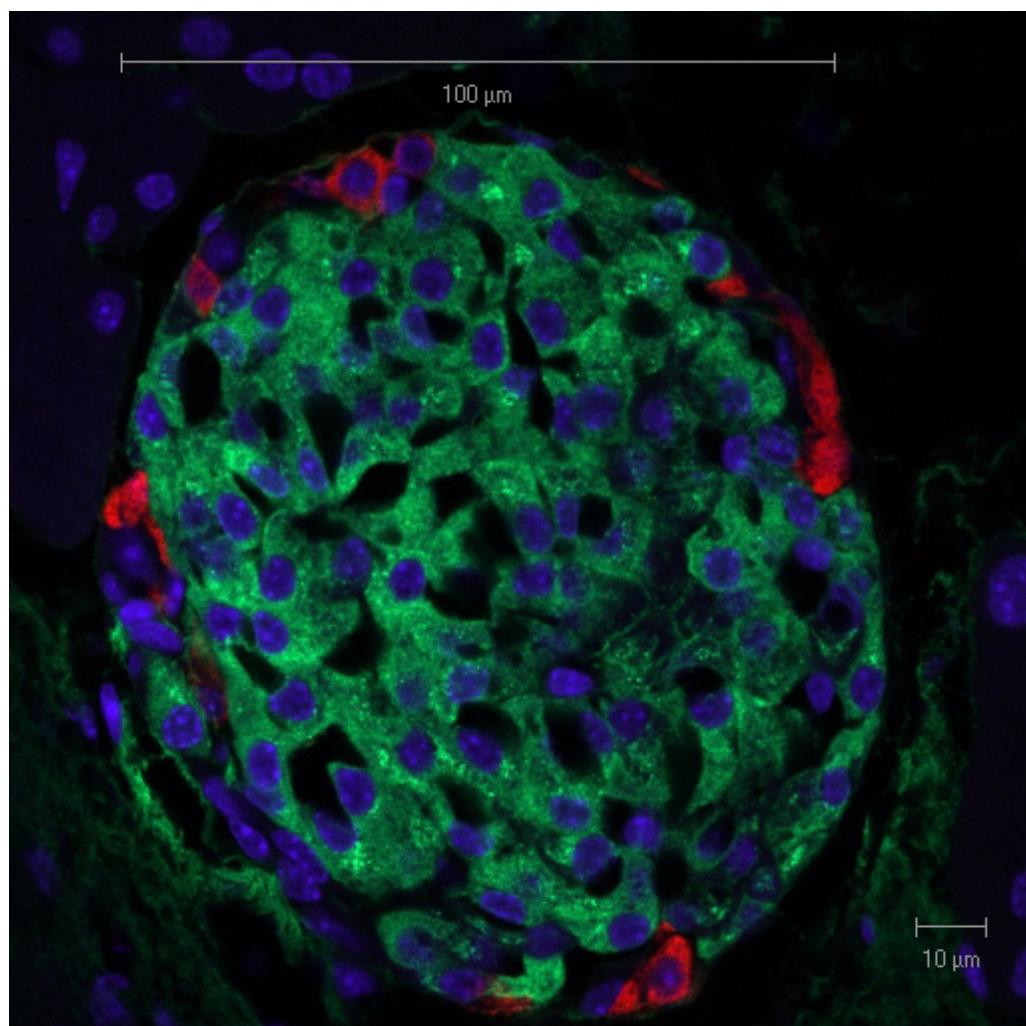


Cell sorter for isolation of insulin producing cells
Rapport



BACHELORPROJEKT
PROJEKTNR: 15137
INGENIØRHØJSKOLEN, AARHUS UNIVERSITET
DEN 16. DECEMBER 2015

11242 ANDERS TOFT ANDERSEN
201270874 ANDERS ESAGER
PROJEKTVEJLEDER: SAMUEL ALBERG THRYSSØE

Forord

Denne rapport er udarbejdet som en del af et syvende semesters bachelorprojekt på Ingeniørhøjskolen, Aarhus Universitet. Rapporten er udarbejdet af en projektgruppe bestående af 2 sundhedsteknologistuderende. Projektet er udført i samarbejde med Søren Gregersen, overlæge på Medicinsk Endokrinologisk Afdeling på Aarhus Universitetshospital med hjælp fra Per. B. Jeppesen, Lektor ved Institut for Klinisk Medicin, Aarhus Universitet. Bachelorprojektet er udført i perioden 28. august 2015 til 16. december 2015, hvor forprojektet er udarbejdet i perioden 26. april 2015 til 15. juni 2015.

Projektgruppen retter en stor tak til Søren Gregersen for samarbejdet, ligeledes skal der gives en tak til Per B. Jeppesen. Ydermere skal der lyde en varm tak til gruppens vejleder Samuel Thrysøe, der har hjulpet og støttet gruppen igennem hele processen. Endelig skal der gives en tak til reviewgruppen bestående af Simon Vammen Grønbæk og Karl-John Schmidt, som har givet konstruktiv kritik og rettelser.

Anders Toft Andersen

Anders Esager

Læsevejledning

Rapporten indeholder primært metoder, resultater og diskussioner til produktet gruppen har udarbejdet. Der vil igennem rapporten fremtræde kildehenvisninger, som vil være samlet i en kildebane bagerst i rapporten. Der er i rapporten anvendt kildehenvisning efter Harvardmetoden, i teksten refereres en kilde med [Efternavn, År]. Denne henvisning fører til kildebanen, hvor bøger er angivet med forfatter, titel, udgave og forlag, mens internetsider er angivet med forfatter, titel og dato. Til sidst i rapporten er bilagsliste, som anskueliggør filnavnene i den afleverede bilagsmappen. Diagrammerne udarbejdet i projektet er skrevet på engelsk.

I bilagslisten forefindes alle filerne, der er afleveret ved siden af rapporten, herunder datablade, Matlab-kode, Eagle kilde-filer og Gerber-filer. Herudover er den udfyldte accepttest og fejlrapport vedlagt som bilag.

Ved siden af rapporten er vedlagt en video, som viser den udviklede prototype.

Ordliste

| Forkortelser | Betydninger |
|--------------|-----------------------------|
| ADC: | Analog Digital Converter |
| BDD: | Block Definition Diagram |
| CMRR: | Common Mode Rejection Ratio |
| GUI: | Graphical User Interface |
| IBD: | Internal Block Diagram |
| KCL: | Kirckhoff Current Law |
| KVL: | Kirckhoff Voltage Law |
| MDD: | Medicinsk Device Direktivet |
| PCB: | Printed Circuit Board |
| PWM: | Pulse Width Modulation |
| SYSML: | SYStems Modeling Language |

Abstract

Background Insulin is a hormone essential for regulating glucose levels in the blood. A decrease in insulin production in the pancreatic langerhans islets can lead to the life-threatening disease *diabetes mellitus*. To further investigate the disease researchers conduct experiments with islets isolated from mice. This is done by surgically removing the pancreas, dissolving the pancreatic tissue with the enzyme *collagenase* and manually picking the islets from a suspension. This procedure is both cumbersome and time-intensive. Hence, new ways are warranted.

Methods Through an agile development process a *Proof of Concept* prototype has been developed for automatic isolation of the islets of langerhans. The development process consisted of four general phases: conceptual phase, specification and verification/test phase, system design phase, and finally implementation phase. The project was managed with a stage gate model where each development phase was given a specific deadline. In cooperation with a review group each phase was reviewed after the deadline. To provide an overview of the different tasks and sprints in the project the scrum based PivotalTracker was used.

Results & discussion The developed prototype consists of a software program developed in *Matlab* and different Hardware components. The system works by processing images taken from a microscope camera. If an islet is detected a mechanical valve will isolate the islet from the remaining fluid. However a unit test showed that the camera wasn't sufficient to detect the islets. In the final prototype a function has been implemented to simulate the camera by feeding a sequence of pre generated images.

In addition a cost-benefit analysis has disclosed the economic benefits of an automatic isolation method.

Conclusion The project has shown that it is possible to detect islets of langerhans through image processing of the generated images. However, the system has failed to isolate objects from the simulation fluid due to an obstruction in the mechanical valve. The project also showed which issues need to be addressed before the system can be implemented in practice. The results of the cost-benefit analysis has given incentive to do further research towards an automatic system for isolation of pancreatic islets.

Resume

Baggrund Insulin er et essensielt hormon til regulering af glukose i blodet. Et fald i insulin produktionen ved nedsat funktion i de langerhanske øer kan føre til den livstruende sygdom *diabetes mellitus*. Til at undersøge sygdommen og funktionen af de insulinproducerende øer laves der videnskabelige forsøg med langerhanske øer fra mus. Denne proces foregår ved operativt at fjerne pankreas for herefter at opnåse den ved hjælp af enzymet *collagenase*. De enkelte øer isoleres herefter manuelt ved at plukke dem fra en petriskål. Denne proces er både besværlig og tidskrævende. Derfor ønskes der nye metoder til isolering af langerhanske øer.

Metoder Igennem en agil udviklingsproces er der udviklet en *Proof of Concept* prototype til automatisk isolering af langerhanske øer. Udviklingsfasen bestod overordnet af fire faser, hhv. konceptanalyse, kravspecifikation og accepttest, design, samt implementering. Projektets overordnede tidsplan er opbygget som en stage gate model, hvor de enkelte udviklingsfaser er fastlagt med deadlines. I samarbejde med projektets reviewgruppe er de enkelte faser reviewed efter hver deadline. For at holde styr på arbejdsopgaverne er scrumværktøjet PivotalTracker anvendt, som har givet et overblik over de enkelte ugers sprints.

Resultater og diskussion Den udviklede prototype består af et software program udviklet i *Matlab*, samt en række hardwarekomponenter. Prototypen virker ved brug af billedprocessering, der detekterer de enkelte øer når de passerer et kamera. Ved detektion isoleres øen vha. en ventil i en separat beholder. En enhedstest af det indkøbte kamera viste, at det ikke var af tilstrækkelig kvalitet til detektion af øerne. Derfor er der i prototypen implementeret en funktion til simulering af kameraet ud fra genererede billeder.

Herudover er en cost-benefit analyse udarbejdet til at belyse, hvilke økonomiske fordele en automatiseret løsning vil have.

Konklusion Projektet har vist, at det er muligt at detektere langerhanske øer vha. billedprocessering ud fra genererede billeder. Det lykkedes ikke i projektet at isolere objekterne i simuleringsvæsken pga. blokering ved ventilen. Projektet har yderligere givet et indblik i hvilke problemstillinger der skal løses inden en fungerende prototype kan implementeres i praksis, herunder hvorvidt øerne tager skade af processen.

Resultatet af cost-benefit analysen giver incitament til at arbejde videre med en automatiseret løsning til isolering af øerne.

Indholdsfortegnelse

| | |
|---|-----------|
| Kapitel 1 Indledning | 1 |
| 1.1 Baggrund | 1 |
| 1.2 Problemformulering | 3 |
| 1.3 Afgrænsning | 4 |
| Kapitel 2 Systembeskrivelse | 5 |
| 2.1 Langerhanske øer | 6 |
| Kapitel 3 Metoder | 7 |
| 3.1 Samarbejds aftale | 7 |
| 3.2 Samarbejdspartnere | 7 |
| 3.3 Udviklingsværktøjer | 8 |
| 3.4 Versionsstyring | 9 |
| 3.5 Projektstyring | 10 |
| 3.6 Udviklingsfaserne | 13 |
| 3.7 Første udviklingsfase: Konceptanalyse | 15 |
| 3.8 Anden udviklingsfase: Kravspecifikation og accepttest | 16 |
| 3.9 Tredje udviklingsfase: Design | 19 |
| 3.10 Fjerde udviklingsfase: Implementering og enhedstest | 25 |
| Kapitel 4 Resultater | 29 |
| 4.1 Det udviklede system | 29 |
| 4.2 Ventil tidsinterval | 34 |
| 4.3 Kamera | 37 |
| 4.4 Simuleringsvæske | 42 |
| 4.5 Accepttest af produktet | 44 |
| 4.6 Cost-benefit analyse | 44 |
| Kapitel 5 Diskussion | 47 |
| 5.1 Billedprocessering | 47 |
| 5.2 Ventil og simuleringsvæske | 47 |
| 5.3 Cost-benefit analyse | 48 |
| 5.4 Hardware | 48 |
| 5.5 Accepttest | 48 |
| Kapitel 6 Perspektivering | 49 |
| 6.1 Kamera | 49 |
| 6.2 Påvirkning af langerhanske øer | 49 |
| 6.3 Pumpe | 49 |
| 6.4 Ventil | 50 |
| 6.5 Parallelle systemer | 50 |

| | |
|---|-----------|
| 6.6 Billedprocessering | 50 |
| 6.7 Omrøring og køling af celleopløsningsbeholderen | 50 |
| 6.8 Medicinsk udstyr | 50 |
| Kapitel 7 Konklusion | 53 |
| Litteratur | 55 |
| Appendiks A Bilag | 59 |
| A.1 Datablade | 59 |
| A.2 Matlab kode | 59 |
| A.3 Arduino Testkode | 60 |
| A.4 Mødereførerater | 60 |
| A.5 Mail korrespondancer | 60 |
| A.6 Accepttest | 60 |
| A.7 Fejlrappport | 60 |
| A.8 Logbøger | 60 |
| A.9 Samarbejds aftale | 60 |
| A.10 Handlingsplan | 60 |

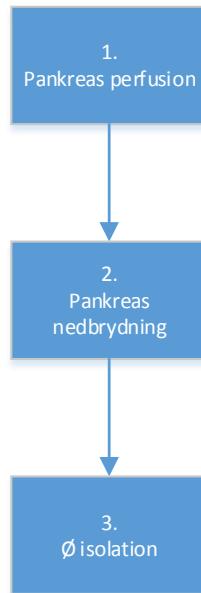
Indledning 1

Hormonet insulin er vigtigt til regulering af glukoseniveauet i blodet. Produktionen af insulin foregår i pancreas, nærmere bestemt i de langerhanske øer. Et fald i insulinproduktionen, grundet nedsat funktion af de langerhanske øer, kan føre til den livstruende sygdom *diabetes mellitus (type 1)*. Prævalensen af diabetes mellitus er i Danmark på 320.545, hvor omkring 10 % lider af type 1 diabetes [Diabetesforeningen, 2015]. På grund af den store udbredelse af diabetes er der stor interesse i forskning på området. Til forskningen bruges der ofte en stor mængde af langerhanske øer.

1.1 Baggrund

For at undersøge sygdommen nærmere, samt øge forståelsen for de mekanismer, der styrer insulinreguleringen i kroppen udføres der videnskabelige forsøg med langerhanske øer.

De øer der anvendes til videnskabelige forsøg stammer typisk fra mus eller rotter. Sortering- og isoleringsprocessen foregår ved tre faser [Jeppesen, 1997], som vist i figur 1.1.



Figur 1.1. Faser i sorteringsprocessen

I første fase, *pankreas perfusion*, sprøjtes enzymet *collagenase V* ind gennem galdegangene og derfra videre til pankreas. Dette enzym starter en nedbrydning af vævet i pankreas. Herefter fjernes pankreas operativt. Enzymet collagenase har den egenskab, at det ikke

nedbryder de langerhanske øer i samme grad som det omkringliggende eksokrine væv.

I anden fase, *pankreas nedbrydning*, nedbrydes pankreas yderligere ved at inkubere pankreas ved 37° i 19 min. Ved denne temperatur er enzymet collagenase særligt aktivt og katalysere derfor nedbrydningen af pankreas. Herefter nedkøles vævet for at stoppe virkningen af collagenase.

I den sidste fase, *ø isolation*, vaskes og rystes øerne først af tre omgange med en vaskebuffer i form af en saltvandsopløsning [HBSS]. Dette gøres for at løsrive øerne fra hinanden inden isolering. Herefter er øerne klar til at blive isoleret fra det eksokrine væv. Der findes en række forskellige metoder til dette, hvor den mest udbredte metode foregår ved manuel isolering af øerne fra en petriskål vha. et mikroskop. Der bruges allerede automatiserede isoleringsmetoder bygget på forskellige teknikker, herunder gradientbaseret centrifugering. Fælles for de anvendte teknikker til isolering af øerne er, at der er stor risiko for skade på øerne.

Den manuelle metode har yderligere ulemper, idet den både er besværlig og tidskrævende. Herudover kan der være stor variation i kvaliteten af isoleringen, da der kan være forskel på den enkelte operatørs håndtering af øerne.

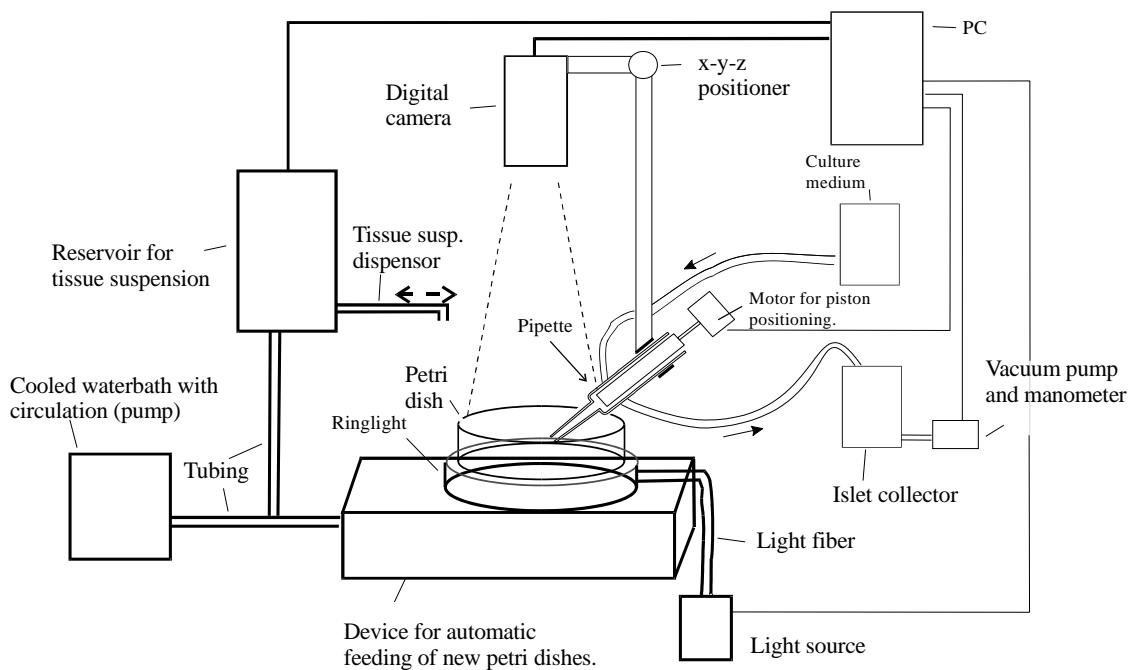
Derfor ønskes der en automatiseret metode til isolering af langerhanske øer, med minimal risiko for skader på øerne. En automatiseret metode vil have følgende fordele Gregersen [2015]:

- Øget sorteringshastighed for højere udbytte
- Reducere variation i de isolerede øer
- Reducere omkostningerne
- Sikre bedre dokumentation
- Forbedret arbejdsmiljø

En automatiseret løsning vil potentielt åbne op for nye muligheder indenfor anvendelse af langerhanske øer. Der forskes bl.a. i transplantation af langerhanske øer, som et led i behandling af type 1 diabetes. Resultaterne af et af disse forskningsprojekter har bl.a. vist at 44 % af modtagerne af denne type behandling var insulinuafhængige 3 år efter transplantation [Bruni et al., 2008]. Til forskning er langerhanske øer bl.a. anvendt til at undersøge hvordan aminosyrer er glukoseafhængige [Zhenping Liu og Hermansen, 2014]. I forskningsforsøget er der brugt 6-10 mus, hvilket cirka har taget 6-10 timer at håndplukke (jvf. Per B. Jeppesen). Det er forskningsprojekter som dette, hvor en automatiseret løsning vil være relevant, da det vil skabe en mulighed for en større testgruppe og frigive ressourcer til andre formål i forsøget.

På Medicinsk Endokrinologisk Afdeling (MEA), Aarhus Universitetshospital, er der tidligere arbejdet hen imod en automatiseret løsning til isolering af langerhanske øer. Den automatiseret løsning vil primært bidrage til forskningen, der foretages på afdelingen. Tidligere blev der udviklet en prototype til automatiseret ø-isolering, der fungerede ved hjælp af kameradetektion. Ved detektion sugede en pipette den detekterede ø op fra en petriskål. Pipetten blev styret i X,Y og Z retning vha. en mekanisk robotarm. Se figur 1.2 for en illustration af den gamle prototype.¹

Schematic drawing of apparatus for isolation of Islets of Langerhans.



Figur 1.2. Tidligere prototype

Prototypen var dog ikke præcis nok, primært pga. bevægelse i væsken, og projektet blev derfor stoppet.

1.2 Problemformulering

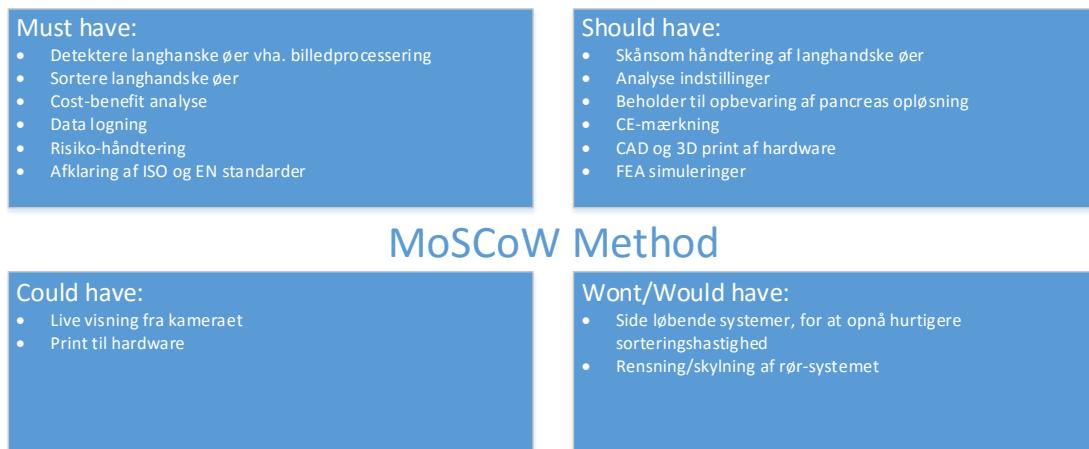
Målet med dette projekt er at udvikle et *Proof of Concept* system til isolering af langerhanske øer, i samarbejde med Medicinsk Endokrinologisk Afdeling. Systemet bygger videre på de erfaringer der er opnået ved den tidligere prototype. Metoden til isoleringen baseres fortsat på kameradetektion, mens den mekaniske isolering i stedet erstattes med et ventilsystem. Det udviklede system skal altså kunne detektere langerhanske øer vha. billedprocessering for herefter at isolere øerne fra resten af opløsningen med en ventil.

Herudover skal en cost-benefit analyse være med til, at belyse hvilke økonomiske og kvalitative fordele det udviklede system kan have, sammenlignet med den anvendte manuelle metode.

¹Kilde: Søren Gregersen

1.3 Afgrænsning

Til afgrænsning i projektet anvendes MoSCoW modellen, som beskriver hvilke dele projektet skal (**Must**), bør (**Should**), kan (**Could** og ikke (**Won't**) indeholde. MoSCoW modellen (figur: 1.3) viser hvordan de enkelte krav og dele af projektet er prioriteret.



Figur 1.3. MoSCoW

De krav systemet skal opfylde er bla. at kunne detektere langerhanske øer vha. billedprocessering, samt sortere øerne ved detektion. Herudover skal systemet kunne gemme data omkring de sorterede øer, herunder størrelse og cirkularitet. Desuden består Projektet af en cost-benefit analyse der beskriver hvilke økonomiske fordele og ulemper det automatiserede system vil have.

Et af de vigtigere krav, der er nedprioriteret i projektet er en skånsom håndtering af langerhanske øer. Denne del vil kræve en validering af den udviklede prototype igennem funktionstest af de isolerede langerhanske øer.

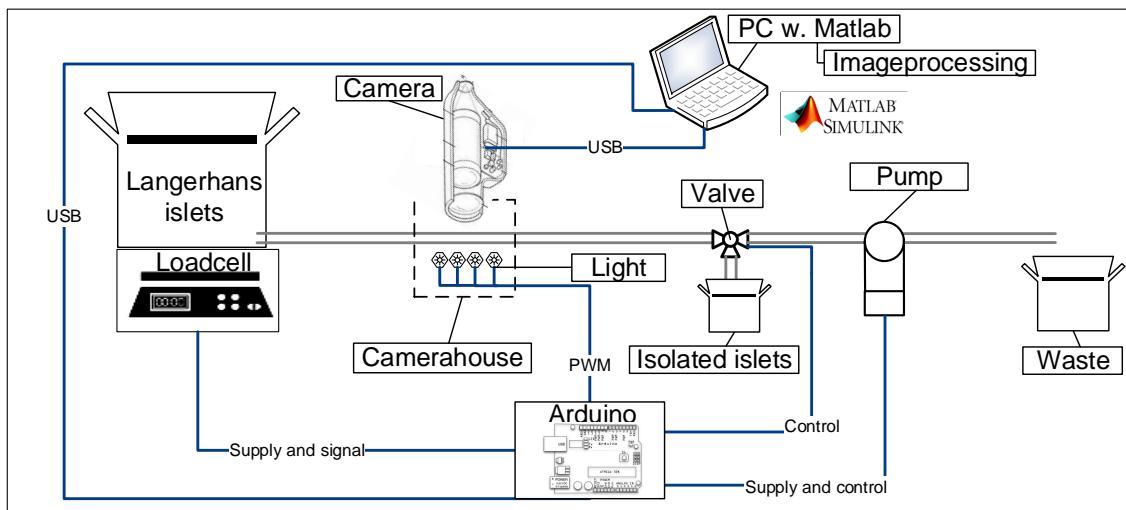
Dette projekt vil derfor i højere grad fokusere på en verificering af den udviklede prototype i form af en accepttest, som tester de funktionelle og ikke funktionelle krav. Derfor er punkterne fordelt på *MoSCoW* modellen som vist. Punkterne i *must have* definerer projektets minimum krav. Det er gjort for at tilpasse arbejdsbyrden i projektet, og at projektet kommer sikkert i mål.

Da de langerhanske øer er svært tilgængelige, skal der findes en simuleringsvæske. En simuleringsvæske skal bruges til at teste systemets mekaniske funktioner. De mekaniske funktioner består i isoleringen processen af de langerhanske øer, hvilket er en elementær del af projektet. Se afsnit 4.4 for uddybelse af simuleringsvæsken.

Systembeskrivelse 2

Dette kapitel indeholder en introducerende beskrivelse af den udviklede prototype. Herudover indeholder kapitlet en beskrivelse af opløsningen med de langerhanske øer.

Figur 2.1 viser den overordnede opbygning af *The Cell Collector*, benævnes herefter som systemet. Grundlæggende består systemet af en beholder, hvor i opløsningen med langerhanske øer er. Opløsningen pumpes herefter igennem en slange, hvor et kamera detekterer om en ø passerer eller ej. Hvis en ø detekteres åbner systemet en ventil som frasortere øen fra resten af opløsningen. Systemet består af et software program udviklet i *Matlab*, hvor operatøren har mulighed for at interagere med systemet. Her er selve logikken og signalbehandlingen i form af billedprocessering implementeret. Programmet giver signal til en microcontroller (Arduino platform), som håndterer den videre styring af hardwarekomponenterne.

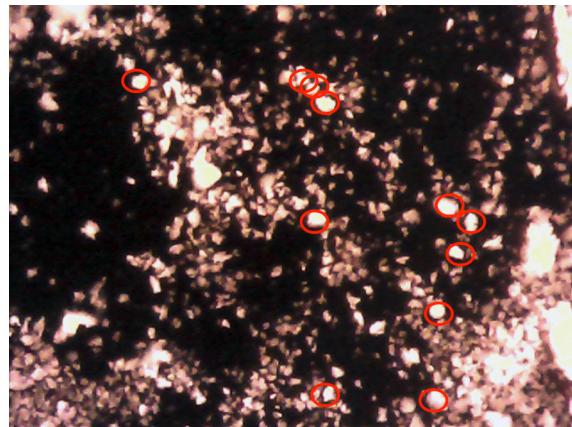


Figur 2.1. Figuren viser den overordnede opbygning af systemet

2.1 Langerhanske øer

Dette afsnit beskriver nærmere opløsningen med de langerhanske øer. Den endelige opløsning består udover øerne af eksokrint væv fra pankreas, samt af en saltvandopløsning (HBSS).

Figur 2.2 viser hvordan opløsningen ser ud, når den er hældt i en petriskål. På billedet er øerne markeret med en rød cirkel.



Figur 2.2. Opløsning med langerhanske øer. Øerne er markeret med en rød cirkel

Et batch af langerhanske øer, som opnåes ved en typisk sorteringsproces på Medicinsk Endokrinologisk Afdeling består af mellem 300 og 400 øer. For at få et udbytte på 300 til 400 øer anvendes typisk mellem 5 og 6 mus. Den samlede mængde opløsningsvæske for sådan et batch er omkring 150 mL.

De generelle fysiske egenskaber for de langerhanske øer er en størrelse mellem 100 og 300 µm, samt at de er forholdsvis runde. Herudover har de en højere lysintensitet end det eksokrine væv. Det vil bl.a. være disse parametre kameraet skal detektere øerne på.

Metoder 3

Dette kapitel indeholder beskrivelser af hvordan projektet er udført, og hvilke metoder, der er brugt. Yderligere indeholder kapitlet projektstyring, samt hvilke modeller, der er fulgt igennem projektforløbet.

3.1 Samarbejdsaftale

For at sikre gruppens enighed i udførelse af projektet, om bla. arbejdsindsats samt timer der skulle bruges på projektet, blev der udarbejdet en samarbejdsaftale. Samarbejdsaftalen bevirker derfor som en forventningsafstemning. Se samarbejdsaftalen i bilag A.9

3.2 Samarbejdspartnere

Gruppens kunde er Søren Gregersen, overlæge på Medicinsk Endokrinologisk Afdeling, Aarhus Universitetshospital. Det er i samarbejde med Søren, at projektet er blevet specifiseret, samt hvilke krav der er til den endelige prototype. Samuel Alberg Thrysøe er gruppens projektvejleder. Der er afholdt ugentlige vejledermøder, hvor gruppen har givet status på projektet og hvor der er diskuteret forskellige problemstillinger. Til hvert vejledermøde har der været en dagsorden med punkter, som formål og begrundelse for punkterne på dagsordenen. Dagsordenerne er sendt til vejleder senest en dag før vejledermødet. Dette er gjort for at give vejleder en chance for at forberede sig på den givne dagsorden. Til hvert vejledermøde er der udført et referat som kan ses i bilag A.4.1. Simon Vammen Grønbæk og Karl-Johan Schmidt har fungeret som projektets reviewgruppe. Der er holdt møde hver tredje uge omhandlende aftalt dagsorden. Formålet med reviewgruppen er, at få konstruktiv feedback på evt. rettelser, opbygning af rapport og generel forståelse. Reviewgruppen har været gode til at sparre med omkring diskussioner, samt andre spørgsmål i gruppen. Dette har været med til at drive projektet igennem udviklingsfaserne, da deadlines har været fastlagt i samarbejdet med reviewgruppen.

Reviewmøderne har forgået ved, at grupperne har udvekslet dokumenterne, der skulle reviews. Grupperne har skrevet kommentar til dokumentet, hvorefter kommentarene og rettelserne er diskuteret på møderne.

3.3 Udviklingsværktøjer

Dette afsnit består af en kort beskrivelse omkring udviklingsværktøjer brugt i projektet.

MATLAB: Softwaren til systemet er udviklet i *MATLAB*. Versionen, der er anvendt, er R2015b. Tilføjelsespakker brugt i projektet er følgende:

- Arduino Support Package (14.2.2)
- Webcam Support Package (15.2)

Disse pakker anvendes til at interagere med Arduinoen og USB mikroskopet.

Fritzing: *Fritzing* er brugt til at illustrere kredsløbsdiagrammer(schematics), samt enhedstest af hardwaren ved illustrationer af *fumlebræt*. Dette har bidraget til dokumentationen af enhedstestene. Derudover har det givet mulighed for, at opstillingerne brugt på *fumlebræt*, let kunne genskabes.

Eagle: *Eagle* er et program til udarbejdelse af print layouts. *Eagle* er brugt til at lave et samlet PCB layout for hardwarelementerne. I *Eagle* er der generet Gerber-filer, som er sendt til PCB fabrikanten.

SolidWorks: *SolidWorks* er et program til at illustrere mekaniske 3D modeller med. *SolidWorks* er brugt til at lave kamerahuset med for senere, at få det 3D printet.

Arduino IDE: Programmet som er udbudt af *Arduino* bruges til at programmere *Arduino* udviklingskortet. *Arduino IDE* er i projektet brugt til enhedstest for at teste hardware opstillingerne.

Microsoft Visio: *Microsoft Visio* er et tegne program, som bruges til at illustrere forskellige modeller. Programmet er valgt til at lave udviklingsdiagrammer og illustrationer med.

Texmaker: Dette program er brugt til at skrive alt dokumentationen i projektet. *Texmaker* gør det muligt at arbejde på samme dokument samtidigt, hvilket har været en stor hjælp.

PivotalTracker: *PivotalTracker* er et scrumbaseret projektstyringsværktøj, der hjælper med at styre arbejdsressourcerne i projektet. Der er i afsnittet 3.5.1 uddybet hvordan *PivotalTracker* er brugt i projektet.

GitHub: Er et versionsstyringsprogram, som i projektet er brugt til versionsstyring af dokumenter og *Matlab* koden til projektet. Se afsnit 3.4.1 for uddybende dokumentation omkring hvordan *Github* er benyttet.

Dropbox: Det cloudbaserede delingsværktøj *Dropbox* er i projektet brugt til at gemme og dele artikler, diagrammer, billeder og generelle noter.

Microsoft Onenote: *Onenote* er et program til at lave noter i. I projektet er programmet brugt til at have fælles noter og huskelister.

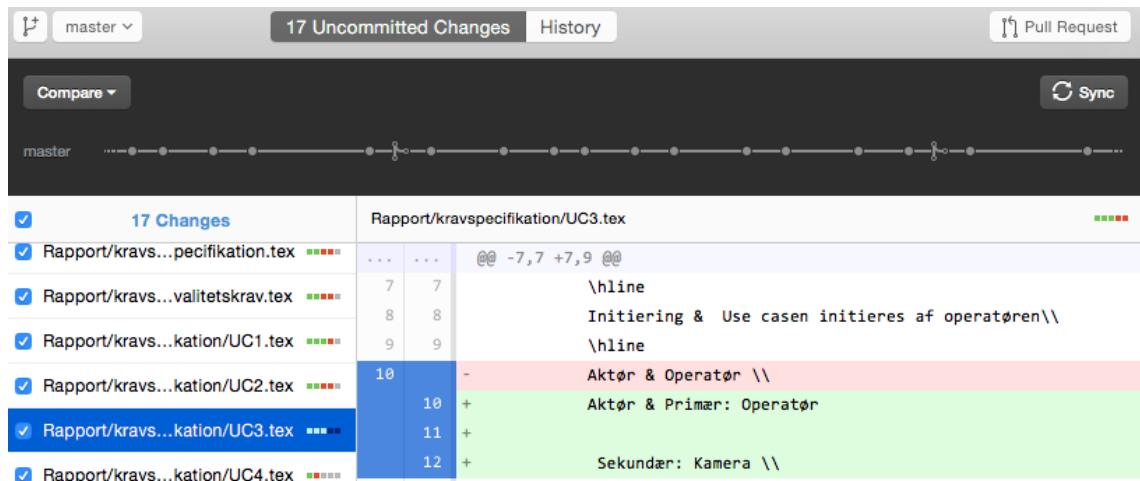
3.4 Versionsstyring

I projektet er der gjort brug af versionsstyringssoftwaren *GitHub*. Til store ændringer har dokumenterne fået et nyt versionsnummer og versionshistorikstabellen er opdateret i hvert dokument, se tabellen nedenfor.

| Version | Dato | Beskrivelse | Initialer |
|---------|------------|--|-----------|
| 0.1 | 19/09 2015 | Dokument sendt til review | AE og AT |
| 1.0 | 19/09 2015 | Rettelser fra reviewmøde og Latex layout | AE og AT |
| 1.1 | 20/10 2015 | Kamerakrav tilføjet | AE og AT |
| 2.0 | 6/11 2015 | Definitioner og layout ændringer | AE og AT |

3.4.1 Github

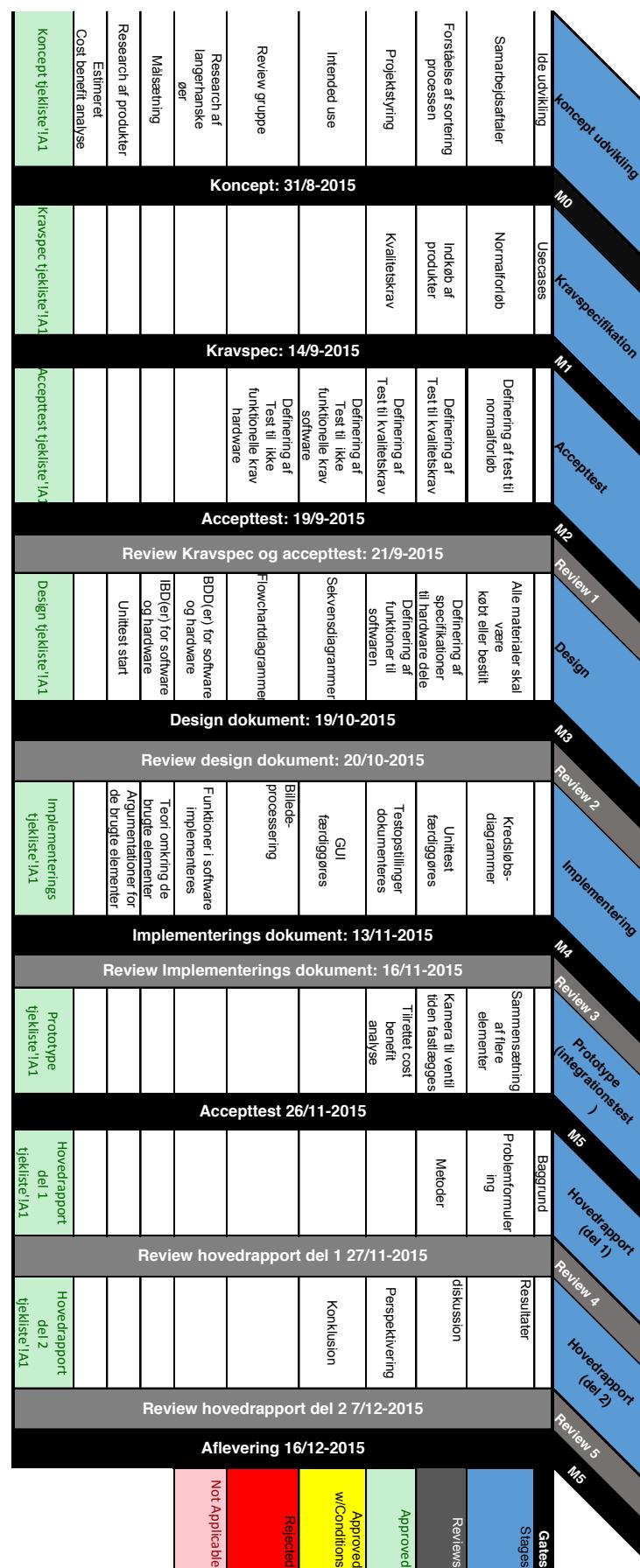
Til versionsstyring af projektdokumentationen og source kode anvendes GitHub, som bygger på open source versionsstyringssystemet Git. Her opdateres der løbende ændringer, så det nyeste dokumentation og source kode altid er tilgængeligt. Som user interface til GitHub anvendes GitHub Desktop (figur: 3.1). I GitHub Desktop vises en tidslinje, for hvornår, der er lavet ændringer. Under de enkelte filer kan det observeres, hvad der er ændret i for den gældende version. Programmet giver yderligere indblik, i hvilke filer, der lokalt er lavet ændringer i, som ikke er tilføjet repositoriet endnu.



Figur 3.1. GitHub Desktop

3.5 Projektstyring

Til projektstyring er der brugt en stage gate model. En stage gate model (figur 3.2) er bestående af nogle udviklingsfaser (stages), hvor der er en deadline for de konkrete faser. For at komme til næste fase/stage, skal der være opfyldt nogle kriterier. Kriterierne opstilles i en tjekliste (se figur 3.3), hvor de kan krydses af. Alle punkter skal være opfyldt for at komme igennem gaten. En stage gate model er god til at få et produkt på markedet, men det har sine svagheder ved en agil udviklingsprocess. Derfor er der i projektet arbejdet med åbne gates, hvilket gør det muligt at gå tilbage for at justere på dokumenter. I projektet er stage gate modellen brugt som tidsplan og en metode, der har været med til at sikre, at fasen var færdig inden næste fase igangsættes.



Figur 3.2. Stage gate model

| Kravspec tjekliste | | |
|---------------------------------|-------------------------------------|------|
| Indledning | <input checked="" type="checkbox"/> | TRUE |
| Læsevejledning og formål | <input checked="" type="checkbox"/> | TRUE |
| Systembeskrivelse | <input checked="" type="checkbox"/> | TRUE |
| Usecase diagram | <input checked="" type="checkbox"/> | TRUE |
| Fully dressed for alle usecases | <input checked="" type="checkbox"/> | TRUE |
| Kvalitets krav | <input checked="" type="checkbox"/> | TRUE |
| Ikke funktionelle krav hardware | <input checked="" type="checkbox"/> | TRUE |
| Ikke funktionelle krav Software | <input checked="" type="checkbox"/> | TRUE |
| Projektafgrænsning | <input checked="" type="checkbox"/> | TRUE |
| | | |
| | | |
| Punkter: | 9 | |
| Udførte punkter: | 9 | |
| Kan Gate gennemføres? | JA | |

Figur 3.3. Stage gate tjekliste

3.5.1 Agil udviklingsproces

I projektet er der brugt en agil arbejdsproces hvor der konstant er fokus på at målrette og prioritere arbejdet, der giver mest værdi for projektet og kunden. Derfor er der løbende prioriteret mellem opgaverne, hvorefter delopgaverne er blevet revurderet og planlagt. Dette gør at produktet og resultater evalueres og testes løbende, hvilket danner grundlag for prioriteringen af opgaverne til næste periode(sprint). Til at sikre arbejdsressourcerne tilrådighed for projektet er brugt på den mest effektive måde, er der valgt at bruge elementer fra SCRUM. SCRUM er en iterativ arbejdsmetode, hvor der er iterationer(sprints), som i dette projekt har haft en periode på en uge. SCRUM er brugt i projektet vha. *Pivotaltracker*.

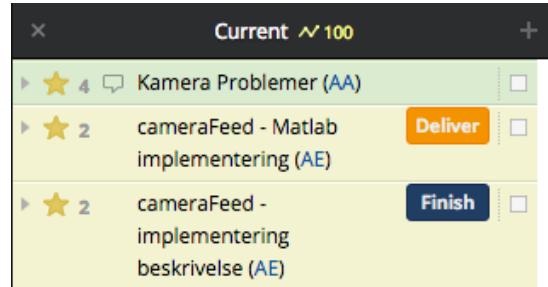
I Pivotaltracker defineres projektets arbejdsopgaver, hvorefter de tildeles point alt efter hvor stor arbejdsbyrden er. De enkelte opgaver prioriteres herefter i projektets backlog, hvor Pivotaltracker automatisk tilføjer opgaver til den igangværende sprint udfra den nuværende “velocity”. Et nyt sprint påbegyndes automatisk når en ny uge starter.

Det betyder, at der er fuldstændig styr på om projektet går for langsomt, eller om udviklingen af projektet er godt med. Dette kan holdes op imod den tidligere nævnte stage gate model.

Herudover giver Pivotaltracker mulighed for en komplet log over projektets udførte opgaver og afsluttede sprints. Her kan man se hvilke opgaver, der er udført i hvilken uge. I projektet anvendes dette som logbog for udførte arbejdsopgaver. Se logbøgerne fra *Pivotaltracker* i bilag A.8.

En opgave kan have forskellige states, som definerer dens status. Når en opgave er afsluttet kan den afleveres til review, hvor den herefter enten kan godkendes eller afvises. Dette er særligt anvendeligt i projektets udviklingsfase, hvor en feature kan testes og godkendes af et andet projektmedlem. Figur 3.4 viser et overblik over tidligere sprints, hvor figur 3.5 viser en igangværende sprint med godkendte, afsluttet og endnu ikke færdiggjorte opgaver.

| ▼ 8 19 Oct | | Pts: 28 | TS |
|------------|-----|---|--------------------------|
| ► | ★ 4 | Billeder af Langerhanske øer (AA) | <input type="checkbox"/> |
| ► | ★ 8 | Indkøb af ventil mm. (AA) | <input type="checkbox"/> |
| ► | ★ 8 | Skriv accepttest over i LaTeX (AA) | <input type="checkbox"/> |
| ► | ★ 8 | Skriv kravspec over i LaTeX (AA) | <input type="checkbox"/> |
| ▼ 9 26 Oct | | Pts: 14 | TS |
| ► | ★ 2 | BDD Software (AE) | <input type="checkbox"/> |
| ► | ★ 2 | IBD Software (AE) | <input type="checkbox"/> |
| ► | ★ 2 | Afsnit omkring udviklingsværktøjer (AA) | <input type="checkbox"/> |
| ► | ★ 4 | Ikke funktionelle krav (AE) | <input type="checkbox"/> |
| ► | ★ 4 | IBD for hardware (AE) | <input type="checkbox"/> |

Figur 3.4. To færdiggjorte sprints*Figur 3.5.* Igangværende sprint

3.6 Udviklingsfaserne

Under udviklingen af projektet er der gennemgået fire faser. Den første fase i projektet har været konceptanalyse. Konceptanalysen bestod bl.a. af litteratursøgning omkring langerhanske øer, dette blev gjort for at opnå tilstrækkelig viden omkring størrelserne, deres egenskaber mm. Der blev søgt efter allerede eksisterende sorteringsmetoder, som blev anvendt på det daværende tidspunkt. Dette var primært for at opnå erfaring inden for området på kort tid. Efter litteratursøgningen blev et overordnet koncept etableret i samarbejde med kunden (*Søren Gregersen*). Ydermere er det i konceptanalysen, at der er identificeret problemstillinger og afgrænsninger af projektet. Samtidigt med konceptanalysen, blev der fokuseret på produktionen af produktet. Det er gjort for at undgå løsninger, som ikke kan produceres eller er besværlige at fremstille. Til sidst i konceptfasen er der udarbejdet en tidsplan for projektet.

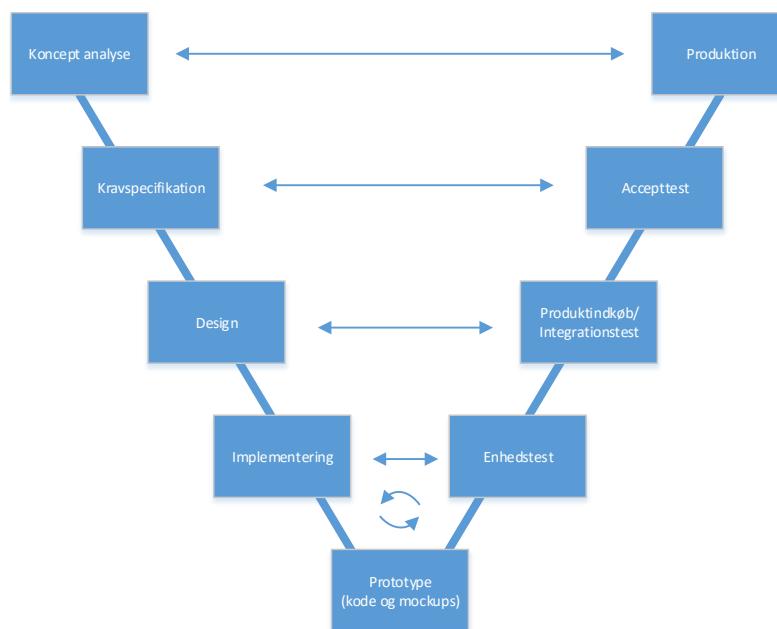
Den anden fase består af kravspecifikationen, hvilket er udarbejdet i tæt samarbejde med kunden. En kravspecifikation har sikret at kunde og projektudviklere er enige om projektets uformning. I kravspecifikationen er der brugt usecasediagram, samt fully dressed beskrivelser af hver usecase. Der laves fully dressed, for at klaregøre normal forløbet for hver usecase, samt undtagelser og udvidelser til dem. Derudover er det i kravspecifikationen, at der er specificeret ikke funktionelle krav og kvalitetskrav. Samtidigt med kravspecifikationen er der udarbejdet en accepttest. Denne test er med til at verificere at alle krav, der er bestemt i samarbejde med kunden, er opfyldte. I accepttesten er det beskrevet hvordan hver enkelt krav skal testes. Accepttesten udføres før produktet afleveres til kunden. Se afsnit 3.8 for eksempler.

Den tredje fase i projektet er designfasen, hvor der udfra kravspecifikationen er lavet overordnede diagrammer. Diagrammerne er brugt til at beskrive systemet og dele systemet op i mindre delsystemer. Det er diagrammerne, der er brugt til at udvikle systemet med. Desuden er de enkelte komponenters specifikationer beskrevet i designdokumentet. Derfor er det i denne fase at komponenterne til projektet er bestilt. Se afsnit 3.9 for eksempler. Da der efter denne fase var dannet et overblik over, hvilke opgaver der var nødvendige i

implementeringen, blev der udarbejdet en handlingsplan med en revurderet tidsplan for projektet. Handlingsplanen kan ses i bilag A.10. Herudover bestod handlingsplanen i at øge arbejdsressourcerne til 50 timer i ugen, hvilket kan ses i *Pivotal Tracker*, hvor målet for *Velocity* har været 100 point.

I den fjerde fase er der produceret en prototype. Derfor er der i denne fase; kodet, monteret og testet. Denne fase er sket efter en iterativ proces, hvor der først kodes, monteres og derefter testes. Dette er gjort ved så små delelementer som muligt, for at sikre at hver delelement virker inden det sættes sammen. Se afsnit 3.10 for eksempler af denne fase.

De fire udviklingsfaser i projektet kan illustreres som på figur 3.6. Modellen har sine fordele og ulemper. Fordelene er, at der sikres dokumentation af projektet fra starten, samt at der hele tiden tænkes på slutresultatet og slutbrugeren. Ulemperne er, at der er meget dokumentation, der bliver ændret løbende. Dermed forekommer dette som spildtid, men det sikrer samtidigt, at projektet bliver veldokumenteret og gennemtænkt fra starten.



Figur 3.6. V-model

I modsætning til v-modellen findes vandfaldsmodellen, hvor hver enkelt fase bliver lavet færdig før næste fase påbegyndes. Dette medfører ofte nedprioritering af test og andre sene deadlines i projektet, grundet at tidligere tidsplaner er overskredet. Det er derfor denne type model ikke er anvendt i projektet.

3.7 Første udviklingsfase: Konceptanalyse

For at eftervise at der er brugt de beskrevne metoder ovenfor, er der valgt at tage eksempler med i rapporten. I afsnittet er der eksempler fra hver af de fire udviklingsfaser. Den første fase er konceptanalysen, som primært har bestået af udarbejdning af forprojektet. I forprojektet er der kort defineret krav, bdd samt en projektplan for projektet. Projektplanen i koncept analysen kan ses på figur 3.7

| Projekt plan | Juni | Julii | August | September | Oktober | November | December | Januar |
|--|------|-------|--------|-----------|---------|----------|----------|--------|
| Forprojekt | | | | | | | | |
| Herunder udkast til kravspecifikation | | | | | | | | |
| Research + indkøb af udstyr herunder USB mikroskop, pumper, microcontroller. | | | | | | | | |
| Opsætning af projekt administrationsprogram | | | | | | | | |
| Samarbejde med Review gruppe | | | | | | | | |
| Opstart på projekt (Herunder accepttest, design dokumenter, use case) | | | | | | | | |
| Udvikling af prototype (både hardware og software) | | | | | | | | |
| Risikovurdering, Cost-benefit analyse og lignende dokumenter | | | | | | | | |
| Test af prototype (Herunder accepttest) | | | | | | | | |
| Færdiggørelse af rapport | | | | | | | | |
| Aflevering | | | | | | | | |
| Projekt forsvar | | | | | | | | |

Figur 3.7. Projektplan i forprojektet

Det er blandt andet projektplanen på figur 3.7, som har henledt til stage gate modellen. Til sidst har konceptanalysen været med til at give et overblik over hvilke komponenter, der kunne være anvendelige og burde undersøges til prototypen.

3.8 Anden udviklingsfase: Kravspecifikation og accepttest

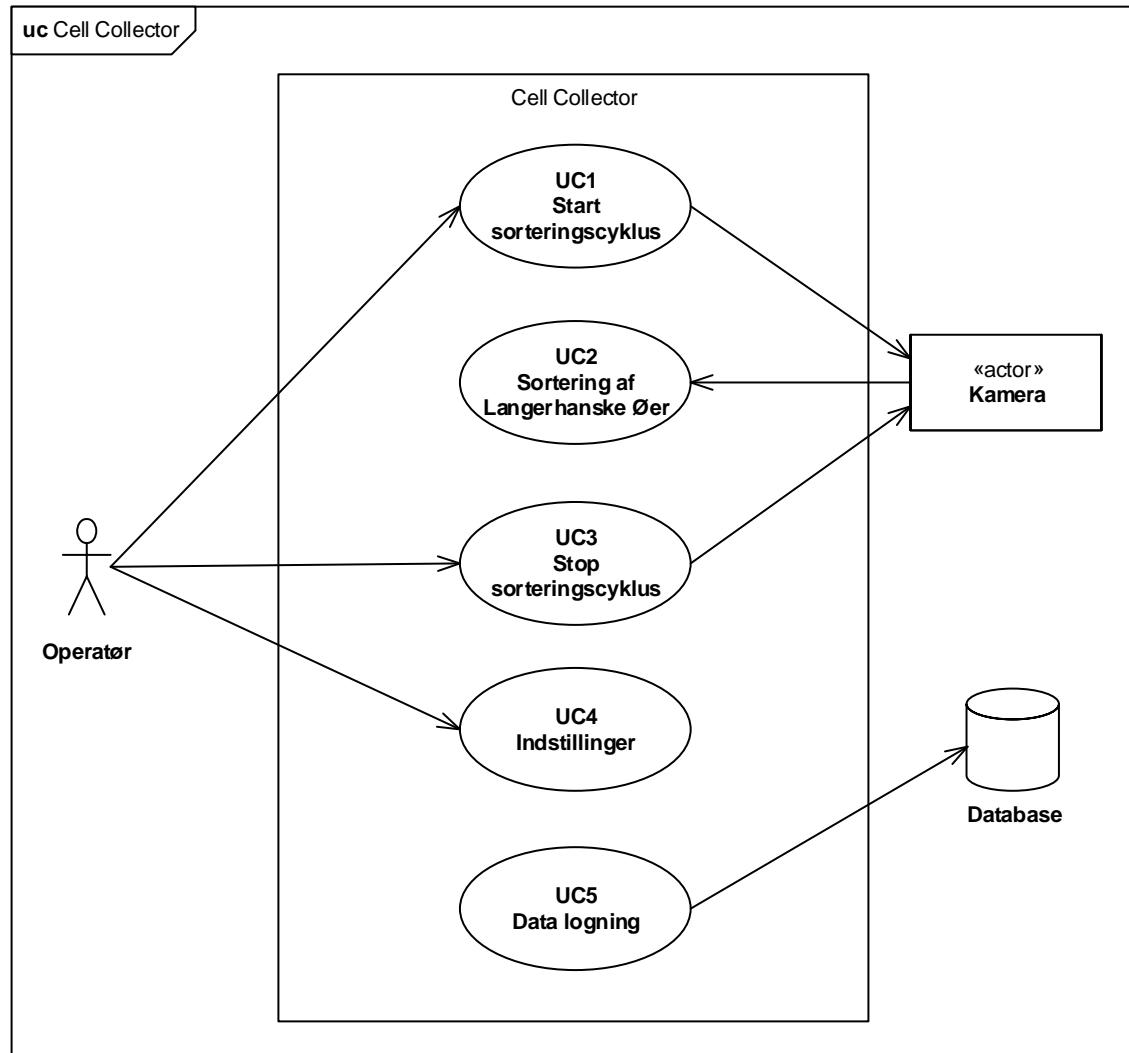
For at vise et eksempel for den anden udviklingsfase i projektet, er der valgt at tage *Aktør beskrivelse*, *use case diagram*, samt et *fully dressed use case* og et udpluk fra accepttesten.

3.8.1 Aktør beskrivelse

Systemets primære aktør er operatøren, som står for påfyldning af celler, samt start og stop af sorteringsprocessen. Operatøren har mulighed for at interagere med systemet via en grafisk brugergrænseflade. Systemets sekundære aktør er kameraet og PC'ens filsystem. Kameraet er systemets interface til detektion af de langerhanske øer. Filsystemet er hvor der gemmes en log over sorteringsprocessen.

3.8.2 Use case Diagram

I Use case diagrammet (figur: 3.8) er der vist, hvilke use cases systemet *The Cell Collector* består af. Yderligere er det vist, hvilke aktører der initierer de enkelte use cases. På venstre side er systemets primære aktør *operatøren* vist, mens systemets sekundære aktører *kamera* og *database* er placeret i højre side.



Figur 3.8. Use Case diagram for The Cell Collector

Efter use case diagrammet var færdigt, blev der udarbejdet fully dressed use cases, som er vist på nedenstående tabel. Tabellen beskriver normalforløbet og undtagelser for *Start af sorteringscyklus*.

3.8.3 Fully dressed use case for use case 1

| | |
|--------------------------------|--|
| Mål | Start sorteringscyklus |
| Initiering | Use casen initieres af operatøren |
| Aktør | Primær: Operatør Sekundær: Kamera |
| Startbetingelser | The Cell Collector programmet er startet på computeren |
| Slutbetingelser ved succes | Systemet starter med sorteringen af Langerhanske øer |
| Slutbetingelser ved undtagelse | N/A |
| Normalforløb | <ol style="list-style-type: none"> 1. Operatør fylder celleopløsningsbeholderen 2. Celleopløsningsbeholderen er fyldt 3. Operatør starter sorteringscyklus ved at klikke på [Start] [Undtagelse 1: Wastebeholder er fyldt] 4. Systemet initialiserer Arduinoen [Undtagelse 2: Ingen forbindelse til Arduino] 5. Systemet kontrollerer celleopløsningsbeholderen ved at konvertere spændingen (V) til ml, og vise beholderens indhold (ml) på GUI 6. Systemet initialiserer kameraet [Undtagelse 3: Kameraet initialiserer ikke] 7. Systemet tænder for kamera lyset 8. Systemet tænder for pumpen |
| Undtagelser | <p>[Undtagelse 1: Wastebeholder er fyldt]</p> <ol style="list-style-type: none"> 1. Systembesked: Tøm venligst Wastebeholder før start 2. Operatøren trykker "OK" 3. Systemet fortsætter opstartprocessen <p>[Undtagelse 2: Ingen forbindelse til Arduino]</p> <ol style="list-style-type: none"> 1. 1. Systembesked: Ingen forbindelse til Arduino, kontrollér forbindelser. <p>[Undtagelse 3: Kameraet initialiseres ikke]</p> <ol style="list-style-type: none"> 1. System fejlmeddelse: Kameraet er ikke initialiseret: 2. Genstart initialisering af Kameraet |

Efter at normalforløbet og undtagelserne er defineret, blev accepttesten lavet for den givne use case. Til hvert punkt i normalforløbet er der forberedt en test, som indeholder et *krav nr.*, *handling* som beskriver hvad der skal gøres for at starte testen. Derudover er der *forventet resultat*, som skal ske for at testen kan godkendes. *Testmetode* beskriver hvordan testen skal udføres og hvordan den godkendes. Denne test er et udsnit af accepttesten, som er udarbejdet i samarbejde med kunden.

3.8.4 Accepttest for krav 1.3

| | |
|---------------------------|--|
| Krav nr. | 1.3 |
| Handling | Operatør starter sorteringscyklus ved at klikke på [Start] |
| Forventet resultat | Opstartsprocessen igangsættes. |
| Testmetode | Knappen [Start] trykkes, observeres ved tekstboks på GUI, med teksten <i>systemet starter op</i> . |
| Resultat | |
| Angiv godkendelse | |
| Initialer | |
| Dato | |

Resten af accepttesten for use casene kan ses i projektdokumentationen afsnit 2.2.

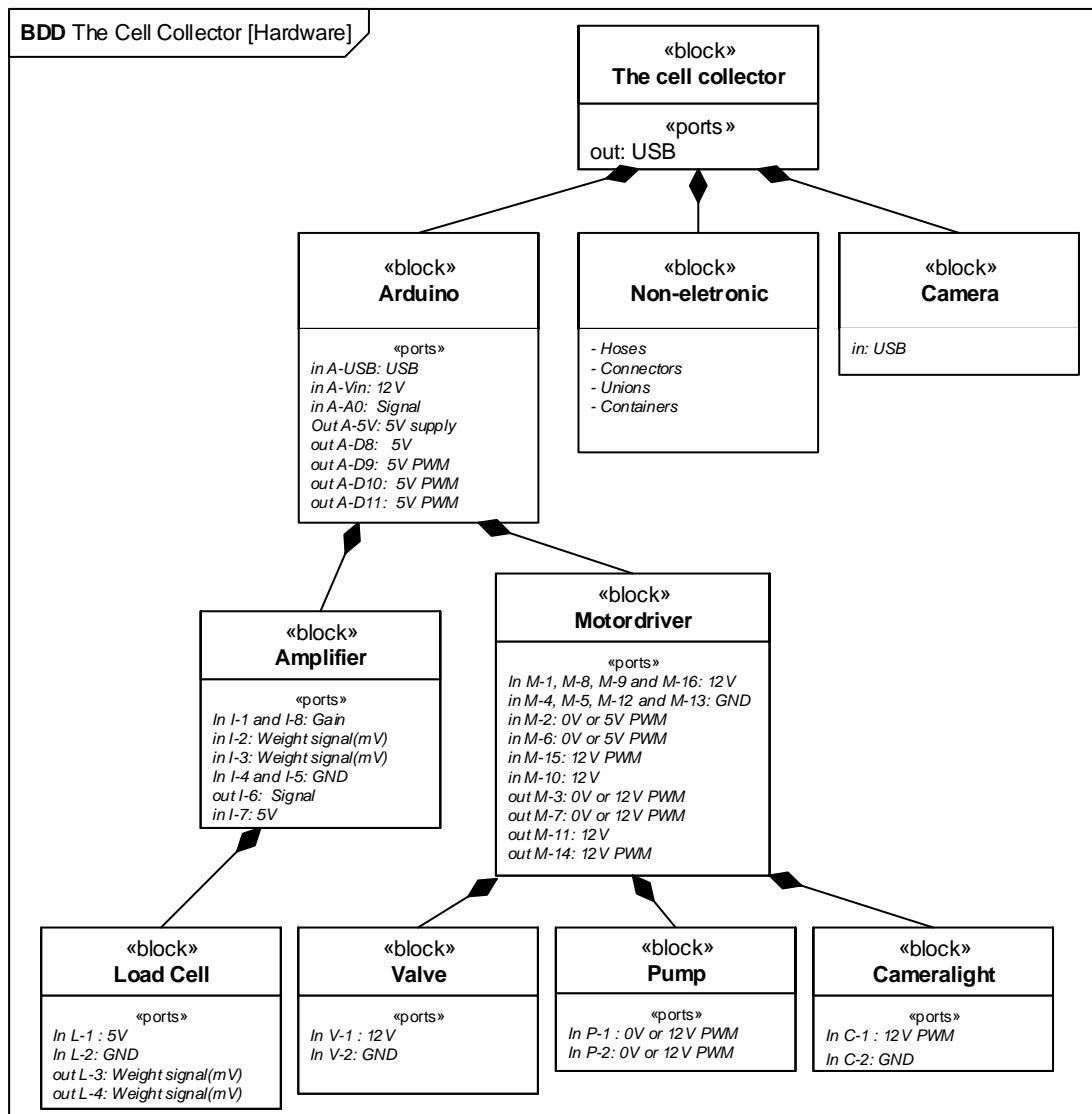
3.9 Tredje udviklingsfase: Design

I dette afsnit er der givet et eksempel på, hvordan projektet er gået fra krav til måder at løse projektet på. Der er blandt andet brugt BDD, IBD, flowchart samt sekvensdiagrammer til at beskrive funktioner og sammenhænge i projektet.

3.9.1 BDD og IBD for hardware

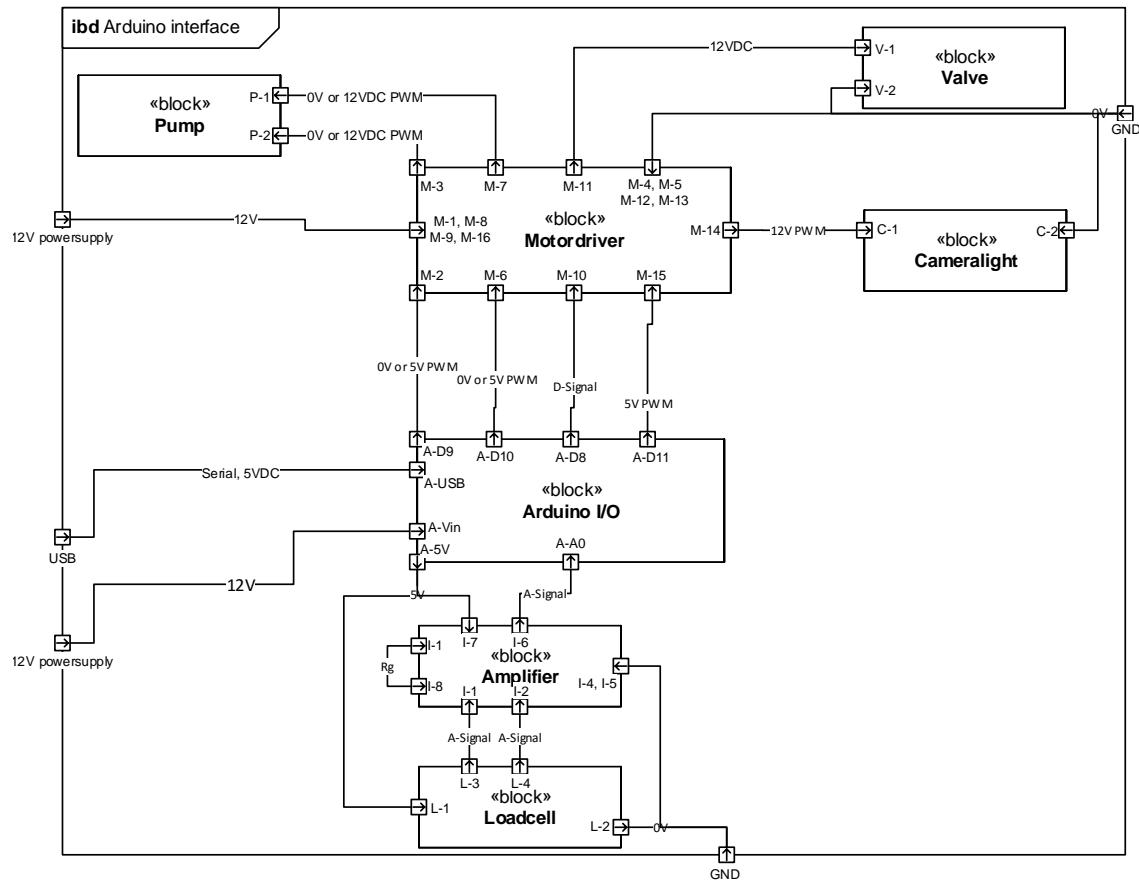
Block definition diagram og interal block diagram over hardwaren er i projektet brugt til at sikre, hver hardware element kan kommunikere med hinanden. Yderligere er det med til at definere specifikationerne, for eksempelvis at indskærpe antallet af strømforsyninger.

Nedenstående BDD 3.9 giver et overordnet overblik i, hvad *The cell collector* indeholder af hardware elementer. Hierarkiet starter øverst med *The cell collector*, som indeholder tre mindre hardware dele. Styreenheden, der er defineret som Arduino, indeholder bl.a. underelementet motordriver. Motordriveren yderligere tre dele, det vil sige at pumpen, ventilen og kameralyset bliver styret her igennem. Udenfor styreenheden er der ikke elektriske dele, som beholdere og føringsveje til opløsningen med langerhanske øer. Den tredje underblok til *The cell collector* er kameraet.



Figur 3.9. BDD - Cell Collector [Hardware]

Nedenstående IBD figur 3.10 beskriver mere præcist, hvordan de forskellige komponenter interagerer med hinanden. Diagrammet er brugt til, at der tidligt i udviklingsforløbet bliver defineret hvilke spændinger og signaltyper systemet skal indeholde. Systemet skal indeholde bestemte typer for, at kunne kommunikere med de interne dele.



Figur 3.10. IBD - Cell Collector [Hardware]

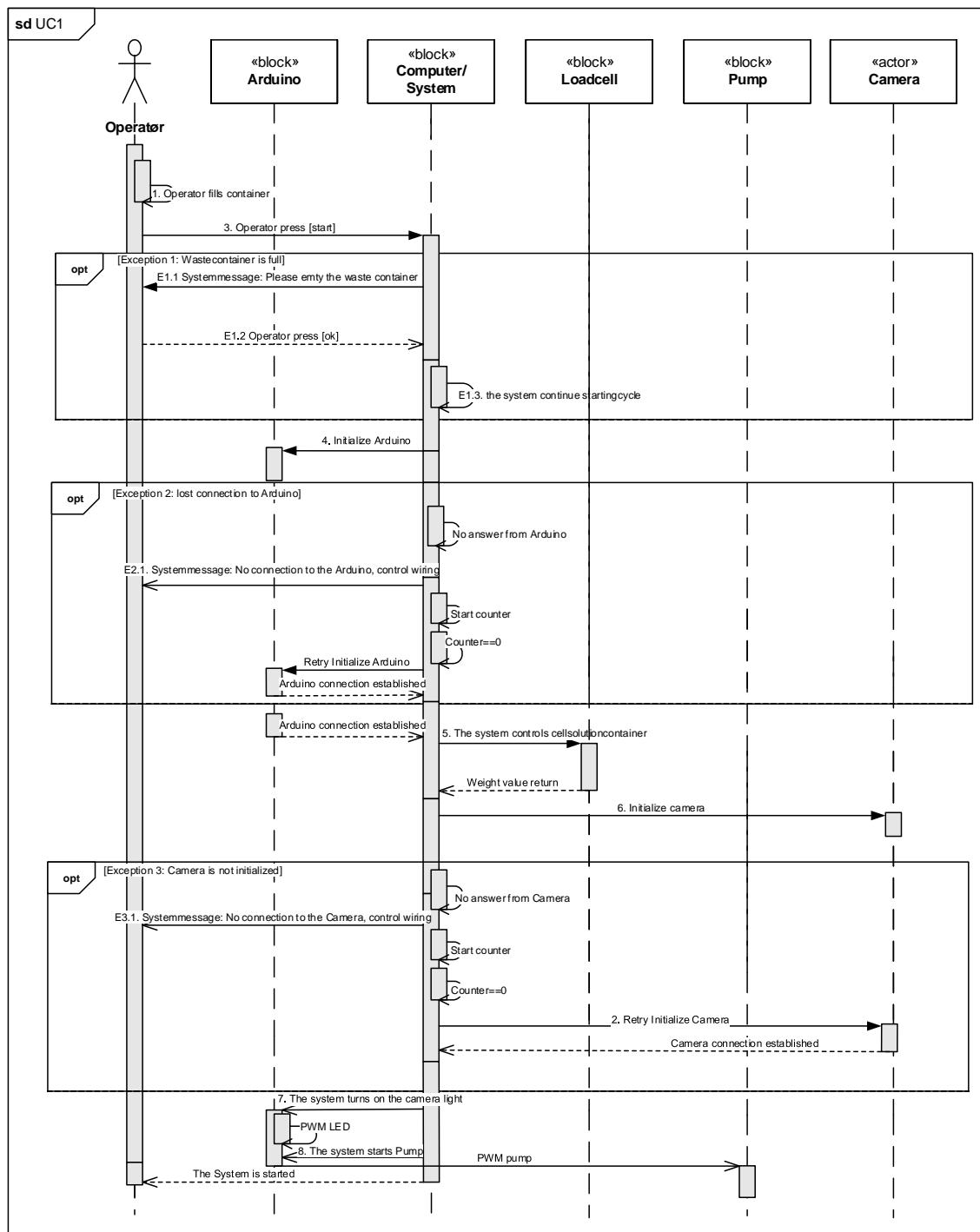
Til ovenstående diagrammer er der udarbejdet tabeller som beskriver signalerne og de enkelte blokke (Se projektdokumentationen afsnit 3.3.4).¹

I designfasen er der udarbejdet specifikationer for de enkelte hardware dele, samt overvejelser omkring komponenten.

¹Fixme Note: Tjek ref

3.9.2 Sekvensdiagrammer

I designdokumentet er der yderligere lavet sekvensdiagrammer, hvilket giver overblik over de sekventielle dele af systemet til hver use case (se figur 3.11 for et eksempel). Sekvensdiagrammerne har været brugt til at klargøre, hvordan softwaren skal implementeres og i hvilken rækkefølge funktionerne skal eksekveres. Derudover har det været med til at danne et overblik over delene, der skal implementeres i projektet, samt dele arbejdsopgaverne op i mindre opgaver.



Figur 3.11. Sekvensdiagram for usecase 1

3.9.3 Vægtcelle

Det følgende afsnit er et eksempel på, hvordan de enkelte hardware komponenter er blevet specificeret i designdokumentet, herunder de overvejelser, der er gjort. Afsnittet viser, hvordan vægtcellen er specificeret.

Vægtcellen skal bruges til at kontrollere om, der er væske i celleopløsningsbeholderen. Systemet stopper sin sorteringscyklus, når der ikke længere er væske i celleopløsningsbeholderen.

Specifikationer for Vægtcelle[Ebay [a]]:

| Specifikation | Værdi |
|----------------------------------|------------------|
| Max belastning: | 1 kg |
| Anbefalet arbejdsspænding | 3-12V |
| Output | 1.0mV/V±0.15mV/V |

Den indkøbte vægtcelle kan veje op til 1 kg, hvilket dækker vægten for celleopløsningsbeholderen på 250ml + beholderens vægt. I designfasen er der udarbejdet et teoriafsnit for at dokumentere den opnåede viden gruppen har fået. Designafsnittet indeholder desuden beregninger og kredsløbsdiagrammer. Det er illustreret i den overstående tabel, at vægtcellens output er i millivolt, hvilket har medført, at signalet skulle forstærkes. Dette er gjort vha. en operationsforstærker. For at vise et eksempel, er der trukket nedenstående afsnit ud fra designdokumentet (se projektdokumentation afsnit 4.2.1² for hele afsnittet). I databladet A.1.9 til INA114 vises det at operationsforstærkeren har en CMRR på 115dB, ved et gain på 1000 og en indgangsmodstand på 10GΩ. Forstærkningen kan regnes ud fra formlen i databladet 3.1

$$G = \left(1 + \frac{50K\Omega}{R_G}\right) \quad (3.1)$$

I dette projekt skal der bruges et gain på $\frac{4,9V}{5mV} = 980$. De 4,9V er for ikke at komme i mætning på Arduinoens ADC og 5mV er den maksimale spænding vægtcellen kan give, ved 5V forsyning.

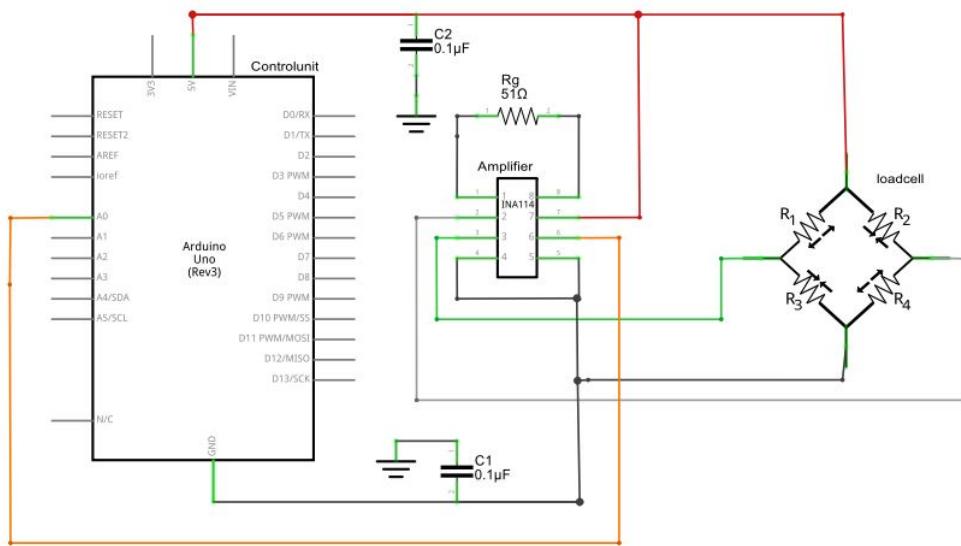
$$R_G = \frac{50k\Omega}{980 - 1} = 51\Omega \quad (3.2)$$

Et gain på 980 giver en $NY_{Maksimalespnding} = 980 * 5mV = 4,9V \pm 0,147V$, dvs at der nu er en opløsning på

$$\frac{1000g}{trin} = \frac{1000g}{1024} = 0,977g/trin \Rightarrow 0,977 * \frac{1024}{5V} = 200g/V \pm 30g \quad (3.3)$$

Kredsløbet for INA114 og vægtcellen til Arduinoen er vist på figur 3.12.

²Fixme Note: Tjek ref



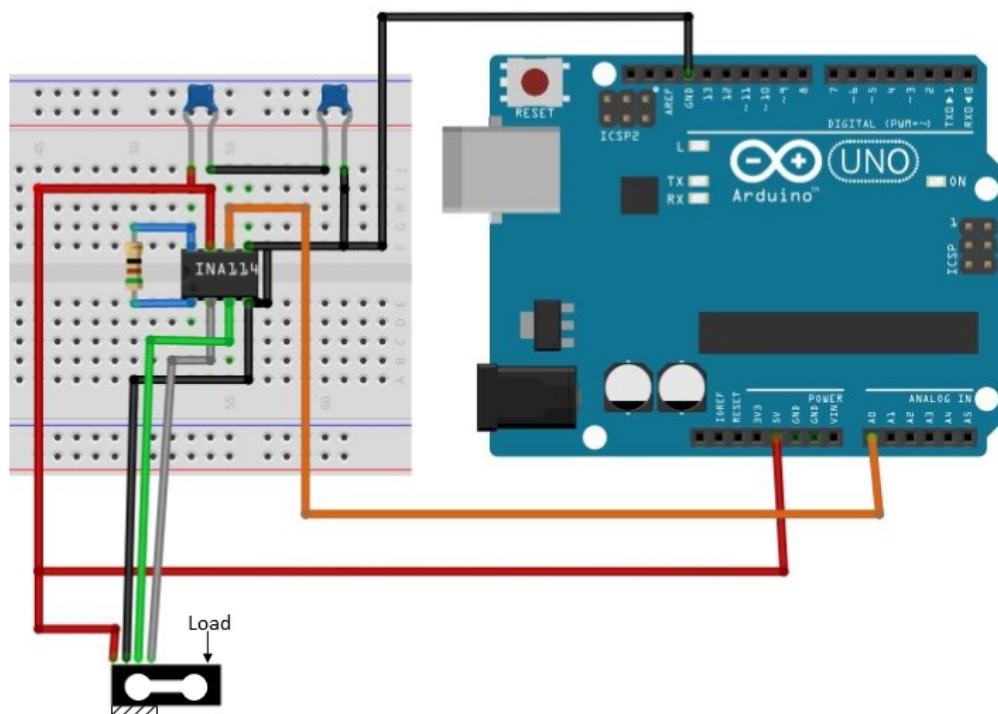
Figur 3.12. Kredsløb for Arduino, INA114 og vægtcelle

3.10 Fjerde udviklingsfase: Implementering og enhedstest

I dette afsnit er der vist et eksempel på hvordan vægtcellen er implementeret, både hardware- og softwaremæssigt. Til sidst i implementeringsfasen er der lavet en integrationstest. Integrationstesten for hardwaren er bestået i udarbejdelse af et shield til *Arduinoen*. Da de enkelte faser er foregået ved en iterativ proces har det været mulighed at justere designdokumentet, hvis en komponent eller funktion blev implementeret anderledes end det var specificeret. På samme måde er designdokumentet ændret, hvis en enhedstest fejlede. Et eksempel på dette er kameraet, hvor det i enhedstesten viste sig at kvaliteten ikke var tilstrækkelig. Enhedstesten for kameraet er vist i afsnit 4.3.1 i projektdokumentation.³

3.10.1 Hardware

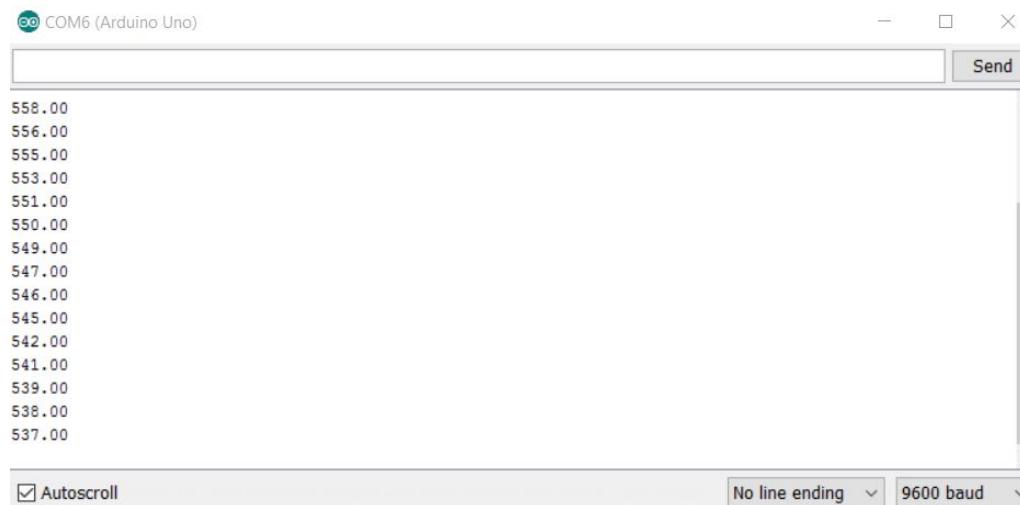
Efter at diagrammet og kredsløbet er fastlagt i designdokumentet kunne vægtcellens kredsløb nu testes på et *fumlebræt*. Se figur 3.13 for testopstilling.



Figur 3.13. Test opstilling for vægtcelle

De primære dele af test koden har været `sensorValue = analogRead(A0);` og `Serial.println(sensorValue);`, hvor ved at inputtet på `A0` er læst i *serial monitor*, som vist på figur 3.14 ved langsom tømning af beholderen.

³Fixme Note: tjek ref



Figur 3.14. Værdier fra A0 i serial monitor

Se bilag A.3.1 for at se hele koden til testen. Til enhedstesten er der brugt et voltmeter til at måle udgangsspændingen på INA114 for at se, om Arduinoens ADC læste rigtigt. Til at sammenligne med voltmeteret, blev formlen 3.4 brugt til at konvertere ADC'ens bits værdi om til spænding.

$$\text{analogRead}(A_0) * \frac{5}{1024} = \text{spænding i volt} \quad (3.4)$$

Testopstilling af vægtcellen ser ud som på figur 3.15 med celleopløsningsbeholderen. I softwaren kræves det en kalibrering for at vægtcellen er præcis, dette er implementeret i projektdokumentation afsnit 4.3.5.3⁴.



Figur 3.15. Illustration af opstilling med vægtcelle og celleopløsningsbeholder

⁴FiXme Note: Tjek ref

3.10.2 Kalibrering af vægtcellen

Funktionen til vægtcellen er implementeret efter beskrivelsen i designdokumentet i projektdokumentation afsnit 4.3.5.3⁵. Dens funktion er at konvertere det analoge input (V) til indholdet (ml) i celleopløsningsbeholderen. Dette er implementeret ved en lineær model:

$$mL = a * \text{input} + b, \text{ hvor } a \text{ er hældningen og } b \text{ er skæringen med y aksen} \quad (3.5)$$

Det analoge input ganges med en faktor a plus et offset b for at konvertere spænding til antal ml. Nedenstående tabel viser indgangsspændingen for forskellige mængder i beholderen. Udfra disse data er der lavet en lineær regression for at finde hældningen a og skæringen b .

| ml i beholder | Analog input |
|---------------|--------------|
| 0 ml | 1,9487 V |
| 25 ml | 2,0440 V |
| 50 ml | 2,1320 V |
| 75 ml | 2,2297 V |
| 100 ml | 2,3109 V |
| 125 ml | 2,4071 V |
| 150 ml | 2,4961 V |
| 175 ml | 2,5821 V |
| 200 ml | 2,6760 V |
| 225 ml | 2,7654 V |
| 250 ml | 2,8587 V |

Tabel 3.5: Kalibreringsdata for vægtcellen

I Matlab er funktionen *fitlm* anvendt til at finde det bedste lineære fit. Regressionen er baseret på Least Square metoden [Matlab, 2015]. Udfra beregningerne i Matlab er hældningen a og skæringen b fundet til hhv:

$$a = 276.14 \quad (3.6)$$

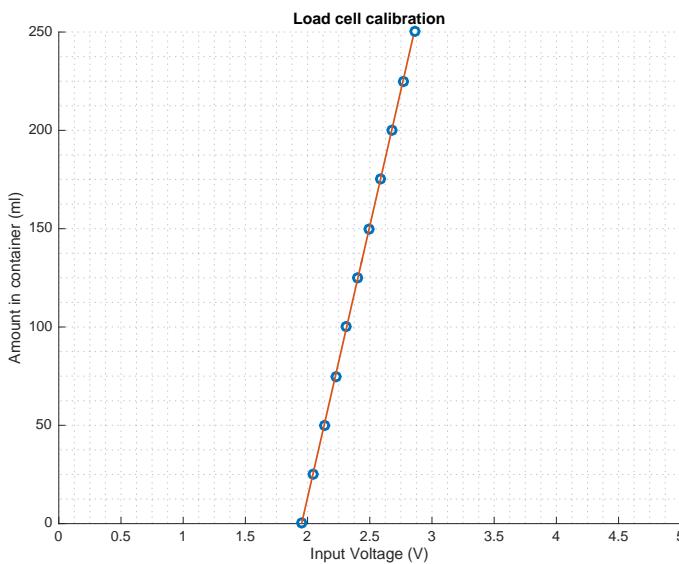
$$b = -539.02 \quad (3.7)$$

Den endelige funktion er givet ved:

$$ml = 276.140 * \text{input} - 539.02 \quad (3.8)$$

Figur 3.16 viser den lineære funktion, samt de enkelte data punkter fra tabellen.

⁵Fixme Note: Tjek ref



Figur 3.16. Kalibrering af load cell

For at reducere støj og mindske følsomheden overfor hurtigere ændringer i indgangsspændingen er der implementeret en midling af de seneste 10 målinger. Dette er med til, at gøre konverteringen mere robust overfor støj.

3.10.3 Integrationstest

Til sidst i denne fase er der udført en integrationstest, hvor hardwaren er samlet, efter alle enkelte enhedstest er udført. Efter samlingen af hardwaren er det verificeret at kredsløbet virker, hvorefter der er designet et PCB til samlingen af hardwaren. Før integrationstest med hardwaren blev softwarens interface med Arduinoen testet vha. LED'er. Lysdioderne er brugt til at koble på de udgange hvor ventil, pumpe og kameralyd skal monteres. Dette gav mulighed for visuelt at se om udgangene på Arduinoen blev sat høj(5V) og lav(0V) som forventet. Herudover blev et potentiometer brugt, som erstatning til vægtcellen. Dermed kunne indgangsspændingen manuelt justeres og det kunne dermed visuelt observeres at beholderens indhold blev ændret. Til sidst i integrationstesten blev de egentlige hardware komponenter koblet på Arduinoens porte og det blev testet om softwaren fortsat kørte efter hensigten.

Resultater 4

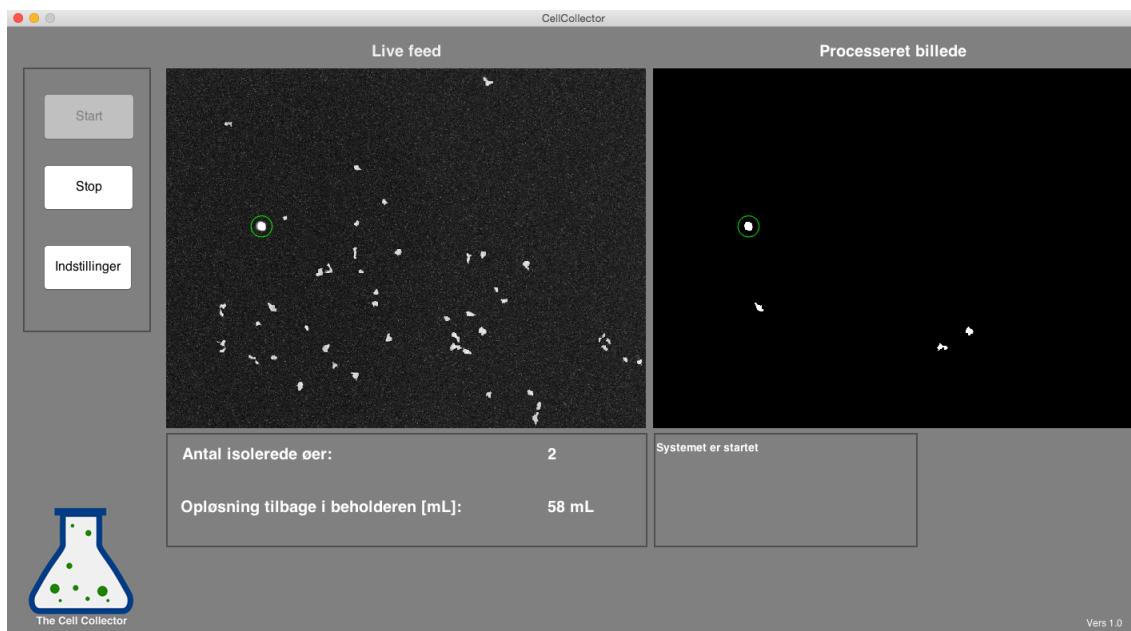
Der er i dette projekt udarbejdet en prototype, som kan undersøge muligheden for en automatiseret metode til at isolere langerhanske øer. Der er udviklet et færdig system, som har potentiale til at isolere langerhanske øer. De mekaniske og elektriske dele er på plads, til videre undersøgelser af konceptet. Ydermere er der i projektet opnået en billedbehandling der understøtter konceptet, i form af processering af de simulerede billeder. Der er brugt simulerede billeder, fordi det indkøbte kamera ikke har været af tilstrækkelig kvalitet.

4.1 Det udviklede system

Det udviklede system til isolering af langerhanske øer består af et softwareprogram udviklet i Matlab og en række hardware komponenter som styres vha. en Arduino.

4.1.1 Brugergrænseflade

Softwaredelen består af en brugergrænseflade, hvor operatøren har mulighed for at interagere med systemet. Figur 4.1 viser, hvordan brugergrænsefladen er opbygget. Operatøren har via en knap mulighed for at starte og stoppe sorteringscyklussen. En række tekstfelter med feedback giver operatøren information om den igangværende sorteringscyklus, herunder antal sorterede øer, indholdet tilbage i opløsningsbeholderen og status for systemet. Et kamera feed er herudover vist, som giver operatøren mulighed for at følge indholdet, der løber igennem slangen. Under kamera feedet vises et behandlet billede af feedet fra kameraet. Hvis en ø er detekteret markeres den på begge feeds med en grøn cirkel.



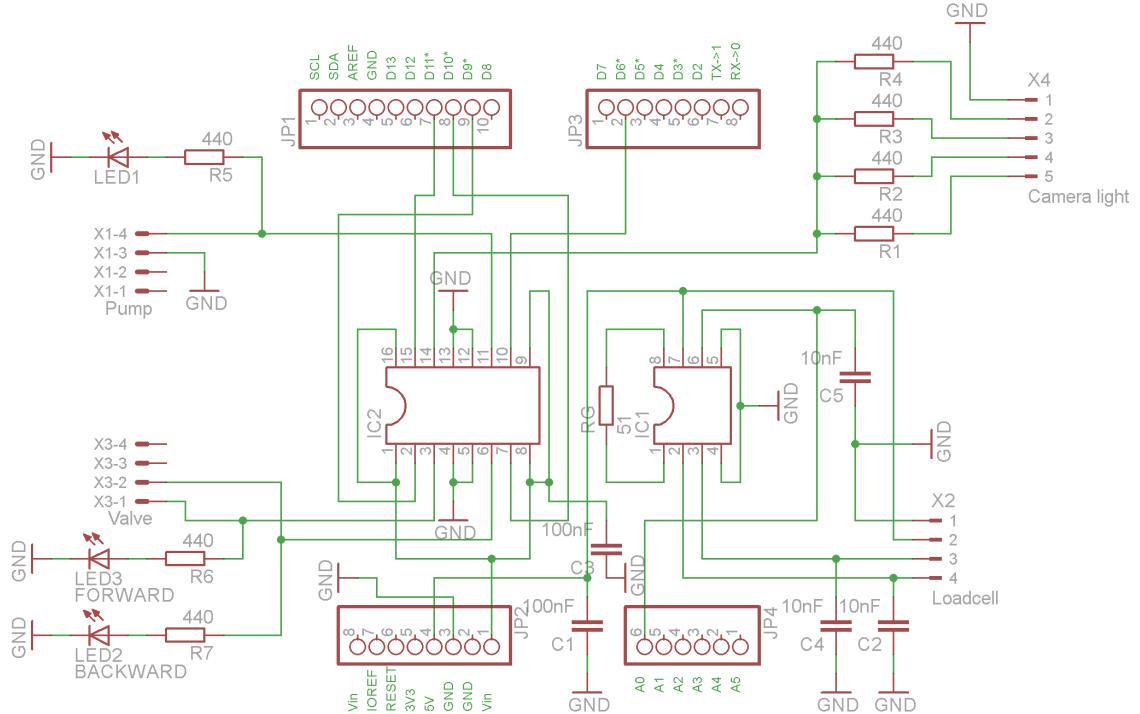
Figur 4.1. Brugergrænsefladen på prototypen

4.1.2 Hardwaren

Den overordnede funktion af hardwaren er at pumpe oplosningsvæsken, forbi kameraet hen forbi ventilen til *wastebeholderen* eller beholderen til de isoleret langerhanske øer. For at fremvise projektets færdige resultater, er kredsløbet og layoutet til det udarbejdede printkort til produktet indsat herunder i figur 4.2. Til et overordnet overblik over hardwaren i projektet henvises der til figur 3.9. Printkortet er lavet som et shield til *Arduino* udviklingskortet. Shieldet kan sættes direkte oven på *Arduinoen*. På printkortet er der stik til at forsyne og videregive signaler til hardwarekomponenterne. Komponenterne er pumpen, der skaber flowet i slangene, ventilen som isolerer øerne, kameralyset der lyser øerne op for, at kameraet bedre kan detektere dem. Derudover er vægtcellen også forsynet igennem printet. Printkortets layout er udarbejdet i *Eagle*, hvor lagene er eksporteret til *Gerber* filer. Filerne er sendt til printfremstilling ved Ingeniørhøjskolens værksted.

4.1.2.1 Det samlede kredsløb

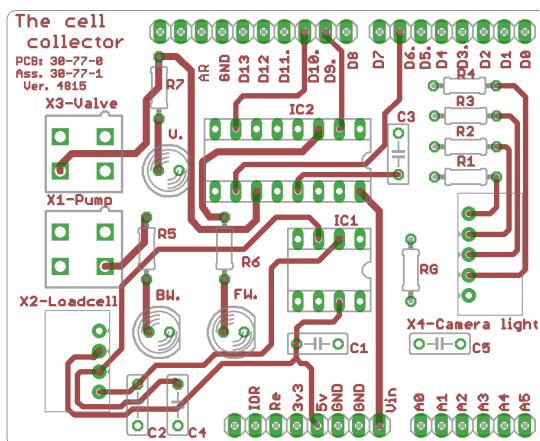
Kredsløbet for det samlede print er indsat nedenfor i figur 4.2 og beskriver hvordan delene er forbundet. Der er 4 pinheaders der forbides til Arduinoen, derudover er der fire stik til pumpen, ventilen, vægtcellen og kameralyset



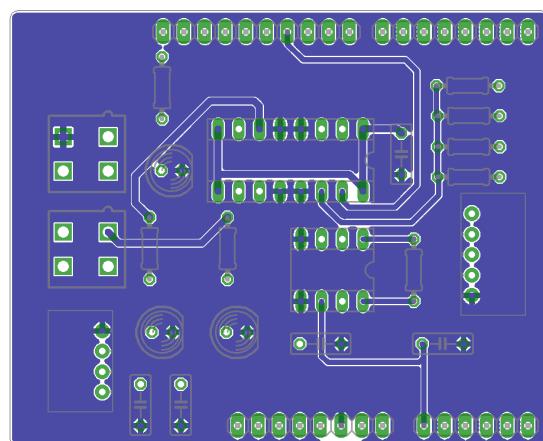
Figur 4.2. Diagram for printkortet

4.1.2.2 Det færdig print layout

Til projektet er der udarbejdet et færdigt printkort, til hardwaren med forbindelserne imellem komponenterne. På printet er der to integrerede kredsløb, den ene(IC2) er en L293D som bla. bliver brugt som motordriver i projektet. Motordriveren forsynes eksternt og bruger denne strømforsyning til at forsyne motor, kameralys og ventil. Kort fortalt tager motordriveren signalet fra Arduinoen og efterligner det, men i stedet leverer effekten igennem den eksterne forsyning. Det andet(IC1) integrerede kredsløb består af en operationsforstærker, som primært har til opgave at forstærke outputtet fra vægtcellen til Arduinoens analog-digital konverter(ADC). Det gøres fordi outputtet fra vægtcellen er i mV og Arduinoens ADC har en arbejdsspænding på 0V til 5V. Nedenfor ses der to figurer, på figur 4.3 er det øverste lag af printet illustreret. På det øverste lag kan der desuden ses lysindikatorer bla. for hvilken vej motoren kører. Herudover er der udarbejdet versionsnummer mm. der kan være med til at hjælpe ved service og generelt sporbarhed. På figur 4.4 er bundplanet vist. På bundplanet er der et groundplane der forebygger EMC, men også virker som køleplade for motordriveren. Der er i bilag A.1.10 anbefalet en køleplade til motordriveren, som implementeret i bundplanet. Yderligere dokumentation af styklister og beregninger til printet kan læses i projektdokumentation afsnit 4.2.6. ¹



Figur 4.3. Toplayout

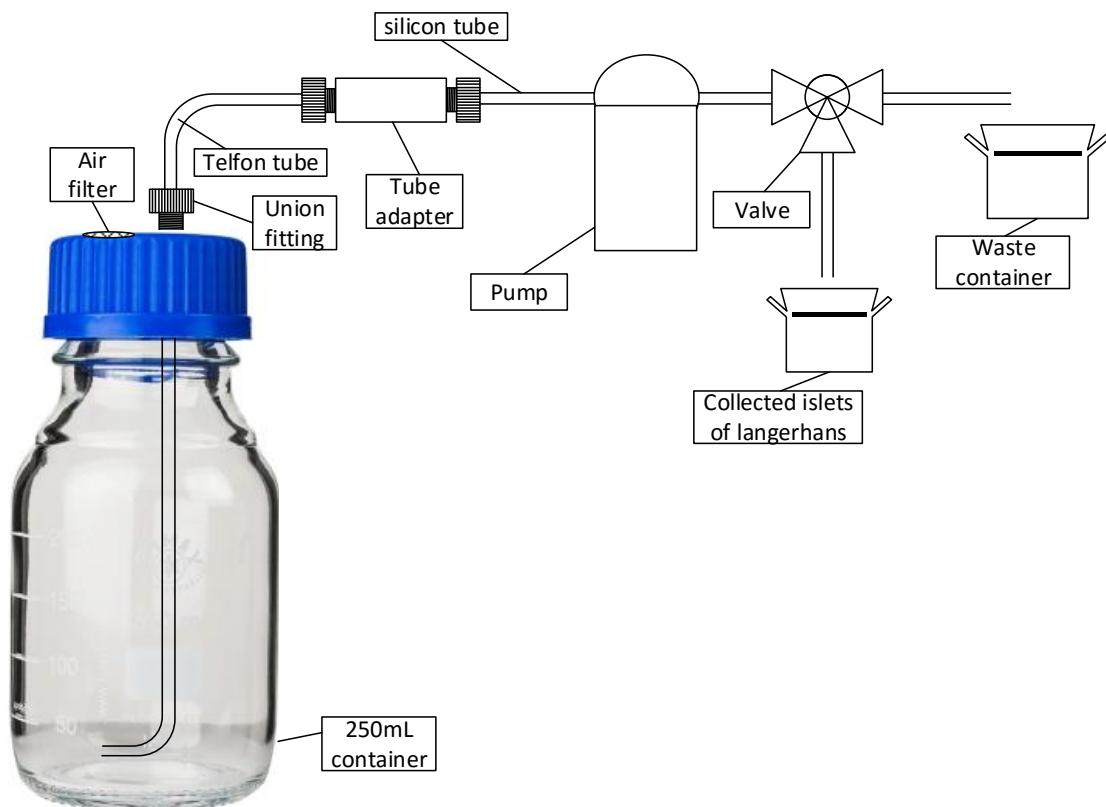


Figur 4.4. PCB bundlayout med groundplane

¹FiXme Note: tjek ref

4.1.3 Ikke-elektroniske dele

De ikke-elektroniske dele består af beholdere, slanger og adapttere. I projektet er der fokuseret på celleopløsningsbeholderen, hvor der er indkøbt en 250ml beholder med et specielt låg. Igennem låget går en teflonslange, for at tætte denne tilslutning er der en forskruning som både aflaster og tætner. I låget er der også et luftfilter, som forhindre støv og andre små partikler i at komme ned i opløsningen. Filteret skal være der for at undgå dannelse af undertryk i beholder, når pumpen suger opløsningsvæsken igennem systemet. Da teflonslangen ikke er eftergivende, er der brugt silikoneslanger til stykket igennem pumpen og resten af systemet. Teflonslangen er for hård til, at pumpen kan sammenpresse den og derved skabe flow i opløsningen. For at kunne konvertere fra teflonslangen til silikoneslangen er der brugt en adapter. Figur 4.5 indeholder en illustration af ikke-elektroniske dele. Se styklisten mm. i projektdokumentation afsnit 4.2.5².

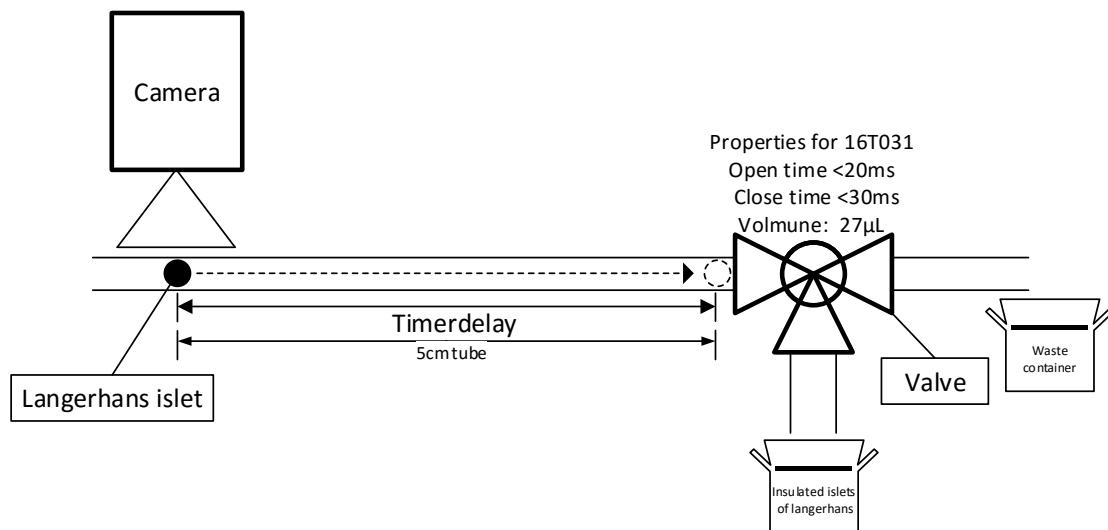


Figur 4.5. Illustration af ikke elektroniske dele

²Fixme Note: Tjek ref

4.2 Ventil tidsinterval

Tidsintervallet mellem kameraet og ventilen er en vigtig faktor i systemet. Intervallet beskriver tiden fra, at kameraet har detekteret en ø til den er kommet hen til ventilen, hvor den skal isoleres. En estimation er beregnet igennem nedenstående formler. Kravet fra kunden hedder 30 øer i minuttet, derfor skal systemet som minimum have en flow hastighed på 11,25 mL/min. Se formel 4.1 for udregningen. På figur 4.6 kan der ses en illustration af problemstillingen.



Figur 4.6. Illustration af ventil timerdelayet

$$\frac{\text{Opløsningsstørrelse}}{\text{Antal øer i opløsning}} = \frac{150 \text{ mL}}{400 \text{ øer}} * 30 \text{ øer/minut} = 11,25 \text{ mL/minut} \quad (4.1)$$

(jf. Søren Gregersen)

For at sorteringen af langerhanske øer er succesfuld, kræver det en beregning af den tid det tager for en langerhansk ø, at komme fra kameraet til ventilen. Det kan estimeres ud fra slange volumenet, imellem kameraet og ventilen 4.2.

$$V = \pi * r^2 * h = \pi * \left(\frac{0,51 \text{ mm}}{2}\right)^2 * 50 \text{ mm} = 0,010\,214 \text{ mL} \quad (4.2)$$

Derfor kan det med et flow på 11,25 mL/minut, som er beregnet ud fra formlen 4.1, bestemmes hvor lang der går fra kamera til ventil ved formlen 4.3.

$$\frac{0,010\,214 \text{ mL}}{\frac{11,25 \text{ mL/minut}}{60 \text{ s}}} \rightarrow \frac{0,010\,214 \text{ mL}}{0,1875 \text{ mL/s}} = 0,054\,475 \text{ s} \quad (4.3)$$

Ifølge bilag A.1.12 har ventilen et volumen på 27 μL. For at beregne tiden det tager, at tömme ventilen bruges formel 4.4.

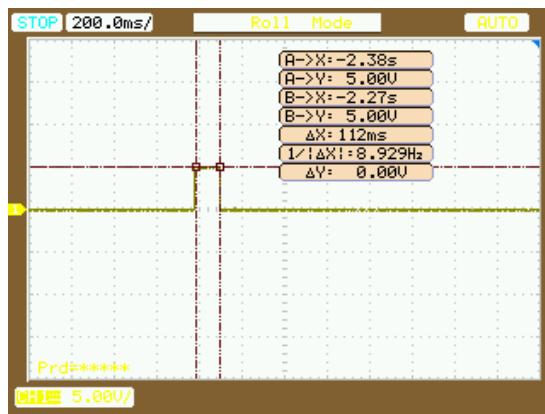
$$\text{Tid for udtømning af ventilen} = \frac{\text{ventil volume}}{\text{ml pr. sekund}} = \frac{27 \mu\text{L}}{0,1875 \text{ mL/s}} = 144 \text{ ms} \quad (4.4)$$

Fra de overstående beregninger kan det observeres, at tidsintervallet mellem kamera og ventilen er 54,475 millisekunder. Til at styre, hvornår ventilen skal åbne og lukke, er der implementeret to timer. Den ene åbner ventilen, mens den anden lukker ventilen. Den første timers tidsforsinkelse er tidsintervallet mellem kamera og ventil. Denne udregning er vist i formel 4.5. Den anden timer startes lige efter den første timer har åbnet for ventilen. Dens tidsforsinkelse beskriver, hvor længe ventilen skal stå åben. Udregningerne for den anden timer er vist i formel 4.6. I begge formler trækkes hhv. ventilens åbning og lukketid fra de beregnede tidsintervaller. I følge bilag A.1.12 har ventilen en åbningstid på op til 20ms og en lukketid på op til 30ms. I de to formler trækkes yderligere tiden fra, som det tager at sende en kommando fra Matlab til Arduinoen (26 ms). Selve implementeringen af timerne kan ses i projektdokumentationen afsnit 4.3.4.5.³

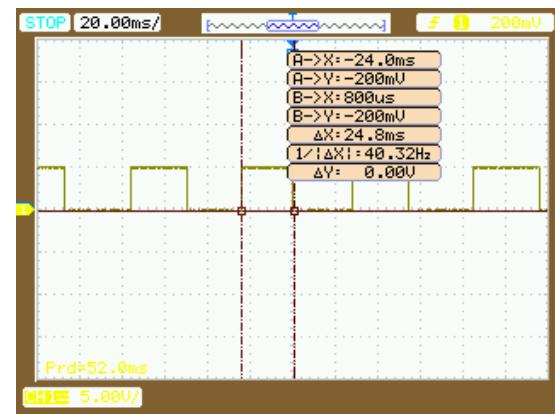
$$\text{Timer 1 delay} = 54,475 \text{ ms} - 20 \text{ ms} - 26 \text{ ms} = 8,475 \text{ ms} \quad (4.5)$$

$$\text{Timer 2 delay} = 144 \text{ ms} - 30 \text{ ms} - 26 \text{ ms} = 88 \text{ ms} \quad (4.6)$$

I figur 4.7 er enhedstesten for ventilen vist. Her er timerne implementeret med ovenstående delays. som det ses er tiden, hvor ventilen er åben målt til 112 ms. Dette resultat stemmer godt overens med timer 2 delay på 88 ms (formel 4.6) plus forsinkelsen på 26 ms som det tager at sende en kommando fra Matlab til Arduinoen. Dette giver en total tid på 114 ms, som ligger tæt på de målte 112 ms. Figur 4.8 viser, hvordan tidsforsinkelsen imellem Matlab og Arduinoen er bestemt. Det ses at en periodetid er på 52 ms, hvilket betyder at en kommando har en tid på 26ms, da en periode består af to kommandoer.



Figur 4.7. Tid hvor ventilen er åben. Timer 1 sætter pin'en høj (5V), for herefter at starte Timer 2. Når Timer 2's tid delay er gået sættes pin'en lav (0V). Den endelige tid ventilen er åben er 112 ms.



Figur 4.8. Tidsforsinkelse over seriell forbindelsen mellem Matlab og Arduino. På figuren sendes skiftevis en kommando til at sætte pin'en høj (5V) og en kommando til at sætte pin'en lav (0V) til Arduinoen. Periodetiden er på 52 ms.

³Fixme Note: Tjek ref

Åbnings- og lukketiden er et kompromis, der skal optimeres, da det ønskes at minimere mængden af eksokrint væv, der kommer over i beholderen med isolerede øer, men samtidigt skal sandsynligheden for, at ventilen er åben når øen kommer være så høj som mulig.

4.3 Kamera

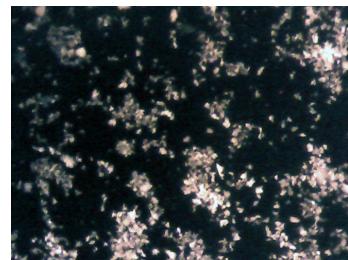
Det indkøbte kamera viste ikke at have tilstrækkelig kvalitet igennem en enhedstest. Der kunne ikke skelnes mellem det eksokrine væv og de langerhanske øer. Yderligere blev det erfaret, at lyset er utrolig vigtigt, derfor er ideen med kamerahuset en nødvendighed for at kontrollere lyset til kameraet. Der kan læses mere om enhedstesten af kameraet i afsnit 4.3.1⁴ i projektdokumentationen. Til at erstatte kameraet blev der i stedet genereret en række billeder, som skulle simulere flowet af opløsningsvæske igennem slangen.

4.3.1 Kamerasimulation

Til at generere et billedsæt, der simulerer langerhanske øer, er der udviklet et Matlab script. Billederne, der er valgt, stammer fra det indkøbte kamera. Grunden til de kan anvendes som grundlag for genereringen af billeder, på trods af kameraets utilstrækkelige kvalitet, er, at øerne og det ekstra væv er adskilt i de enkelte billeder. Dette muliggør en separat segmentering af øer og væv, som herefter kan sammensættes til billeder, der ligger tæt op af det der observeres gennem et almindeligt mikroskop. De 3 billeder, der danner grundlag for billedgenereringen, er vist nedenfor.



Figur 4.9. Billede indeholdende langerhanske øer



Figur 4.10. Billede indeholdende ekstra væv



Figur 4.11. Baggrundsbillede

Flowsimuleringen er opbygget på en måde, hvor den består af henholdsvis en sekvens indeholdende en langerhansk ø efterfulgt af en sekvens uden en ø. I selve programmet indlæses et nyt billede hvert 0,1 sekund. Derfor skal der generes en passende mængde billeder, som programmet kan indlæse. Simuleringen er implementeret, så der minimum genereres 252 eller maksimalt 432 billeder, hvilket giver en samlet sekvenslængde på 25,2 sekunder eller 43,2 sekunder. Grunden til, at antallet af billeder varierer, er, at længden af sekvensen uden en langerhansk ø bestemmes udfra en random variabel. Dette gøres for at simulere, at der kan være variabel tid mellem en ny ø kommer igennem slangen. I scriptet genereres der i alt 18 fulde sekvenser. Det betyder, at der passerer imellem 25 og 43 øer i minuttet. Antallet af øer, der passerer pr. minut, er bestemt ud fra formel 4.7:

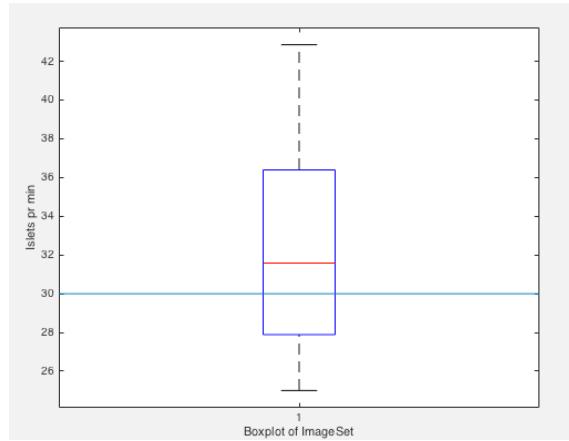
$$\frac{18 \text{ sekvenser}}{\frac{n}{10}} * 60\text{s} = \text{Antal øer pr. minut}, \text{ hvor } n \text{ er antallet af billeder i sættet}$$

(4.7)

I figur 4.12 er vist et boxplot, som viser distributionen af hvor mange øer der passerer i minuttet. Det ses at medianen ligger over 30 øer pr. minut, hvilket betyder, at der

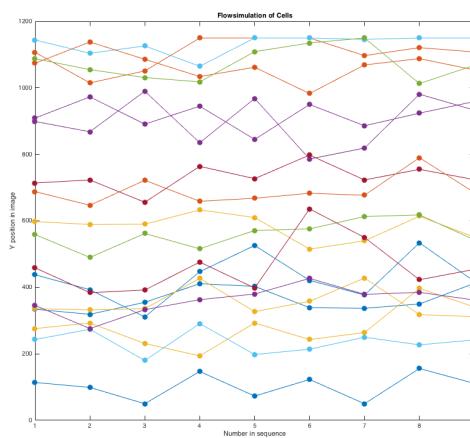
⁴Fixme Note: Tjek ref

i gennemsnit vil blive sorteret over 30 øer pr minut. De 30 øer pr. minut stammer fra hastighedskravet fra systemets kvalitetskrav (Se kravspecifikationen afsnit 1.4.1.1 i projektdokumentationen) ⁵



Figur 4.12. Boxplot af distributionen af øer pr. minut

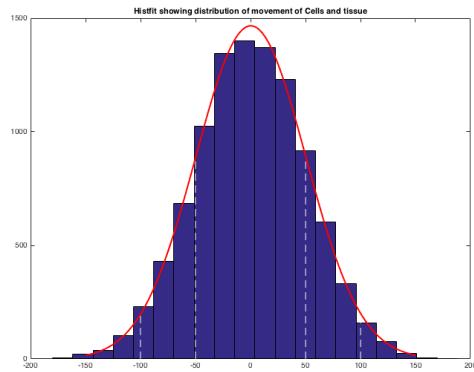
Selve flowsimuleringen sker i et for loop, hvor positionen for øen flyttes for hver iteration. I figur 4.13 er flow simulationen illustreret for de i alt 18 sekvenser, med en graf for hver ø. Det ses at øen flytter sin Y position tilfældigt, mens X positionen springer med et fast interval for hver iteration i forloopet.



Figur 4.13. Illustration over flow simuleringen

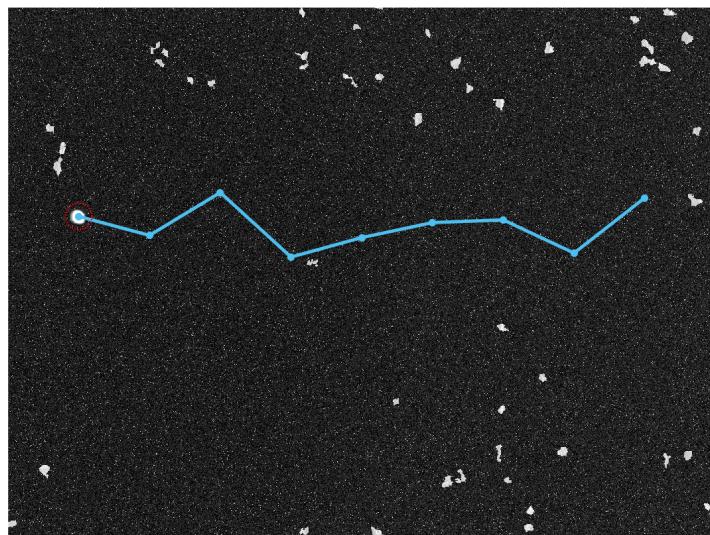
Øens position på billedet ændres ved en normalfordelt random variabel med en middelværdi på 0 og standard afvigelse(σ) på 50 pixels. Nedenstående figur 4.14 viser fordelingen af pixels øen flyttes for hver iteration. På figur 4.14 er der vist markører for standard afvigelsen (σ) og 2σ for hver side af mean. Imellem $-\sigma$ og σ er der 68,26 % sandsynlighed for, at den nye Y position ville ligge inden for dette område. For 2σ er der 95,45 % for, at den nye Y position vil ligge indenfor dette område.

⁵Fixme Note: tjek ref



Figur 4.14. Histogram over fordelingen af ny Y position

Det endelige resultat af billedgenereringen er vist i figur 4.15. Til at illustrere hvordan øen flytter sig er der tilføjet en graf, som viser dens position ændres for hver iteration. Øen er markeret med en rød ring på billedet.



Figur 4.15. Endelige resultat for flowsimulering

4.3.2 Billedprocessering

Segmenteringen sker via en række morfologiske operationer, hvor billedet fra kamera simulationen konverteres til en logisk maske. Detektionen baseres på, at øerne har en mere cirkulær form end resten af vævet. Derfor hentes der egenskaber fra objekterne i masken vha. *Matlab* funktionen *regionsprops*. De egenskaber, der hentes, er med til at beskrive hvor cirkulært objektet er. Egenskaberne er; arealet, centerpositionen, omkredsen og excentriciteten. Excentriciteten beskriver hvor langstrakt objektet er. Hvis excentriciteten er 0 er det en perfekt cirkel mens det vil være en langstrakt ellipse hvis verdien er 1.

Ud fra arealet og omkredsen af objekterne anvendes de 2 nedenstående formler til bestemmelse af 2 værdier for radiusen af objektet:

$$Areal = R1^2 * \pi \Rightarrow R1 = \sqrt{\frac{Areal}{\pi}} \quad (4.8)$$

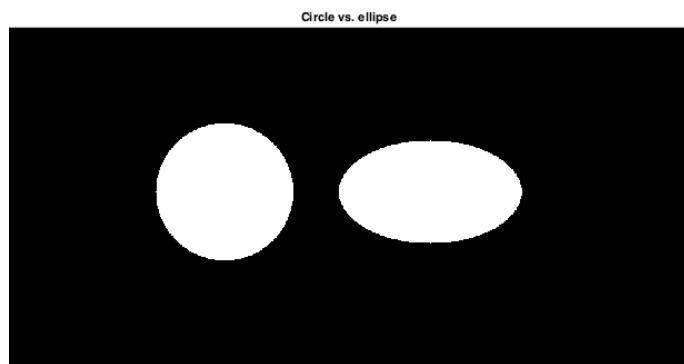
$$Omkreds = 2 * R2 * \pi \Rightarrow R2 = \frac{Omkreds}{2 * \pi} \quad (4.9)$$

Ud fra disse 2 radius værdier, vil det objekt, hvor der er mindst forskel imellem de 2, være det objekt der er mest cirkulært. Til at illustrere dette er der i figur 4.16 vist en cirkel og en ellipse, begge med samme areal. Ud fra *regionsprops* og de 2 formler for radius kan der bestemmes 2 radiuser for henholdsvis cirklen og ellipsen. Forskellen bestemmes ved at trække de 2 beregnede værdier fra hinanden. I udregningerne nedenfor er den første kolonne værdier for cirklen, mens den anden er for ellipsen. Som det ses, er forskellen mindst ved cirklen, hvilket indikerer, at den er mest cirkulær.

```

1 r1 = 100.0017 99.9380
2
3 r2 = 99.7978 105.7890
4
5 rDif = 0.2039 5.8510

```



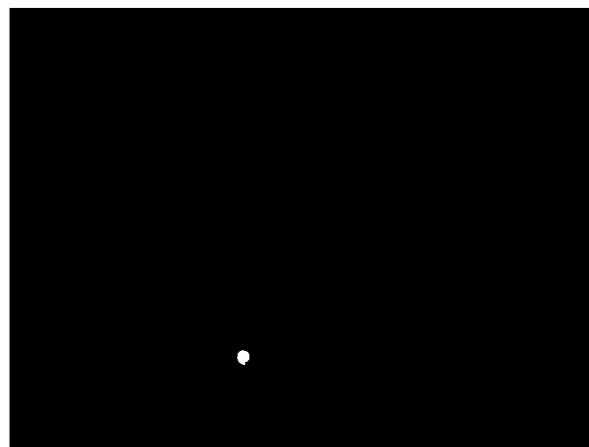
Figur 4.16. Cirkel sammenlignet med ellipse

Indeksset for det objekt med mindst forskel i radius og indekset for det objekt med den mindste excentricitet gemmes i variabler. Til sidst i segmenteringen af den langerhanske ø kontrolleres det om radius forskellen og excentricitet ligger under nogle fast definerede

grænseværdier. Værdierne er fundet ved at analysere en række billeder og deres objekters radius og excentricitet. Radius og excentriciteten bliver gemt i logfilen, så de kan analyseres for senere at justere grænseværdierne. Hvis objektets værdier er under disse grænseværdier er en ø detekteret. Når en ø er detekteret illustreres det med en grøn ring på GUI.

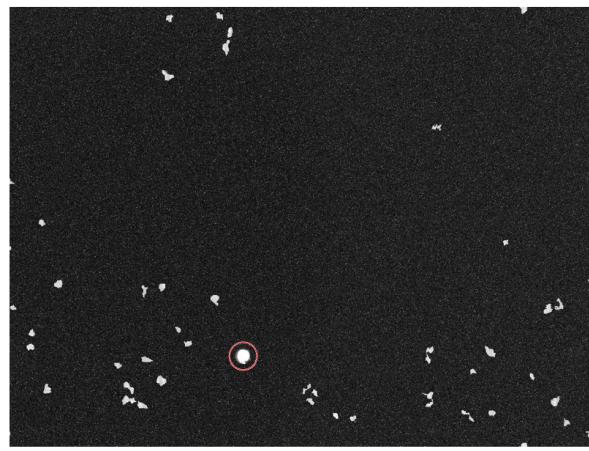
For at løse udfordringen med, at den samme ø vil optræde på det næste billede, er der implementeret et smalt detekteringsvindue (100 pixels bredt). Opløsningen på billederne er 800*600 pixels. Når øen er detekteret inden for dette område bliver variablen isletDetected sat til true. Denne variabel anvendes til styring af ventilen.

Figur 4.17 viser slutresultatet af segmenteringen, hvor masken kun indeholder den langerhanske ø fra det oprindelige billede.



Figur 4.17. Slutresultat af segmentering

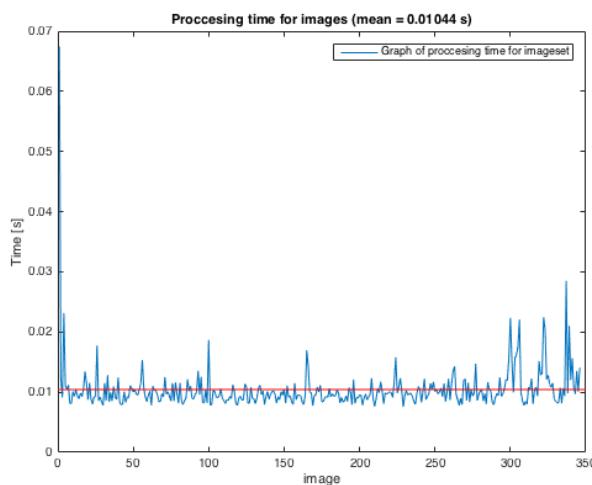
I figur 4.18 er det oprindelige billede vist, med markering af den detekterede ø.



Figur 4.18. Oprindelige billede med detekteret ø

Test af billedprocessering

Udfordringen med billedprocesseringen er, at hastigheden ikke må være for langsom i forhold til frame raten for kameraet. Da et nyt billede indlæses hvert 0,1 sekund skal billedprocesseringen dermed være hurtigere end dette for at følge med. Til test af dette er Matlab funktionen tic/toc anvendt, som kan måle hvor langt tid billedprocessingen tager om at eksekvere. I figur 4.19 viser grafen hvor lang tid det har taget, at behandle de enkelte billeder. Den røde streg viser den gennemsnitlige tid for processeringen. For den nuværende implementering tager billedeprocessingen 0,0104 sekund pr. billede, hvilket er indenfor grænsen på 0,1 sekund. Tiden vil variere alt efter tilgængelig processorkraft og om der skal udføres andre opgaver, eksempelvis opdatere figurer på GUI.



Figur 4.19. Test af tidsforbrug for billedeprocessing

4.4 Simuleringsvæske

Da de langerhanske øer ikke er lettilgængelige, har det været nødsaget at finde et objekt til at agere aktør for cellerne. Simuleringsvæsken skal bruges til at teste de mekaniske dele, samt tidsintervallet mellem kameraet og ventilen. Egenskaberne af de langerhanske øer der er brugt til at finde simuleringsvæsken med er at de er runde og lyse, størrelsen, samt at de bundfælder i collagenase opløsningen. Der har været to iterationer for at finde en aktør for cellerne, den første bestod i at finde perfekte runde og hvide objekter med en *massefylde* > 1. Til dette blev der brugt polysterene, som er lettilgængelige, samt runde, hvide og kan fås med en massefylde lige over 1 [Wunsch, 1999]. Dog blev det fundet svært at finde polysterene kugler i størrelsen 0.1-0.3mm, derfor blev der skabt kontakt til forhandlere af flamingo kugler. Dog var de tilsendte kugler for store, samt at de flød ovenpå grundet massefylden og overfladespændingen.

Den anden iteration bestod i at finde et biologisk materiale, hvor faktorerne runde og hvide blev nedprioriteret for at finde et objekt. For at finde en celleaktør med lettilgængelighed blev plantefrø den næste mulighed. Trods begrænset data omkring plantefrø, lykkes det gruppen at finde plantefrø i størrelsen 0.1-0.3mm vha. forhandlere af plantefrø og andre eksperter A.5.3. De indkøbte frø består af timian frø og rævehale frø, frøene kan ses på figur 4.20 og 4.21.



Figur 4.20. Billede af timianfrø



Figur 4.21. Billede af rævehalefrø

Udover de indkøbte frø kan der ved videre bearbejdelse af projektet overvejes transparent testa (tt4 mutant) frø. De skulle i følge *Carsten Meier* (se bilag A.5.3) være hvide, forholdsvis runde, men ikke lige så tilgængelige, som de allerede indkøbte frø. Dette har været grunden til, at de ikke er brugt i projektet indtil nu. Dermed består den anvendte simuleringsvæske af demineraliseret vand og timian plantefrø som aktør for de langerhanske øer.

En test af systemet med simuleringsvæsken viste, at frøene ikke kun komme igennem systemet. Frøene der blev anvendt i simuleringsvæsken havde en størrelse på 0,4 mm. Testen viste, at ventilen blokerede for frøene på trods af en specificeret diameter på 1 mm (Se bilag: A.1.12).

4.5 Accepttest af produktet

Til sidst i udviklingsforløbet blev der udført en accepttest af systemet. Den udførte accepttest er vedlagt i bilag A.6. En række af kvalitetskravene blev ikke testet grundet kameraet ikke blev implementeret i prototypen. For hvert af kravene der ikke blev godkendt er der udfyldt en fejlrappport, hvor en handlingsplan beskriver hvordan og hvornår fejlen rettes. De udfyldte fejlrapporter er ligeledes vedlagt i bilag A.7.

4.6 Cost-benefit analyse

Dette afsnit indeholder cost-benefit analysen, som belyser hvilke omkostninger der er forbundet med sorteringsproces. Analysen sammenligner omkostningerne forbundet med den manuelle sorteringsproces og den automatiserede metode. Formålet med analysen er at beregne, hvad stk. prisen pr. ø er, ved henholdsvis den ene og den anden metode. Dette anvendes til at give en tidshorisont for, hvornår en investering af den automatiserede løsning vil være tjent hjem igen. Udgangspunktet for analysen bygger på det tidsmæssige forbrug, der er forbundet med isolering af et batch på 400 øer fra 6 mus.

Fælles for begge sorteringsmetoder er, at der fortsat foretages manuel perfusion, nedbrydning og vaskning af pankreas. Nedenstående tabel opsummerer, hvilke basisudgifter der er forbundet ved disse processer. Timelønnen er beregnet ud fra en PhD studerendes månedsløn på 25.040 kr [Aarhus Universitet, 2015].

| | Antal | Pris | Total |
|-----------------------------------|--------------|-------------|------------------|
| Mus | 6 stk | 60 kr | 360 kr |
| Tid for perfusion | 1 time | 171,62 kr | 171,62 kr |
| Tid for nedbrydning + vask | 1 time | 171,62 kr | 171,62 kr |
| Total | | | 703,24 kr |

Tabel 4.1: Basis omkostninger for sorteringsproces

Nedenstående tabel 4.2 viser hvilke omkostninger, der er forbundet med selve isoleringsprocessen af de langerhanske øer. Den halve time, der bruges ved den automatiske metode er et estimat af, hvad der skal bruges på påfyldning af beholdere og start af sorteringsproces.

| | Antal | Pris | Total |
|-----------------------|--------------|-------------|--------------|
| Tid manuel | 6 timer | 171,62 kr | 1029,73 kr |
| Tid automatisk | 0,5 timer | 171,62 kr | 85,81 kr |

Tabel 4.2: Omkostninger ved sorteringsproces

Omkostningerne ved de to sorteringsmetoder er opsummeret i tabel 4.3, herunder en samlet pris for et batch på 400 øer, og stk. pris pr. ø for hver sorteringsmetode.

| | Manuelt | Automatisk |
|--|----------------|-------------------|
| Basis omkostninger | 703,24 kr | 703,24 kr |
| Isoleringsomkostninger | 1029,73 kr | 85,81 kr |
| Total pris for sorteringsprocesen | 1732,97 kr | 789,05 kr |
| Pris pr. ø ved batch på 400 stk | 4,33 kr | 1,97 kr |

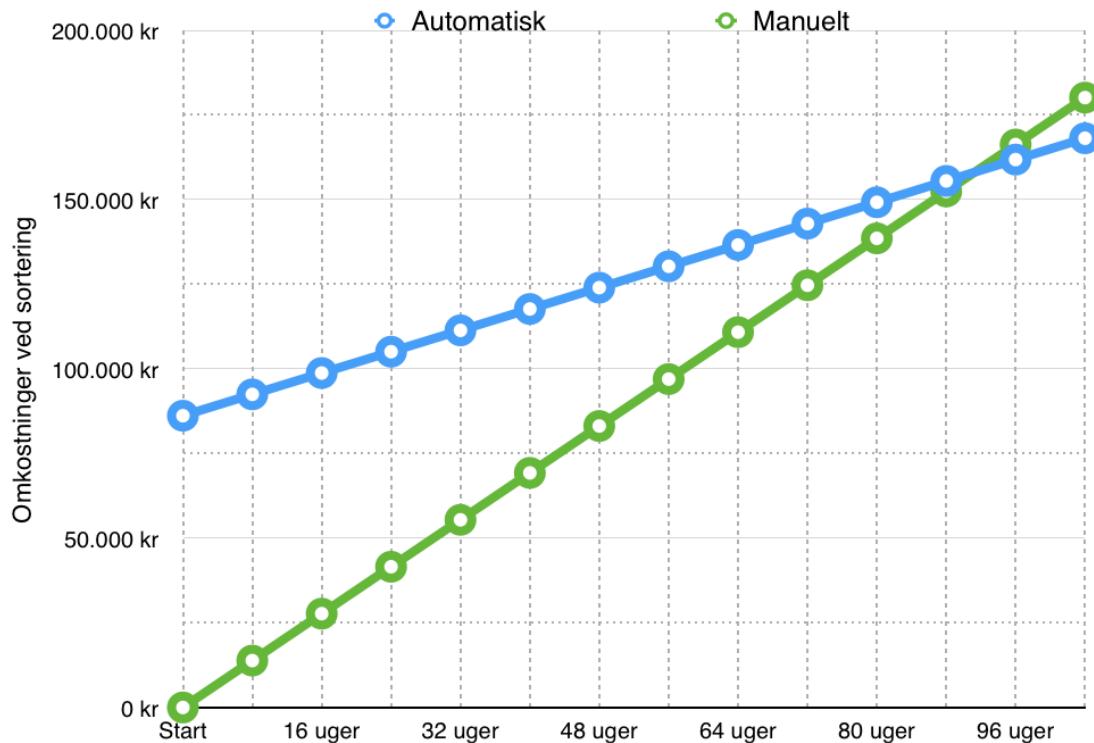
Tabel 4.3: Opsummering af omkostninger for sorteringsprocesen for de to sorteringsmetoder

Da der er forbundet omkostninger ved en investering i et system til automatisk isolering, er udgifterne til den udviklede prototype opsummeret i tabel 4.4. I tabellen er de enkelte hardware komponenters pris noteret, samt hvad en licens til Matlab koster. Herudover er der vurderet en profit på 400 % af udgifterne til hardware komponenter og Matlab licens. Denne udgift dækker udgifter til support og vedligehold, produktion, monteringsomkostninger, garanti og udvikling. Som det ses i tabellen er udgiften til den udviklede prototype beregnet til 86175 kr.

| | Pris |
|---------------------|------------------|
| Kamera | 400 kr [Farnell] |
| Ventil | 845 kr (A.5.3) |
| Pumpe | 80 kr [Ebay, b] |
| Arduino | 150 kr [Arduino] |
| Andet hardware | 210 kr |
| Beholdere + slanger | 425 kr (A.5.3) |
| 3d print | 125 kr |
| Matlab licens | 15000 kr |
| Profit på 400% | 68940 kr |
| Total pris | 86175 kr |

Tabel 4.4: Udgifter til automatiseret system

Udgiften ved de to sorteringsmetoder er illustreret i figur 4.22 over tid. Figuren viser omkostningerne set over en periode på 2 år, ved sorteringen af et batch pr. uge.



Figur 4.22. Udvikling i omkostninger

I figur 4.22 vises det, at efter 92 uger overstiger omkostningerne ved den manuelle sorteringsproces omkostningerne ved den automatiske. Dermed vil investeringen være tjent hjem efter 92 uger.

Diskussion 5

5.1 Billedprocessering

Da den udviklede prototype segmenterer øer ud fra billeder, der er genereret, kan det ikke konkluderes, at detektionsalgoritmen vil være lige så effektiv ved rigtige øer. Algoritmen vil givetvis skulle indeholde flere trin i segmenteringen for at give en præcis detektering af øerne. Dette vil gå ud over den tid det tager at processere de enkelte billeder, som vil give en lavere framerate. En lavere framerate kan have betydning for, hvorvidt en ø detekteres eller ej, da der mistes noget information. Dette vil have betydning for, om en ø bliver isoleret af ventilen. Flowhastigheden i væsken kan godt sænkes, men dermed opfyldes kravet om 30 øer i minuttet ikke.

Herudover vil en anden mængde eksokrint væv fra opløsningsvæsken end den fra de genererede billeder, kunne bidrage med yderligere støj. Dette vil potentielt kunne påvirke præcisionen af detektionsalgoritmen og derved detektere flere øer end der reelt er.

Ideelt set skulle billedprocesseringen være baseret på billeder af øerne igennem slangen, men grundet den første test af kameraet blev det klart, at kvaliteten af billederne ikke havde tilstrækkelig kvalitet. Derfor blev billedprocesseringen baseret på de genererede billeder.

5.2 Ventil og simuleringsvæske

Prototypen har vist, at det ikke er muligt at frasortere objekter fra simuleringsvæsken vha. ventilen. Det skal undersøges nærmere, hvorfor ventilen blokerer for objekterne selvom ventilens diameter er specificeret til 1 mm. Enhedstesten af ventilen viste, at Matlab timerne er præcise i forhold til styring af, hvornår ventilen skal åbne og lukke. Tidsforsinkelserne er dog baseret på udregninger, og derfor skal det optimeres, hvordan øernes væskeegenskaber har indvirkning for, hvornår ventilen skal åbne og lukke. Herudover skal tiden optimeres i en endelig prototype for at minimere mængden af eksokrint væv, som bliver isoleret. En anden vigtig tidsfaktor er, at tallene fra ventilens åbnings- og lukningstider er den maksimale tid det vil tage. Derfor kan der komme eksokrint væv med, hvis ventilen åbner hurtigere. Ligeledes kan en langerhansk ø blive fanget inde i ventilen, hvis den lukker hurtigere. Til sidst er slangen til beholderen med isolerede øer ikke regnet med. Det betyder, at en ø bliver i slangen, hvis denne er for lang. Derfor vil det være at foretrække, at slangen hertil er så kort som mulig.

5.3 Cost-benefit analyse

Resultatet af cost-benefit analysen viser, at der er økonomiske fordele ved at anvende en automatiseret løsning til isolering af øerne. Da det ikke har været muligt at lave en egentlig test af prototypen til sortering af øerne er en række af tallene estimeret. Derfor vil resultatet af cost-benefit analysen givetvis være anderledes ved en egentlig implementering af systemet. Herudover er systemets profit til dækning af ekstra udgifter sat til 400 %, hvilket ligeledes er et estimat af den udgift der vil være til ekstra udgifter. Yderligere er tilbagebetalingsperioden på 92 uger baseret på, at der laves et batch af øer pr. uge. I perioder hvor der foretages flere batch pr. uge vil tilbagebetalingsperioden dermed forkortes ved investering af en automatiseret løsning.

En af de ting, der ikke er belyst i cost-benefit analysen er om en automatiseret løsning vil have en betydning for arbejdsmiljøet. Da der bruges 6 timer på sorteringsprocessen ved den manuelle metode, vil den automatiserede løsning potentielt kunne forbedre arbejdsmiljøet. Det kan være svært at svare på, i hvilken grad arbejdsstillingen ved den manuelle metode medfører arbejdsskader. Der, hvor en automatiseret løsning vil kunne forbedre arbejdsmiljøet, er i højere grad ved at frigøre den tid den PhD studerende bruger på sorteringsprocessen til eksempelvis at forberede videnskabelige forsøg.

5.4 Hardware

Resultatet af hardwaren er mundet ud i et shield til Arduino platformen. I stedet kunne printet indeholde en microcontroller. Dette vil gøre Arduino platformen udnødvendig, men det vil kræve en bootloader til microcontrolleren for at kunne virke med *Matlabs* Arduino Support pakke. I samme ombæring kan det overvejes om, der i stedet for pakken, kan anvendes den serielle forbindelse til opsætning af microcontrolleren. På denne måde kan en string sendes over den serielle forbindelse til opsætning af systemet. Dette vil skabe mere kontrol over Arduinoen eller microcontrolleren. Det vil dog kræve en ændring i software implementeringen, som vil være en større arbejdsopgave på noget, der allerede fungerer. Økonomisk vil der ikke være meget at indtjene ved at integrere microcontrolleren i printet, grundet Arduino Unoens favorable pris. Produktet vil derimod opnå et mere professionelt udseende end ved brug af Arduino platformen.

5.5 Accepttest

Den udførte accepttest blev ikke godkendt, da kammeraet ikke levede op til den ønskede kvalitet. Kameraet blev derfor i prototypen erstattet af en funktion i Matlab, som simulerede kameraet. Derfor var det ikke muligt at teste kvalitetskravene i accepttesten, da detektionen ikke blev foretaget på rigtige øer. Dermed kunne det heller ikke testes om ventilen ville frasortere øerne, som beskrevet i kravspecifikationen. I en endelig prototype, hvor et kamera er implementeret ville en test af disse krav skulle testes igen for at verificere, at systemet lever op til de specificerede kvalitetskrav.

Perspektivering 6

Dette kapitel indeholder perspektivering af projektet, som kan udføres i en videreudvikling af projektet.

6.1 Kamera

I en videreudvikling af projektet bør der laves en test af en række forskellige kameraer for, at denne del af systemet har en tilstrækkelig kvalitet. Dermed kan kvaliteten af kameraerne testes inden et bestemt kamera indkøbes. Testen skal foretages ved, at der tages en række billeder af langerhanske øer, ideelt hvor øerne pumpes igennem slangen for at komme tæt på brugsscenariet.

6.2 Påvirkning af langerhanske øer

I et senere stadie af projektet bør der undersøges, hvor meget tryk og stress de langerhanske øer kan tåle. Området indeholder mange faktorer, der stadig er ukendte, hvilket bør undersøges. Derfor kræves et mindre studie omkring dette. Metoden, det kan gøres på, er ved at teste, hvor meget insulin øerne producerer efter, at de har været påvirket på forskellige måder. På den måde kan de isolerede øer testes inden og efter, at de pumpes igennem systemet, hvorefter resultatet sammenlignes. Det skal blandt andet undersøges, hvordan øerne opfører sig ved en peristaltisk pumpe, som brugt i projektet. Derudover bør ventilen undersøges. Hvis testen giver negative resultater, bør systemet deles op for at teste pumpen og ventilen hver for sig.

6.3 Pumpe

Den peristaltiske pumpe virker ved at sammenpresse slangen med roterende ruller, hvor slangen efter sammenpresningen udvider sig og dermed trækker væsken til sig. Væsken indespærres af den næste rulle, som sammenpresser slangen, og væsken drives derfor ud af slangen. Fordelene ved en peristaltisk pumpe er, at væsken ikke forurenes, da væsken er isoleret fra pumpen vha. slangen. Derudover skabes der et præcist flow, og pumpen kan ikke løbe "tør", da den er selvspændende. Ulemplen ved en peristaltisk pumpe er, at den netop sammenpresser slangen, hvilket kan risikere at klemme på de langerhanske øer. For at forhindre, at øerne bliver beskadiget, kan en sprøjtepumpe overvejes, fordi der bør være mindre påvirkning af øerne. Det vil være ideelt, hvis der et konstant flow med henblik på at opnå kravet om de 30 isolerede øer i minuttet. Derfor kan en tocylindret sprøjtepumpe med tilhørende ventiler bruges for at holde hastigheden konstant.

6.4 Ventil

Ved ventilen skal der undersøges, hvor lang tid der går fra kameraet har detekteret en langerhanske ø til den er ved ventilen. I projektet er forsøget kun lavet med simuleringsvæsken, men sandsynligheden for, at de langerhanske øer har samme væskeegenskaber som frøene brugt i simuleringsvæsken, er forholdsvis lav. Det er derfor ikke sikkert, at tidsintervallet er det samme. Grunden til, at forsøget bør laves er, at det er vigtigt, at ventilen isolerer de langerhanske øer, men det er heller ikke anvendeligt at få for meget eksokrint væv med. Yderligere vil et mindre volumen i ventilen være favorabel for at gøre udtømningstiden af ventilen mindre.

6.5 Parallelle systemer

For at opnå en højere hastighed kan parallelle rørsystemer ført forbi kameraet overvejes. Da kameraet sandsynligvis har et større synsfelt end én slange, kan systemerne føres forbi det samme kamera. Det kræver dog, at billedprocesseringen er hurtig nok.

6.6 Billedprocessering

I projektet er billederne til billedbehandling simuleret ved at bruge billeder taget af isolerede øer, hvor der er tilføjet tilfældigt støj og eksokrint væv. I videreudvikling af projektet bør billedprocesseringen optimeres til de faktiske øer i slangen.

6.7 Omrøring og køling af celleopløsningsbeholderen

Til videreudvikling af de ikke-elektroniske dele bør en omrøring i opløsningsvæsken skabes, da de langerhanske øer bundfælder. Samtidigt bør omrøringen håndteres, så skånsomt som muligt så øerne ikke beskadiges. En mulighed kunne være, at den tidligere omtalte sprøjtepumpe roteres, hvor der på den måde vil blive skabt en skånsom omrøring af opløsningsvæsken. Ydermere ville det være at foretrække, hvis systemet kunne nedkøle opløsningsbeholderen for at sikre, at enzymet ikke aktiveres uønsket. Det kunne være bestående af et kølelement.

6.8 Medicinsk udstyr

Da systemet primært er fokuseret til forskningsbrug, kræver udstyret ikke medicinsk godkendelse. Dog kan systemet på sigt muligvis bruges til ø implantationer, for lindring og som præparat mod diabetes. Skal systemet bruges til dette formål, vil det kræve en medicinsk godkendelse. For at et produkt kan opnå medicinsk godkendelse skal produktet overholde en række direktiver, som kan opnås igennem harmoniseret standarder. Det første punkt ved medicinsk godkendelse er udstyrets formål (intended use), da det er formålet der styrer klassifikationen for udstyret. I Europa klassificeres der efter 4 klasser I, IIa, IIb og III, hvor klasse III er den med de strengeste krav. Uanset klasse skal udstyret opfylde de væsentlige krav i MDD direktivet, som bl.a. er rapportering af utilsigtede hændelser og overvågning af systemet på markedet. Ydermere skal der foretages en risikoanalyse af produktets risikoer. For at et medicinsk udstyr kan CE mærkes skal det alt efter

klassificering, gennemgå trin igennem bilagene i MDD direktivet. Det skal indeholde teknisk dokumentation, væsentlige krav og risikoanalyse. Derudover skal der udarbejdes kvalitetsstyringssystemer til at dokumentere produktets forløb. Dokumentationen skal godkendes af et bemyndiget organ, hvis udstyret vurderes til at være højere end klasse I. Ydermere er der høje krav til versionsstyring af produktet, for at give en gennemsigtig sporbarhed. Før der begyndes på arbejdet med medicinsk godkendelse, er det vigtigt, at intended use klarlægges. Det skal derfor være sikker på, om udstyret går ind under MDD direktivet. Det bør undersøges, om produktet kan sammenlignes med en insulinpen, da den går ind under det farmaceutisk direktiv. Det gør den, fordi det er insulinen som virker.

Konklusion 7

Der er i dette projekt udviklet et *Proof of Concept* system i samarbejde med Medicinsk Endokrinologisk Afdeling, Aarhus Universitetshospital. Systemet skal anvendes til automatisk isolering af langerhanske øer i forbindelse med forskningsprojekter. Den udviklede prototype er baseret på kameradetektion og mekanisk isolering af øerne. Projektet har vist, at det er muligt at detektere langerhanske øer vha. billedprocessering ud fra genererede billeder. Det lykkedes ikke i projektet at isolere objekterne i simuleringsvæsken. Projektet har yderligere givet et indblik i, hvilke problemstillinger der skal løses inden en fungerende prototype kan implementeres i praksis, herunder hvorvidt øerne tager skade af processen.

En cost-benefit analyse har herudover belyst, hvilke økonomiske og kvalitative fordele den udviklede prototype vil have i sorteringsprocessen. Resultaterne af denne analyse giver incitament til at arbejde videre med en automatiseret løsning til isolering af øerne.

Litteratur

Aarhus Universitet, 2015. Aarhus Universitet. *PhD stipendium.*
<http://www.au.dk/forskudd/stip/>, 2015. Downloadet: 02-12-2015.

Arduino. Arduino. *Microcontrollerboard— Arduino, UNO.* URL
<https://www.arduino.cc/en/Main/ArduinoBoardUno>. [Online; accessed 30-October-2015].

Bruni et al., 2008. Anthony Bruni, Boris Gala-Lopez, Andrew R Pepper, Nasser S Abualhassan og AM James Shapiro. *Islet cell transplantation for the treatment of type 1 diabetes: recent advances and future challenges.* Diabetes Metab Syndr Obes, pages s. 211–223, 2008.

Diabetesforeningen, 2015. Diabetesforeningen. *Fakta om diabetes i Danmark.*
<http://www.diabetes.dk/presse/diabetes-i-tal/diabetes-i-danmark>, 2015.
Downloadet: 16-07-2015.

Ebay, a. Ebay. *Vægtcelle- Ebay.* URL <http://www.ebay.com/itm/281311660424>.
[Online; accessed 30-October-2015].

Ebay, b. Ebay. *Pumpe- Ebay.* URL <http://www.ebay.com/itm/6V-Peristaltic-Pump-Dosing-Water-Pump-DC-Motor-Tube-For-Aquarium-Lab-Analytical-/131367703927?hash=item1e96201577>. [Online; accessed 30-October-2015].

Farnell. Farnell. *Farnell — Kamera, microscope digital.* URL <http://dk.farnell.com/duratool/bw788/microscope-digital-usb-25x-200x/dp/2319420>. [Online; accessed 30-October-2015].

Gregersen, 2015. Søren Gregersen. *Powerpoint - projektoplæg*, 2015.

HBSS. HBSS. *Hanks Balanced Salt solution.* <http://www.sigmapellici.com/catalog/product/sigma/h6648?lang=en®ion=DK>.
Downloadet: 03-12-2015.

Jeppesen, 1997. Per Bendix Jeppesen. *Master Thesis - The effect of the diterpene glucoside stevioside and the aglucon steviol, on the endocrine pancreas*, 1997.

Matlab, 2015. Matlab. *Least Square Fitting.*
<http://se.mathworks.com/help/curvefit/least-squares-fitting.html>, 2015.
Downloadet: 10-12-2015.

Wunsch, 1999. J. R. Wunsch. *Polystrene Synthesis, production and Applications.* ISBN: 978-1-859-57191-0, Paperback. Rapra Technology Ltd, 1999.

Zhenping Liu og Hermansen, 2014. Søren Gregersen Xiaoping Chen Zhenping Liu, Per B. Jeppesen og Kjeld Hermansen. *Dose- and Glucose-Dependent Effects of Amino*

Acids on Insulin Secretion from Isolated Mouse Islets and Clonal INS-1E Beta-Cells.
Diabetes Studies, pages s. 232–244, 2014.

Rettelser

| | |
|--------------------------|----|
| Note: Tjek ref | 21 |
| Note: Tjek ref | 23 |
| Note: tjek ref | 25 |
| Note: Tjek ref | 26 |
| Note: Tjek ref | 27 |
| Note: tjek ref | 32 |
| Note: Tjek ref | 33 |
| Note: Tjek ref | 35 |
| Note: Tjek ref | 37 |
| Note: tjek ref | 38 |

Bilag A

A.1 Datablade

A.1.1 Operationsforstærker - INA114

A.1.2 Motordriver - L293D

A.1.3 Lysdiode - L5-W55N-BVW

A.1.4 Ventil - 161T031

A.1.5 Ventil - 161T031

A.1.6 Kamera - BW788

A.1.7 Molex 5268

A.1.8 Molex 5566

A.1.9 Operationsforstærker - INA114

A.1.10 Motordriver - L293D

A.1.11 Lysdiode - L5-W55N-BVW

A.1.12 Ventil - 161T031

A.1.13 Kamera - BW788

A.1.14 Molex 5268

A.1.15 Molex 5566

A.1.16 DIL sokkel

A.2 Matlab kode

A.2.1 initArduino.m

A.2.2 cameraFeed.m

A.2.3 detectIslets.m

A.2.4 loadCell.m

A.3 Arduino Testkode

A.3.1 Kode til enhedstest til vægtcelle.pdf

A.3.2 Kode til enhedstest til pumpe.pdf

A.3.3 Kode til enhedstest til ventil.pdf

A.4 Møderefereater

A.4.1 Vejledermøder

A.4.2 Kundemøder

A.4.3 Reviewmøder

A.4.4 Andre

A.5 Mail korrespondancer

A.5.1 Kunde

A.5.2 Vejleder

A.5.3 Leverandør

A.5.4 Andre

A.6 Accepttest

A.7 Fejlrapport

A.8 Logbøger

A.9 Samarbejds aftale

A.10 Handlingsplan