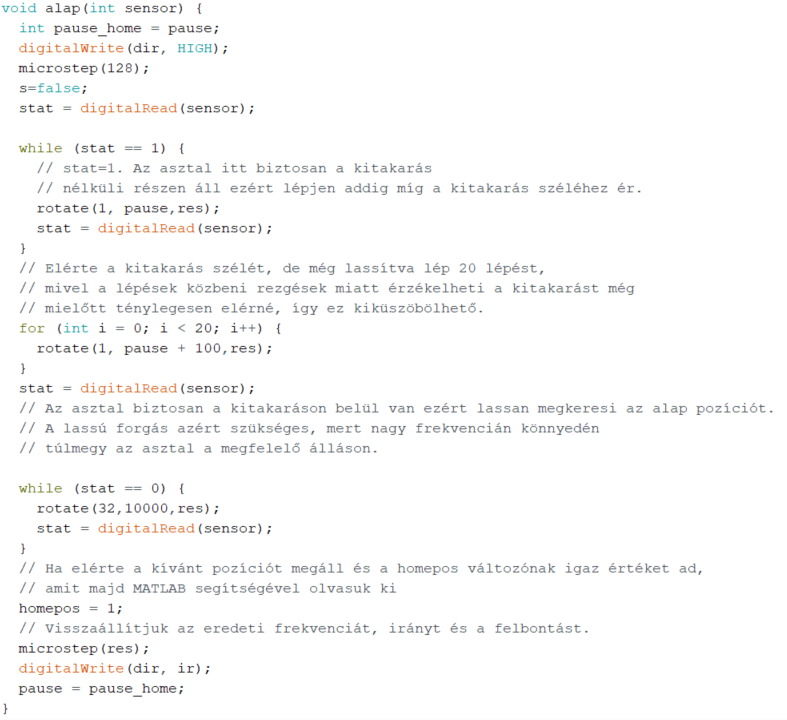
A küldött karakter meghatározza mit szeretnénk csinálni az asztallal.

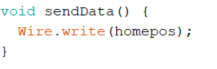
Ha a küldött karakter ’t’ akkor az asztal annyit lép amennyi a küldött szám volt, emellett, ha az ’s’-sel a gyorsítás, lassítás engedélyezve van akkor azt is elvégzi a rotate() függvény:



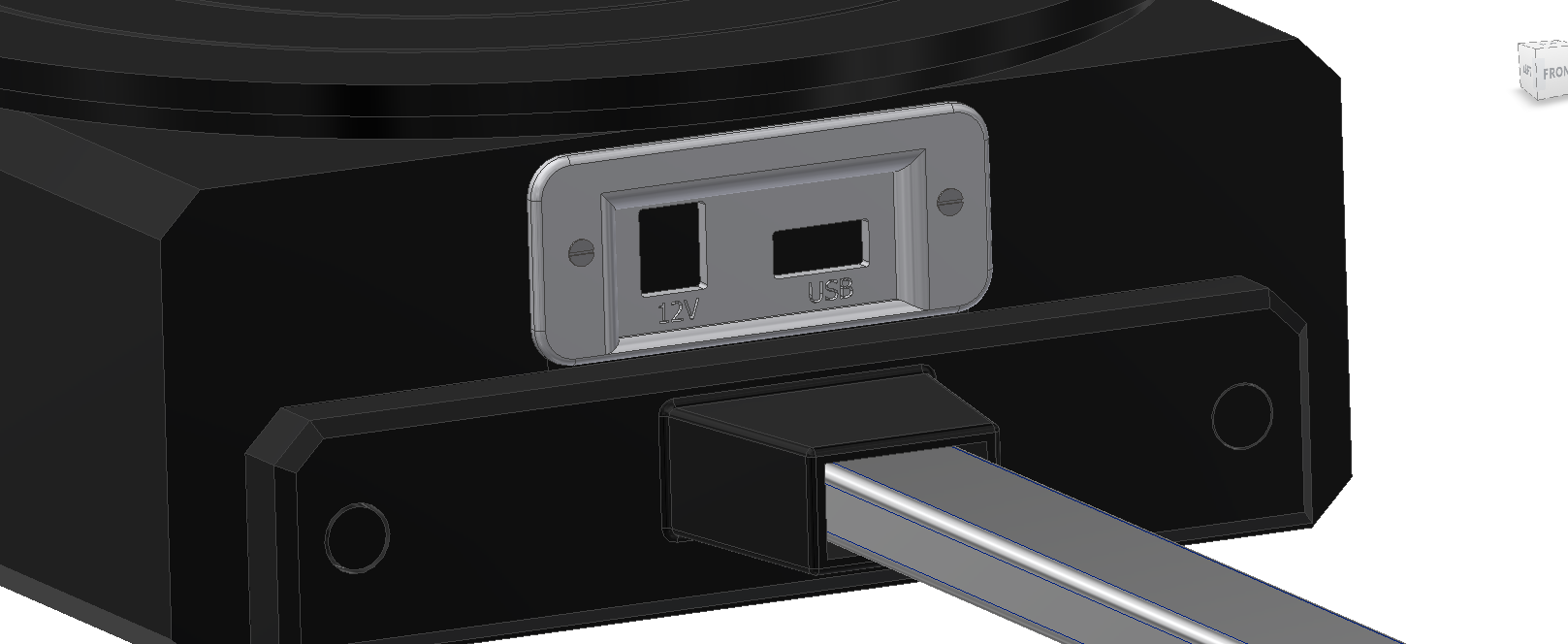
Az ’r’ (resolution) esetén a felbontást szeretnénk megváltoztatni, így a küldött szám meghatározza milyen léptetéssel dolgozunk tovább, a microstep() függvény ez alapján átírja a driver M0-M2 lábaira adott logikai értékeket és felvesz egy alap lépésidőt.

Az ’f’ karakter a frekvenciát a ’d’ az irányt változtatja meg. A driver kikapcsolható, ha az Enable lábára igaz értékű logikai jelet küldünk ez megtehető az ’e’ karakterrel. Az asztal alaphelyzetbe állítása ’a’ karakter segítségével történik, ekkor lefut az alap() függvény, ami megkeresi optokapu fényének kitakarására szolgáló alkatrész segítségével az alaphelyzetet.



Ha a mestereszköz kéri az asztal visszaküldi, hogy elérte az alaphelyzetet.

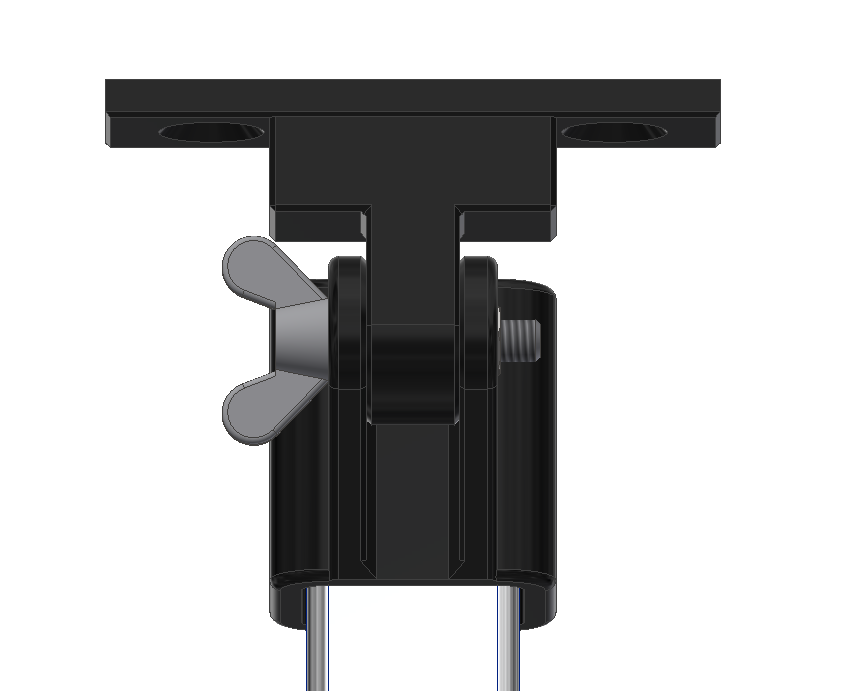
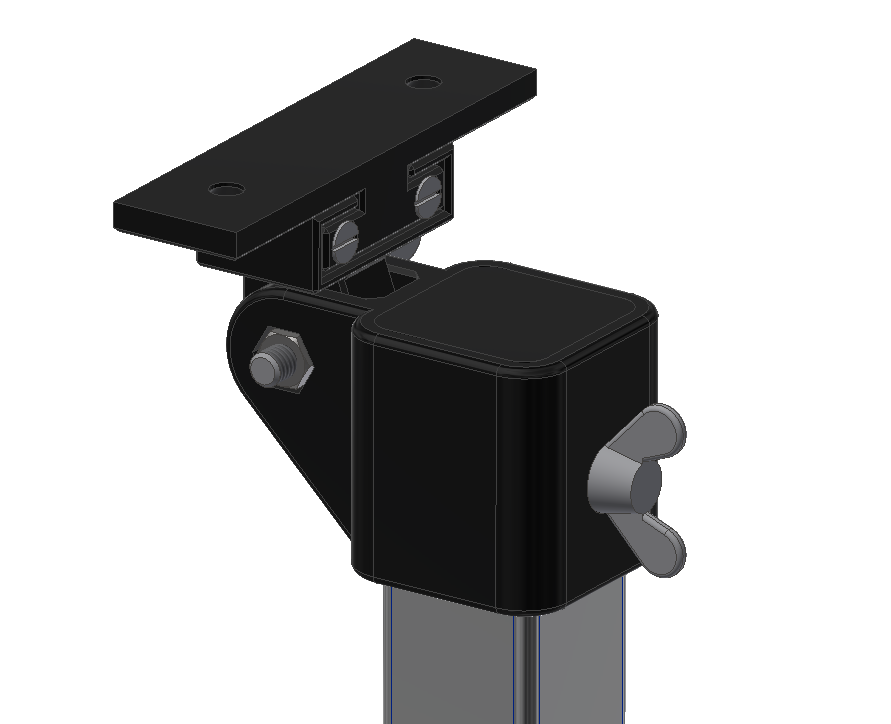
## Állvány

Miután elkészült az elektronika és kialakultak csatlakozók pozíciói, minden adott a szkenner és az állvány megtervezéséhez. Elsődlegesen az asztalba került be a motor táplálásának csatlakozója és a kommunikációhoz szükséges USB csatlakozó egy közös csatlakozóablakba rögzítve.

4‑12. ábra: Csatlakozóablak

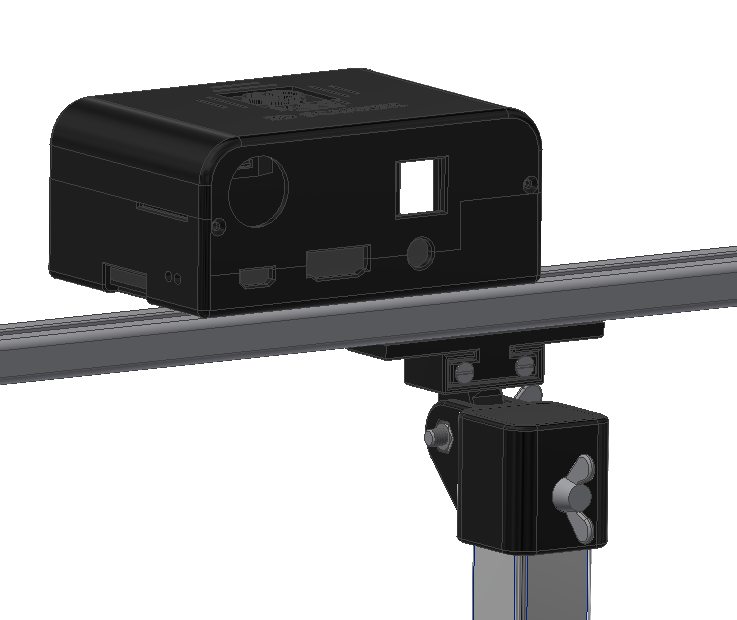
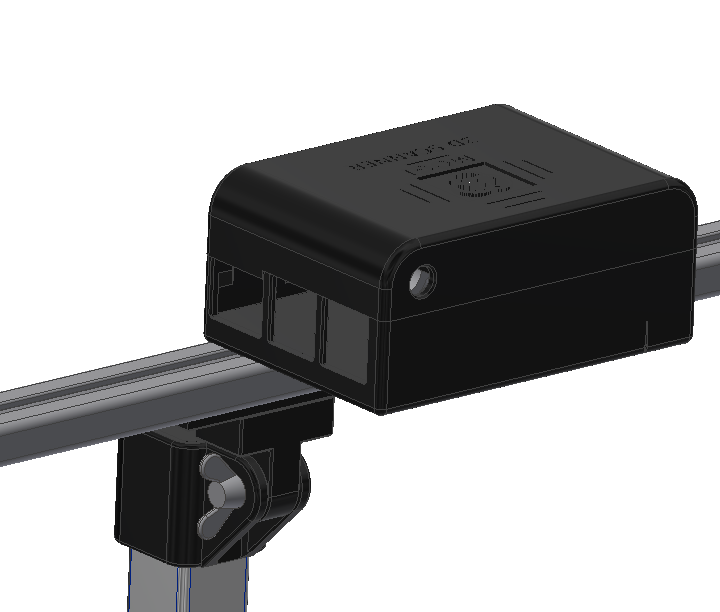
Ezek után a Raspberry és a kamerák elhelyezése következett. Az állvány harmadig tengelyének dönthetőnek és könnyen leszerelhetőnek kell lennie úgy, hogy a magassága is állítható legyen. Egy olyan a kamera állványokhoz hasonló a tengelyeket összekapcsoló elem került nyomtatásra, aminél a rögzítés szárnyascsavarokkal történik. A csavarok meghúzásakor az oldalak közti súrlódás megakadályozza a beállított tengely elfordulását, de kézzel könnyedén lazítható, ha szükség van rá.

4‑13. ábra: Tengelyeket összekapcsoló elem



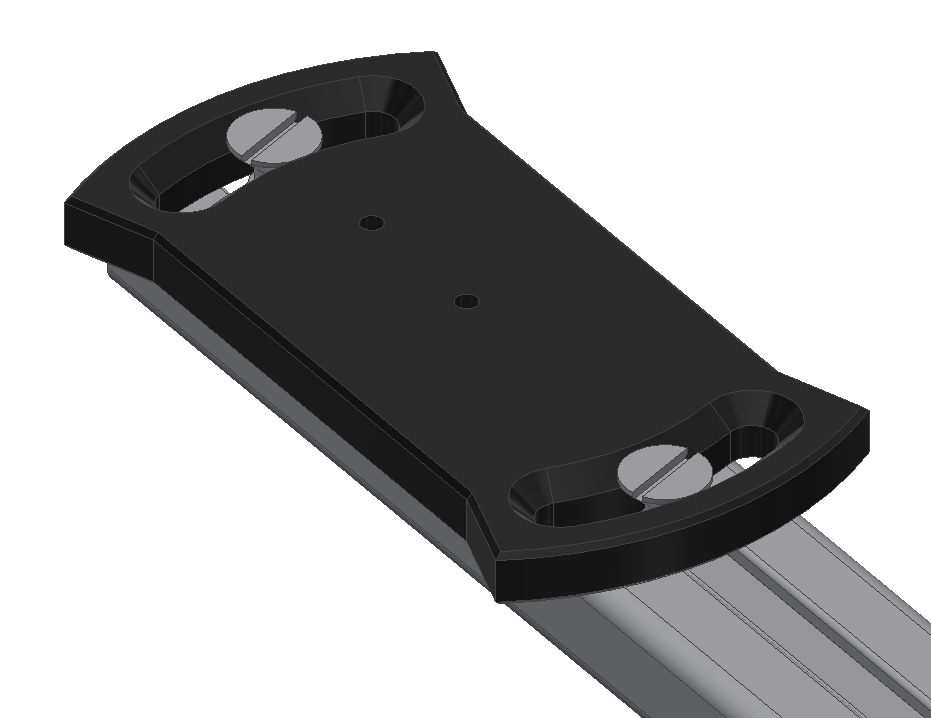
A vízszintes tengely ismét egy alumínium profilból lett kialakítva. A profilt középen két csavarral lehet rögzíteni az előbb bemutatott műanyag elemhez. Raspberry esetében bonyolult volt a ház tervezése, mivel sok porttal és csatlakozási lehetőséggel rendelkezik. Ezeket jó döntés nem elrejteni, mert az eszköz képes arra, hogy HDMI kimenetét monitorra kötve megjelenítsen egy desktop-ot és egeret, valamint billentyűzetet kezeljen, így későbbiekben akár külön számítógép nélkül is működhet. A ház két részből csavarozható össze és helyet kap benne az elektronika és a lézer.

4‑14. ábra Szkenner ház



A kamerák a Raspberry két oldalára kerülnek, ezeknek a profil mentén elcsúsztathatónak és elforgathatónak kell lenniük, erre megoldást nyújtanak a 4-14. ábrán látható alkatrészek

4‑15. ábra: Balra a webkamerát tartó jobbra a kamera modult tartó alkatrészek





4‑16. ábra: Elkészült szkenner

4‑16. ábra Összeállítás modellje

# Képfeldolgozó szoftver

A szoftver MATLAB nyelven íródik és a felhasználói felület a MATLAB által szolgáltatott App Designer-ben készül. A kód túl hosszú ahhoz, hogy minden részlete bemutatásra kerüljön, ezért csak a fontosabb, képfeldolgozással kapcsolatos részekre összpontosít a leírása.

## Csatlakozás

A Raspberryhez való csatlakozás a MATLAB Rasbian módosított operációsrendszere segítségével egyszerűen megvalósítható. A Raspberry memóriájára fel kell tölteni a módosított lemezképfájlt, majd az operációs rendszer telepítése után könnyedén csatlakozhatunk és adhatunk parancsokat a MATLAB-on keresztül az alábbi függvényekkel []:

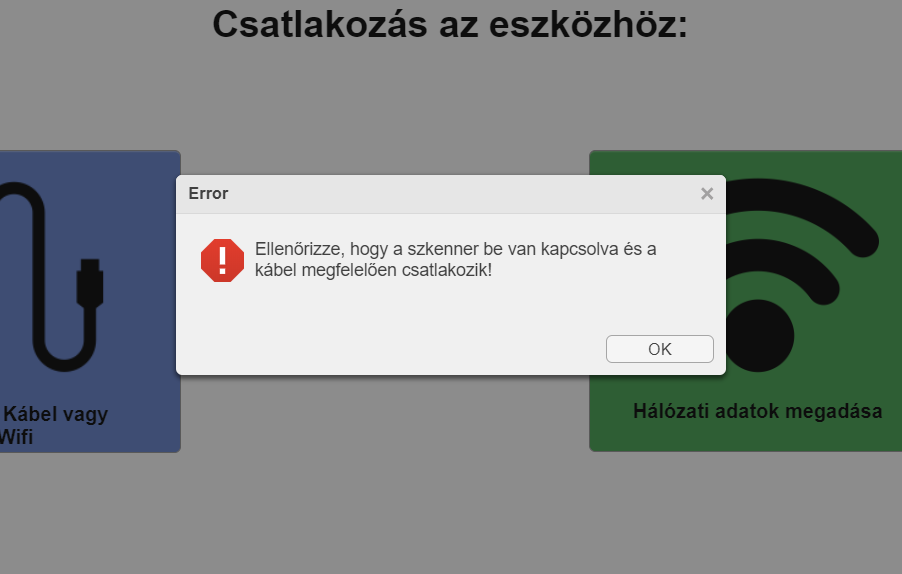
* raspi(3DScanner) – csatlakozás az eszközhöz.
* writeDigitalPin() - Raspberry GPIO lábának vezérlése
* i2cdev() – I2C eszközhöz való csatlakozás
* write() – Adat küldés I2C eszköznek
* read() – Adat fogadása I2C eszközről.

A program megnyitása után az 5-1. ábrán jelölt ablak látható.



‑. ábra: Nyitó ablak

Itt a kék gombra kattintva lefut a csatlakozáshoz szükséges összes függvény, valamint az eszköz nem észlelése esetén hibaüzenet ugrik fel.

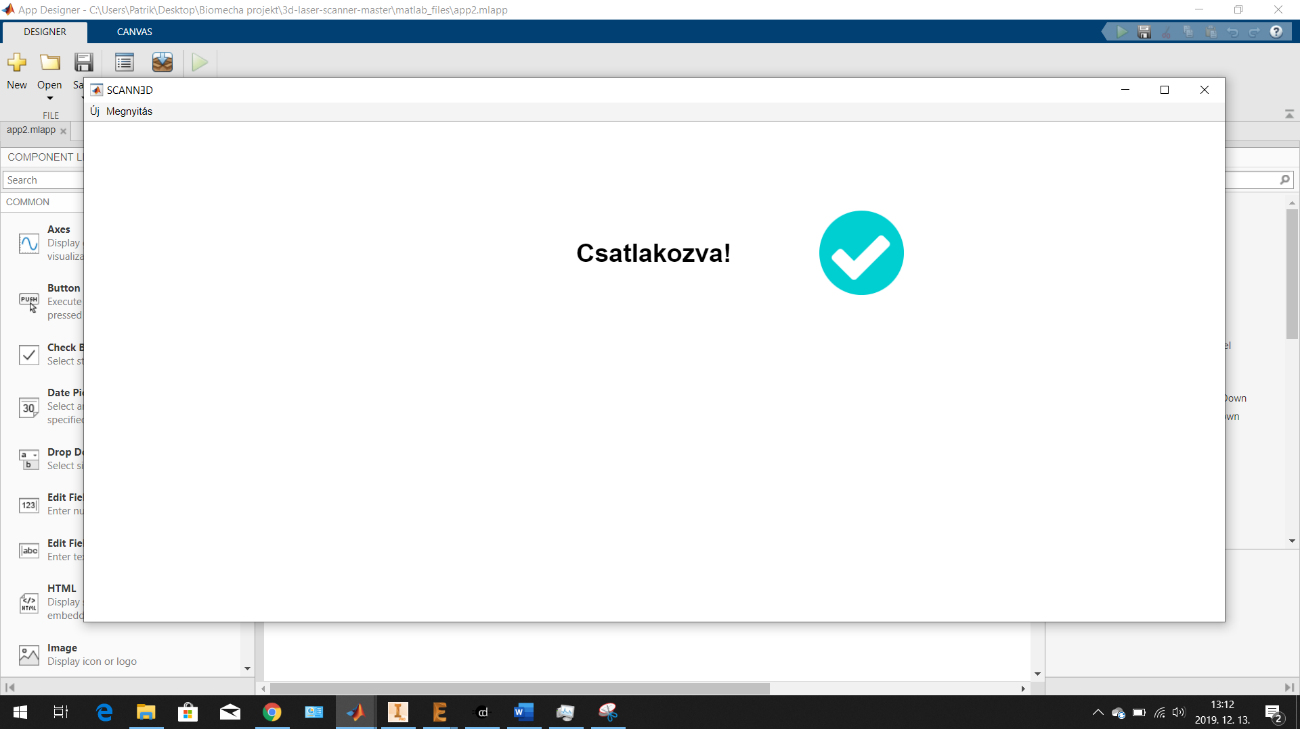


‑. ábra: Hibaüzenet

Ezt a hibakezelés egyszerű beépített dialogboxokkal és a try-catch használatával készült az eddig felismert minden hibatípusra.

A Raspberry képes ethernet kábelen és hálózaton csatlakozni a számítógéphez és ha már egyszer csatlakoztunk kábelen akkor a nyitó oldalon található zöld gomb megnyomásával megadhatók a hálózat adatai. Ehhez a Raspberry linux alapú rendszerében kell dolgozni, a MATLAB, ezt is támogatja és kiadhatók a parancsok a system függvény segítségével[].

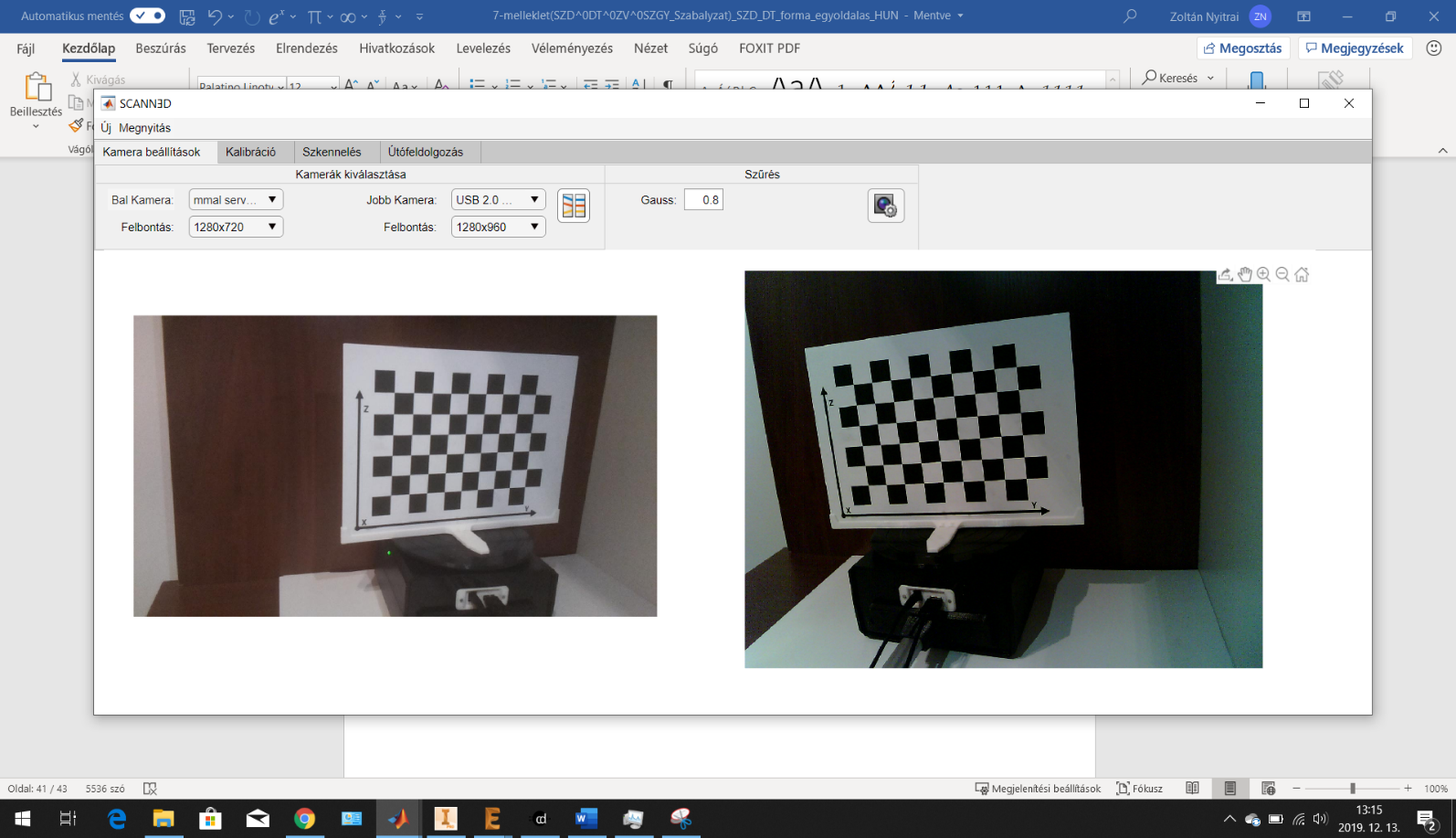
Pl.: system(rpi,'iwlist wlan0 scan ','sudo') – Hálózat keresése a Raspberry-n

Ha már kommunikál a számítógép a panellal, akkor jelezve ezt villogtatja gomb LED-jeit és kiírja a felhasználónak.

5‑3. ábra: Csatlakozási dialogbox

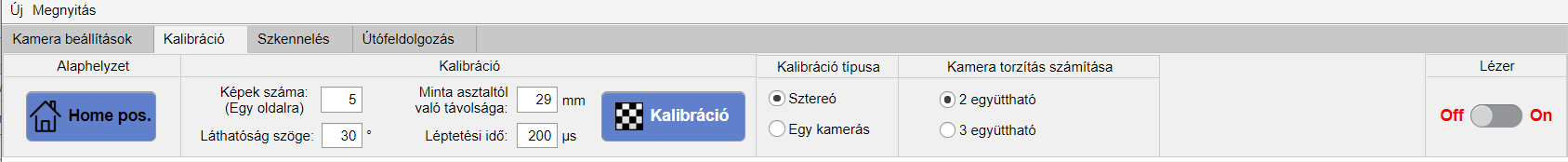
## Kamerák

Csatlakozás után megjelenik a főoldal, ahol kiválasztható, mely kamerákat szeretnénk használni az elérhetőek közül és milyen felbontásban. Fontos, hogy a később használt kalibráció típustól függően, lehet, hogy csak két azonos felbontású kamerával működik a szkennelés. A kamerák külön kalibrálása esetén nincs ilyen megszorítás, csak sztereókalibrációnál. Ugyanezen az oldalon beállítható, ha szükséges, a lézer láthatóságát növelő képszűrés. A kamera ikonnal rendelkező gombokkal előképet vagy fotót tehetünk a képernyőre, hogy láthatóvá váljon a változás.



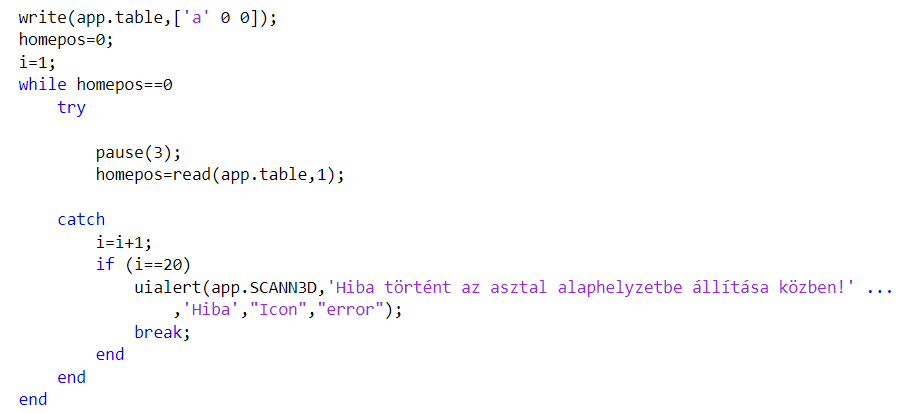
5‑4. ábra: kamerák beállítása

## Kalibráció



5‑5. ábra: kalibráció menüje

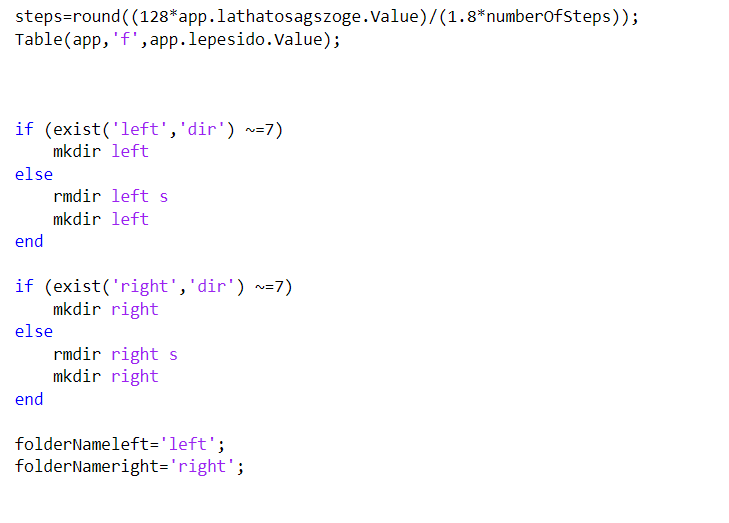
Ezen az oldalon az első fontos lépés az asztal alaphelyzetbe állítása a Home pos. Gombbal. Ennek megnyomásakor lefutó függvény legfontosabb részlete látható az ábrán ekkor az asztalnak ’a’ karaktert küldve az alaphelyzetbe állást adjuk parancsba.



Három másodpercenként lekérdezi az asztal helyzetét és ha nem sikerül a művelet kilép és értesíti a felhasználót.

A kalibráció során kiválaszthatjuk, hogy egy vagy két kamerás kalibrációt szeretnénk végezni, utóbbi esetében a program meghatározza a két kamera egymáshoz képesti helyzetét is. Mivel a két kamera tulajdonképpen két ugyanazt a fényforrást használó külön szkenner ezért választásnak ebben a helyzetben nincs jelenősége. Csupán az számít, hogy az lézer síkja hogyan helyezkedik el. Későbbiekben azonban, a két kamera képén közös pontmegfeleltetéssel és kétkamerás háromszögeléssel egy másik mérési módszer is használható lehet, ezért érdemes beépíteni a sztereókalibrálást is.

Az automatikus kalibrációnál a mintát az asztal bizonyos pozícióba forgatja, hogy a minta különböző helyzetében talált képpontokkal elvégezze a kamera paramétereinek meghatározását a matematikai háttérben leírtak szerint. A programot tartalmazó mappába elkészít a program két almappát a kamerák által alkotott képek tárolására, ezeket később újra fel lehet használni így nem szükséges az újra kalibráció minden szkennelés elején.

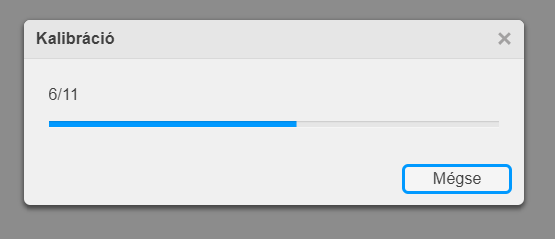


Csak a sztereókalibráció bemutatására kerül sor, mivel a másik típusnak is ugyanez az alapelve. A menüben kiválaszthatjuk a forgatás teljes szögét és lépésidejét azt, hogy hány képet készítsen a kamera és hogy a minta milyen távol van az asztallaptól. Ezek után a forgatás során készített képeken (a programban iml és imr) detektáljuk a pontokat, majd megbecsüljük a belső és külső paramétereket. A belső paraméterek számításánál megadható, hogy a kamera torzításának meghatározásához hány együtthatót használjon az algoritmus.



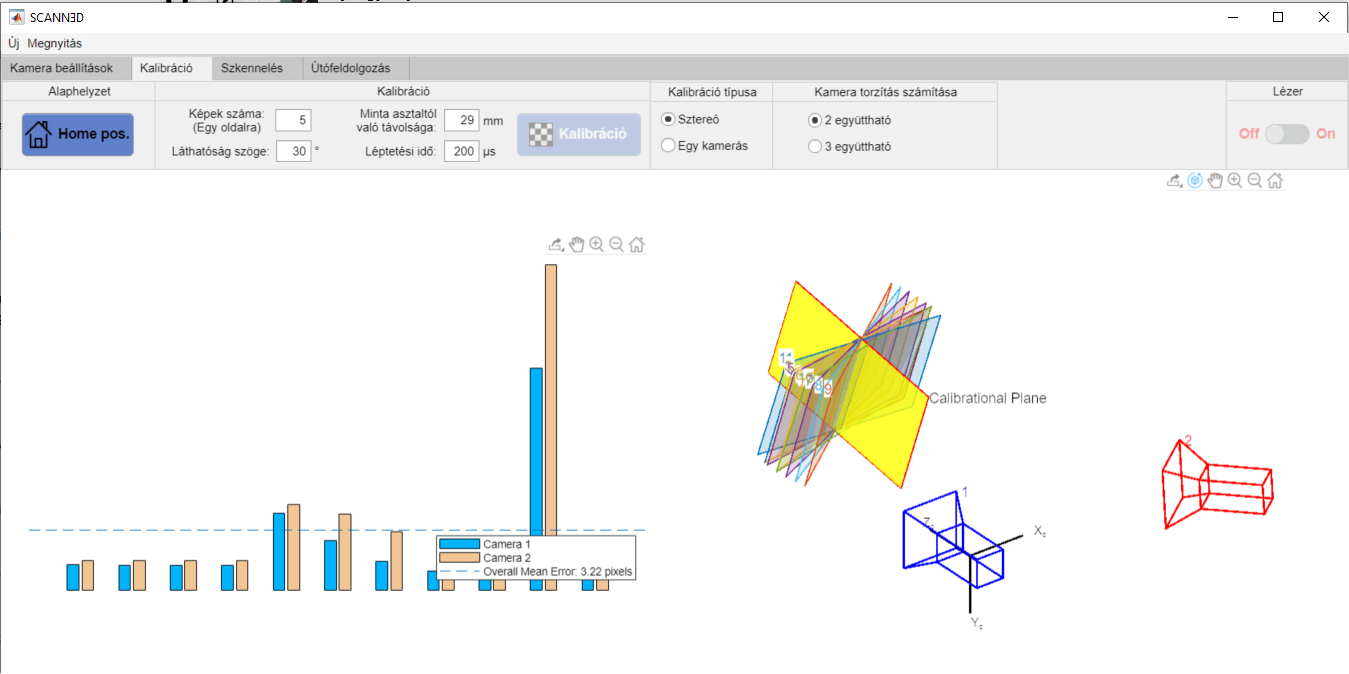
Mivel sztereókalibrációról van szó, olyan helyzetbe kell állítani a mintát, hogy mindkét kamera detektálni tudja a pontokat a minta azonos helyzetében készült képeken. Ezek után ki kell számítani a lézer síkjának helyét a minták közé lineáris regresszióval síkot illesztve. Az ezeket megvalósító függvények láthatók alább.



A kalibráció során a program jelzi mennyi képet kell még elkészíteni.

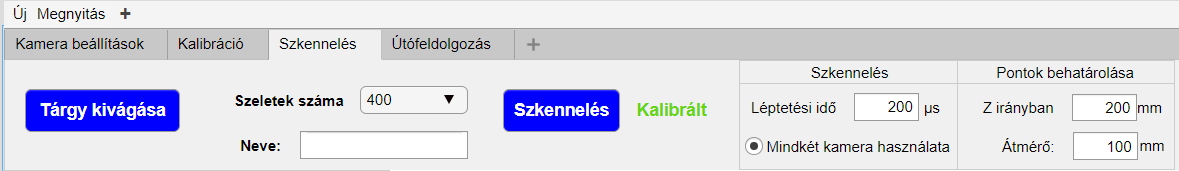
5‑6. ábra: Kalibrációs dialogbox

Ha befejeződött a művelet a program megmutatja a kalibráció becsült hibáit és az észlelt síkok elhelyezkedését. Ezzel a kalibráció kész.



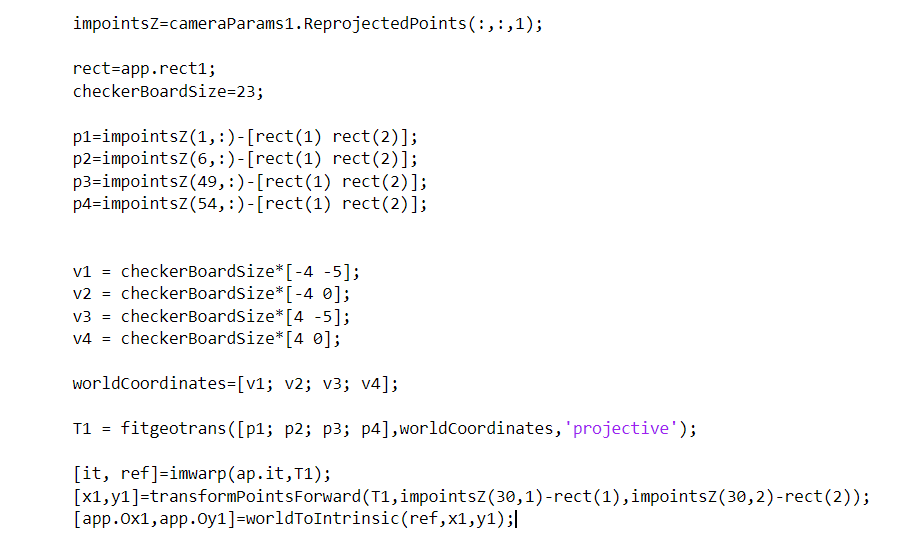
‑. ábra: Kalibrációs eredmény

## Szkennelés

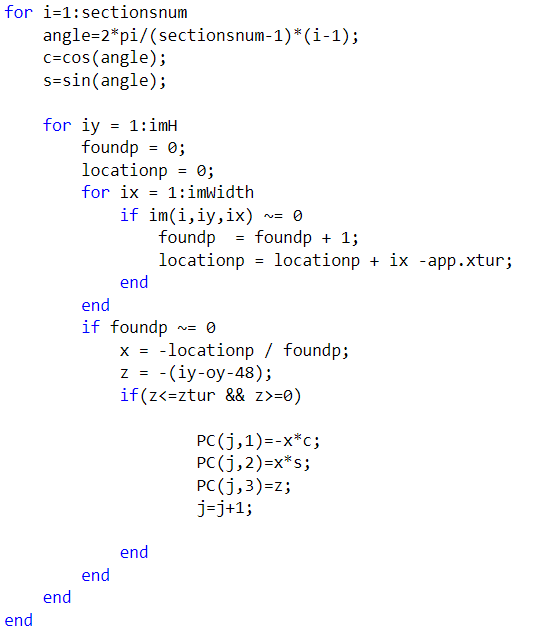
 A kalibráció után el lehet kezdeni a szkennelést.

5‑8. ábra: Szkennelés menüje

Itt a Tárgy kivágása gombbal az első készült képen kijelölhető az a képterület, ami fontos számunkra, hogy a többivel ne kelljen számolnia a szoftvernek. Kiválasztható, milyen felbontással dolgozzon a szkenner. A folyamat első lépéseként az előzőleg meghatározott lézersík és a hozzá tartozó kamerán észlelt pontok között megkeresi a transzformációt.

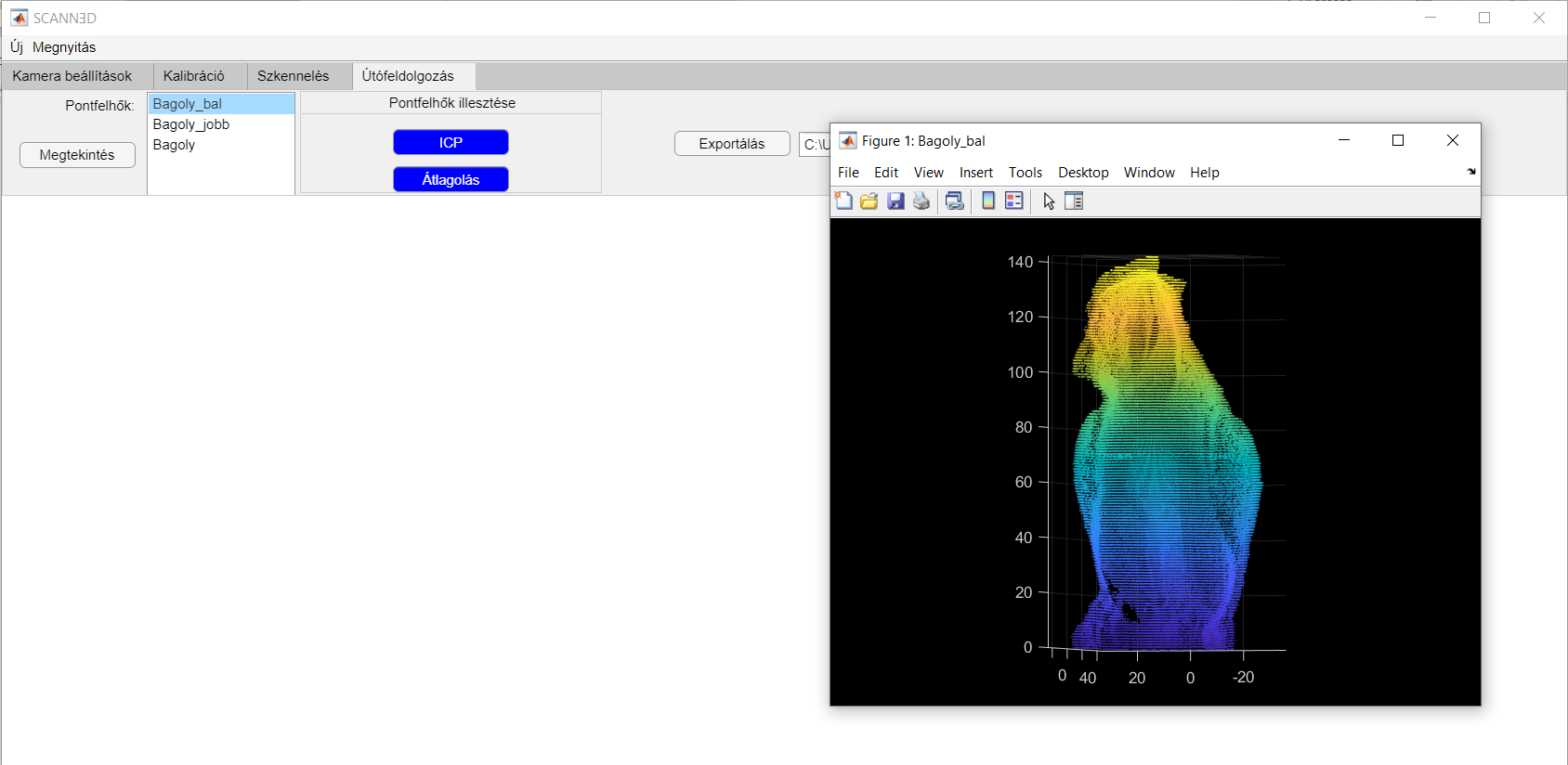


A transzformált képen mérhető a tárgyakra vetülő lézer és a tengelyek közötti távolság. A szkenner egyszer körbe forgatja a tárgyat a szeletek számánál megadott lépésszámmal, minden pozícióban képet készítve. Majd a transzformációt az egyes képekre alkalmazva elkészülnek a mérésre alkalmas minták. A képek egyes pixelei 1 mm távolságnak felelnek meg így a program csupán megszámolja a képen egy sorban talált lézer pixeltávolságát a transzformált origótól. Ezeknek a távolságoknak az ismeretében a pontok elhelyezhetők egy közös polárkoordináta rendszerben.

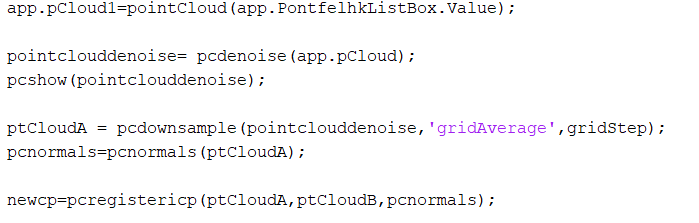


## Utómunka

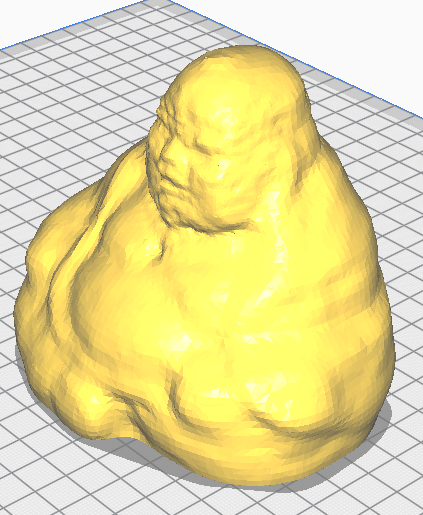
Az így elkészült két pontfelhő megtekinthető az utómunkák fülben egy új ablak nyitásával.



**5‑9**. ábra: Utómunka felülete

A két kamera által készített pontok illesztéséről a MATLAB-ba épített pontfelhő kezelését megkönnyítő csomag segített. Ebben találhatóak különböző algoritmusok, amelyek segítségével a kis eltérésekkel rendelkező pontfelhők könnyedén egy közös koordinátarendszerbe konvertálhatók. Fontos megjegyezni, hogy nagy eltérések esetén a program nem alkalmas az illesztésre, mivel ezt ilyenkor a két modellen kijelölhető pontmegfeleltetéssel szokták megoldani, ehhez interaktívabb ablak szükséges a fel használóval.

A keletkező pontfelhő kiexportálható .ply fájl formátumban az export gombbal. Ehhez a pcwrite(ptCloud,filename,'PLYFormat','binary') beépített parancs került felhasználásra[].

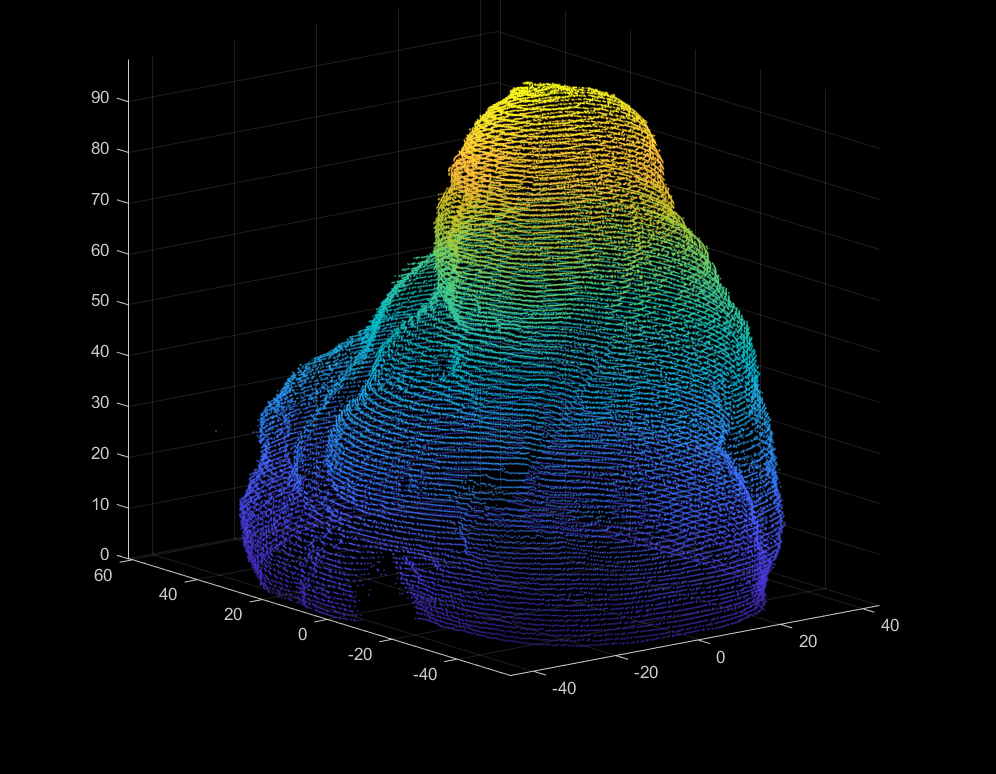
Ez a fájl formátum már lehetővé teszi a pontok megnyitását egy másik program számára. Egy ilyen program például a Meshlab amely képes a megnyitásra és a pontok hálózásával, háromszögelésével stl. fájl készíthető[]. Ez fájl típus alkalmas arra, hogy megnyissuk CAD programban vagy egy 3D nyomtatható modellt készítsünk.

# Eredmények

A program megírása után néhány tesztnek vettetem alá szkennert az első ilyen teszt egy 30x35x60mm es téglatest szkennelése volt a méretpontosság megállapítása miatt. Ennek eredménye alább látható.

|  |  |  |  |
| --- | --- | --- | --- |
|  | X (mm) | Y (mm) | Z (mm) |
| 500mm-ről szkennelve (1280x720p) | 29,2 | 35,3 | 62,1 |
| 400mm-ről szkennelve (1280x720p) | 29,5 | 34,8 | 60,9 |
| 300mm-ről szkennelve (1280x720p) | 29,1 | 33,9 | 61,1 |

A tesztekből jól látható, hogy 400 mm-es távolságból mm alatti szkennelés végezhető el. A következő teszt egy bonyolult felületű test beszkennelése és nyomtatása volt.

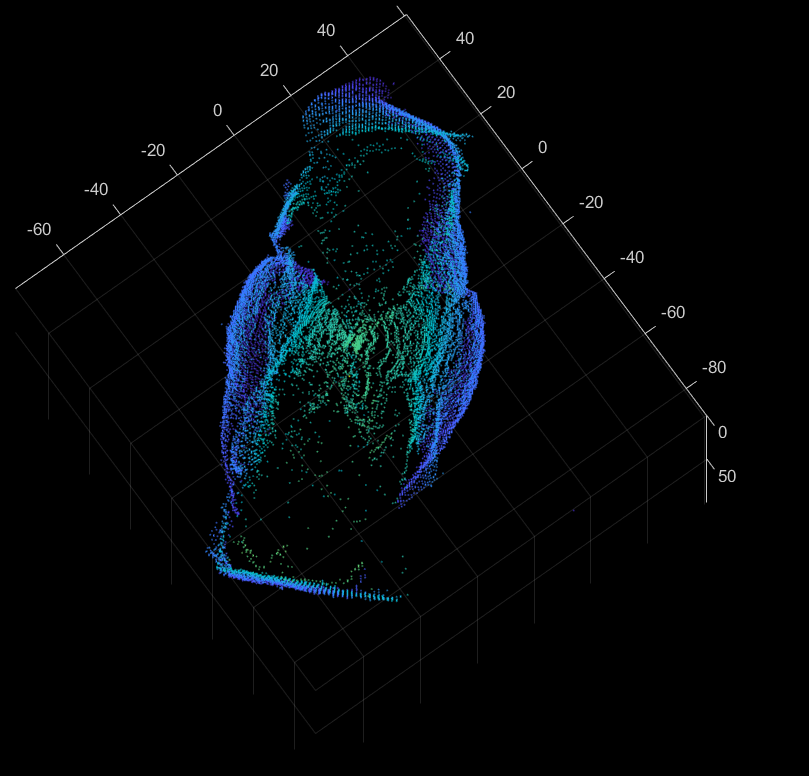


6‑1. ábra: Szkennelt szobor

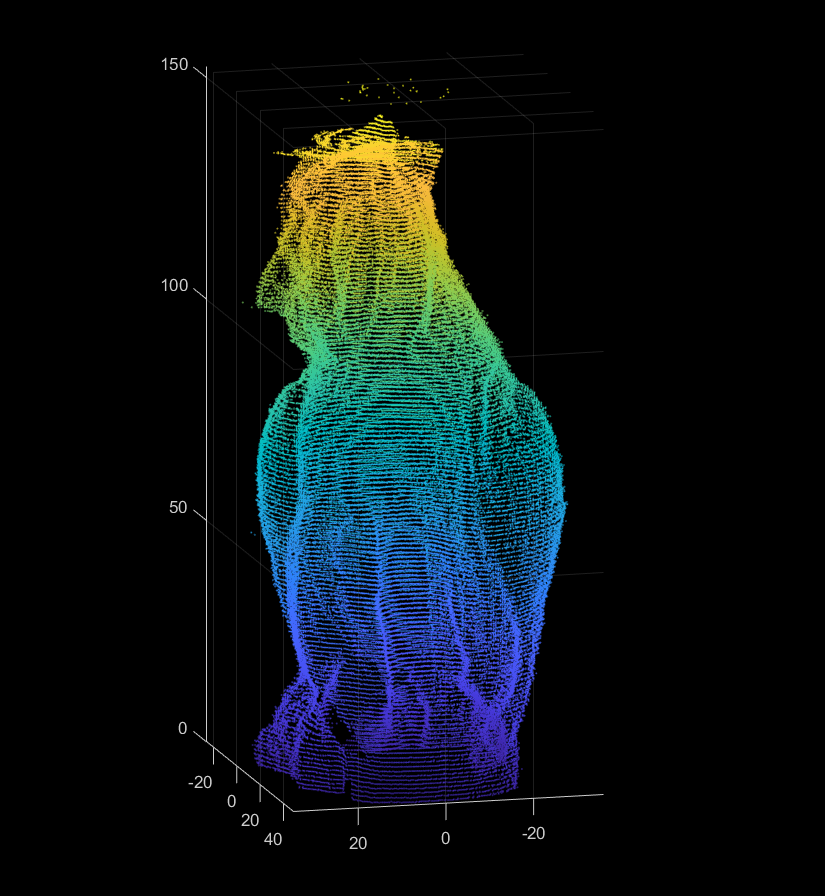
A generált pontokat Meshlab-ba helyezve először újra számoltam a pontokhoz illeszkedő normálisokat majd egy Poisson Surface Reconstruction nevű paranccsal stl fájlként menthetővé tettem a modellt. Ezek után Cura nevű Slicer programban megalkottam a nyomtatási szeleteket.



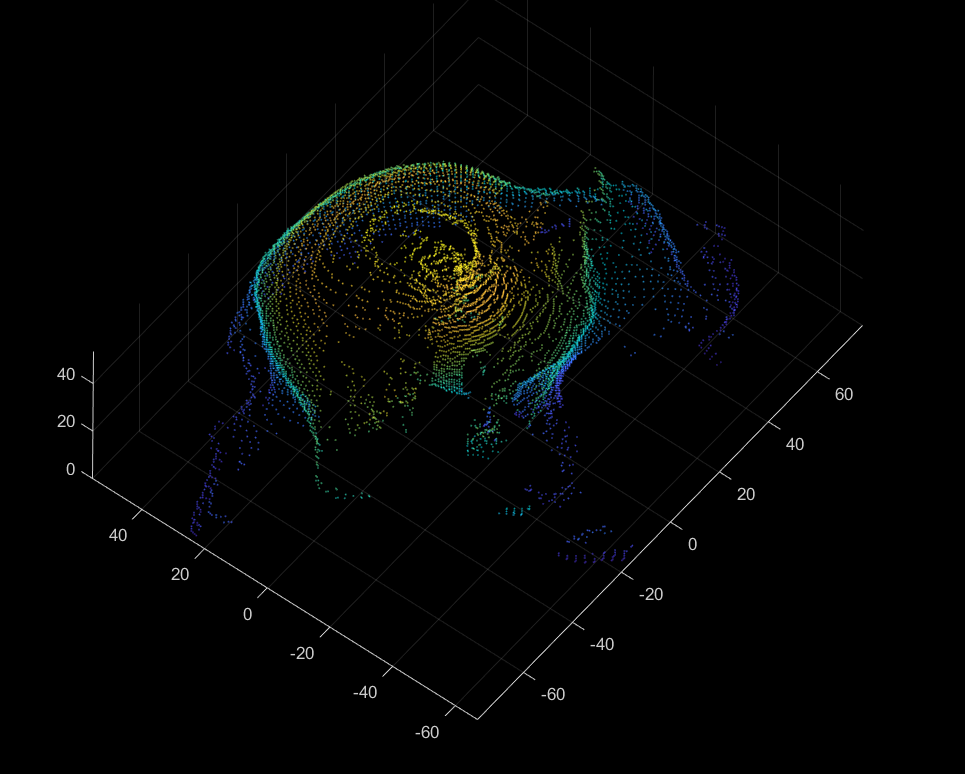
6‑2. ábra: Nyomtatási eredmény

Az elkészült elemen jól látszik, hogy méret és alakpontos, a mélyebb redők láthatóságát még a technika finomításával növelni kell.

6‑3. ábra: Szkennelt bagolyszobor fektetve



6‑4. ábra: Szkennelt bagoly szobor álló-fekvő ponthalmaz illesztve



6‑5. ábra: Szkennelt elefánt szobor

## Következtetések és javaslatok

Az elkészült szkenner első prototípushoz mérten jól működik, bizonyos körülmények között és hozzáértéssel szép modelleket lehet készíteni. További fejlesztésre leginkább a képfeldolgozó szoftver szorul. Első lépésként érdemes lenne a már működőképes elemeket MATLAB-ról átírni egy olyan nyelvre, ami szintén rendelkezik hasznos könyvtárakkal a képfeldolgozás terén, de ugyanakkor képes jobban, gyorsabban kezelni a szkenner és a számítógép kapcsolatát. Fontos, hogy könnyen kezelhető grafikus megjelenítést lehessen megvalósítani. Erre alkalmas lenne egy Python-ban írt szoftver, ami esetleg kisebb-nagyobb módosításokkal a Raspberryre is felkerülne így a felhasználó közvetlenül csatlakoztathat monitort és egeret a szkennerhez.

# Felhasznált források

1. <https://www.aniwaa.com/best-3d-scanner/>w
2. <https://www.3dnatives.com/en/3d-scanner-laser-triangulation080920174-99/>
3. <http://www.mogi.bme.hu/letoltes/BIOMECHATRONIKAI%20&%20BIOMECHANIKAI%20T%c3%81RGYAK/Biomechatronika_Projekt/Info/3D_lezerszkenner_dokumentacio_tard_bognarm_vargar.pdf>
4. <https://www.3dnatives.com/en/structured-light-projection-3d-scanning/>
5. A távérzékelés fogalma, a fotogrammetria és a távérzékelés kapcsolata - Balázsik Valéria (2010)
6. <https://3dee.hu/product-category/3d-scanner/>
7. <https://matterandform.net/scanner>
8. Zhengyou Zhang, A Flexible New Technique for Camera Calibration. Microsoft Research Technical Report MSR-TR-98-71, Microsoft Corporation, Redmond, WA 98052, 2 December, 1998.
9. A gépi látás és képfeldolgozás párhuzamos modelljei és algoritmusai - Dr. Rövid András, Dr. Vámossy Zoltán, Dr. Sergyán Szabolcs (2014) Typotex Kiadó
10. Számitógépes Látás-Kató Zoltán Czúni László (2011) Typotex Kiadó
11. <https://uk.mathworks.com/discovery/raspberry-pi-programming-matlab-simulink.html>
12. <http://www.meshlab.net/>
13. <https://ultimaker.com/software/ultimaker-cura>

# Summary

Ez a dolgozat azért íródott, hogy bemutassa egy 3D lézerszkenner prototípustervezésének fázisait, az alapötletek mérnöki megvalósításának menetét és az eredmények kritikai elemzését.

Az iparban, egészségügyben és a művészet világában is találkozhatunk 3D szkennerekkel. Ezek különböző elven működnek attól függően, hogy milyen feladatot látnak el. Például egy épület digitalizálásához más berendezés szükséges, mint egy kisebb részletes alkatrész vagy az emberi test szkenneléséhez. Azonban alapvetően minden eszköz kamerából vagy kamerákból és legtöbbször valamilyen speciális fényforrásból áll. Kamerák által rögzíthető a tárgyról visszaverődő fény, ami hasznos információval rendelkezik annak alakjáról vagy megjelenéséről. Ez az információ a képekből képfeldolgozó szoftver segítségével nyerhető ki.