

Niko Viitake

VÄRINTUNNISTUSKONENÄKÖJÄRJESTELMÄ

Automaatiotekniikan koulutusohjelma

2017

VÄRINTUNNISTUSKONENÄKÖJÄRJESTELMÄ

Viitake, Niko
Satakunnan ammattikorkeakoulu
Automaatiotekniikan koulutusohjelma
Huhtikuu 2017
Ohjaaja: Leino, Mirka
Sivumäärä: 55

Asiasanat: konenäkö, värintunnistus

Tämän opinnäytetyön aiheena oli suunnitella ja rakentaa konenäköjärjestelmä, jonka avulla pystytään tunnistamaan kappaleiden värejä tarkasti. Konenäköjärjestelmän tuli olla soveltuva esittelytarkoituksiin, jotta pystytään havainnollistamaan konenäön toimintaa värintunnistuksessa. Järjestelmän tuli olla näyttävä, mielenkiintoinen, helposti kuljetettava, sekä konenäköä helposti havainnollistava, jotta se olisi helposti ymmärrettävä myös konenäöstä vähemmän tietäville ihmisille.

Opinnäytetyön teoriaosuudessa käsitellään yleisesti kameroiden kuvanmuodostamista, sekä konenäkökameroita ja valaistuksen perusteita.

Opinnäytetyön käytännön osuus sisältää järjestelmän mekaanista suunnittelua, ohjelmointia, sähköistä suunnittelua ja järjestelmän rakentamisen vaiheet.

COLOUR IDENTIFICATION WITH MACHINE VISION SYSTEM

Viitake, Niko

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in Automation Technology

April 2017

Supervisor: Leino, Mirka

Number of pages: 55

Keywords: machine vision, colour identification

The purpose of this thesis was to design and build a machine vision system that is capable of identifying object's colour very precisely. The machine vision system was meant to be suitable for demonstrating the use of machine vision system in colour identification. Requirements for the system were that it should look appealing and interesting as well as to be easy to transport and easy to understand even for those who know little or nothing of machine vision systems.

The theory part of the thesis consists of general camera technology, machine vision generally and illumination basics.

The practical part consists of system's mechanical designing, programming, electrical designing and steps of building the system.

SISÄLLYS

1	JOHDANTO.....	5
1.1	Tavoite	5
2	KONENÄÖN TEORIAA.....	6
2.1	Mitä on konenäkö ja mihin sitä käytetään?.....	6
2.2	Optiikka.....	7
2.3	Kameratyypit.....	9
2.4	Kamerakennot.....	10
2.5	Valaistus.....	10
2.6	Värien tunnistaminen kamerassa	13
2.7	Kuvan analysointi	14
3	JÄRJESTELMÄN SUUNNITTELU	15
3.1	Yleinen suunnittelu	15
3.2	Mekaaninen suunnittelu ja 3D-mallinnus	16
3.3	Sähköiset komponentit ja niiden kytkennät	26
3.3.1	Kamera	26
3.3.2	Valaisin	28
3.3.3	Raspberry PI ja käyttöliittymä.....	29
3.3.4	Piirikaaviot	32
4	OHJELMOINTI	35
4.1	Älykameran ohjelmointi	35
4.2	Raspberry PI -ohjelmointi.....	40
4.2.1	Ohjelman toiminnallisuus.....	41
4.2.2	Miten ohjelma toimii?	42
5	RAKENTAMINEN.....	47
6	KÄYTTÖOHJEET.....	52
7	POHDINTA.....	53
	LÄHTEET.....	54

1 JOHDANTO

Opinnäytetyöni tarkoituksena oli suunnitella ja rakentaa Satakunnan ammattikorkeakoulun TKI-laboratorioon värintunnistuskonenäköjärjestelmä, jota käyttäen voidaan havainnollistaa konenäön vahvuuksia värien tarkastelussa.

1.1 Tavoite

Opinnäytetyön tavoitteena oli suunnitella ja rakentaa tarkkaan värintunnistukseen kykenevä konenäköjärjestelmä. Järjestelmän toivottiin olevan kiinnostava ja näyttävä. Siinä pitäisi myös olla jotakin liikettä ja sitä pitäisi olla helppo siirtää paikasta toiseen. Tiivistettynä tavoite oli siis rakentaa sellainen järjestelmä, joka herättäisi ihmisten mielenkiinnon konenäöllä tehtävään värintunnistukseen ja tiedottaa yrityksiä tämän kaltaisesta teknologiasta, jotta yritykset voisivat mahdollisesti parantaa omaa toimintaansa konenäön avulla.

2 KONENÄÖN TEORIAA

2.1 Mitä on konenäkö ja mihin sitä käytetään?

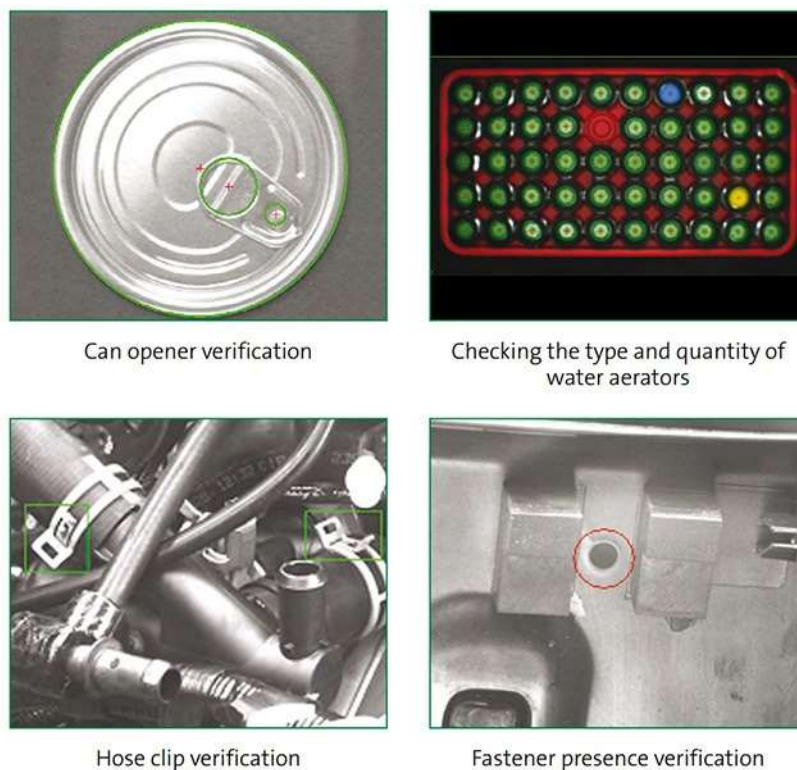
Toiminnaltaan konenäkö muistuttaa ihmissilmää, mutta se on nopeampi, tarkempi, luotettavampi ja väsymätön. Konenäöllä on kaikki samat elementit kuin ihmissilmällä eli optiikka, ohjausyksikkö, kuvanmuodostuselin, ohjausalgoritmi, sekä valaistuksen säätö. Konenäön avulla voidaan parantaa toiminnan kannattavuutta, sillä se ei pidä taukoja, eikä sille makseta palkkaa niin kuin ihmisille. Konenäkö parantaa laatua ja nopeutta, sillä parhaimmillaan kamerat ovat ihmissilmään verrattuna moninkertaisesti tarkempia ja nopeampia. Konenäkö vapauttaa työvoimaa turhasta tarkastustoiminnasta sellaisiin työtehtäviin, joita vain ihminen pystyy hoitamaan. Konenäön yleisimpiä sovelluskohteita ovat mm. tarkastustehtävät ja mittaustehtävät. (Savonautomaatio WWW-sivut, STEMMER-Imaging WWW-sivut)

Tarkastustehtävät voivat sisältää esimerkiksi:

- juotoksen tarkastus
- paketoinnin todentaminen
- tölkin tai pullon korkin tarkastus
- piirilevyn kokoonpanon tarkastus
- maalipinnan värin tai eheyden tarkastus

Mittaustehtävät voivat sisältää esimerkiksi:

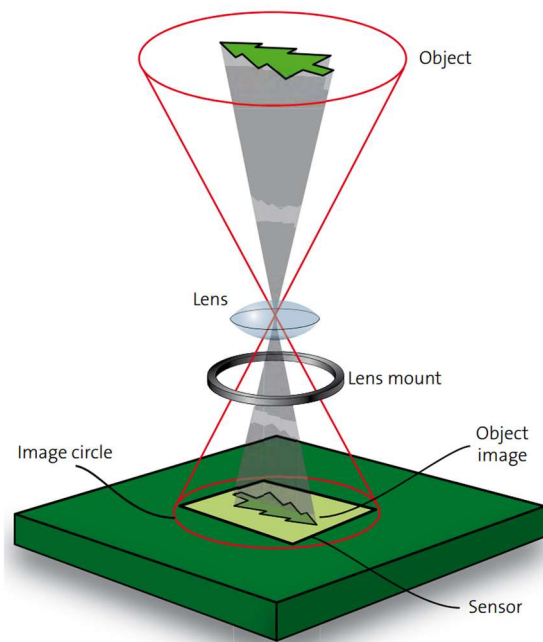
- jousen mittaus
- sylinterin mittaus
- tiivisteiden mittaus
- reikien oikean sijainnin mittaus



KUVA 1. Esimerkkejä käyttökohteista (STEMMER-Imaging WWW-sivut)

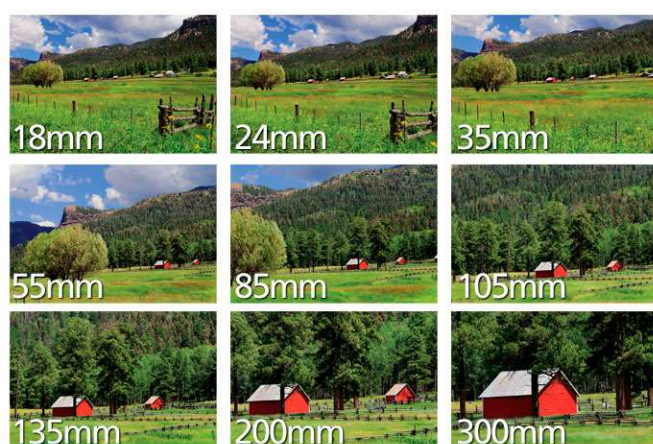
2.2 Optiikka

Kaikki kamerat tarvitsevat objektiivin kerätäkseen valon kohteesta ja johtaakseen kuvan objektiivin toiselle puolelle, valoherkälle pinnalle, joka on yleensä CCD- tai CMOS-kenno. Kennolle muodostuvan kuvan kokoa saadaan säädeltä eri-tyyppisiä objektiivin polttovälejä käyttämällä. Useimmat nykyaikaiset kamerat sisältävät objektiivin, jonka polttoväliä pystytään muuttamaan. (Nikonusa WWW-sivut, STEMMER-Imaging WWW-sivut)



KUVA 2. Objektiivin periaate (STEMMER-Imaging WWW-sivut)

Objektiivit voidaan jakaa kahteen luokkaan polttovälin säädettävyyden mukaan. Nämä ovat kiinteäpolttovälinen objektiivi ja muuttuvapolttovälinen objektiivi. Muuttuvapolttovälisen objektiivin etuna on monipuolisuus. Ne ovat ihanteellisia kuvatessa monia eri kohteita eri etäisyyksillä, jolloin kuvaukset voidaan tehdä vain yhdellä objektiivilla. Kiinteiden objektiivien suurimmat edut ovat mm. niiden koko, paino ja valotusaukon suuruus. Valotusaukon suuruus kuvastaa sen valonkeräyskykyä, eli mitä suurempi se on, sitä parempia kuvia saadaan hämärissä olosuhteissa. (Nikonusa WWW-sivut, STEMMER-Imaging WWW-sivut)

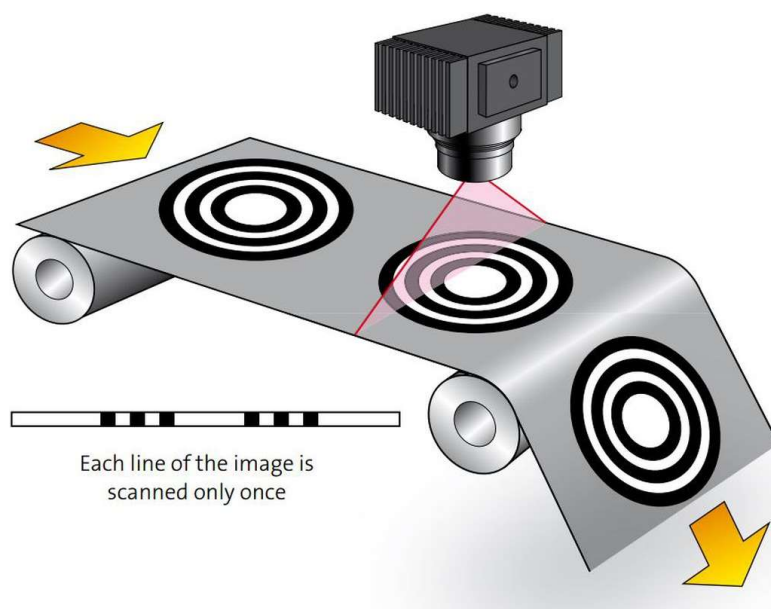


KUVA 3. Eriaisien polttovälin omaavien optiikoiden avulla otettuja kuvia (Nikonusa WWW-sivut)

2.3 Kameratyypit

Kamerat voidaan jakaa muun muassa kahteen eri tyyppiin, jotka ovat matriisi- ja viivakamerat. Matriisikennoiset kamerat ovat yleisimmin käytettyjä kameroita ja lähes poikkeuksetta kaikki kuluttajien kamerat ovat sellaisia. Niillä saadaan kaksiulotteinen suorakulmion muotoinen kuva, jonka tarkkuus riippuu kameran ominaisuuksista. Matriisikameran kennossa on pikseleitä vaaka- ja pystysuunnassa, jolloin koko kuvattava kappale saadaan kuvaan yhdellä otolla. (SAMK automaation tutkimusryhmän WWW-sivut)

Viivakameralta saadaan viivamainen kuva, jossa on 1-3 pikseliä pystysuunnassa ja sivusuunnassa 1024 – 16 000 pikseliä, riippuen kameran ominaisuuksista. Viivakameroita käytetään usein liikkuvien kohteiden kuvaamiseen ja erityisesti liikkuvien yhtenäisten kohteiden kuvaamiseen. Käyttökohteita ovat mm. paperin valmistus, postien lajittelu, viivakoodien luku, sekä painokoneiden tulostuksen tarkastus. (STEMMER-Imaging WWW-sivut, baslerweb WWW-sivut, SAMK automaation tutkimusryhmän WWW-sivut)



KUVA 4. Viivakennoinen kamera (STEMMER-Imaging WWW-sivut)

2.4 Kamerakennot

Kuvaamisessa käytetään tyypillisesti joko CCD-kennoa (Charge-Coupled Device) tai CMOS-kennoa (Complementary Metal Oxide Semiconductor). (STEMMER-Imaging WWW-sivut)

CCD-kennoissa on miljoonia fotodiodeja, jotka ovat valolle herkkiä. Optiikan kautta kennolle tulevat fotonit muuttavat fotodiodin varausta. Valon intensiteetti siis vaikuttaa fotodiodin varauksen muutokseen. Kennolta luetaan jokaisen diodin varaus ja se muunnetaan digitaaliseksi, jolloin saadaan luotua kuva. Kuvassa pikseli on sitä vaa-leampi, mitä enemmän fotodiodin varaus on muuttunut. Varaukset siirtyvät pikseliltä toiselle, pyrkien lopulta kennon ulkopuolella olevalle A/D muuntimelle luettavaksi. CCD-kennojen etu on ainakin toistaiseksi parempi kuvanlaatu verrattuna CMOS-kennoihin. Huonoja puolia ovat suuri virran kulutus, sekä hankala valmistustekniikka, mikä nostaa hintaa. (STEMMER-Imaging WWW-sivut)

CMOS-kennoissa jokaisessa pikselissä itsessään tehdään varauksen muunnos jännit-teiseksi transistorien avulla. Pikselit usein sisältävät myös vahvistimen, melunkorjauk-sen ja digitalisointipiirejä. Tämän ansiosta itse kameraan tarvitaan vähemmän piirejä lukemaan signaalia. Lisäksi muita hyviä puolia ovat sen alhainen virran kulutus ja hal-vempi valmistushinta verrattuna CCD-kennoihin. (STEMMER-Imaging WWW-sivut)

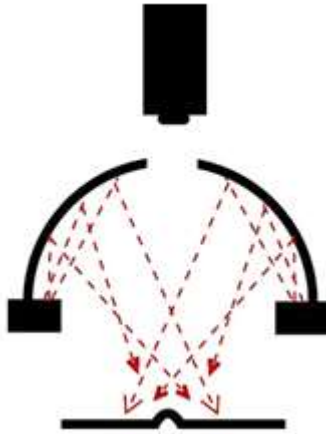
2.5 Valaistus

Oikeanlaisen valaistuksen valitseminen on yksi konenäköjärjestelmän tärkeimmistä tekijöistä. Mikäli on valittu väärä valaistus, voidaan menettää hyvin tärkeää tietoa kappaleesta, koska kuvan laatu on heikko. Puuttuvaa tai väärää tietoa ei välttämättä saada mitenkään takaisin. Esimerkkinä huono kontrastiero voi johtaa kappaleen reunan tun-nistuksen virheeseen. Hyvin tärkeää valaistuksen suunnittelussa on se, että ulkoiset tekijät eivät muuta kuvaa, vaan valaisuolosuhteet pysyvät mahdollisimman samanlai-sina koko ajan. Oikeanlaisen valaisun saavuttamisessa tulee pohtia mm. kohteen väriä, materiaalia, muotoa, yksityiskohtia ja mitä kuvasta halutaan erottaa.

(STEMMER-Imaging WWW-sivut)

Yleisimpiä valaistustekniikoita:

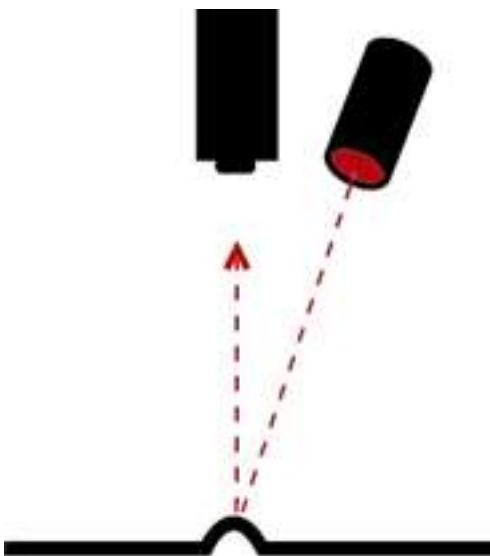
Diffuusikupolivalaisin



KUVA 5. Diffuusikupolivalaisimen periaate (National Instruments WWW-sivut)

Diffuusikupolivalaisimet poistavat varjot ja heijastukset, minkä takia ne sopivat hyvin kaareville ja kiiltäville pinnoille. Hyvä kuvauskohde on esimerkiksi laakeri. Diffuusikupolivalaisimen sisäreunaan on upotettuna ylöspäin suunnattuja ledejä, jotka lähettävät valonsäteet kupolin sisäpinnalle, josta ne heijastuvat kohteen pintaan. (National Instruments WWW-sivut)

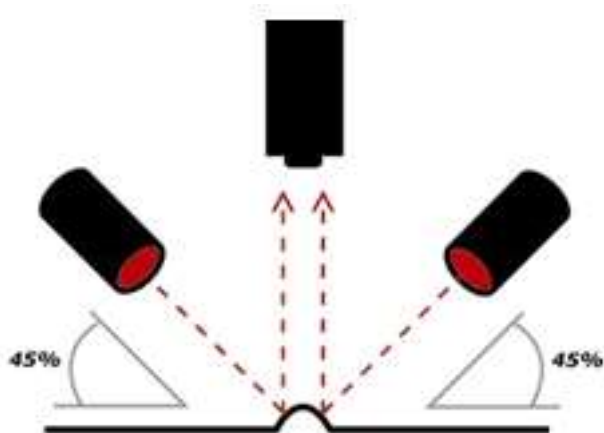
Kohtisuora valaisu



KUVA 6. Kohtisuoran valaisun periaate (National Instruments WWW-sivut)

Kohtisuora valaisu on yleisin valaistustekniikka ja käytämme sitä päivittäin, sillä aurinko on eräänlainen kohtisuora valaisin. Se on hyvä tapa luoda kontrastia kuvattavaan kohteeseen, mutta ei toimi hyvin kiiltävien pintojen kanssa. (National Instruments WWW-sivut)

Sivuväläisy



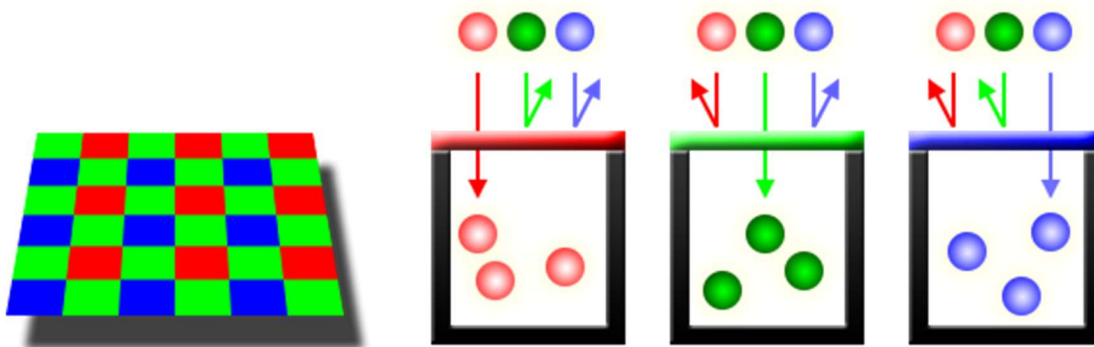
KUVA 7. Sivuväläisyyden periaate (National Instruments WWW-sivut)

Sivuväläisyydellä saadaan erotettua hyvin pinnan muodot. Tarkoituksena on asentaa valaisimet niin, että ne valaisevat lähes suoraan sivulta tai pienessä kulmassa. (National Instruments WWW-sivut)

2.6 Värien tunnistaminen kamerassa

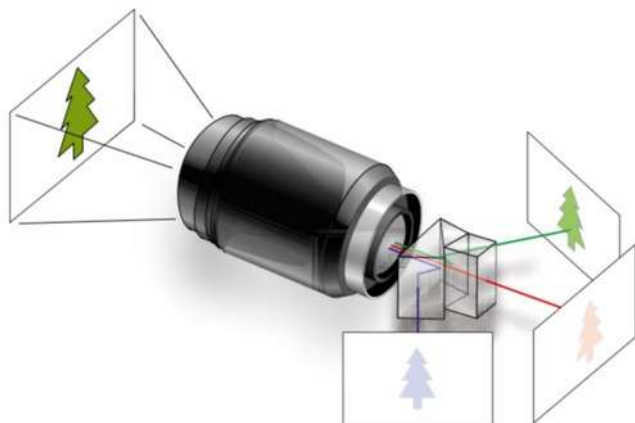
Kuten useimmat ihmiset näkevät maailman eri väreillä perusodotuksena on, että kameralla otetut kuvat ovat värillisiä. Konenäössä väreillä ei yleensä ole merkitystä, jos halutaan vain tunnistaa kontrastieroja. Mikäli konenäköjärjestelmällä ei tutkita värejä, kannattaa aina käyttää harmaasävykameraa, koska se vangitsee jopa kolme kertaa enemmän valoa, sekä antaa tarkempaa tilallista tietoa kohteesta, kuin yleisimmät värikamerat. (STEMMER-Imaging WWW-sivut)

Yleisimmät värikamerat perustuvat siihen, että CCD- tai CMOS-kennon pikseleiden päällä on suodatin, joka päästää vain tietyn väristä valoa läpi. Yleisin käytetty suodatin on Bayer-Filter (KUVA 8). Yksittäinen pikseli varautuu vain punaisesta, vihreästä tai sinisestä valosta, riippuen suodattimen väristä. Muut värit kuin punainen, vihreä, sininen lasketaan yksittäistä pikseliä ympäröiville pikseleille tulleen valon arvosta. Tällä menetelmällä jokainen pikseli saa vain kolmasosan sille tulevasta valosta, koska muun väriset valot suodatetaan (KUVA 8). (Red digital cinema WWW-sivut, STEMMER-Imaging WWW-sivut)



KUVA 8. Bayer-Filter (Red digital cinema WWW-sivut)

Kun halutaan mahdollisimman tarkkaa väritietoa, on paras vaihtoehto kolmikennon kamera eli niin sanottu 3-chip colour camera (KUVA 9), jossa on prisma erottamassa valon eri värit. Kamerassa on kaikille kolmelle päävärille erilliset kennot, jolloin valoa ei mene hukkaan niin kuin perinteisissä värikameroissa. Näiden kameroiden huono puoli on se, että ne ovat kalliita ja isompia, koska niissä on kolme kennoa. Sen lisäksi näissä kameroissa ei voida käyttää normaaleja objektiiveja vaan erikoisvalmisteisia värikorjattuja objektiiveja. (STEMMER-Imaging WWW-sivut)

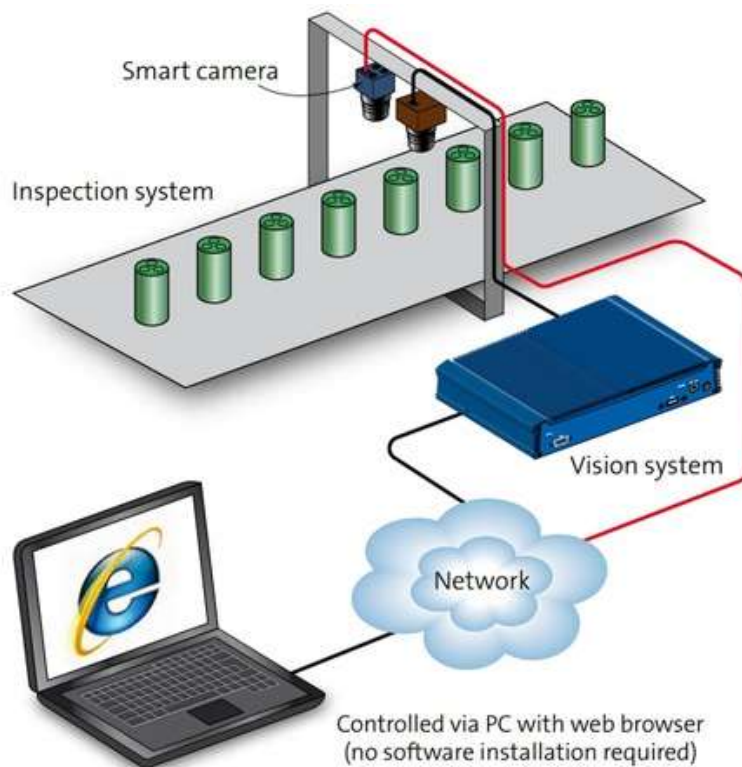


KUVA 9. Kolmikennokameran toimintaperiaate (STEMMER-Imaging WWW-sivut)

2.7 Kuvan analysointi

Jokaisen konenäön analysointiohjelmiston perustana on kyky käskää kameraa, siirtää dataa ja manipuloida sitä, sekä tulkita ohjelmistolle tulevaa dataa. Ohjelmistoilla pystytään tekemään hyvin yksinkertaisia asioita kuten esimerkiksi vain tallentamaan kuva kovalevylle, tai hyvin monimutkaisia kuviontunnistustoimintoja. (STEMMER-Imaging WWW-sivut)

Älykameroiksi kutsutaan sellaisia kameroita, jotka ottavat kuvan ja analysoivat sen kamerassa olevalla ohjelmalla (KUVA 10). Ne ovat hyvin käteviä, sillä niiden ohjaamiseen ei tarvita ulkoista tietokonetta tai ohjauspiiriä. Kameraa on myös mahdollista ohjata ulkoisella tietokoneella tai ohjauspiirillä. Ohjelmistot sisältävät yleensä erilaisia valmiiksi rakennettuja toimintoja, joita käyttäjä pystyy valitsemaan tarpeensa mukaan. Kun haluttu kuva on analysoitu ohjelmistossa, on mahdollista lähettää komento ulkoiselle järjestelmälle esimerkiksi tuotteen hyväksymisestä tai hylkäämisestä. (STEMMER-Imaging WWW-sivut)



KUVA 10. Esimerkki konenäköjärjestelmän rakenteesta (STEMMER-Imaging WWW-sivut)

3 JÄRJESTELMÄN SUUNNITTELU

3.1 Yleinen suunnittelu

Suunnittelun perustana oli, että järjestelmän tulisi olla kiinnostava, näyttävä, siinä olisi jotakin liikettä, sekä sen tulisi olla helposti siirrettävä. Suunnittelussa oli siis hyvin vapaat kädet ja ensin lähdettiin pohtimaan, minkälaista liikettä järjestelmään suunniteltaisiin. Päädyttiin ratkaisuun, että rakennettaisiin pyörivä alusta, jossa on lokeroita. Kameralla kuvattaisiin yläpuolelta alustaa lokero kerrallaan ja lokeroihin voisi laittaa erilaisia kuvauskohteita. Pyörivään alustaan päädyttiin siksi, että se täytti kaikki annetut kriteerit kuten; kiinnostavuus, näyttävyys ja siinä on liikettä. Todettiin, että pyörivä alusta olisi siksi kiinnostava, että ihmiset voisivat laittaa lokeroihin käytännössä lähes mitä vain ja se olisi hyvin helppo ymmärtää myös ihmisille ilman teknistä taustaa.

Pyörivä alusta ja kamera tarvitsivat jonkinlaisen tukirungon, johon ne saataisiin kiinnitettyä siististi. Pohdinnan jälkeen päädyttiin kuutiomaiseen ratkaisuun, eli kuution päälle tulisi pyörivä alusta ja kuution sivuilta saataisiin rakennettua tukirunko kameralle, joka ylettäisi pyörivän alustan yläpuolelle. Pyörivän alustan alapuolelle saataisiin hyvin suunniteltua akseli, joka menee kuution sisäpuolelle ja akselia pyöritettäisiin moottorin avulla. Kuutioon päädyttiin myös siksi, että se olisi näyttävä, sillä sen sisään voidaan asentaa ja suunnitella kaikki elektroniikka. Kuutio olisi myös helposti siirrettävä, koska kaikki osat ovat suoraan siinä kiinni.

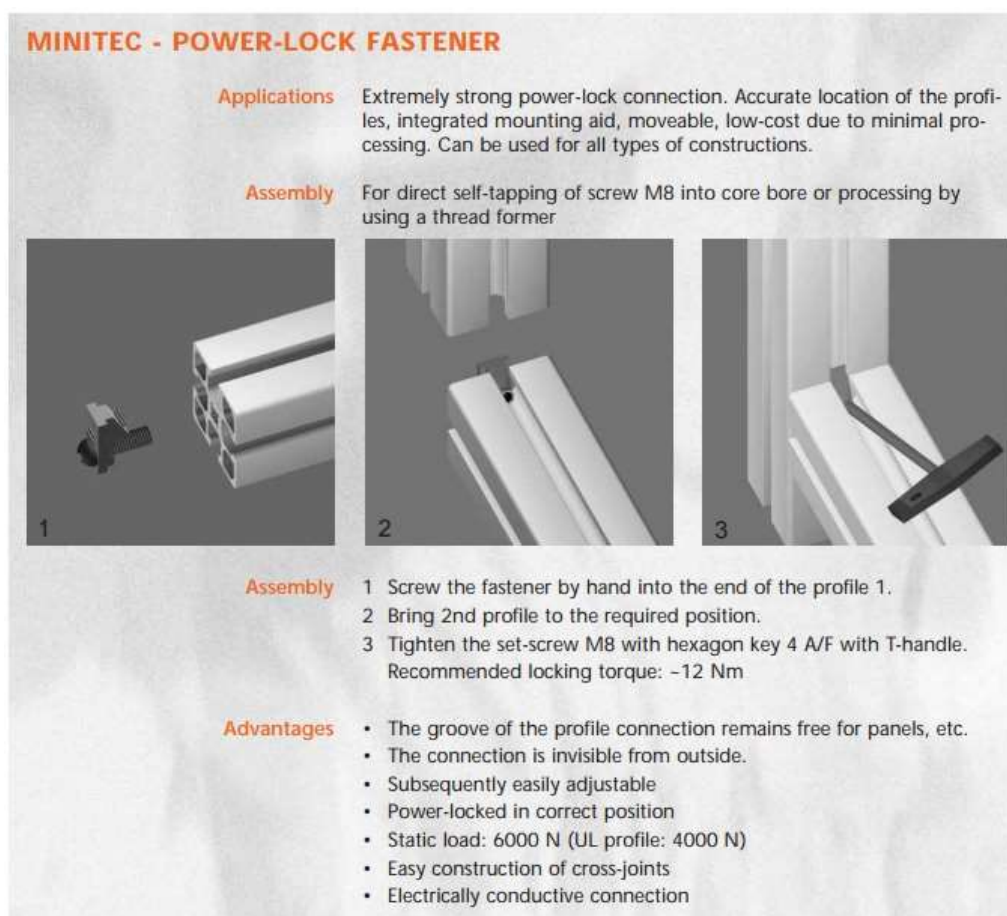
Seuraavaksi alettiin ideoimaan järjestelmän elektronista puolta, koska moottorin tulisi pystyä toimimaan yhteistyössä kameran kanssa. Tarvittaisiin siis käyttöliittymä ja ohjauspiiri, joka toimii järjestelmän aivoina eli kykenee ohjaamaan moottoria, sekä lähettämään ja vastaanottamaan dataa älykameralta. Todettiin, että järjestelmän kuutiomainen runko on täydellinen, koska kuution päälle saisi asennettua pienen kosketusnäytön, joka toimisi käyttöliittymänä.

Järjestelmän ominaisuuksia suunniteltiin yleisellä tasolla ja päädyttiin siihen, että alustan tulisi pyöriä niin, että se pysähtyisi aina kunkin lokeron kohdalla. Kamera sitten ottaisi kuvan, jonka jälkeen se analysoitaisiin ja siirrettäisiin tieto käyttäjälle. Todettiin, että paras ja selkein tapa esittää data käyttäjälle olisi siirtää se käyttöliittymälle, mutta tulisi olla myös mahdollista nähdä kameran dataa suoraan tietokoneelta. Pohdinnan jälkeen todettiin siis, että kameran, ohjauspiirin ja käyttöliittymän välillä pitää pystyä siirtämään dataa ja ohjauskomentoja.

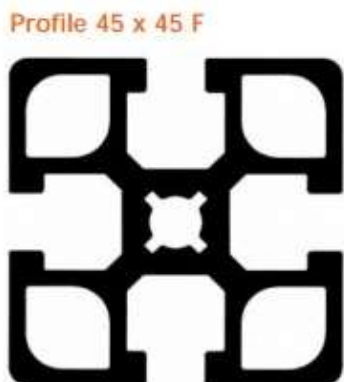
3.2 Mekaaninen suunnittelu ja 3D-mallinnus

Kaikki järjestelmän mekaaniset osat 3D-mallinnettiin Solidworks ohjelmalla. Mekaaninen suunnittelu aloitettiin suunnitteleamalla ensin kuution muotoinen järjestelmän runko. Päädyttiin käyttämään alumiiniprofiileja, koska ne ovat kestäviä, siistin näköisiä ja helppo liittää yhteen kuutiomaiseen muotoon. Kuution sivujen pituudeksi päätettiin 590mm ja korkeudeksi 200mm (KUVA 13). Tähän kokoon päädyttiin, jotta pyörivän alustan kooksi saatiin kohtalaisen iso eli halkaisijaltaan 300mm. Näillä mitoilla jäi tilaa kuution päälle tulevalle käyttöliittymälle. Todettiin, että kuution koko

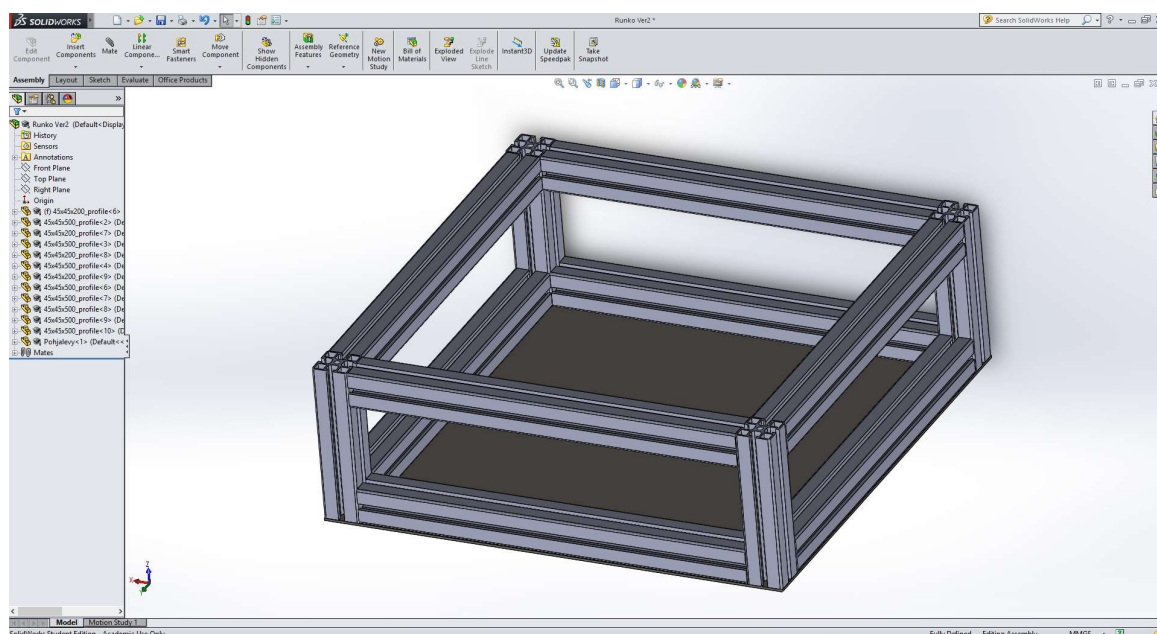
olisi tarpeeksi myös siihen, että sen sisään mahtuisi kaikki elektroniikka, moottori, akseli ja rattaat. Tässä vaiheessa ei ollut vielä tarkkaa tietoa kuinka paljon kuution sisällä olevat komponentit tulisivat viemään tilaa, mutta todettiin, että tämä olisi tarpeeksi. Rungon sisäpuolen tilavuudeksi laskettiin noin 50 kuutiodesimetriä. Alumiiniprofiileita tarvittiin siis 8 kappaletta 590mm mittaisia, sekä 4 kappaletta 200mm mittaisia ja niiden tyypiksi valittiin 45x45mm (KUVA 12). Profiilit päätettiin kiinnittää toisiinsa Mitec power-lock fastener:illa (KUVA 11), koska ne ovat nopeita asentaa ja silti kestäviä. Profiilit ja niiden kiinnikkeet tilattiin Movetec Oy:ltä.



KUVA 11. Profiilien kiinnitys Power-Lock fastener (Movetec WWW-sivut)



KUVA 12. Alumiiniprofiilin tyyppi (Movetec WWW-sivut)

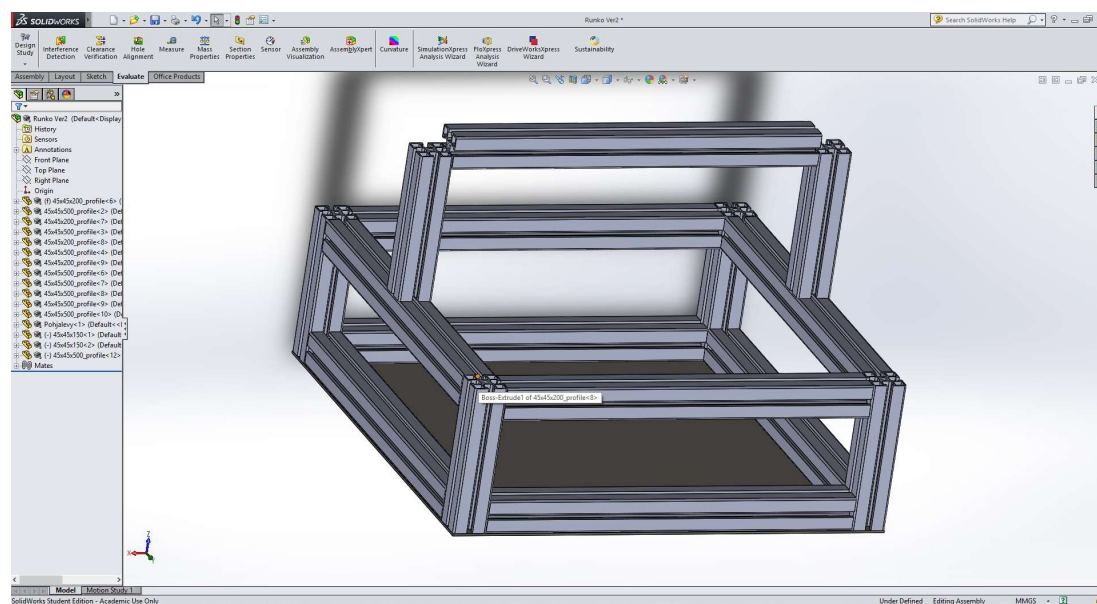


KUVA 13. Alumiiniprofiili rungon ja pohjalevyn 3D-malli

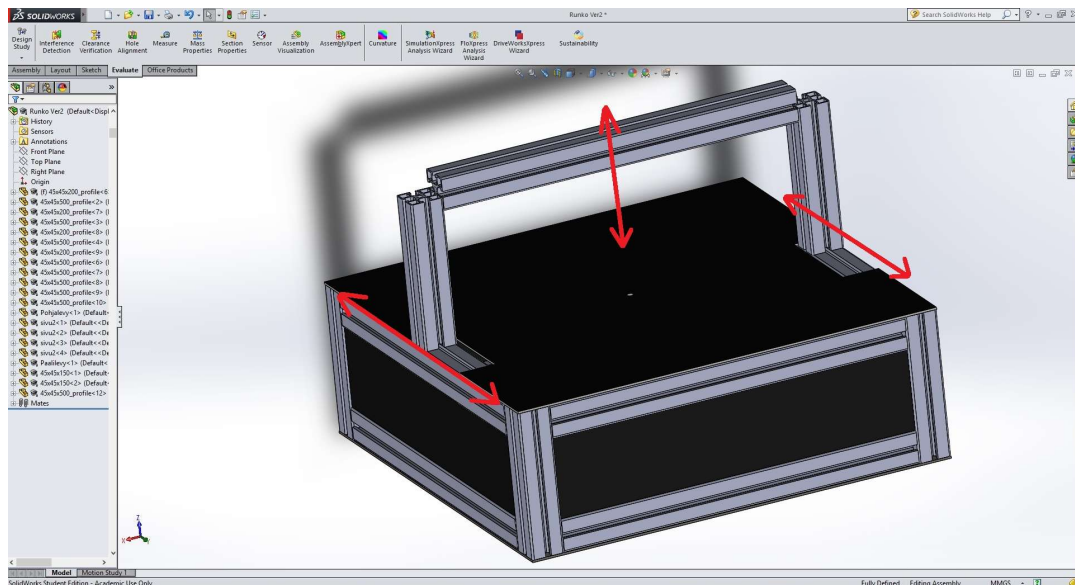
Seuraavaksi suunniteltiin kameralle tukirakenne, johon se kiinnitetään. Tukirakenteen tuli olla sellainen, että se olisi suoraan pyörivän alustan yläpuolella. Päädyttiin jälleen käyttämään alumiiniprofiileja niin, että kaksi profiilia on kiinnitetty vaakatasossa olevan profiilin päälle ja niitä yhdistää kolmas profiili (KUVA 14). Profiileihin päädyttiin myös tässä sen vuoksi, että niitä on helppo yhdistää toisiinsa. Näin suunnitellun kameran tukirakenteen sijainnin muuttaminen ja korkeuden säätäminen on myös hyvin helppoa. Tukirakenteen korkeutta pystytään säätämään liikuttamalla vaakatasossa olevaa profiilia pystysuunnassa. Tukirakenteen sijaintia on myös helppo säätää, koska sitä

voidaan liikuttaa vaakatasossa (KUVA15). Nämä säätömahdollisuudet ovat tärkeitä, jotta kamera saadaan oikeaan kohtaan.

Pohjalevyksi suunniteltiin 9mm paksua puulevyä, jonka mitat olivat 590x590mm. Pohjalevyn suunnittelussa ei otettu mitään muuta huomioon kuin, että levyn tulisi olla helposti saatavilla, levyä pitäisi olla helppo porata ja sen tuli olla tarpeeksi paksua kestämään sisällä olevien elektroniikkakomponenttien painon. Pääilevyksi suunniteltiin samojen kriteerien mukaisesti puulevy, jonka mitat olivat täysin samat kuin pohjalevyn. Pääilevyyn suunniteltiin sivuille lovet, jotta kameras tukirakenteen siirtäminen on mahdollista (KUVA15). Rungon sivuille suunniteltiin 500x110mm muovilevyt, joiden ideana oli vain peittää näkyvyys sisään ja niihin voitaisiin asentaa elektronisia liittimiä tai painikkeita (KUVA15).

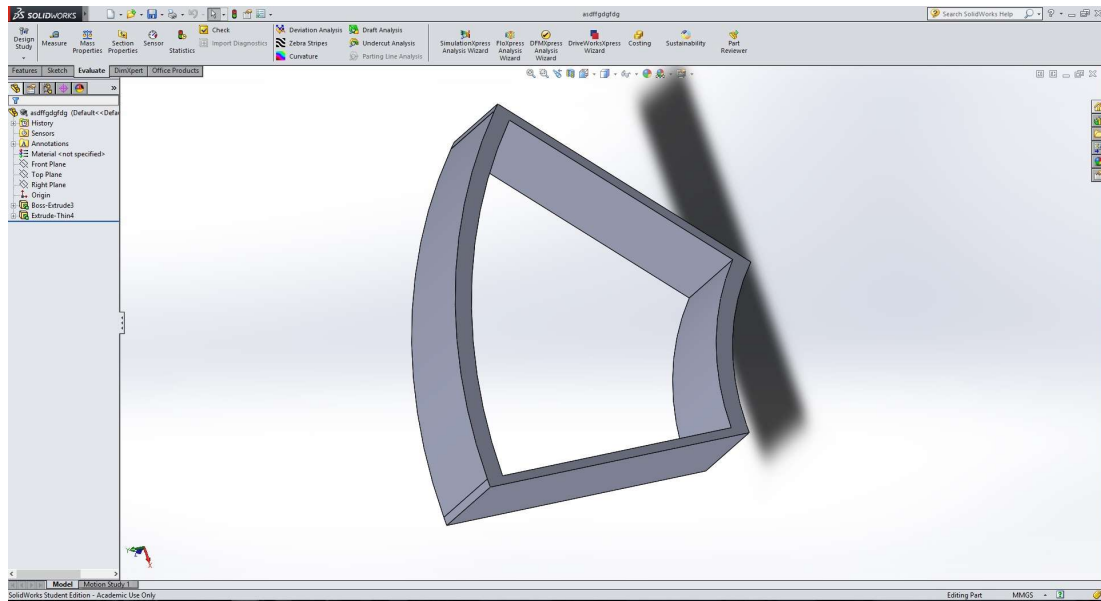


KUVA 14. Alumiiniprofiili rungon, sekä kameras tukirakenne

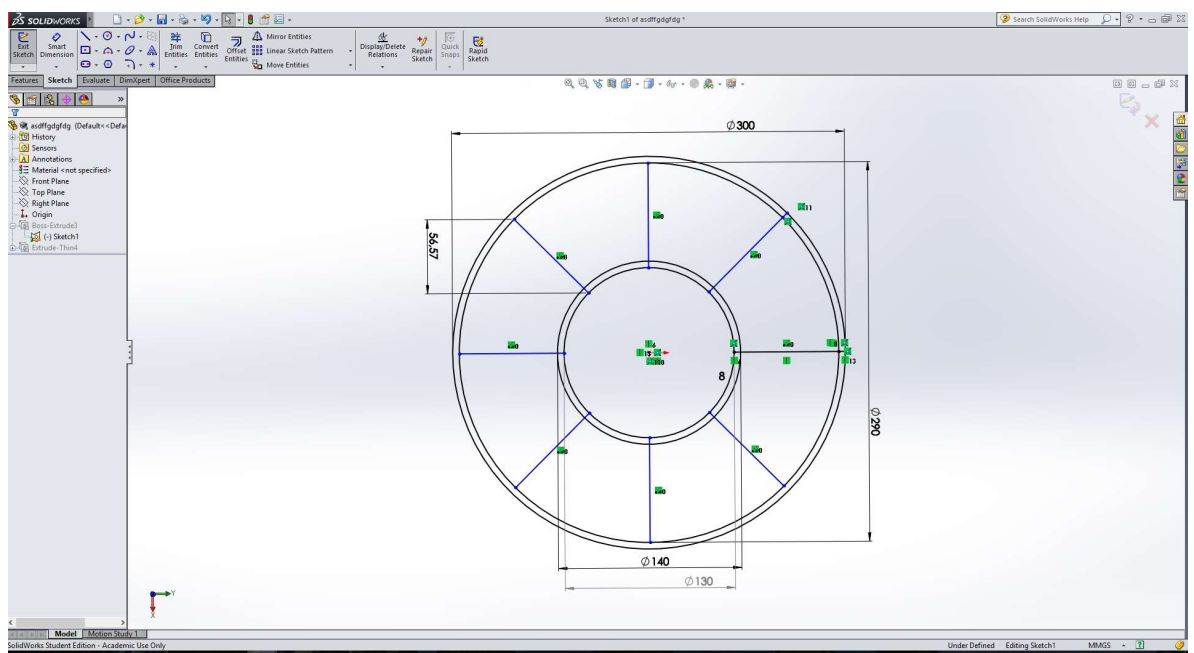


KUVA 15. Kameran tukirakenteen säätö

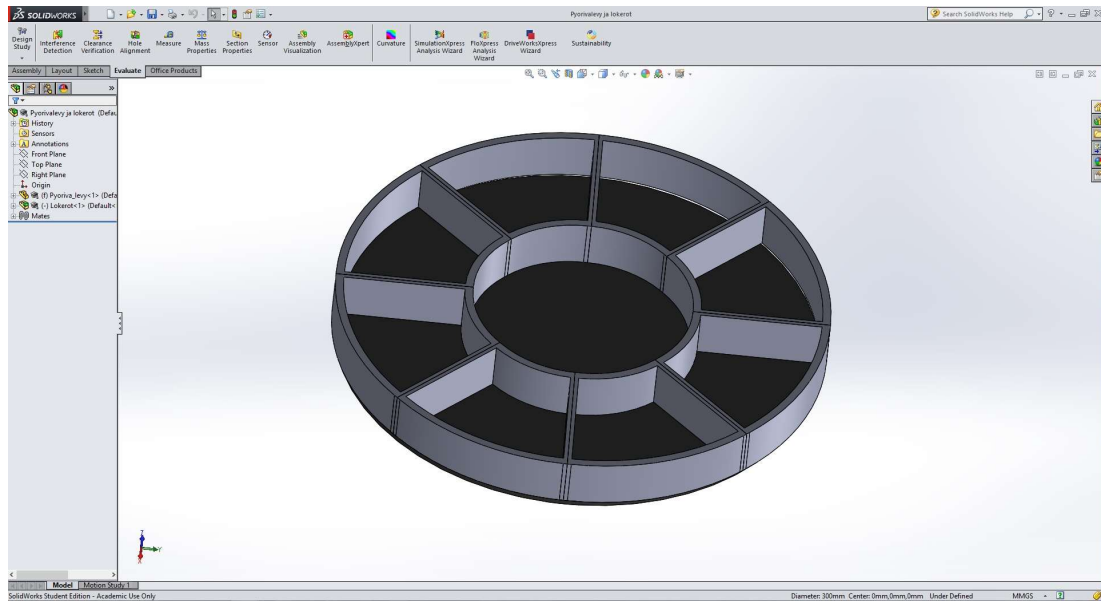
Seuraavaksi suunniteltiin ja mallinnettiin pyörivä alusta, sekä sen lokerot. Päätettiin, että lokeroita tulisi olla kahdeksan kappaletta. Lokerikko suunniteltiin, jotta kappaleet ovat aina samassa paikassa, eli lokeron rajojen sisällä. Lokeron tarkoituksena oli myös, että sen sivut estävät ulkoisen valon pääsyn kappaleelle, jolloin saadaan hyvälaatuisia kuvia kameralla ja valaistusolosuhteet pysyvät stabiileina. Pyörivän alustan halkaisijaksi valittiin 300mm, koska se oli sopivan kokoinen rungolle ja jätti kuitenkin tilaa vielä käyttöliittymälle. Lokerot suunniteltiin itse, koska mitään tähän tarkoitukseen sopivaa ei löydetty valmiina. Satakunnan ammattikorkeakoululla oli käytössä 3D-tulostimia, joten niitä päätettiin käyttää hyödyksi. Mallinnuksen tuloksena oli kuvan 16 mukainen lokero, joita sitten liimattaisiin yhteen 8 kappaletta. Yhteen liimatut lokerot muodostavat halkaisijaltaan 300mm olevan lokerikkorakenteen (KUVA 18). Lokerojen sisäpuolen pinta-ala on noin 81 neliösenttimetriä. Yksittäisen lokeron korkeus on 30mm. Lokeroissa ei ole pohjaa. Lokeron sivujen paksuus on 5mm lukuun ottamatta niitä kahta sivua, jotka liimataan yhteen (KUVA 17). Niiden paksuus on yksittäin 2,5 mm jolloin, kaksi lokeroa yhteen liimaamalla tulee kaikkien sivujen paksuudeksi 5 mm. Lokeroista saatiin näin erittäin siistit ja symmetrisen näköiset. Sivujen paksuuden arvioitiin olevan tarpeeksi, jotta ne eivät mene helposti rikki käytössä. Lokerikon keskelle jätettiin tyhjä ympyrän muotoinen tila, johon kiinnitetään ruuvein alle tuleva akseli.



KUVA 16. Yksittäinen lokero

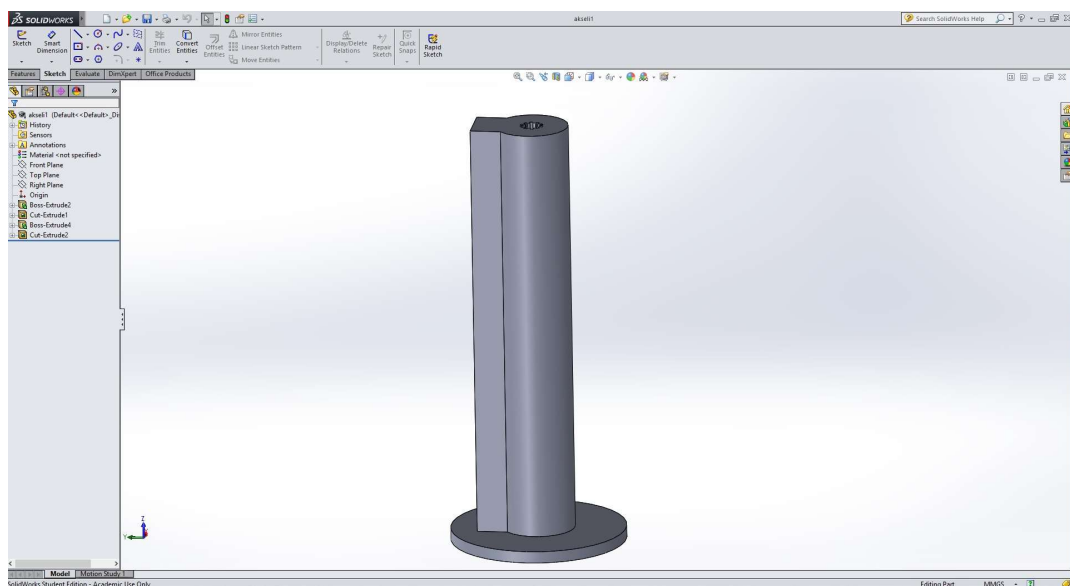


KUVA 17. Lokeron mittoja

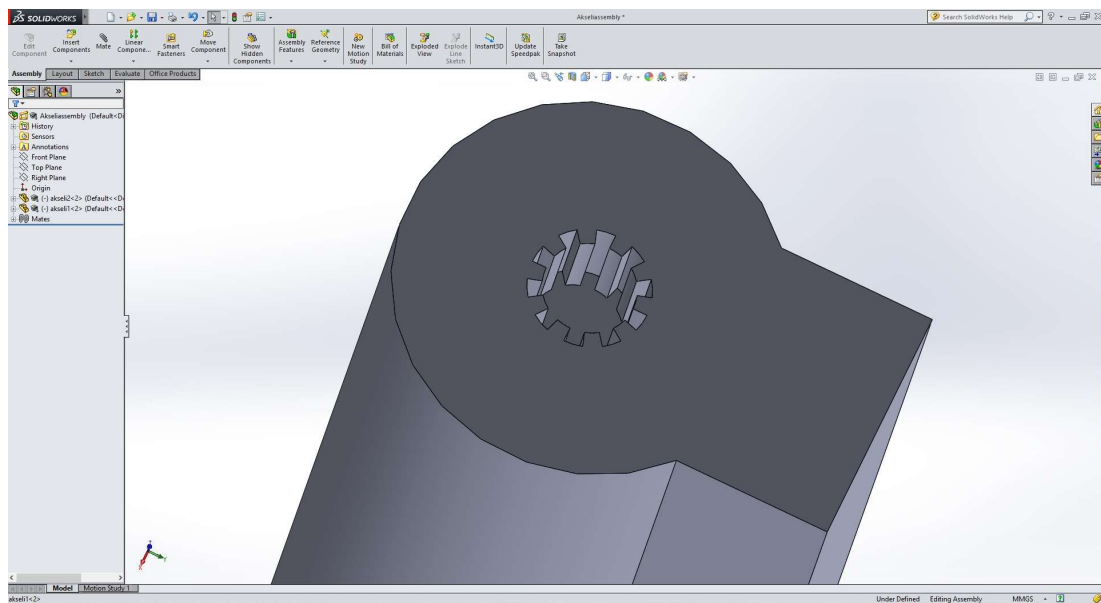


KUVA 18. Kahdeksan lokeroa, sekä alusta

Seuraavaksi suunniteltiin voimansiirto moottorilta pyörivälle alustalle. Kaikki nämä osat suunniteltiin tulostettavaksi 3D-tulostimella, koska niitä ei ollut mistään valmiina saatavilla tähän tarkoitukseen. Ensimmäinen akseli suunniteltiin niin, että se liitettäisiin vanhasta kissan lelusta otettuun pyörivään alustaan ja tähän akseliin tulisi myös kiinni iso ratas, jota pyöritetään moottorin pienellä rattaalla (KUVA 19). Akselin alosaan suunniteltiin pyöreä alue, johon saisi porattua helposti reiät kiinnitystä varten. Akselin yläosaan suunniteltiin kuvan 20 mukainen kiinnityskohta kahden akselin välille.

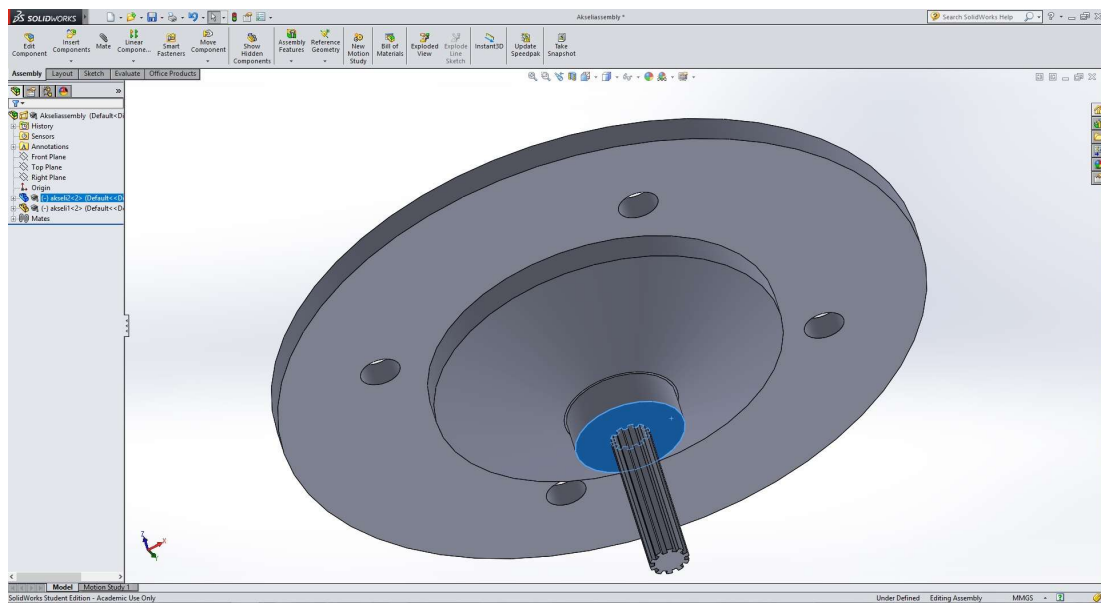


KUVA 19. Ensimmäinen akseli

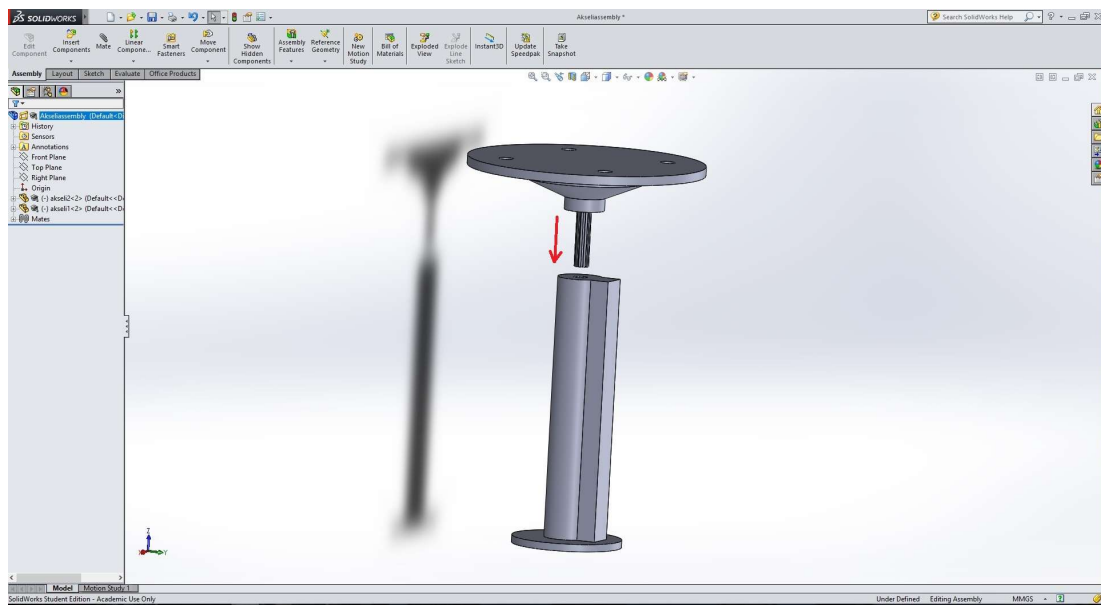


KUVA 20. Ensimmäisen akselin kiinnityskohta toiseen akseliin

Toinen akseli (KUVA 21) suunniteltiin niin, että sen saa kiinnitettyä lokerikon alustaan (KUVA 18) neljän ruuvin avulla. Akselin tuli olla mahdollisimman tukeva, jotta sen yläpuolella oleva lokerikko ja pyörivä alusta pysyisivät paikallaan. Akselin kiinnitys ensimmäiseen akseliin suunniteltiin niin, että kuvassa näkyvä tappi työnnetään ensimmäisen akselin sisään (KUVA 22). Tämän akselin oli siis tarkoitus ottaa koko lokerikon paino päälleen ja kuvassa 21 näkyvä sininen alue lepää laakerin päällä.

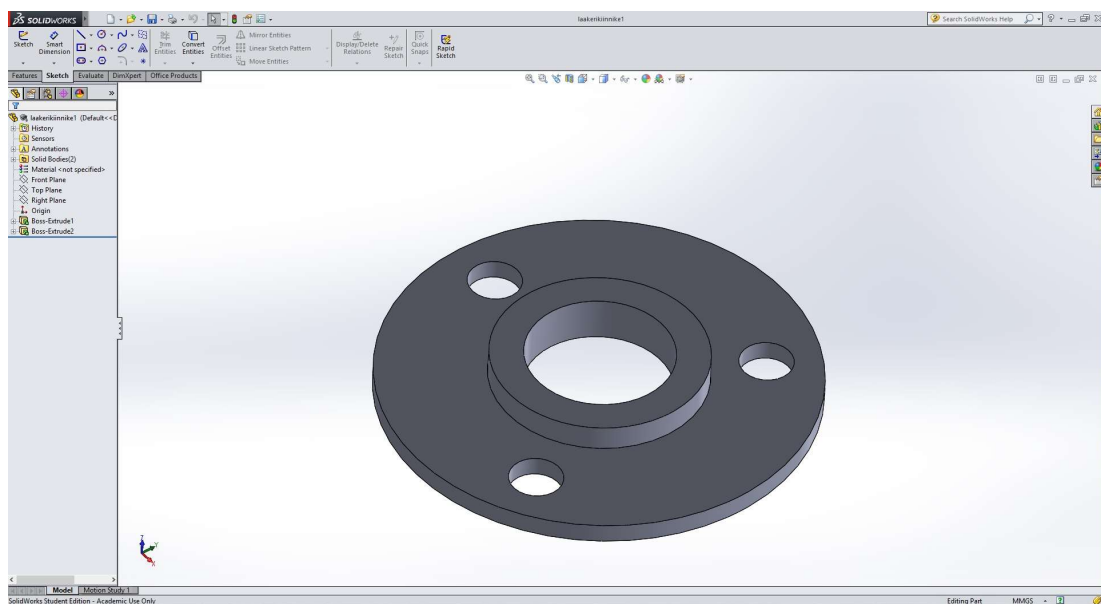


KUVA 21. Toinen akseli



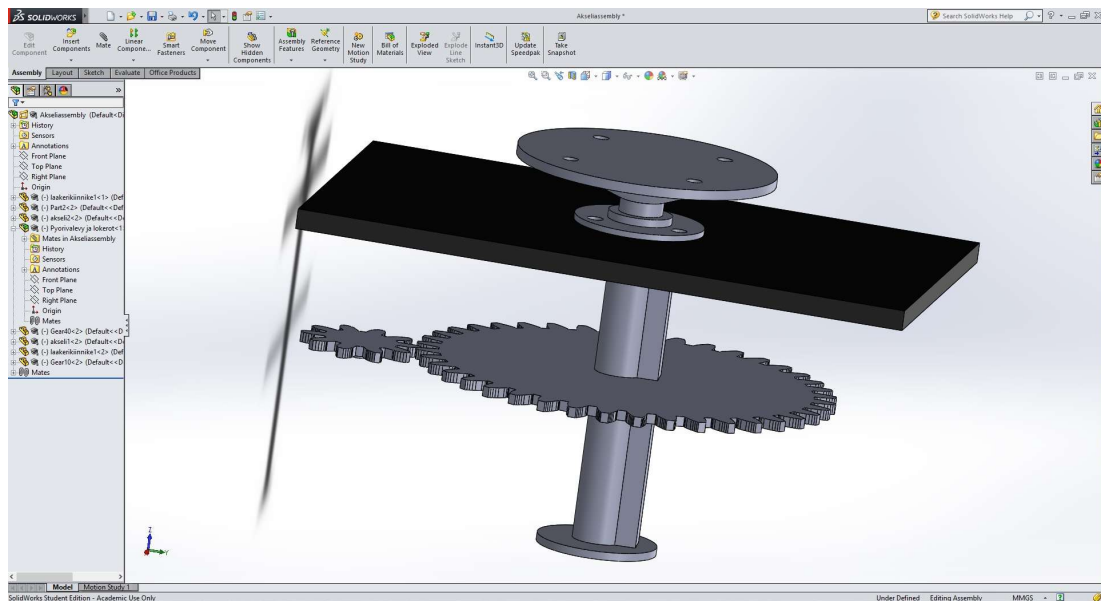
KUVA 22. Akselit ja niiden liittäminen toisiinsa

Päälilevyn molemmille puolille suunniteltiin laakereille kiinnikkeet, joilla varmistettiin, että lokerikkolevy pyöri hyvin (KUVA 23). Laakerien kiinnike suunniteltiin asennettavaksi kolmella ruuvilla ja sen keskiosaan asennettaisiin itse laakeri.



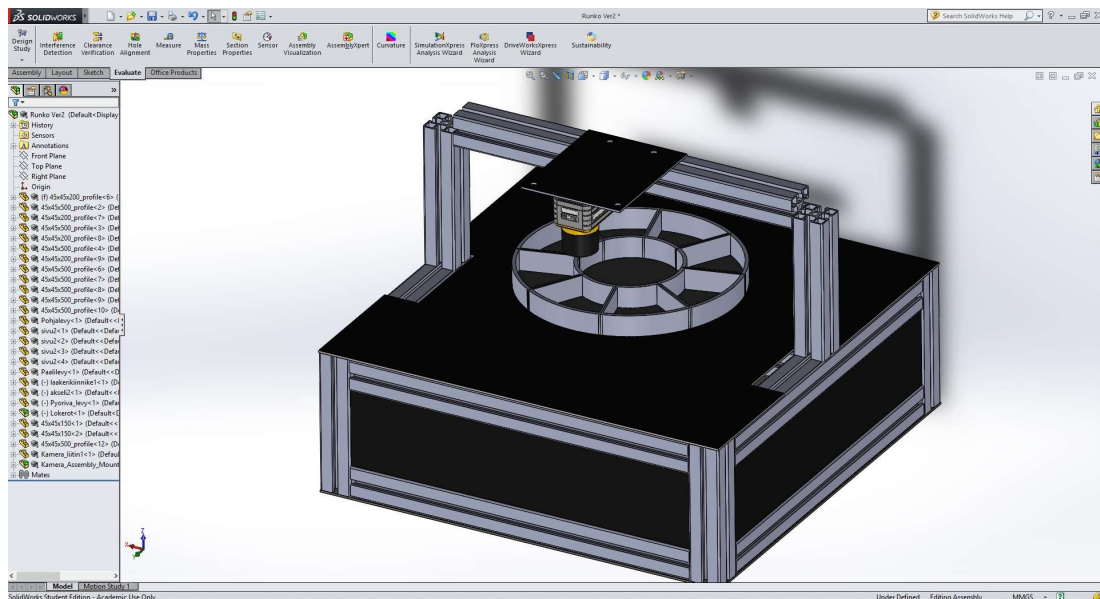
KUVA 23. Laakerien kiinnikkeet

Seuraavaksi suunniteltiin ja mallinnettiin iso hammasratas, jossa on 40 hammasta, sekä pieni hammasratas, jossa on hampaita 10 kappaletta. Näin pienen rattaan pyörähtäessä kerran on isoratas pyörähtänyt vain $\frac{1}{4}$ kierrosta. Koko akseliston ja rattaiden idea selviää kuvasta 24.



KUVA 24. Akselisto, hammasratat ja laakerien tuet yhdessä kokonaisuudessa

Seuraavaksi suunniteltiin kameras kiinnitys toteutettavaksi metallilevyn avulla. Levy kiinnitettäisiin alumiiniprofiiliin kahdella ruuvilla ja levyyn porattaisiin 3 reikää, joiden avulla itse kamera kiinnittyi levyyn. Levyn mitoiksi suunniteltiin 180x150 mm, jotta levyyn jää valaistuksen kiinnitykselle tilaa ja kameraan kytkettävät kaapelit pysytään kiinnittämään. Lopulta, kun kaikki osat oli suunniteltu ja mallinnettu ne yhdistettiin yhteen malliin (KUVA 25).



KUVA 25. Mekaanisen suunnittelun kokonaisuus

3.3 Sähköiset komponentit ja niiden kytkennät

Järjestelmän elektroninen puoli koostuu älykamerasta, ohjauspiiristä, virtalähteestä, moottorista ja kosketusnäyttöisestä käyttöliittymästä.

3.3.1 Kamera

Järjestelmässä päätettiin käyttää Cognex 7200C -älykameraa (KUVA 26). Kameraan päädyttiin siksi, että se on värikamera ja sopii ominaisuuksiltaan kyseiseen tarkoitukseen. Kameran ominaisuuksia on listattu kuvassa 27. Ominaisuuksista tärkeimpiä ovat mahdollisuus ohjelmoida kameraa, kameran 24-bittinen värienerottelukyky, mahdollisuus ohjata ulkoista valaisinta, sekä mahdollisuus ohjata kameran kuvan ajastusta ulkoisesti. Kameran resoluutio on 800x600. (Cognex WWW-sivut)

Kamerassa on Ethernet-liitäntä tiedonsiirtoa varten, light-liitäntä, jolla pystytään ohjaamaan ulkoista valaisinta, sekä 12-pin virta- ja I/O-liitäntä. Kamerassa on 1/1.8 tuuman CMOS-kenno. (Cognex WWW-sivut)



KUVA 26. Cognex Insight-7000 sarjan kamera (Cognex WWW-sivut)

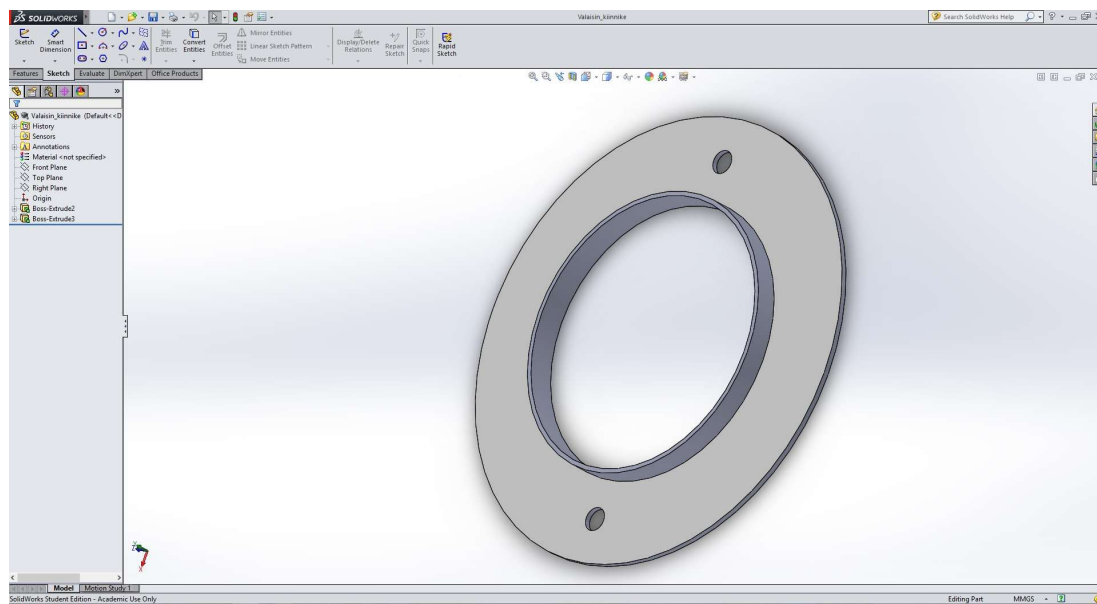
Specifications	In-Sight 7010/7020/7050/7200/7210/7230/7400/7410/7430	In-Sight 7010C/7200C/7400C	In-Sight 7402/7412/7432	In-Sight 7402C
Minimum Firmware Requirement	In-Sight Version 4.7.1/4.7.3 ¹	In-Sight Version 4.8.0	In-Sight Version 4.7.1/4.7.3	In-Sight Version 4.8.0
Job/Program Memory	512MB non-volatile flash memory; unlimited storage via remote network device.			
Image Processing Memory	256MB SDRAM			
Sensor Type	1/1.8-inch CMOS			
Sensor Properties	5.3mm diagonal, 5.3 x 5.3µm sq. pixels		8.7mm diagonal, 5.3 x 5.3µm sq. pixels	
Resolution (pixels)	800 x 600		1280 x 1024	
Electronic Shutter Speed	16µs to 950ms			
Acquisition	Rapid reset, progressive scan, full-frame integration.			
Bit Depth	256 grey levels (8 bits/pixel).	24-bit color.	256 grey levels (8 bits/pixel).	24-bit color.
Image Gain/Offset	Controlled by software.			
Frames Per Second ²	102 full frames per second.	50 full frames per second.	60 full frames per second.	30 full frames per second.
Lens Type	M12 or C-Mount.			
Image Sensor Alignment Variability ³	±0.127mm (0.005in), (both x and y) from lens C-Mount axis to center of imager.			
Trigger	1 opto-isolated, acquisition trigger input. Remote software commands via Ethernet and RS-232C.			
Discrete Inputs	3 general-purpose inputs when connected to the Power and I/O Breakout cable. (Eight additional inputs available when using the optional CIO-MICRO or CIO-MICRO-CC I/O module.)			
Discrete Outputs	4 high-speed outputs when connected to the Power and I/O Breakout cable. (Eight additional outputs available when using the optional CIO-MICRO or CIO-MICRO-CC I/O module.)			
Status LEDs	Network link and activity, power and 2 user-configurable.			
Internal LED Ring Light	Red, Green, Blue, White, IR (M12 lens configuration only).			
Network Communication	Ethernet port, 10/100 BaseT with auto MDI/MDIX. IEEE 802.3 TCP/IP protocol. Supports DHCP (factory default), static and link-local IP address configuration.			
Serial Communication	RS-232C: 4800 to 115,200 baud rates.			
Power Consumption	24VDC ±10%, 2.0 amp.			

KUVA 27. Cognex Insight-7000 sarjan kameroiden ominaisuuksia (Teksit WWW-sivut)

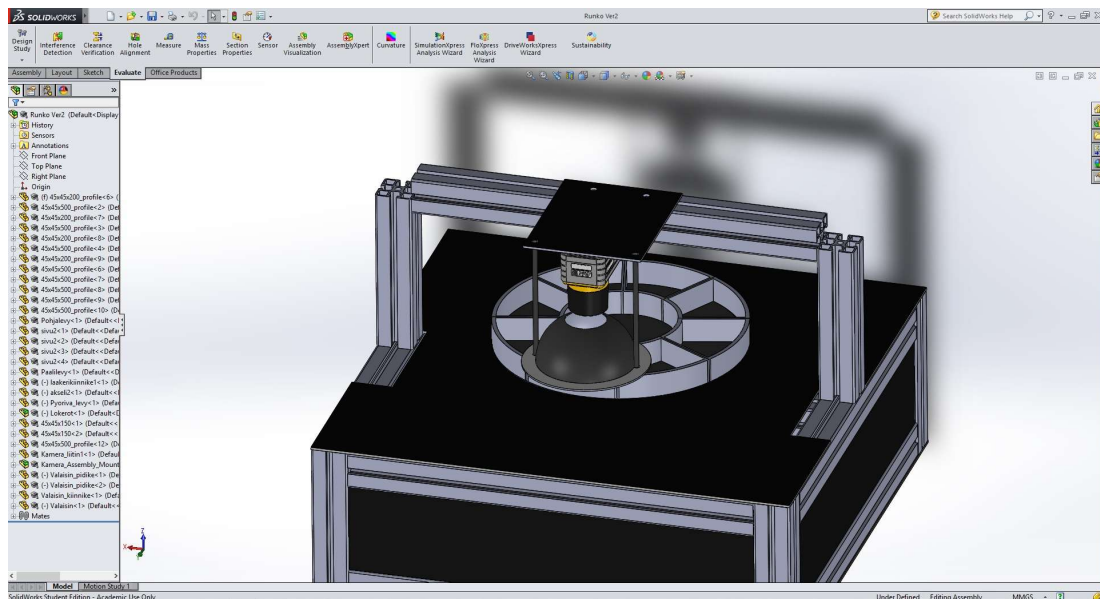
3.3.2 Valaisin

Älykameran valaisussa päätettiin käyttää diffuusiokupolivalaisinta. Tämä valaisintyyppi soveltuu järjestelmään hyvin, koska sen avulla kappaleen ja varjojen ja heijastusten syntyminen saadaan minimoitua vaihtelevienkin kappaleiden pinnalla. Valaisin suunniteltiin asennettavaksi niin, että se tulee mahdollisimman lähelle pyörivän alustan lokerikkoa. Tällä saavutetaan hyvin vähäinen ympäröivän, häiritsevän valon pääsy itse kuvattavan kappaleen pinnalla. Myös älykamera suunniteltiin asennettavaksi mahdollisimman lähelle valaisimen yläosassa olevaa aukkoa, jotta estetään häiritsevän valon pääsy kohteelle.

Valaisimen asennus suunniteltiin niin, että Solidworks-ohjelmistoa käyttäen mallinnettiin kuvassa 28 oleva osa, jonka päälle valaisin voidaan asentaa. Valaisin ja sen kiinnityskomponentti suunniteltiin kiinnitettäväksi yläpuolella olevaan metallilevyyn kahden kierretapin avulla (KUVA 29). Valaisimen kiinnityskomponentti 3D-tulostettiin. Kierretapit on kiinnitetty yläpuolella olevaan metallilevyyn muttereilla.



KUVA 28. Valaisimen kiinnityskomponentti



KUVA 29. Valaisimen kiinnityskokonaisuus

3.3.3 Raspberry PI ja käyttöliittymä

Järjestelmän “aivoiksi” valittiin “Raspberry PI 2 model B” piirilevy (KUVA 30). Piirilevyn tärkeimpiä ominaisuuksia tämän järjestelmän kannalta ovat:

- 4 USB porttia
- 40 ohjelmoitavaa I/O-pinniä
- HDMI-portti
- Ethernet-portti
- Micro SD -muistikorttipaikka
- Piirilevyyn on mahdollista kytkeä kosketusnäyttö

Piirilevyssä on myös 900 MHz:n prosessori ja 1GB RAM-muistia, mikä tekee siitä varsin nopean ja tähän tarkoitukseen täydellisen. (Raspberrypi WWW-sivut)

Ethernet-kaapeli voidaan kytkeä piirilevyn ja kameran välille ja näin saadaan haluttu data siirrettyä laitteelta toiselle. Ohjelmoitavien I/O-porttien avulla voidaan ohjata järjestelmän toimintaa. Piirilevyn muistikortille voidaan tallentaa kameran tuleva kuva ja käyttöjärjestelmän ajurit.

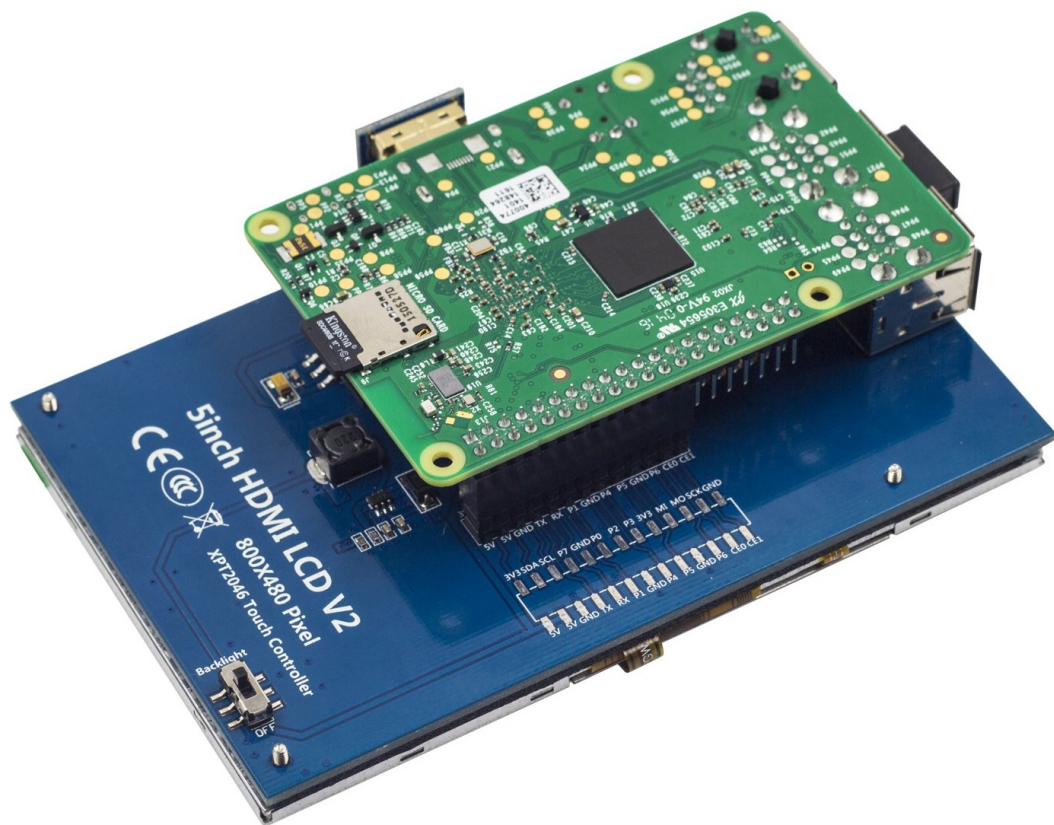
Nopean prosessorinsa ansiosta piirilevy pystyy pyörittämään esimerkiksi Linux- tai Windows 10 -käyttöjärjestelmää. Piirilevyssä on valmiiksi asennettuna esimerkiksi Java- ja Python-ohjelmointikielet. (Raspberrypi.org WWW-sivut)



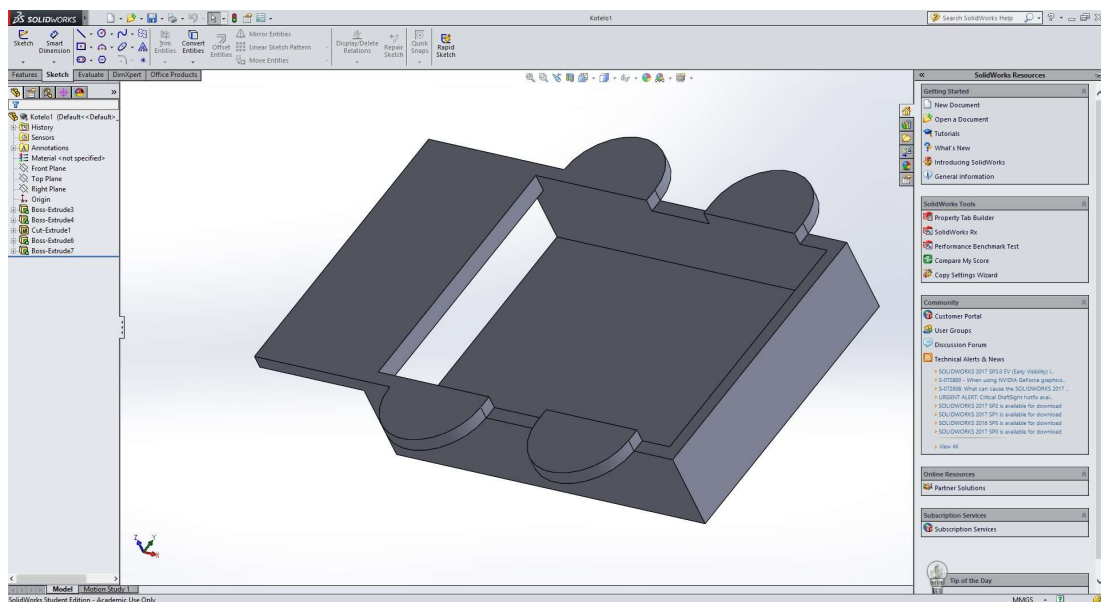
KUVA 30. Raspberry PI 2 model B (Raspberrypi WWW-sivut)

Järjestelmän käyttöliittymä suunniteltiin 5 tuuman kosketusnäyttöä käyttäen (KUVA 31). Näyttö kytketään suoraan piirilevyn I/O-pinneihin ja kuva siirretään niiden välillä HDMI-adapterilla. Näytön tarkkuus on 800x480 pikseliä.

Kosketusnäytölle ja piirilevylle suunniteltiin kuvan 32 mukainen kotelo. Piirilevy suunniteltiin kiinnitettäväksi ruuveilla kotelon pohjaan. Kotelon toinen pääty suunniteltiin avonaiseksi, jotta pystytään kytkemään kaapelit piirilevylle.



KUVA 31. Kosketusnäyttö (Sunfounder.com WWW-sivut)



KUVA 32. Kotelokosketusnäytölle ja piirilevylle

3.3.4 Piirikaaviot

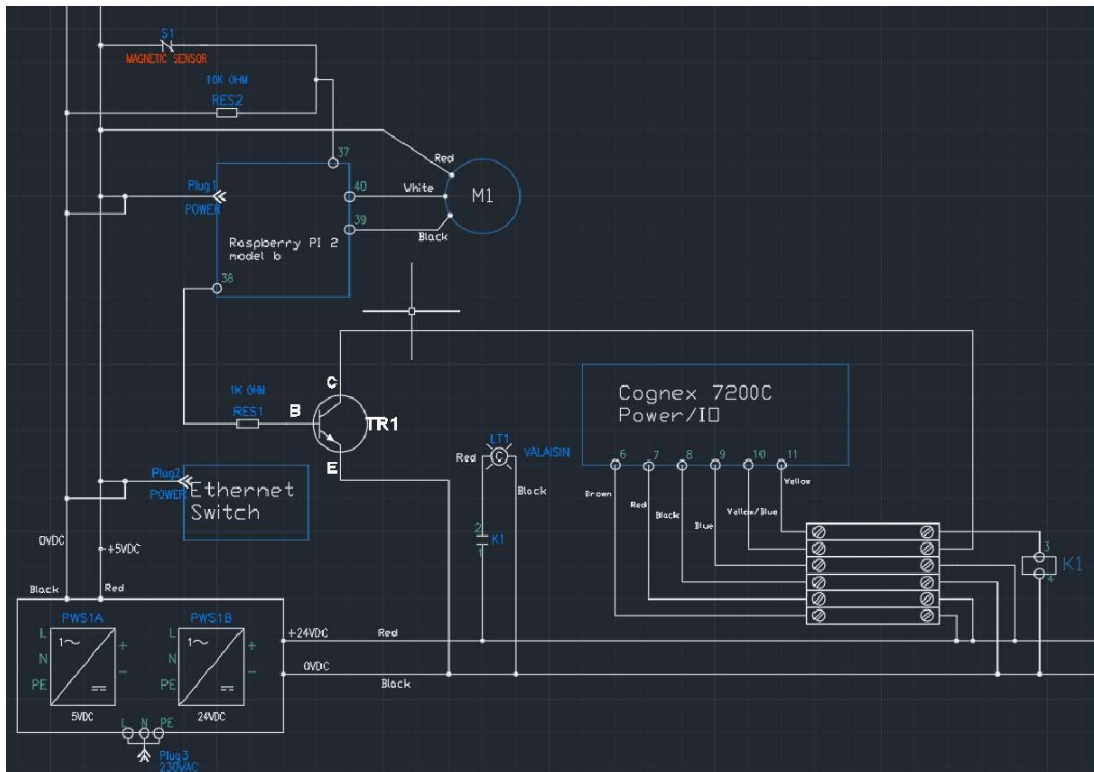
Järjestelmän sähkökuvat ja suunnittelu tehtiin Autocad Electrical -ohjelmistolla. Piirikaavioon on merkitty kaikki sähköiset komponentit ja niiden kytkennät (KUVA 33).

Järjestelmän virtalähteenä toimii yksikkö, jossa on 5 VDC:n lähtö ja 24 VDC:n lähtö. Virtalähde vaatii 230 VAC toimiakseen. Virtalähde, jossa on kaksi eri jännitelähtöä valittiin siksi, että järjestelmässä on molemmilla jännitteillä toimivia komponentteja ja olisi epäkäytännöllistä, käyttää kahta eri virtalähdettä.

Käydään ensin läpi 24 VDC:lla toimivat komponentit ja kytkennät. Kamera toimii tällä jännitteellä ja myös kameran I/O-liitännät tarvitsevat tämän jännitteen. Kamerasta lähtee Ethernet-kaapeli, sekä power- ja I/O-kaapeli. Tähän piirikaavioon on piirretty vain power- ja I/O-kaapelin kytkentä. Kameran pinneistä käytössä on pinnit 6-11 ja näiden pinnien nimet ovat nähtävissä kuvassa 34. Pinnit 6 ja 9, jotka normaalisti kytketään nollaan, on tässä tapauksessa kytketty +24 VDC:iin, PNP-konfiguraation mukaisesti.

Kameran pinnillä numero 11 ohjataan valaisinta päälle ja pois releen avulla. Sähkökuvissa olevan transistorin tarkoituksena on mahdollistaa, että Raspberry PI -piirilevyn output-signaalilla pystytään ohjaamaan kameran input-signaali päälle tai pois. Transistori tarvitaan väliin siksi, että piirilevyn output-signaalin jännite on 3,3 V ja kamera tarvitsee 24 V, joten transistorikytkennän avulla ongelma ratkaistiin. Piirilevyn ja transistorin kannan välille laskettiin yhden kiloohmin vastus, jotta kytkentä ei ole oikosulussa.

Järjestelmän moottori toimii 5 V:n jännitteellä ja sitä ohjataan suoraan piirilevyltä. Moottori on tyypiltään servomoottori, mutta pystyy pyörimään 360 astetta ympäri toisin kuin normaalit servomoottorit. Magneettianturi toimii ikään kuin kytkimenä, joka on normaalisti kiinni ja havaitessaan magneetin kytkin aukeaa, jolloin signaali muuttuu nolaksi. Kytkimen perässä on 10 k Ω vastus, jotta kytkentä ei ole oikosulussa. Raspberry PI ja Ethernet-kytkin tarvitsevat 5 V:n jännitteen toimiakseen kuten kuvasta selviää.



KUVA 33. Järjestelmän piirikaavio

COGNEX

Power and I/O Breakout Cable Specifications

The Power and I/O Breakout cable provides connections to an external power supply, the acquisition trigger input, general-purpose inputs, high-speed outputs, and RS-232 serial communications. The Power and I/O Breakout cable is not terminated.

Table 3-8: Power and I/O Breakout Cable Pin-Out

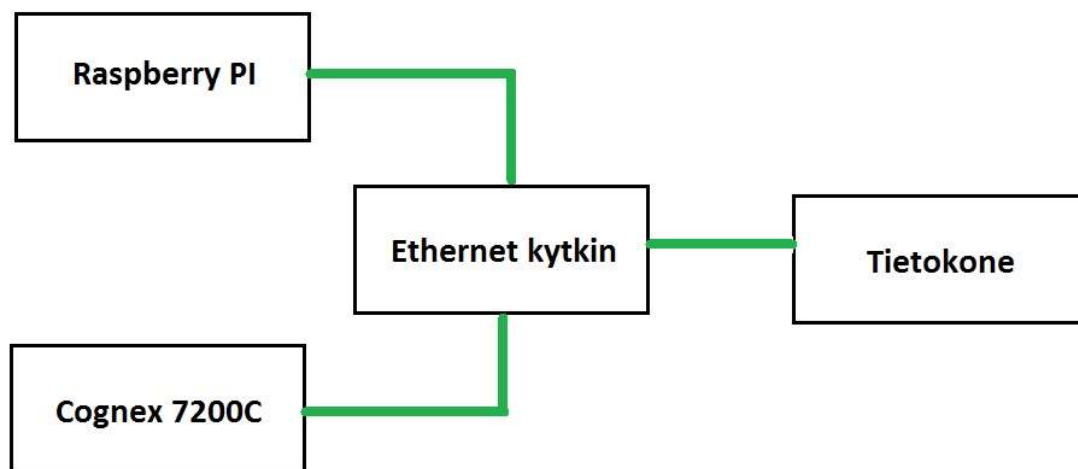
Pin#	Signal Name (I/O Mode)	Wire Color
1	IN 2	Yellow
2	IN 3	White/Yellow
3	HS OUT 2	Brown
4	HS OUT 3	White/Brown
5	IN 1/RS-232 RECEIVE ¹	Violet
6	INPUT COMMON	White/Violet
7	+24VDC	Red
8	GROUND	Black
9	OUTPUT COMMON	Green
10	TRIGGER	Orange
11	HS OUT 0	Blue
12	HS OUT 1/RS-232 TRANSMIT ²	Grey
Shell	SHIELD	Bare Wire

Note:

- Cables are sold separately.
- Unused bare wires can be clipped short or tied back using a tie made of non-conductive material. Keep all bare wires separated from the +24VDC wire.

KUVA 34. Cognex power- ja I/O-pinnit (Teksis.co.uk WWW-sivut)

Jotta tiedonsiirto olisi mahdollista kameran ja Raspberryn välillä, tuli ne olla kytkettyinä yhteen Ethernet-kaapelilla (KUVA 35). Järjestelmä suunniteltiin niin, että haluttaessa voidaan kytkeä myös tietokone järjestelmään Ethernet-kaapelia käyttäen. Järjestelmä toimii täysin normaalisti joko ilman tietokonetta tai sen kanssa, mutta tietokoneen etuna on se, että voidaan esitellä kamerassa olevaa ohjelmaa. Tietokoneesta nähdään reaaliajassa, mitä ohjelma tekee ja kameran ottama kuva nähdään hieman paremmin tietokoneelta kuin itse järjestelmän pieneltä käyttöliittymästä.

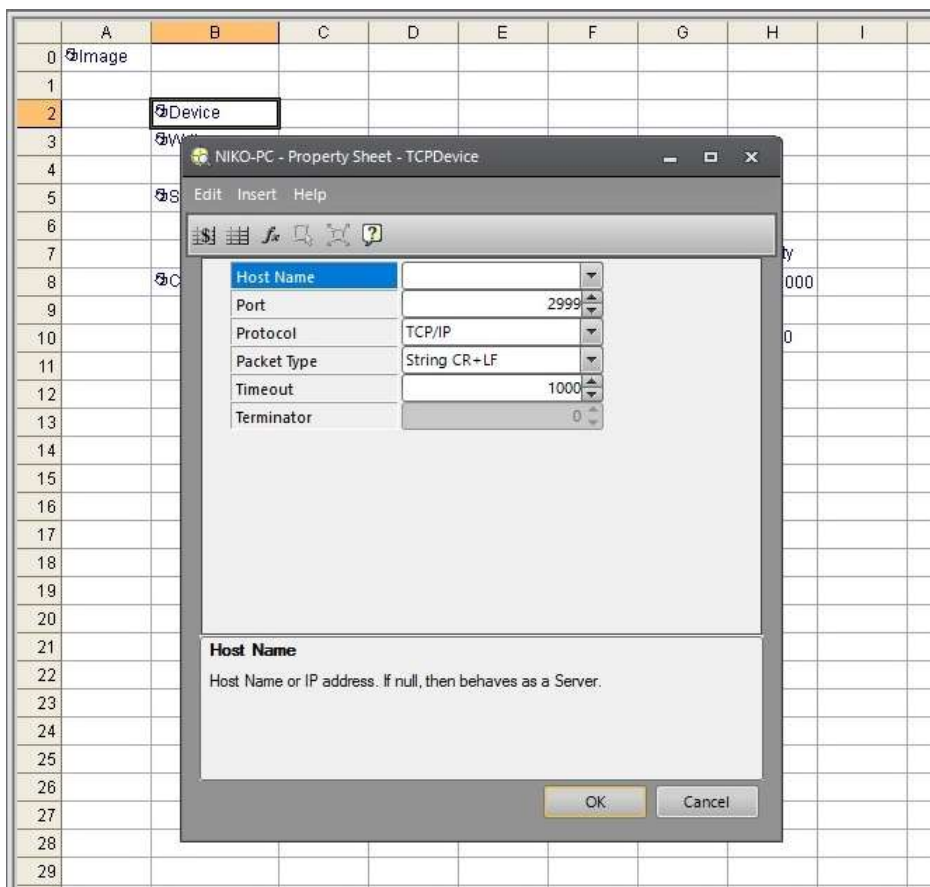


KUVA 35. Ethernet-kytkennät

4 OHJELMOINTI

4.1 Älykameran ohjelmointi

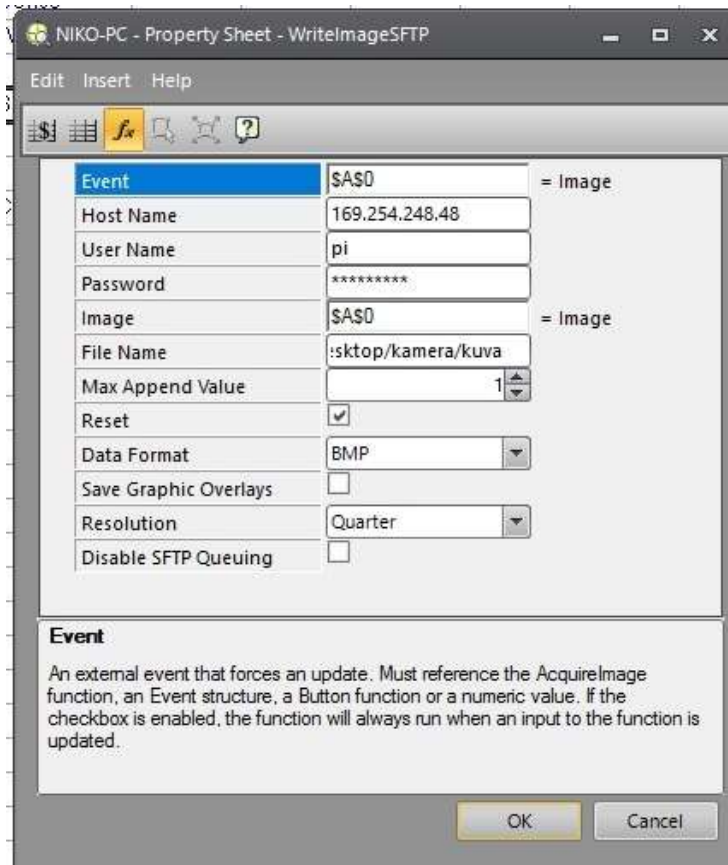
Älykameran ohjelmointi tehtiin In-Sight Explorer -sovelluksella, joka on tarkoitettu Cognex-älykameroiden ohjelmointiin. Ensimmäisessä ohjelmassa pohjustettiin TCP/IP yhteyden luonti kameran ja Raspberry PI -piirilevyn välille käyttämällä työkalua ”TCPDevice” (KUVA 36). Työkalun parametreista ”Host name” jätettiin tyhjäksi, jotta kamera toimii niin sanottuna serverinä ja portti numeroksi määriteltiin 2999, joka voi olla käytännössä mikä tahansa. Raspberry PI:llä pystytään ottamaan yhteys kameran luomaan serveriin ja näin pystytään vastaanottamaan dataa kameralta.



KUVA 36. ”TCPDevice” työkalu ja asetetut parametrit

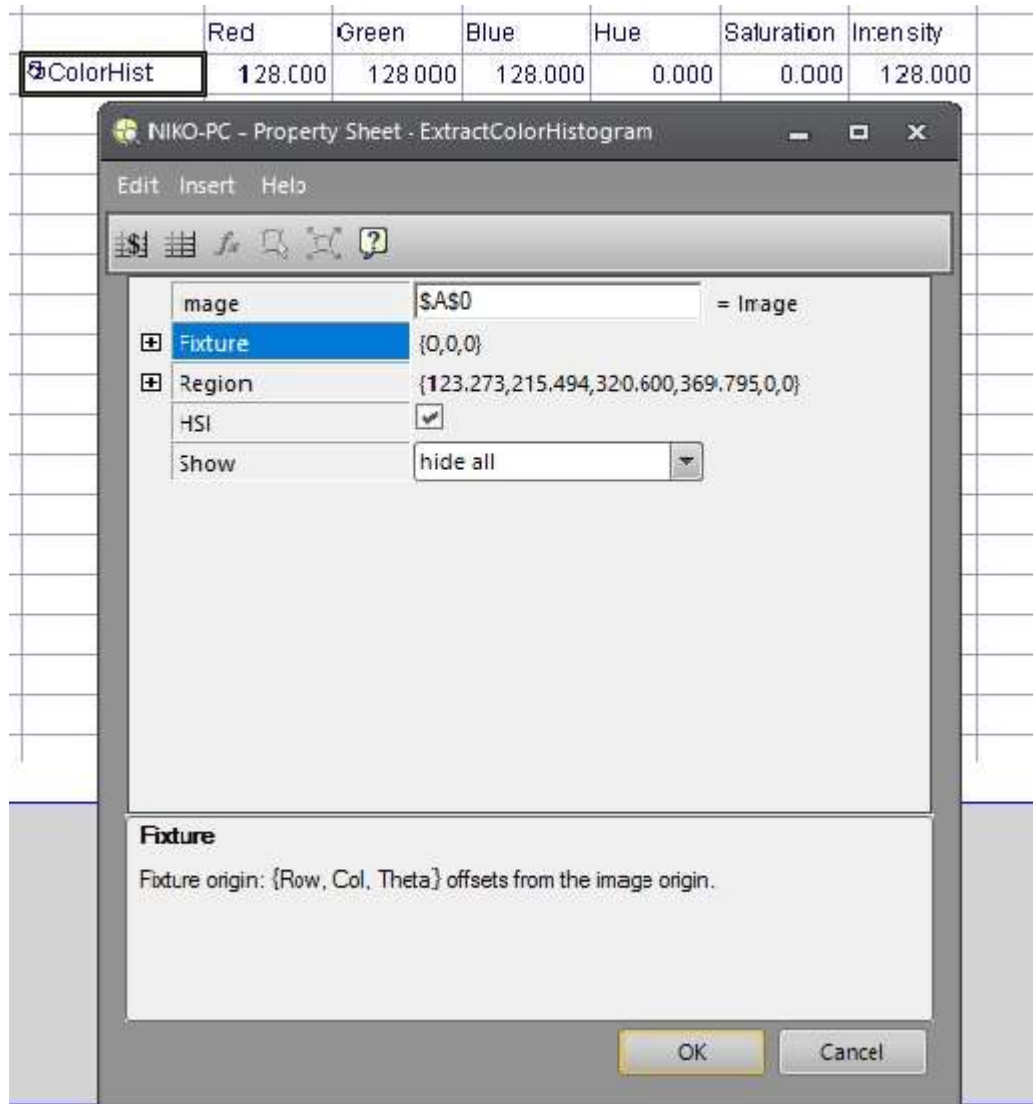
Seuraavaksi luotiin ”WriteDevice” -työkalua käyttäen toiminto, joka aktivoituessaan kirjoittaa määritetyn datan ”TCPDevice” -työkalulle. ”WriteDevice” -työkalun parametrit olivat: WriteDevice(\$A\$0,B2,C10,D10,E10,F10,G10,H10). Työkalun aktivoituminen eli datan kirjoitus määriteltiin alkavaksi kameran ottaessa kuvan. Parametreihin määriteltiin datan kirjoituskohde ”TCPDevice”, joka sijaitsee ohjelman sarakkeessa ”B2”. Lopuksi määriteltiin datan sisältö, joka sijaitsee sarakkeissa ”C10 – H10”.

Seuraava työkalu ”WriteImageSFTP” (KUVA 37), jolla saadaan siirrettyä kameran ottama kuva Raspberry PI -piirilevyn muistiin. Työkaluun määriteltiin kohteen IP-osoite, sekä sen käyttäjänimi ja salasana, jotta kamera kykenee muodostamaan yhteyden oikean laitteen kanssa. Piirilevyn käyttäjänimi oli oletuksena ”pi” ja salasana ”raspberrypi”. Kuvan tallennuskohteeksi määritettiin ”Desktop/kamera/kuva”. ”Max Append Value” asetettiin yhdeksi ja ”reset” päälle, jolloin kamera luo aina tiedoston ”kuva0.bmp” ja poistaa vanhan. Resoluutioksi valittiin pienin mahdollinen, jotta kuvan siirtoajat saatiin pidettyä kohtuullisena.



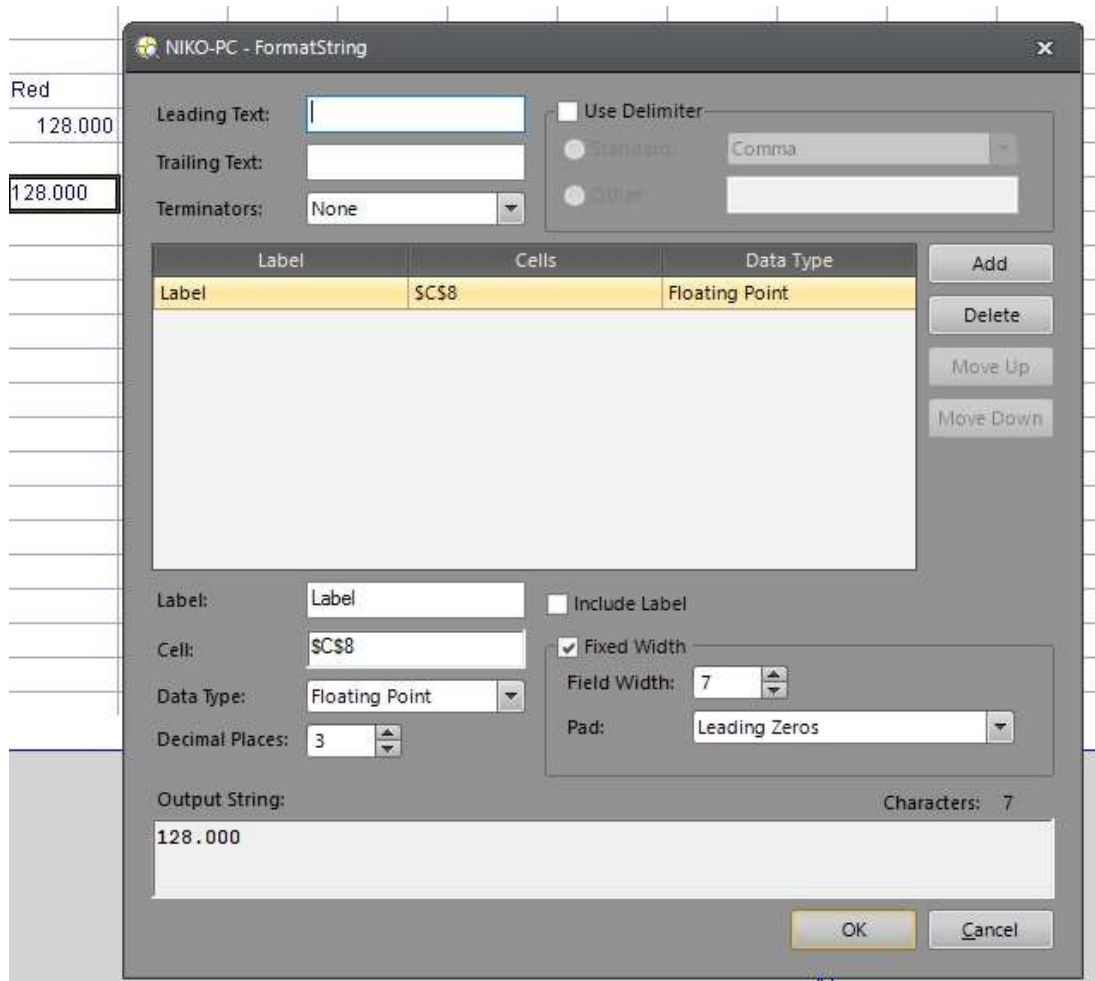
KUVA 37. ”WriteImageSFTP” -työkalun parametrit

Seuraavaksi luotiin värinanalysointityökalu ”ExtractColorHistogram” (KUVA 38). Työkalu analysoi RGB -arvot, värisävyn, kyllästyksen ja intensiteetin. Värin tarkastelualue määriteltiin niin, että se on pyörivän alustan lokeroiden sisällä.



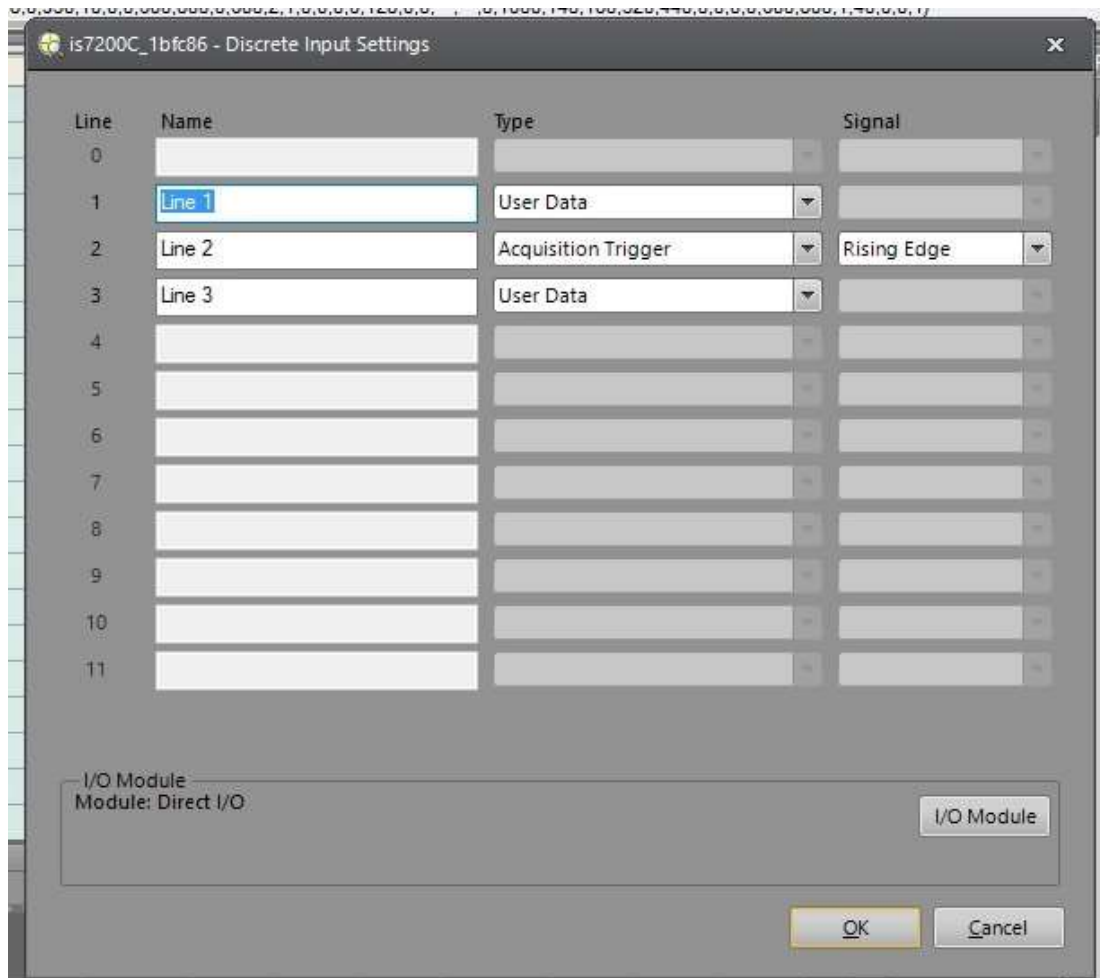
KUVA 38. ”ExtractColorHistogram” -työkalun parametrit

Jokaiselle ”ExtractColorHistogram” väriarvolle luotiin FormatString-työkalut (KUVA 39). ”FormatString” -työkalulla muotoiltiin string-muodossa olevat värien arvot niin, että ne esitetään aina 7 merkillä. Siirrettäessä väriarvot ”WriteDevice” -työkalulla varmistetaan, että data on aina 7 x 6 eli 42 merkkiä pitkä. Näin pystytään lukemaan data Raspberry PI:lla, kun tiedetään tarkalleen, missä muodossa data lähetetään sille.

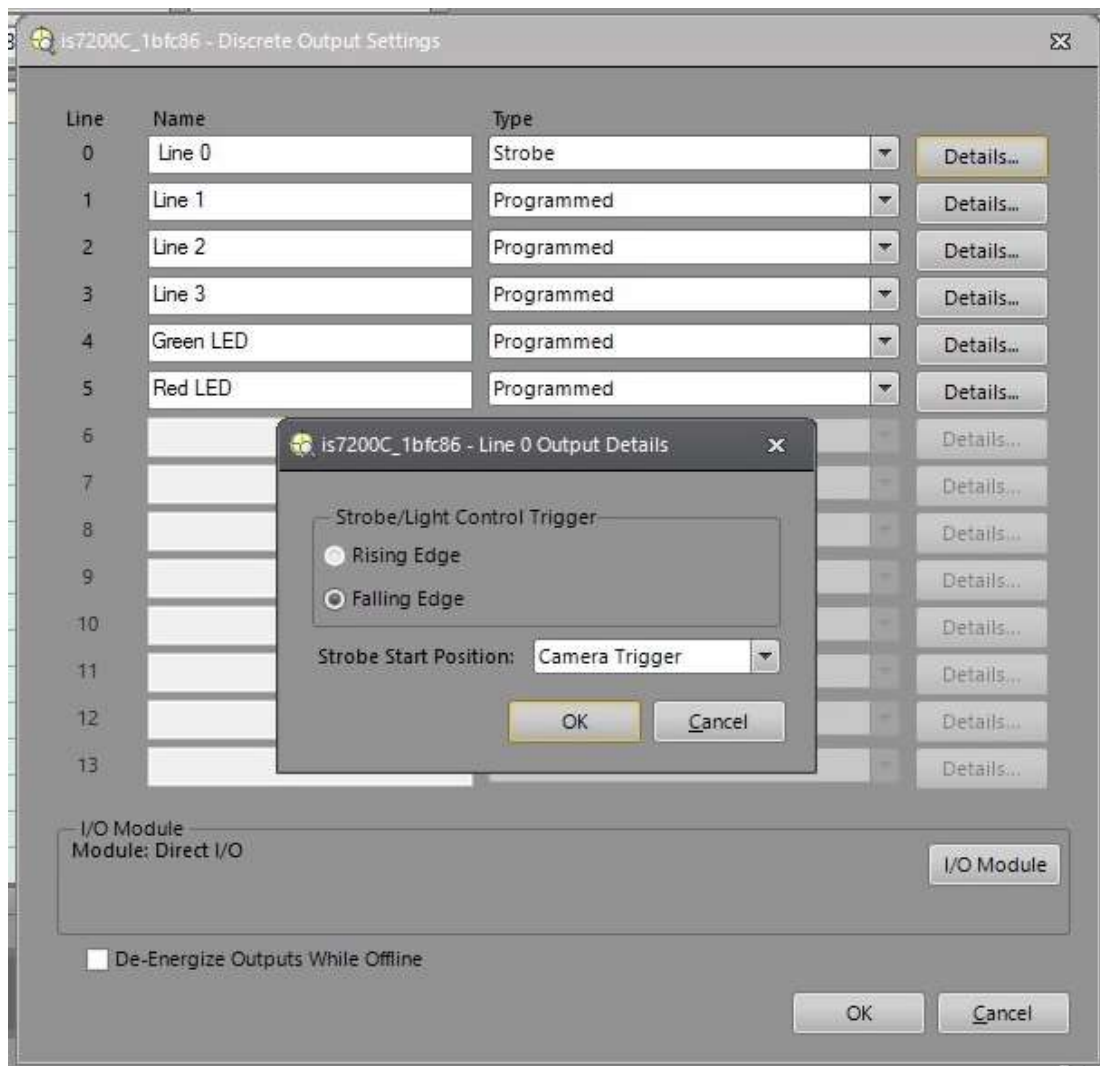


KUVA 39. ”FormatString” -työkalun parametrit punaiselle väriarvolle

Lopuksi ohjelmoitiin input- ja output-signaalit kameralle (KUVA 40, 41). Input-signaaliiksi määriteltiin ”Acquisition Trigger” nouseva reuna. Tästä input-signaalista kamera ottaa kuvan kappaleesta. Output-signaaliiksi määriteltiin ”Line 0” ja tyypiksi ”strobe”, jolloin ulkoinen valaisin välähtää. Outputin tyypiksi määriteltiin laskeva reuna ja kappaleen valaisu alkaa, kun kamera saa trigger-signaalin. Kameran ”AcquireImage” -asetuksista määriteltiin ”trigger source; camera”.



KUVA 40. Input-asetukset



KUVA 41. Output-asetukset

4.2 Raspberry PI -ohjelmointi

Raspberry PI ohjelmoitiin Python-ohjelmointikielellä. Käytössä oli Pythonin versio 2.7. Raspberry PI toimii järjestelmän aivoina ja sillä hoidetaan kaikki toiminnot moottorinohjauksista kameran kuvan ajastukseen, tiedonsiirtoon ja käyttäjärjestelmän toimintaan. Ohjelma jaettiin kahteen osaan eli pääohjelmaan, sekä toiseen ohjelmaan, joka suorittaa tiedonsiirron kameran piirilevyille. Tiedonsiirto-ohjelmaa kutsutaan pääohjelmasta, joka teki ohjelmistosta hieman helpommin ymmärrettävän ulkopuoliselle ihmiselle, mikäli joskus halutaan esimerkiksi muuttaa ohjelmaa.

Python on korkean tason ohjelmointikieli, joka painottuu olio-ohjelmointiin. Python ohjelmointikielen ensimmäinen versio julkaistiin vuonna 1990 ja sen on kehittänyt

Guido van Rossum. Python ohjelmointikielessä painotetaan syntaksin yksinkertaisuutta, joka tekee siitä helposti ymmärrettävää. Python tunnetaan hyvin sen korkean tason tietorakenteiden vuoksi ja siitä, että kieltä ei tarvitse kääntää ennen ohjelman ajoa, mikä mahdollistaa ohjelmistojen nopean testauksen ja kirjoittamisen. Pythonilla pystytään luomaan käyttöliittymiä ja sitä voidaan käyttää hyvin vaativassakin tieteellisessä laskennassa. Ohjelmointikieltä voidaan käyttää mm. proseduraalisessa, funktionaalisessa ja oliopohjaisessa ohjelmoinnissa. Python-ohjelmointikielestä löytyy samankaltaisia ominaisuuksia mm. Java-, JavaScript-, Perl-, Tcl- ja Smalltalk-ohjelmointikielten kanssa. (Python.org WWW-sivut)

4.2.1 Ohjelman toiminnallisuus

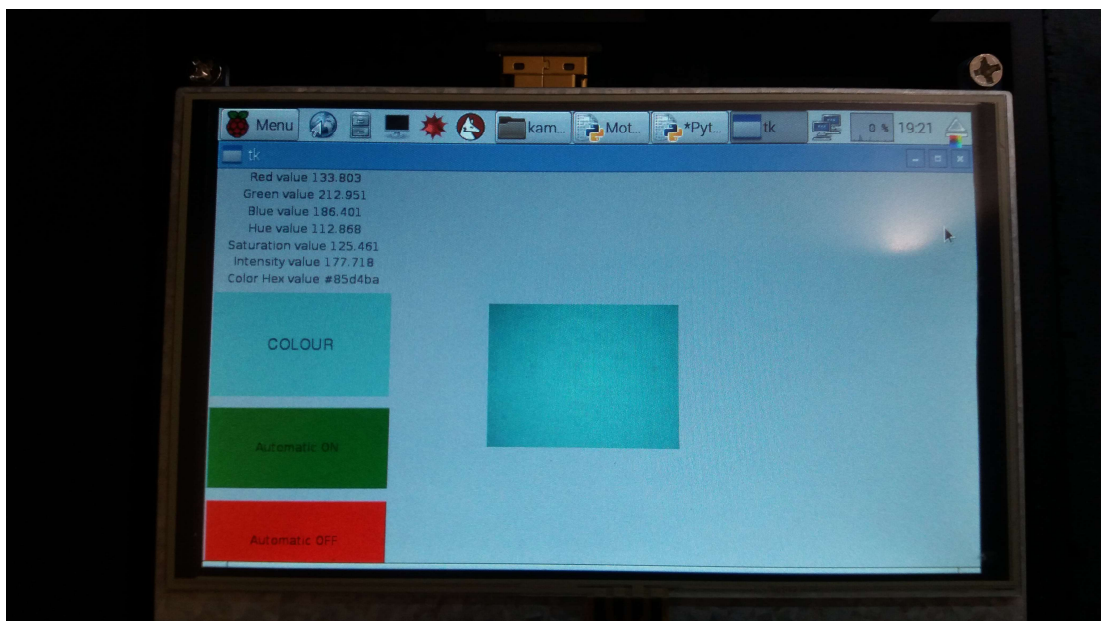
Kun pääohjelma käynnistetään se avaa käyttöliittymän kosketusnäytölle, jossa näkyy ”Automatic ON” ja ”Automatic OFF” -painikkeet, sekä viimeksi kameran muistiin tallennettu kuva (KUVA 42). Mitään väriarvoja ei ole vielä näkyvissä, koska dataa ei ole vastaanotettu.

Laitteistoa ohjataan käyttöliittymästä ”Automatic ON” ja ”Automatic OFF” -painikkeilla. ”Automatic ON” -painikkeella järjestelmä aloittaa ohjelmakierron ja ”Automatic OFF” -painikkeella ohjelmakierto pysähtyy. ”Automatic ON” -painikkeella moottori alkaa toimia ja se liikkuu, kunnes jokin pyörivän alustan alla olevista magneeteista saavuttaa magneettianturin, jolloin moottori pysähtyy välittömästi. Kun moottori on pysähtynyt, annetaan piirilevyltä signaali kameran trigger-inputille, jolloin kamera ottaa kuvan. Kun kamera ottaa kuvan, se ohjaa ulkoisen valaisimen hetkellisesti päälle. Kuvan otettuaan kamera analysoi väritiedot ja ne ovat valmiina siirrettäväksi piirilevylle.

Tämän jälkeen piirilevy avaa TCP/IP-yhteyden kameraan, josta luetaan data. Seuraavaksi python ohjelma laskee väriarvoista niiden vastaavan heksa-arvon, joka talletetaan muuttujaan myöhempää käyttöä varten. Kymmenen sekunnin ajastin käynnistyy tämän jälkeen, minkä aikana kameran ottama kuva siirretään piirilevyn muistiin. Piirilevy on nyt saanut kaiken datan kameralta ja se on valmis siirtämään datan ja kuvan käyttöliittymälle. Käyttöliittymälle tuodaan kameran kuva, värin RGB-arvot, värisävy,

kylläisyys, sekä intensiteetti. Saaduista arvoista lasketaan väri ja käyttöliittymässä oleva väriä muuttava alue vaihtaa väriään laskettujen arvojen mukaan. Näin käyttäjä näkee näytöltä värin numeeriset arvot sekä itse värin, jota arvot vastaavat. Lopuksi ohjelma odottaa 5 sekuntia ennen kuin se lähtee uudestaan alusta ja toistaa samat edellä mainitut vaiheet.

Ohjelmakierron voi myös pysäyttää painamalla ”Automatic OFF” -painiketta, mutta se toimii vain silloin, kun kaikki data on siirretty ja moottori on paikoillaan. ”Automatic ON” -painike muuttuu ohjelman viimeisen 5 sekunnin ajaksi vihreäksi, jolloin käyttäjä pystyy pysäyttämään ohjelman. ”Automatic OFF” -painike ei sulje käyttöliittymää vaan halutessaan käyttäjä voi painaa uudelleen ”Automatic ON” -painiketta, jolloin ohjelmakierto alkaa taas.



KUVA 42. Kosketusnäytöllinen käyttöliittymä

4.2.2 Miten ohjelma toimii?

Tiedonsiirto-ohjelmalla (KUVA 43) luodaan TCP/IP-yhteys älykameran luomalle serverille. Tähän ohjelmaan on määritetty kameran IP-osoite ja portti, sekä 3 funktiota. ”SetupSocket” -funktiolla määritellään mihin kohteeseen yhteys muodostetaan. ”Receive” ja ”transmit” -funktioilla avataan ja suljetaan yhteys, sekä luetaan dataa ja tallennetaan se muuttujaan, jota käytetään pääohjelmassa.



```

File Edit Format Run Options Window Help
import socket
import time

host = '169.254.7.248'
port = 2999

def setupSocket():
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect((host, port))
    return s

def receive(s):
    datavalue = s.recv(1024)
    s.close()
    return datavalue

def transmit():
    s = setupSocket()
    data = receive(s)
    return data

```

Ln: 21 Col: 0

KUVA 43. Pythonin TCP/IP-yhteyden muodostusohjelma

Pääohjelman ensimmäiset 14 riviä (KUVA 44) ovat erilaisia muuttujien alustuksia, otetaan käyttöön raspberry PI:n I/O-pinnejä ja alustetaan niitä, joko inputeiksi tai outputeiksi. Ensimmäisillä riveillä kutsutaan myös tiedonsiirto-ohjelmaa ja erilaisia python-kirjastoja.

Seuraavaksi ohjelmassa on luotu monia funktioita, joilla kullakin on eri käyttötarkoitukset. Funktioita pystytään kutsumaan ja suorittamaan muualta ohjelmasta. Ensimmäisenä on ”updatergb” (KUVA 44) -funktio, jolla saadaan päivitettyä käyttöliittymän väriarvokenttiä, kun sinne tuodaan uutta dataa vanhan tilalle. Monissa näistä kentistä käytetään ”rgbdata” -nimistä string-tyypin muuttujaa, joka on kameran tuleva data. Esimerkiksi punaisen arvo saadaan ”rgbdata[0:7]” -komennolla, joka lukee muuttujasta ensimmäiset 8 merkkiä. Näihin ensimmäisiin merkkeihin on tallennettu punaisen arvo. Tämä tiedetään siitä, että kameran ohjelmassa kaikki väriarvot tallennetaan yhteen muuttujaan, johon määritettiin arvojen tallennusjärjestys ja määritettiin myös, että jokainen väriarvo esitetään aina 7 merkillä.

Funktiolla ”imageupd” (KUVA 44) päivitetään käyttöliittymän kuvakenttä, jossa näytetään kameralta tuleva kuva. Funktioilla ”motoron” ja ”motoroff” (KUVA 44) käynnistetään moottori ja pysäytetään. ”Sensor” (KUVA 44) -funktiolla odotetaan, että magneettianturi tunnistaa magneetin läsnäolon. ”Ptrigger” (KUVA 44) -funktio asettaa output-signaalin päälle ja tämä on kytkettynä kameran trigger-inputille, jolloin kamera saa komennon ottaa kuva. Seuraava funktio ”motorfunc” (KUVA 44) kutsuu moottorin käynnistys- ja lopetus-funktioita niin, että moottori käynnistyy ja pysähtyy, kun magneettianturi havaitsee magneetin. Funktio ”count” (KUVA 45) laskee kameran väriarvoista värin heksaluvun ja tallettaa sen muuttujaan, jota käytetään muualla ohjelmassa.

”Main” -funktio (KUVA 45) on nimensä mukaisesti ohjelman runko ja siitä kutsutaan toisia funktioita halutusti. Mikäli ”running” -muuttuja on tosi niin ”main” -funktio ajetaan läpi. Jos taas ”running” -muuttujan tila on epätosi niin funktio hyppää loppuun ja odotetaan 5s, jonka jälkeen funktiota yritetään ajaa uudestaan. Tämä toiminto toistuu loputtomasti. Funktioiden ”start” ja ”stop” (KUVA 45) ainoa tarkoitus on muuttaa muuttujan ”running” tilaa, joko todeksi tai epätodeksi.

Funktioiden jälkeen viimeiset 25 ohjelmariviä (KUVA 46) ovat kaikki käyttöliittymään liittyviä toimintoja. Käyttöliittymä ohjelmoitiin käyttäen ”tkinter” kirjastoa. Ensin luodaan itse käyttöliittymän ikkuna ja tähän sitten luodaan painikkeet, tekstikentät ja kuvakenttä. Riveillä määritellään mm. kenttien koot, sijainnit ja tyypit, jotta ne tulevat hallitusti käyttöliittymän näytölle. Painikkeissa ”Automatic ON” ja ”Automatic OFF” on toiminnot, jotka kutsuvat funktioita ”start” ja ”stop”.

```

File Edit Format Run Options Window Help
import RPi.GPIO as GPIO
import time
from Tkinter import *
from PIL import Image, ImageTk
from Connection import transmit

GPIO.setmode(GPIO.BCM)
GPIO.setup(20,GPIO.OUT)#Kameran triggaus
GPIO.setup(26, GPIO.IN, pull_up_down = GPIO.PUD_UP)#Magnetic sensor pin 26
GPIO.setup(21,GPIO.OUT)#Motor output pin 21 setup
pwm = GPIO.PWM(21, 50) #Pin 21, 50Hz

running = False
rgbdata = ""
cal = ""
mycolour = ""

#Päivitetään väriarvokenttä
def updatergb():
    red.configure(text= "Red value" + " " + rgbdata[0:7])
    green.configure(text= "Green value" + " " + rgbdata[7:14])
    blue.configure(text= "Blue value" + " " + rgbdata[14:21])
    hue.configure(text= "Hue value" + " " + rgbdata[21:28])
    satr.configure(text= "Saturation value" + " " + rgbdata[28:35])
    intens.configure(text= "Intensity value" + " " + rgbdata[35:42])
    hexval.configure(text= "Color Hex value" + " #" +hex(cal) [2:])
    colorep.configure(bg=mycolour, font='helvetica 12')

#Päivitetään kuva
def imageupd():
    img2 = ImageTk.PhotoImage(Image.open("kuva0.bmp"))
    panel.configure(image = img2)
    panel.image = img2

def motoron():
    pwm.start(1)

def motoroff():
    pwm.stop()

def sensor():
    GPIO.wait_for_edge(26, GPIO.FALLING)

def ptrigger():
    GPIO.output(20, 1)
    time.sleep(0.1)
    GPIO.output(20, 0)

def motorfunc():
    motoron()
    time.sleep(0.5)
    sensor()
    motoroff()
    print ("Motorfunc ran till end")

```

KUVA 44. Python-pääohjelma osa 1

```

File Edit Format Run Options Window Help
def count():
    global cal
    global mycolour
    red1 = int(float(rgbdata[0:7]))
    green1 = int(float(rgbdata[7:14]))
    blue1 = int(float(rgbdata[14:21]))
    cal = (red1*65536)+(green1*256)+blue1
    mycolour = '#%02x%02x%02x' % (red1,green1,blue1)

def main():
    global rgbdata
    if running:
        motorfunc()
        ptrigger()
        rgbdata = transmit()
        count()
        time.sleep(10.0)
        updatergb()
        imageupd()
        frame.after(5000, main)#odotus ja main funktio alkaa uudelleen

def start():
    global running
    running = True

def stop():
    global running
    running = False

```

KUVA 45. Python-pääohjelma osa 2

```

#Create Frame
frame = Tk()
frame.geometry("800x480")

#RGB Labels
red = Label(frame, text="")
red.grid(row=0, column=0)
green = Label(frame, text="")
green.grid(row=1, column=0)
blue = Label(frame, text="")
blue.grid(row=2, column=0)
hue = Label(frame, text="")
hue.grid(row=3, column=0)
satr = Label(frame, text="")
satr.grid(row=4, column=0)
intens = Label(frame, text="")
intens.grid(row=5, column=0)
hexval = Label(frame, text="")
hexval.grid(row=6, column=0)
colorep = Label(frame, text="COLOUR", height =5, width=20)
colorep.grid(row=7, column=0, pady=5)

#Create Buttons
Button(frame, text="Automatic ON", command = start, bg="green", height=5, width=20).grid(row=8, column=0, pady=5)
Button(frame, text="Automatic OFF", command = stop, bg="red", height=5, width=20).grid(row=9, column=0, pady=5)

#Create Image
img = ImageTk.PhotoImage(Image.open("kuva0.bmp"))
panel = Label(frame, image = img)
panel.grid(row=0, column=2, rowspan=10, padx=100)

frame.after(1000, main)
frame.mainloop()

```

Ln: 75 Col: 66

KUVA 46. Python-pääohjelma osa 3

5 RAKENTAMINEN

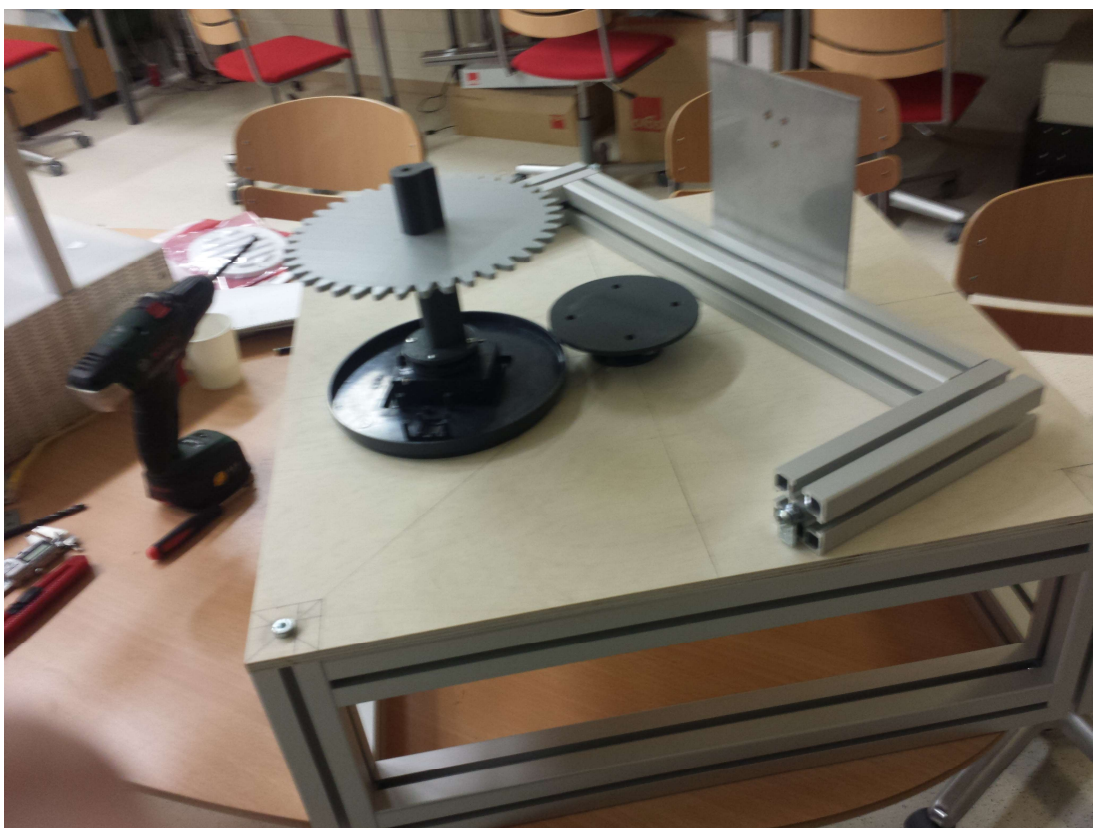
Laitteiston rakentaminen aloitettiin kasaamalla alumiiniprofiilirunko (KUVA 48), jonka jälkeen asennettiin valmiiksi mitaan sahattu pohjalevy ruuvien avulla. Seuraavaksi sahattiin päälilevyn lovet, jotta kameran tukirunkoa on mahdollista liikuttaa, sekä porattiin levyn keskelle akselin mentävä reikä. Porattiin myös kosketusnäytölle sopiva aukko levyn mahdollisimman reunaan, ottaen huomioon alumiiniprofiilit. Kun sahaukset olivat valmiit, maalattiin levy mustaksi spray-maalilla. Lopuksi kasattiin kameran tukirunko kolmesta profiilista ja kameran kiinnittämiseen tarkoitetulle metallilevylle porattiin reiät kameran kiinnitystä varten.

Koulun 3D-tulostinta käyttäen tulostettiin lokerot, akselit, laakerin kiinnikkeet, kameran tuki, sekä kosketusnäytölle suunniteltu lokero. Lokerot liimattiin yhteen pikaliiman avulla (KUVA 47) ja valmis lokerikko liimattiin kiinni pyöreään alustaan, joka maalattiin mustaksi. Laakerien kiinnikkeet porattiin päälilevyn kiinni ja asennettiin laakerit paikalleen. Pyörivään alustaan kiinnitettiin ylempi akseli myös ruuveilla.



KUVA 47. 3D-tulosteita

Seuraavaksi porattiin pienet reiät alemman akselin kantaan ja se kiinnitettiin kissan lelusta otettuun pyörivään alustaan kiinni. Haettiin alustalle ja alemmalle akselille oikea paikka niin, että akseli on suoraan päälilevyn keskellä olevan reiän alapuolella ja kiinnitettiin akseli, sekä sen pyörivä alusta pohjalevyyn kiinni. Lopullinen päälilevyn kiinnitys runkoon ja akselien liittäminen toisiinsa tehtiin vasta aivan lopussa, jotta rungon sisäpuolelle oli helpompaa tehdä sähköiset kytkennät.



KUVA 48. Runko, 3D-tulosteita ja kameran tukirakenne

Sähköiset kytkennät tehtiin hyödyntämällä DIN-kiskoa, johon asennettiin riviliittimiä ja rele. Riviliittimet asennettiin 24 voltin kytkentöjä varten, sekä 5 voltin kytkennöille. Seuraavaksi tehtiin suunnitellun piirikaavion mukaiset kytkennät. Kaikkien johtimien pituudet mitoitettiin silmämääräisesti niin, että ne olivat hieman tarvittua pidemmät, jotta niissä olisi joustovaraa. Kytkennät tehtiin ensin rungon ulkopuolella (KUVA 49), josta ne siirrettiin sen sisäpuolelle ja kiinnitettiin. Kaikkiin johtimien päihin puristettiin holkit, jotta ne saatiin helposti ja tukevasti asennettua esimerkiksi riviliittimiin.

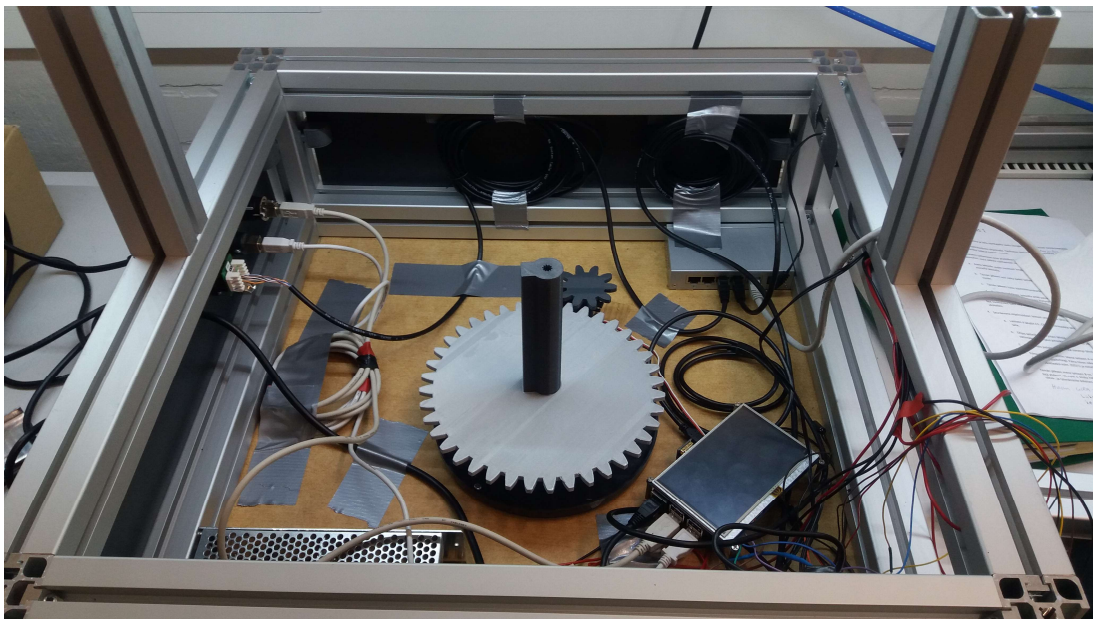
johtimet liittimen terminaaleihin. Päälilevyyn porattiin pieni reikä, jonka läpi magneettianturi vedettiin. Reikä porattiin niin, että se on pyörivän lokerikon ulkoreunalla. Päälilevyyn porattuun aukkoon upotettiin piirilevyille 3D-tulostettu lokero ja kiinnitettiin ruuveilla.



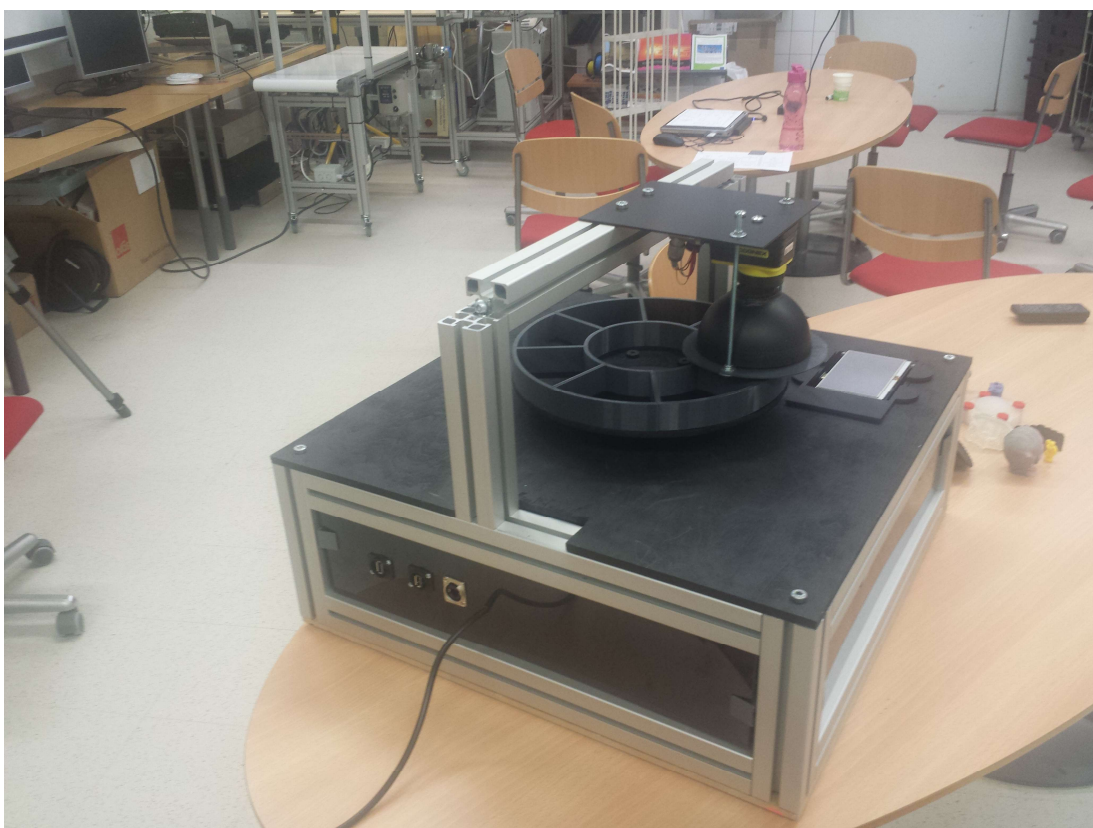
KUVA 50. Liittimet ulkopuolelta

Seuraavaksi kytkettiin kaikki Ethernet-kaapelit kytkimelle, sekä yksi kaapeli sivulevyyn Ethernet-liittimeltä. Moottori liimattiin pienen korokkeen päälle, jotta se oli samassa tasossa rataan kanssa. Moottori startattiin ja sen pyöriessä etsittiin sille sopiva kohta, jossa ratas pyörii sulavasti, jonka jälkeen moottori liimattiin pohjalevyyn kiinni (KUVA 51).

Kun kaikki rungon sisäpuolella oli valmista, laitettiin päälilevy takaisin paikoilleen ja liitettiin akselit yhteen. Metallilevyyn, jossa kamera on kiinni, porattiin reiät valaisimen kiinnitykselle. Reiät porattiin kameran molemmille puolille ja asennettiin valaisimen kiinnike kierretankojen avulla paikoilleen. Lopuksi liimattiin magneetit lokerikon pohjalle niin, että moottori pysähtyy oikeaan kohtaan. Magneettien oikeat sijainnit haettiin käynnistämällä moottori ja magneetteja siirrettiin sen mukaan, mihin moottori pysähtyi. Kuvassa 52 on valmis laite.



KUVA 51. Rungon sisäpuolella olevat kytkennät



KUVA 52. Valmis laite

6 KÄYTTÖOHJEET

Järjestelmää voidaan käyttää kahdella tapaa; ilman tietokonetta tai sen kanssa. Tietokoneen etuna on, että siitä voidaan katsella kameran ohjelman toimintaa.

Laitteen käyttö tietokoneen kanssa alkaa kytkemällä järjestelmään virrat mukana tulevalla johdolla, joka kytketään 230 VAC pistokkeeseen. Kun kosketusnäytössä näkyy työpöytä, voidaan halutessa kytkeä näppäimistö ja hiiri käyttämällä laitteen sivussa olevia USB paikkoja. Hiiren käyttäminen ei ole pakollista, mutta hieman helpottaa ohjelman käynnistystä. Kytketään laitteen sivussa olevaan Ethernet-liittimeen kaapeli ja kytketään se tietokoneeseen. Seuraavaksi voidaan avata tietokoneesta In-Sight Explorer ja odotetaan, että kamera saa luotua yhteyden.

Kun kamera on saanut yhteyden ja ruudulla näkyy ohjelma, varmistetaan, että se on värintunnistusohjelma eikä mikään muu. Mikäli ruudulla on toinen ohjelma, tarvitsee se vaihtaa asetuksista. Kun oikea ohjelma on valittuna, etsitään kameran ”network” -asetuksista sen IP-osoite. Kun IP-osoite on esillä, voidaan avata Raspberry PI:sta ”connection” -ohjelma, joka sijaitsee työpöydällä ”kamera” -kansiossa ja verrataan ohjelmassa olevaa IP-osoitetta kameran sen hetkiseen osoitteeseen. Mikäli ne eivät täsmää niin, muutetaan ”connection” ohjelmaan kameran IP-osoite, tallennetaan ja suljetaan ohjelma. Tätä vaihetta ei tarvitse toistaa joka kerta, mutta aina silloin, jos on muutettu kameran IP-osoitetta.

Seuraavaksi avataan pääohjelma, joka sijaitsee työpöydän kansiossa ”kamera” ja painetaan valikosta ”Run”. Ohjelma käynnistyy ja ruutuun ilmestyy käyttöliittymä. Seuraavaksi In-Sight-ohjelmasta laitetaan kamera online-tilaan, mikäli se ei ole. Laite on valmis käyttöön. ”Automatic ON” -painikkeesta ohjelman suoritus käynnistyy ja ”Automatic OFF” -painikkeesta pysähtyy. ”Automatic OFF” -painiketta on mahdollista painaa vain, kun moottori on pysähtynyt ja kaikki data on siirretty kameralta. Tämän tietää siitä että ”Automatic ON” -painike on muuttunut vihreäksi.

Kun järjestelmää halutaan käyttää ilman tietokonetta, tulee varmistaa, että kameran IP-osoite on sama kuin mikä on ”Connection” -ohjelmassa. Muuten kamera ja piirilevy

eivät pysty muodostamaan yhteyttä. Tulee myös varmistaa, että kameran käynnistyessä se automaattisesti suorittaa värintunnistusohjelman. Mikäli kamera käynnistää väärän ohjelman, se ei tietenkään toimi. Ennen Raspberry PI -pääohjelman käynnistämistä tulee odottaa, että kamera on saanut luotua yhteyden. Yhteys on luotu, kun kamera päästää äänen. Järjestelmä on valmis käyttöön.

7 POHDINTA

Mielestäni tämän opinnäytetyön tavoitteet täyttyivät. Suunniteltiin ja rakennettiin toimiva konenäköjärjestelmä, joka kykenee tunnistamaan kappaleiden värin ja ilmoittamaan sen käyttäjälle. Mielestäni laitteisto on sekä kiinnostava, että näyttävä ja helposti kuljetettava.

Laitteiston rungon alumiiniprofiilit olisivat voineet olla mielestäni pienemmät, jolloin laitteisto ei olisi aivan niin painava. Sen pystyy kantamaan yksin paikasta toiseen, koska siitä saa erittäin helposti tukevan otteen. Parannettavaa jäi mielestäni myös pyörivän alustan mekaanisessa suunnittelussa, koska alusta pyörii hieman jähmeästi, sekä se on hieman epävakaa. Alustalle ei tästä syystä pysty laittamaan mitään kovin painavia kappaleita. Kaiken kaikkiaan laitteisto kuitenkin toimii hyvin ja täyttää vaaditut kriteerit.

Järjestelmän suunnittelu oli kohtalaisen hankalaa, koska siinä on paljon eri asioita, joita tuli ottaa huomioon. Järjestelmän suunnittelu vaati osaamista ja perehtymistä monilta osa-alueilta. Erityisesti ongelmia tuotti Python-ohjelmointi, koska minulla ei ollut mitään kokemusta siitä vaan opiskelin sen itse ja useiden virheiden kautta sain toteutettua toimivan ohjelman. Erityisen vaikeaa oli myös moottorin akselien suunnittelu, koska minulla ei ollut oikeastaan minkäänlaista mekaanista suunnittelutaustaa. Helpointa oli Solidworksin 3D-mallinnusohjelman käyttö, koska olin ennen käyttänyt sitä. Laitteiston kytkentöjen tekeminen ja rakentaminen olivat kohtalaisen helppoja, mutta ne vaativat paljon tarkkuutta ja aikaa.

LÄHTEET

Savonautomaatio Oy. N.d. Viitattu 18.2.2017. <http://www.savonautomaatio.fi/palvelut/konenako/>

Stemmer-Imaging GmbH: Application areas for vision systems. N.d. Viitattu 18.2.2017. <http://www.stemmer-imaging.co.uk/en/knowledge-base/vision-systems-application-areas/>

Diane Berkenfeld, Dave Black, Mike Corrado & Lindsay Silverman. Understanding focal length. N.d. Viitattu 18.2.2017. <http://www.nikonusa.com/en/learn-and-explore/article/g3cu6o2o/understanding-focal-length.html#!>

Nikonusa: Understanding Maximum Aperture. N.d. Viitattu 19.2.2017. <http://www.nikonusa.com/en/learn-and-explore/article/g3cu6o1r/understanding-maximum-aperture.html>

SAMK automation tutkimusryhmä: Matriisikenno – vai viivakamerakuvaus. N.d. Viitattu 22.2.2017 http://automaatio.samk.fi/?page_id=80

Basler AG: Areas of Application for Line Scan Cameras. N.d. Viitattu 22.2.2017. <http://www.baslerweb.com/en/products/cameras/line-scan-cameras>

Stemmer-Imaging GmbH: Cameras. N.d. Viitattu 22.2.2017. <http://www.stemmer-imaging.co.uk/en/knowledge-base/cameras/>

National Instruments Corp: A Practical Guide to Machine Vision Lighting. N.d. Viitattu 22.2.2017. <http://www.ni.com/white-paper/6903/en/>

Red Digital Cinema: The Bayer sensor strategy. N.d. Viitattu 22.2.2017. <http://www.red.com/learn/red-101/bayer-sensor-strategy>

Stemmer-Imaging GmbH: Configuration and programming. N.d. Viitattu 22.2.2017. <http://www.stemmer-imaging.fi/en/knowledge-base/vision-systems-configuration-and-programming/>

Movetec Oy: Catalogue 2017. Viitattu 30.3.2017. <http://www.movetec.fi/images/pdf/Profilesystem.pdf>

Cognex: In-Sight 7000 Integrated Vision System. N.d. Viitattu 30.3.2017. <http://www.cognex.com/products/machine-vision/in-sight-7000-series-integrated-vision-systems/?id=13608&langtype=2057>

TekSis: In-sight 7000 series vision system installation manual 2011 - 2012. Viitattu 30.3.2017. <http://www.teksis.co.uk/downloads/datasheets/IS7010-01.pdf>

Python: What is Python? Executive summary. N.d. Viitattu 30.3.2017. <https://www.python.org/doc/essays/blurb/>

Raspberry PI 2 Model B: Specifications. N.d. Viitattu 30.3.2017. <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

Sunfounder: 5 Inch HD 800x480 HDMI LCD Touch Screen Monitor for Raspberry Pi. N.d. Viitattu 4.4.2017. <https://www.sunfounder.com/sf-5-hdlcdscr.htm>