



# CHALMERS

---



## Kalibrering av ljussensor för Parans solpanel

Examensarbete inom Högskoleingenjörsprogrammet i datateknik

PÄR SVEDBERG  
OSKAR ÅKERGREN



EXAMENSARBETE

# **Kalibrering av ljussensor för Parans solpanel**

PÄR SVEDBERG  
OSKAR ÅKERGREN

Institutionen för data- och informationsteknik  
CHALMERS TEKNISKA HÖGSKOLA

Göteborg, Sverige 2015

**Kalibrering av ljussensor för  
Parans solpanel**  
PÄR SVEDBERG  
OSKAR ÅKERGREN

© PÄR SVEDBERG, OSKAR ÅKERGREN, 2015

Institutionen för data- och informationsteknik  
Chalmers tekniska högskola  
SE-412 96 Göteborg  
Sverige  
Telefon: +46 (0)31-772 1000

Omslag:  
Parans solpaneler modell SP3 installerade på taket till Centrum för vuxnas lärande i Växjö  
*Fotograf: Oskar Åkergren*

## Abstract

Parans Solar Lighting, a company based in Göteborg, have developed a solar panel used to bring sunlight indoors, by focusing light into optical fibres. A photosensor is used to adjust the panel's direction, in order to maximize its intake of light, and the necessary calibration of the photosensor was previously performed manually. This degree project report describes the development of an automatic procedure for the calibration of the photosensor, and focuses on two main problems. The first problem was the development of an algorithm that performs the calibration, using values of measured luminosity, and the second problem was to find a communication solution to transfer these values from the illuminated room to the panel. The project resulted in a calibration software application and two suggestions of solutions regarding the communication, where the solar panels fibre cables are used as communication medium.

**Keywords:** calibration, serial communication, optical communication, fibre optics, microcontroller, indoors lighting, Parans, light sensor

## Sammandrag

Företaget Parans Solar Lighting har utvecklat en solpanel som leder solljus genom optisk fiber för att användas till inomhusbelysning. Solpanelen använder en fotosensor för att maximera ljusintaget genom att justera dess riktning och en kalibrering av fotosensorn är nödvändig, vilket tidigare har utförts manuellt. Rapporten beskriver framtagandet av en automatisk kalibrering av solpanelens fotosensor och fokuserar på två huvudområden. Det första området är en kalibreringsalgoritm som använder värden av uppmätt ljusstyrka och det andra området är en lösning för kommunikation mellan solpanelen och det upplysta rummet, i syfte att förmedla dessa värden. Resultaten av projektet var en mjukvarulösning som kalibrerar panelen och två förslag på lösningar gällande kommunikationen, där det rekommenderades att använda panelens egna fiberkablar som överföringsmedium.

**Nyckelord:** kalibrering, seriell kommunikation, optisk kommunikation, fiberoptik, mikrokontroller, belysning, Parans, ljussensor

## **Förord**

Detta examensarbete skrevs inom ramen för kursen LMTX38, Examensarbete vid Data- och informationsteknik vårterminen 2015 på Chalmers tekniska högskola.

Författarna vill rikta ett stort tack till Parans Solar Lighting AB för tillhandahållande av utrustning, arbetsytor och stöd under projektets gång. Ett särskilt tack riktas till vår handledare från Parans, Karl Nilsson, för förslag till examensarbetets problemområden och avsatt tid för handledning. Vi riktar även ett tack till Simon Larsson hos Parans för diskussioner om idéer och stöd under utvecklingsarbetet. Vidare vill vi tacka handledare Lennart Hansson vid Chalmers för hans stöd under rapportens framställning och förslag om att driva arbetet framåt. Ett tack skickas också till Sakib Sistek vid Chalmers för hans förmedling av kontakten till Parans och för hans hjälp vid förberedelsen till detta examensarbete. Avslutningsvis vill vi tacka familj och vänner för visat stöd under arbetets gång.

## Beteckningar

8N1	Parameterinställning för seriell kommunikation
C	Programmeringsspråk
I/O	Input/Output
I <sup>2</sup> C	Standard för synkron seriell datakommunikation
lm	lumen, SI-enhet för ljusflöde
lux	SI-enhet för belysning, $1 \text{ lux} = 1 \text{ lm/m}^2$
UART	Universal Asynchronous Receiver/Transmitter, gränssnitt för seriell kommunikation

# Innehåll

<b>1</b>	<b>Introduktion</b>	<b>1</b>
1.1	Bakgrund . . . . .	1
1.2	Syfte . . . . .	1
1.3	Mål . . . . .	2
1.4	Frågeställning . . . . .	2
1.5	Avgränsningar . . . . .	2
1.5.1	Hårdvara . . . . .	2
1.5.2	Mjukvara . . . . .	2
<b>2</b>	<b>Metod</b>	<b>3</b>
2.1	Vetenskaplig metod . . . . .	3
2.2	Arbetsmetodik . . . . .	4
<b>3</b>	<b>Teknisk bakgrund</b>	<b>5</b>
3.1	Parans SP3 . . . . .	5
3.1.1	SP3 mikrokontrollerkort . . . . .	5
3.1.2	Fiberoptik . . . . .	5
3.1.3	Ljussensor . . . . .	6
3.2	Luxmätare . . . . .	6
3.2.1	Adafruit TSL2591 Digital Light Sensor . . . . .	6
3.2.2	Yoctopuce Yocto-Light-V3 . . . . .	6
3.3	Arduino . . . . .	7
<b>4</b>	<b>Genomförande</b>	<b>9</b>
4.1	Fas 1 . . . . .	9
4.2	Fas 2 . . . . .	10
4.3	Fas 3 . . . . .	10
4.3.1	Kommunikationslösning . . . . .	11
4.3.2	Avläsning av luxmätare . . . . .	12
4.3.3	Algoritm . . . . .	13
4.3.4	Applikation . . . . .	15
4.4	Fas 4 . . . . .	17
<b>5</b>	<b>Resultat</b>	<b>19</b>
5.1	Algoritm . . . . .	19
5.2	Optisk kommunikation . . . . .	19
5.2.1	Förutsättningar . . . . .	19
5.2.2	Tillförlitlighet . . . . .	20
5.3	Applikation . . . . .	21



<b>6</b>	<b>Slutsats</b>	<b>23</b>
6.1	Sammanfattning . . . . .	23
6.2	Diskussion . . . . .	23
6.3	Arbetsgång . . . . .	26
6.4	Vidareutveckling . . . . .	26

	<b>Referenser</b>	<b>27</b>
--	-------------------	-----------

	<b>Bilagor</b>	<b>I</b>
--	----------------	----------

<b>A</b>	<b>Simuleringsresultat av kalibreringsalgoritm</b>	<b>I</b>
----------	--	----------

<b>B</b>	<b>Flödesschema av kalibreringsalgoritmen</b>	<b>II</b>
----------	---	-----------

<b>C</b>	<b>UML-Diagram</b>	<b>III</b>
----------	--------------------	------------

<b>D</b>	<b>Kopplingsschema</b>	<b>IV</b>
----------	------------------------	-----------

<b>E</b>	<b>Specifikationer</b>	<b>V</b>
----------	------------------------	----------

E.1	Arduino . . . . .	V
-----	-------------------	---

E.2	Sändare . . . . .	V
-----	-------------------	---

E.3	Mottagare . . . . .	V
-----	---------------------	---

<b>F</b>	<b>Uppmätt fokuspunkt från panel</b>	<b>VI</b>
----------	--------------------------------------	-----------

<b>G</b>	<b>Tidsplan</b>	<b>VIII</b>
----------	-----------------	-------------

## Figurer

2.1	Utformning av anpassad metod . . . . .	3
-----	--	---

4.1	Exempel på förväntade fokuspunkter . . . . .	9
-----	--	---

4.2	Översikt av sändare/mottagare . . . . .	11
-----	---	----

4.3	Översikt av rundgångskoppling . . . . .	12
-----	---	----

4.4	Jämförelse algoritmernas antal söksteg . . . . .	14
-----	--	----

4.5	Uppmätt ljusstyrka . . . . .	15
-----	------------------------------	----

4.6	Skiss av det grafiska gränssnittet . . . . .	17
-----	--	----

## Tabeller

1	Minskning av antal söksteg mellan algoritmversioner . . . . .	14
---	---	----



# 1 Introduktion

## 1.1 Bakgrund

Parans är utvecklare av en produkt som via optiska fibrer levererar naturligt solljus in i byggnader som ett alternativ till dagens traditionella ljuskällor. Bolaget är baserat i Göteborg men levererar systemen globalt och har flera installationer runt om i världen.

Produkten fokuserar in solljus i optiska fibrer och styrs med hjälp av två stegmotorer för att följa solens bana. Styrningen sker med en algoritm som ger solposition i grader, baserat på installationsplatsens geografiska position och tid, och för finare styrning av panelen då solen är framme inhämtas data från en ljussensor med fotocell. Detta för att alltid maximera det solljus som fokuseras in i fibern.

Styrkortet och motorerna till panelen drivs av en spänning om tolv (12) volt och kortet är en egen design kring mikrokontrollen PIC32. Källkoden till panelen är skriven i C och kommunikation till enheten sker via seriell förbindelse över USB, där en USB till UART-omvandlare är integrerad på styrkortet. För att skicka instruktioner till panelen används en terminalemulator.

Ljussensorn som används i solpanelen kan representeras som ett koordinatsystem där sensorn som standard förväntar sig att ljuset genom en lins fokuseras till en punkt som träffar origo. Parans problem är att vid tillverkning av panelen kan linsen fokusera ljuset något vid sidan av sensorns origo, vilket leder till sämre ljusintag till de optiska fibrerna. Efter fabriksmontering åtgärdas detta genom att kalibrera sensorn, genom att flytta punkten på det koordinatsystem som ljuset fokuseras ned till. Parans använder en manuell metod där man via en terminalemulator anger kommandon som vrider solpanelen och sedan kontrolleras värdet på en separat luxmätare.

Vid installation hos slutkund har det i ibland visat sig att en enstaka panelers kalibrering inte har varit korrekt och vad detta beror på är fortfarande inte klarlagt. Sensorn måste då kalibreras om på installationsplatsen, vilket idag vanligtvis sker manuellt av två personer. Den ena personen befinner sig då på taket vid panelen och justerar ljussensorns kalibreringsvärden, via den seriella anslutningen, samtidigt som den andra personen uppskattar den upplevda ljusstyrkan i det upplysta rummet.

## 1.2 Syfte

Syftet med projektet är att möjliggöra en helt automatisk process som kan kalibrera fotosensorn i Parans solpaneler så att maximalt ljusflöde från panelen kan uppnås. Syftet med processen är att minska tidsåtgången och höja precisionen jämfört med dagens manuella metod. Vidare syftar projektet till att möjliggöra kommunikation mellan panelen och en luxmätare inne i byggnaden, så att kalibreringstekniken kan nyttjas till systemen generellt, oavsett om de är tagna i bruk eller i fabrik.

### 1.3 Mål

Målet med det här projektet är att ta fram en produkt som justerar fokuspunkten på ljussensorn, vilket då vrider på solpanelen för att lokalisera de x- och y-värden där intaget av solljus är som störst. Produkten ska kunna användas vid ett kalibrerings-tillfälle och därefter avlägsnas från installationsplatsen. Ljusstyrkan mäts med hjälp av en luxmätare som levererar ljusintaget till en dator eller till en annan programmerbar enhet. När det maximala ljusintaget är uppmätt registreras x- och y-värdena som den nya fokuspunkten för ljussensorn istället för det förinställda värdet på origo. Vidare är målet med produkten att den ska stödja kommunikation mellan en luxmätare inne i byggnaden och en panel som befinner sig på taket.

### 1.4 Frågeställning

- Vilken algoritm kan anses vara lämplig för kalibreringen?
- Vilka förutsättningar för kommunikation finns mellan solpanelen och det upplysta rummet?
- Hur tillförlitligt är det valda kommunikationssättet?

### 1.5 Avgränsningar

#### 1.5.1 Hårdvara

Redan existerande hårdvara kommer att användas, det vill säga sådan avsedd att användas för de ändamål nödvändiga för projektet. Den primära hårdvaran, solpanel och luxmätare, kommer att tillhandahållas av uppdragsgivaren och inga alternativ till dessa kommer att undersökas. Eventuell övrig hårdvara kan antingen vara helhetslösningar eller sådana som löser delproblem och kombineras. De lösningar som kommer att undersökas och utvecklas är begränsade till att stödja företagets panel SP3.

#### 1.5.2 Mjukvara

Mjukvara kommer att utvecklas för att nå projektets uppsatta mål. Denna kan komma att inkludera användning av både medföljande och externa ramverk och bibliotek för att lösa olika delproblem, exempelvis grafisk framställning och kommunikation mellan olika enheter.

## 2 Metod

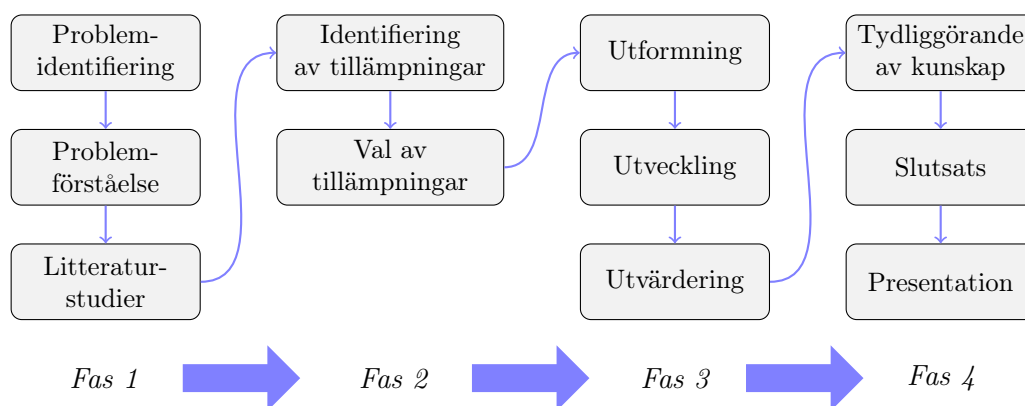
### 2.1 Vetenskaplig metod

Detta projekt har tillämpat en variant av den vetenskapliga metoden Design Science Research (DSR), en metod som anses lämplig till problemlösande forskning där redan existerande produkter ska vidareutvecklas [1, s. 13]. Målet med DSR är att skapa artefakter, exempelvis en praktisk lösning, metod eller lösningsförslag, som löser de problem som identifierats inom projektet.

Design Science valdes då dess mål stämmer bra överens med det som projektet syftar till att göra. Detta kan sättas i kontrast med mer traditionella vetenskaper som snarare syftar till att utforska, förklara eller förutse fenomen [1, s. 13]. Att DSR valdes som metod över fallstudier eller action research är återigen att målen överensstämmer med projektet och även att typen av kunskap som anskaffas stämmer bättre överens än de andra två alternativen [1, s. 95].

Dresch et al. rekommenderar, baserat på studier av flera metoder för DSR, en metod i 12 steg [1, s. 118–126]. De tre inledande stegen är en analys av de problem som ska lösas, problemidentifiering, problemförståelse och litteraturstudier. Denna inledande fas mynnar ut i att hitta eventuella befintliga lösningar som kan vara lämpliga och att sedan föreslå en vidareutveckling och tillämpning av denna eller att föreslå en ny lösning. Steg sex till åtta är sedan att utforma, utveckla och utvärdera lösningen. Därefter ska den kunskap som givits av tidigare steg tydliggöras och slutsatser dras. Tidigare nämnda steg itereras vid behov för att uppnå önskat resultat. Slutligen ska generalisering av lösningen utformas och resultatet presenteras.

Ovan nämnda metodik har för detta projekt förenklats något för att anpassas till projektets storlek och omfattning. Den anpassade metoden indelas i fyra faser med totalt 11 steg, enligt figur 2.1, där fas 1–3 itereras efter behov.



Figur 2.1: Utformning av anpassad metod

I den första fasen, problemidentifikationsfasen, läggs grunden till projektet. Här strävas efter att skapa en grundläggande förståelse av det problem som projektet avser lösa, samt att möjliggöra fördjupning inom områdets rådande forskningsläge. Efter att problem har identifierats och litteraturstudier har genomförts tar nästa fas vid. I den andra fasen läggs förslag till hur problem kan lösas och ett sådant förslag väljs sedan att föras över till nästkommande fas. Fas 3 syftar till själva utvecklingen av artefakter, det vill säga att faktiskt framställa någonting som går att använda. Den fjärde och avslutande fasen avhandlar inte arbetsgången som sådan, utan visar på att resultatet från den tredje fasen ska analyseras och delges i syfte att sprida kunskapen om vad som har uppnåtts vidare.

## 2.2 Arbetsmetodik

Projektets arbetsmetodik utgick ifrån versionshanteringsverktyget `git` som var den mjukvara som användes för att hantera utveckling och dokumentation. För att få tillgång till redundant och centraliserad lagring använde projektet sig av *GitHub.com* vilket även bistod med ett grafiskt gränssnitt till `git`, som i sig självt endast har ett textbaserat gränssnitt. Detta gav också uppdragsgivaren en tydlig inblick i det pågående arbetet.

Vidare var arbetsmetodiken inspirerad av Scrum där större mål sattes upp och bröts ner till mindre så kallade *issues* [2, kap. 8]. Dessa issues sattes upp på en virtuell tavla med hjälp av verktyget *Waffle.io* för att få en bättre överblick kring hur projektet utvecklades och vad som behövde göras.

Anledningen till att inte hela Scrum-metodiken anammades var att projektet utfördes av få personer så den rollfördelning som hör till i Scrum gick ej att utföra på ett meningsfullt vis [2, kap. 6], samt att ovanan vid denna typ av utveckling gjorde att kostnaderna för varje issue var svårt att bestämma. Vidare var projektets omfång väl avgränsat av uppdragsgivaren så istället för de föreslagna användarberättelserna användes avgränsningarna som milstenar (inbyggd funktion i GitHub) [2, kap. 9].

## 3 Teknisk bakgrund

### 3.1 Parans SP3

SP3 är tredje generationens solpanel utvecklad av Parans [3]. Panelen monteras på utsidan av en byggnad, ofta på taket, och fokuserar solljus genom linser in i optisk fiber för att sedan genom armatur lysa upp inomhus. Varje panel har sex utgående kablar med fiberoptik, vardera ansluten till en armatur, vilkas räckvidd är upp till 20 meter. Panelen är utformad så att ultraviolett och infrarött ljus avskärmas från det ljus som leds in i fibern. Vid fabriksmontering limmas fibrerna fast i panelen och dessa kan sedermera ej enkelt avlägsnas. Diagnostik av ljusintaget är därför inte möjligt vid panelen, utan kan endast ske vid fiberkablarnas ändar.

Två stegmotorer används för att justera panelens riktning horisontellt och vertikalt så att linserna alltid är vända mot solen. Motorernas rörelser styrs av ett mikrokontrollerkort där en algoritm i mjukvaran räknar ut solens nuvarande position. Algoritmen kombinerar mätvärden från en ljussensor med solens förväntade himlposition, baserat på tid, datum och installationsplatsens geografiska position. Mjukvaran som körs på mikrokontrollern är skriven i C.

#### 3.1.1 SP3 mikrokontrollerkort

Mikrokontrollerkortet som används i panelen är konstruerat av Parans och är baserat på en PIC32-mikrokontroller. PIC32 är en kategori mikrokontroller tillverkade av Microchip Technology för användning i inbyggda system och ger tillgång till bland annat I/O-anslutningar och UART för seriell kommunikation [4]. För att kommunicera med mikrokontrollerkortet via en dator finns en USB-port som ger en seriell anslutning som hanteras av en USB till UART-krets från Silicon Laboratories, CP2102. Detta kräver att den anslutna datorn har en drivrutin för CP2102 installerad och möjliggör anslutning via en terminalemulator för installation, diagnostik och underhåll.

#### 3.1.2 Fiberoptik

Den fiberoptiska kabel som i dagsläget används av Parans är en plastfiber som har en ljusöverföring om 96 % per meter. Panelen saluförs med fiberkablar i fyra fasta längder, 5, 10, 15 och 20 meter [5]. Var kabel består av sex stycken fibrer och ger ett ljusflöde om 730 lm till 430 lm från solpanelen i fullt solljus, beroende på längd av kablagen. Vid fibrernas paneländar finns IR-speglar monterade som reflekterar den infraröda delen av solljuset och de används för att fibern annars riskerar att smälta.

### 3.1.3 Ljussensor

För att optimera ljusintaget i panelens fibrer finjusteras vinkeln till solen med hjälp av en ljuskänslig krets monterad på panelens front. Kretsen skyddas av ett gråfilter med 10 % ljusgenomsläpp för att dämpa solljusets intensitet. Via en lins fokuseras genomträngande ljus till en fokuspunkt på sensorn som ger ett utslag och kretsen omvandlar denna data till fyra strömmar, vilkas värden representerar avståndet från sensorns mittpunkt. Strömmarna omvandlas och representeras som x- och y-värden i ett koordinatsystem, värdena kan sedan avläsas och manuellt justeras vid kommunikation med panelen. Själva ljussensorn är tillverkad av Hamamatsu, med beteckningen S5901, och sitter integrerad på ett egendesignat mönsterkort. Databladet för den använda sensorn har ej kunnat lokaliseras men enligt information inom Parans kan den antas arbeta efter samma princip som sensorerna S5990-01 och S5991-01 [6].

## 3.2 Luxmätare

För att mäta upp ljusstyrkan från panelen används en luxmätare kopplat till en av de fiberkablar som leder ner till det upplysta rummet. De luxmätare som stöds av projektet är mätare avsedda för privatbruk och är inte att anse som professionella i det avseende att ljusmiljön i ett rum kan bestämmas med hjälp av dem. Syftet med mätarna är istället att registrera skillnaden i ljusstyrkan från panelen. När panelen är rätt kalibrerad kommer luxmätaren leverera ett högre värde än vid en felkalibrerad panel. Det exakta värdet är i detta fall inte av intresse, det är istället möjligheten att hitta den inställning på panelen som ger luxmätarens maximala värdet.

### 3.2.1 Adafruit TSL2591 Digital Light Sensor

Sensorkortet från Adafruit innehåller ljussensorn TSL2591 från tillverkaren ams och används till att uppmäta ljusintensitet upp till 88 000 lux och kan anslutas till en mikrokontroller via I<sup>2</sup>C. På sensorn finns två fotodioder, där den ena reagerar på IR-ljus och den andra reagerar på fullspektrumljus [7]. Fotodiodernas avläsning kan ske oberoende av varandra eller kombineras vilket möjliggör avläsning av endast synligt ljus. Sensorns värden avläses digitalt.

### 3.2.2 Yoctopuce Yocto-Light-V3

Yocto-Light-V3 är en luxmätare i form av ett kretskort, baserad på ljussensorn BH1751FVI från ROHM, som är avsedd att mäta synligt ljus upp till 100 000 lux [8]. Kretskortet har en USB-port för anslutning till dator och kräver ingen extra drivrutin mer än de som medföljer vanliga operativsystem för att användas. Tillverkaren Yoctopuce tillhandahåller kodbibliotek till flera vanligt förekommande programmeringsspråk som möjliggör avläsning av sensorvärden.



### 3.3 Arduino

Arduino Uno är ett mikrokontrollerkort byggt kring mikrokontrollern ATmega328. På kortet finns bland annat USB-anslutning och 20 pins för att ansluta externa enheter och kringutrustning, 6 analoga och 14 digitala. [9]. Mikrokontrollern är en del av AVR-serien från Atmel och den har en RISC-baserad processor, 32 KB flashminne för lagring av programkod, 2 KB internminne och en UART som möjliggör seriell kommunikation. Plattformen för Arduino, där Uno är en implementering, är öppen och det är fritt att bygga mikrokontrollerkort enligt tillgängliga scheman. Till Arduino tillhandahålls även en tillhörande utvecklingsmiljö som möjliggör programmering via en dator och ger tillgång till exempelkod och färdigskrivna kodbibliotek. Kod till Arduinoenheter skrivs i Arduinos egen implementation av språken C och C++ [10, 11]. För fullständig specifikation, se bilaga E.1.



## 4 Genomförande

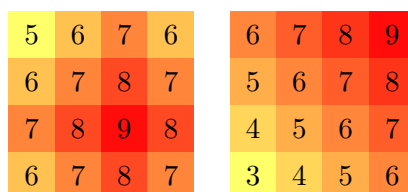
### 4.1 Fas 1

Projektet genomfördes med stöd av den valda metoden. Fas 1 innebar att skapa en överblick över projektets problem och att inhämta information till stöd för arbetet. För att initiera arbetet behövdes stegen problemidentifikation och problemförståelse genomföras, vilket i detta fall innebar möten med uppdragsgivaren i syfte att få en enhällig uppfattning om vad företaget efterfrågade och att formalisera de praktiska problem till vilka företaget sökte lösning. Vidare var det även i denna fas som introduktionen till projektets rapport utvecklades, som ett ytterligare led i problemidentifieringen och att fastslå vad projektet avsåg utföra.

Den initiala problemanalysen resulterade i att projektet i stort kom att vara uppdelat i två mindre delar, dels den algoritm som kalibrerar panelen och dels en kommunikationslösning mellan panelen och rummet som den levererar ljuset till.

Förutsättningen vid litteraturstudien, gällande kommunikationen mellan taket och byggnadens innandöme, var att den trådlösa kommunikationen skulle ske med standardiserade protokoll. Detta för att underlätta mottagandet av den trådlösa sändningen, i syfte att undvika tidssänken i felsökning då projektet hade en relativt snäv tidsram.

För kalibreringsalgoritmens del bestod problemförståelsesteget av att undersöka vilka typer av datastrukturer som skulle komma att beröras. Det insågs när problemet analyserades att de värden som samlas in kan representeras som en tvådimensionell matris (eng. 'array'), där det finns ett unikt maxvärde och kring detta minskande värden som blir lägre ju längre från maxvärdet de befinner sig, se figur 4.1.



Figur 4.1: Exempel på förväntade fokuspunkter

## 4.2 Fas 2

Metodens andra fas innebar att finna lösningar till de identifierade problemen. Möjliga förslag på lösningar diskuterades för att finna de som ansågs lämpliga att föra vidare till utveckling.

Litteraturstudien resulterade i en förståelse av att standarder för trådlös datakommunikation såsom 802.11-standarderna har problem att sända när betongkonstruktioner hindrar utspridningen av radiovågor och speciell apparatur krävs för att klara av att skicka data under sådana förhållanden [12]. Detta medför att trådlös kommunikation inte är lämplig för företaget, då de på förhand inte kan veta ifall deras kommunikation kommer att fungera på plats hos deras kunder. Inköp av nämnda apparatur är inte aktuellt.

Ett lämpligare medium att kommunicera via är istället de fiberoptiska kablar som redan är dragna, då rummet lysas upp av just dessa kablar. Enligt företaget kommer det finnas mer än en fiberkabel dragen till varje rum, vilket öppnar upp för möjligheten att koppla in apparatur för kommunikation i en fiberkabel, medan den eller de andra kablarna kan fortsätta hämta in ljus till rummet. Med de svårigheter som den trådlösa kommunikationen medförde, i kombination med att ett fungerande alternativt medium redan finns draget, valde projektet att fokusera på det senare. Vid val av kommunikationstillämpning konstaterades att panelens fiberändar har ett reflekterande skydd mot infrarött ljus, enligt 3.1.2, och att ultraviolett ljus inte leds genom panelens yttre glasskiva, vilket leder till att kommunikation över fiberkabeln måste ske med ljus i det synliga spektret.

Det framgick vid datorkörningar att en kalibreringsalgoritm som itererar över hela matrisen för att leta det högsta värdet kommer att vara väldigt ineffektiv. Ett beslut fattades om att utveckla en algoritm som kräver så få steg som möjligt, detta då den praktiska implementationen kommer att innebära fördröjningssteg vid två punkter i körningen, dels när panelen flyttar på sig, och dels när ljuset ska hämtas in från luxmätaren. Algoritmen ska kontinuerligt söka efter ett högre värde tills det maximala värdet är funnet, likt figur 4.1.

## 4.3 Fas 3

Projektet spenderade större delen av projekttiden i denna fas, då varje artefakt utvecklades genom iterationer av de tre delstegen utformning, utveckling och utvärdering. Detta innebar att utveckla en del av artefakten och sedan utvärdera delen för att undersöka hur den beter sig och vad som kan göras bättre.

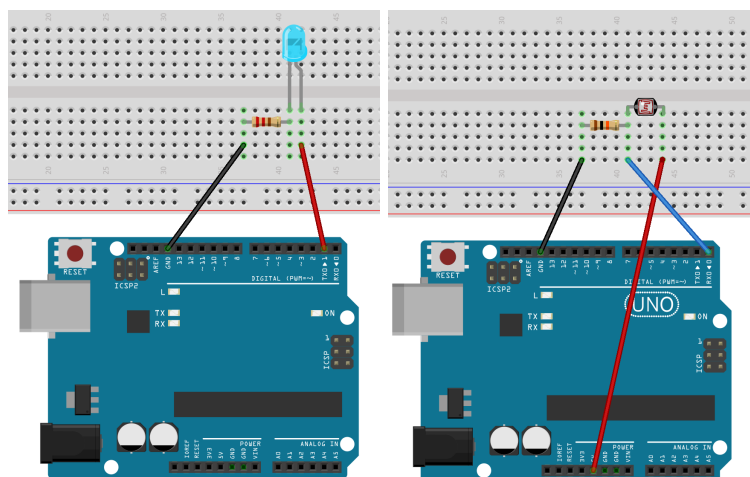
### 4.3.1 Kommunikationslösning

Kommunikationsdelen i projektet har genomgått flera iterationer av fas 3 och två huvudlösningar utformades.

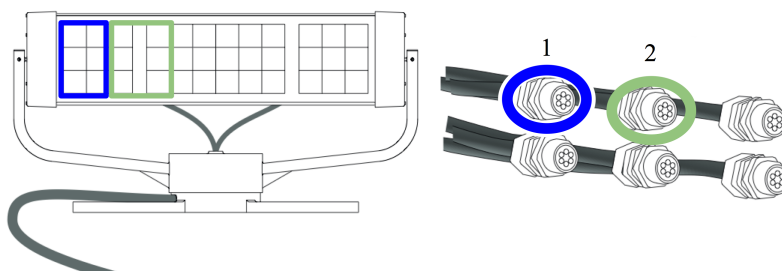
Den första lösningen bestod av två uppsättningar av mikrokontrollerkortet Arduino Uno, för en fullständig specifikation se bilaga E.1 [13]. En lysdiod kopplades till sändarens gränssnitt som skickar seriell data, vilket då omvandlade den seriella utgångens höga och låga läge till ljus på och ljus av. Till mottagaren kopplades en fotoresistor som ändrar motståndet när den träffas av ljus. När den kopplades in till det seriella gränssnittet skapade resistorn spänningsförändringar som registrerades som högt eller lågt värde av standarden. Vid försök i labbmiljö kunde data sändas med hjälp av lysdioden, driven av 5 V och 20 mA, via fiberkabel om 16 meter och där tolkas från en enskild fiber. För en enkel översikt av kopplingen se figur 4.2, kopplingsschema finns i bilaga D. Fullständig specifikation finns i bilaga E.

För att möjliggöra överföringen användes den i förhållandevis låga överföringshastigheten om 300 baud, vilket gjorde att komponenterna (fotoresistorn och lysdioden) hann ändra sina värden under den tid de förväntades göra det. Komponenterna var inte särskilt utformade för denna uppgift och innehöll vissa fördröjningar vid skifte av tillstånd.

Den data som avsågs skickas i denna kommunikation bestod av ett 16 bitars värde, det vill säga två byte. Med en överföringshastighet om 300 baud ges en teoretisk möjlighet att skicka  $\frac{300 \text{ baud}}{8 \text{ bit} \cdot 2} = 18,75$  värden per sekund. Den faktiska hastigheten blir något lägre då extra bitar skickas enligt 8N1, men överföringshastigheten var inte att anse som ett hinder i denna implementation.



Figur 4.2: Översikt av sändare/mottagare.  
Figuren skapad med hjälp av verktyget Fritzing [14]



Figur 4.3: Översikt av rundgångskoppling

Den andra lösningen använde sig endast av en luxmätare uppe på taket för att inhämta den data som efterfrågades. Genom att koppla samman två fibrer nere i rummet skickas ljuset åter upp till taket och en rundgång i systemet kan skapas. Figur 4.3 visar på linsernas samband med fiberkablarna och genom att koppla ihop kabel 1 med kabel 2 har en återkoppling skapats. Kopplingen kan lämpligen göras med en form av mutter, i fallet att kablarna befinner sig nära varandra, eller med en kopplingskabel. För att minska ljusbortfall rekommenderas lösningen med en mutter när det är möjligt, men om en kopplingskabel behöver nyttjas bör den vara kortast möjligast.

För att nyttja den här metoden behöver samtliga linser som tillhör en fiberkabel täckas över på panelen för att förhindra att dubbelt intag av solljus transporteras i de båda fiberkablarna, det är även i denna övertäckningsanordning som luxmätaren placeras. Kablarna är anpassade för att hantera den värme som produceras av det ljus som normalt transporteras vid fullt dagsljus men om den dubbla mängden skulle transporteras riskerar fibrerna att smälta på grund av värmebildningen. Lösningen använder sig inte av fiber som bärare av genererad information utan endast det solljus som panelen tar in skickas, vilket minskar både komplexiteten i systemet och risken för störningar.

### 4.3.2 Avläsning av luxmätare

De båda luxmätarna skiljer sig åt med avseende på det gränssnitt som förmedlar data. Mätaren från Adafruit kräver koppling till en mikrokontroller för tolkning av data, medan mätaren från Yoctopuce kopplas direkt till en enhet via USB, vilket underlättar anslutning till en persondator. Även om luxmätarnas gränssnitt skiljer sig åt beter sig de båda snarlikt vid användning. Båda genererar ett positivt heltal som representerar uppmätt luxvärde och skickar detta när en metod i respektives bibliotek anropas. Från en programmerares perspektiv är resultatet från och användningen av de båda mätarna i praktiken identisk.

Vid kalibrering av panelen är det av intresse att inhämta så mycket av panelens ljus som möjligt när luxvärdet ska avläsas, samtidigt som bakgrundsljus från andra eventuella ljuskällor undviks, för att skapa mätbara skillnader mellan panelens

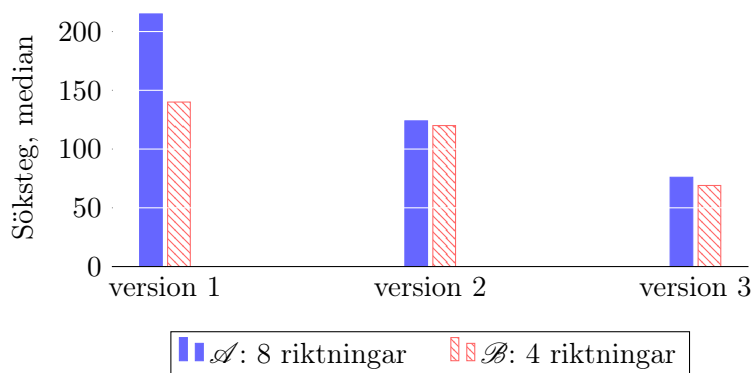
olika kalibreringssteg. Luxmätaren placeras så nära fiberändan som möjligt för att säkerställa maximalt ljusintag från den panel som ska kalibreras, vilket i praktiken för detta projekt innebär tre till fyra centimeter från fibern. För att hålla mätaren stabil används olika metoder beroende på mätare, där den ena har monterats in i ett metallhölje som skruvas fast på fiberändan, medan den andra vilar i en armatur. Att metoderna skiljer sig åt beror på olika utformning av luxmätarnas mönsterkort och leder till att avlästa värden inte direkt kan jämföras.

Testning av luxmätarnas inrapporterade värden visade att resultaten från solpanelen fluktuerar, vilket kan bero på orsaker så som att panelen ständigt justerar för att följa solen eller att något som momentant skuggar panelen passerar förbi. Dessa skiftningar i värde är inget som det mänskliga ögat uppfattar men blir synligt i mätningarna. Kompensation för detta kan implementeras med hjälp av mjukvara och projektets lösning beskrivs i avsnitt 4.3.4.

### 4.3.3 Algoritm

Utformningen av algoritmen skedde genom flera iterationer av steg tre i den valda metoden. I den första iterationen utvecklades en sökalgoritm, algoritm  $\mathcal{A}$ , som genomförde medurs sökning i åtta riktningar med motsvarande österut som utgångsriktning. När ett större värde påträffats uppdateras nuvarande position och sökningen återupprepas. För jämförelse utvecklades även en variant med fyra sökriktningar, algoritm  $\mathcal{B}$ . Efterkommande iterationer var samtliga en vidareutveckling av den förra. Den andra iterationen gav algoritmen möjlighet att lagra information om tidigare besökta koordinater, så att dessa ej undersöks vid upprepade tillfällen. Ytterligare förbättringar implementerades i den tredje iterationen, då riktningen på den senaste förflyttningen registreras om ett nytt större värde påträffats. Med hjälp av denna information undersöks först samma riktning som den senast lyckade förflyttningen innan medurs sökning återupptas. Samtliga iterationer innebar märkbara förbättringar vid simulering. Efter tredje iterationen ansågs algoritmen vara funktionsduglig, den använder betydligt färre söksteg än tidigare versioner och hittar effektivt maximipunkten. Ett flödesschema över algoritmen finns i bilaga B. Den utvecklade algoritmen kom att bli en variant av *hill-climbing* [15], vilket är en algoritm som söker upp ett lokalt maximum genom att stegvis flytta till närliggande positioner med högre värde tills inga närliggande större värden träffas på.

För att jämföra sökalgoritmerna genomfördes simuleringar som mäter antal steg från utgångspositionen till positionen där maximalt värde påträffas. Simuleringarna använde sig av 10 000 stycken  $100 \times 100$ -matriser, där varje matris hade en slumpvis genererad utgångsposition och maximalt värde. Algoritmerna genomsökte identiska matriser med samma utgångsposition. Samtliga positioners värden var strängt avtagande med avseende på avståndet till det maximala värdet. För varje iteration av algoritmen undersöktes sökning med både fyra och åtta sökriktningar. Sökning i fyra riktningar visade sig mer effektivt i samtliga fall, enligt figur 4.4 och bilaga A.



Figur 4.4: Jämförelse algoritmernas antal söksteg

Skillnaderna i antal steg mellan samma algoritm med olika antal sökriktningar vara störst i den första iterationen, med 54 % fler steg, men de visade sig även i övriga iterationer. I den tredje och slutgiltiga iterationen var skillnaden 10 % fler steg. Varje iteration av algoritmen minskade det antal steg som behövdes för att finna det största värdet. Största förändringen mellan iterationer skedde mellan iteration två och tre, där antalet steg minskades med 43 %, se tabell 1. Minskningen mellan iteration ett och tre var 51 %.

I händelse av längre tids molnighet deaktiveras ljussensorn av panelens mjukvara och kalibrering går då ej att genomföra. För att hantera detta lades ytterligare funktionalitet till i algoritmen. Om så skulle ske under pågående kalibrering återställs panelens korrigeringsvärden till de värden som var aktuella innan den automatiska kalibreringen startade. För att ytterligare motverka oförutsedda problem vid kalibreringstillfället har kontroller för timeout och korrigeringsvärdernas rimlighet implementerats, där båda kontrollerna vid utslag avbryter sökningen och korrigeringsvärdena återställs.

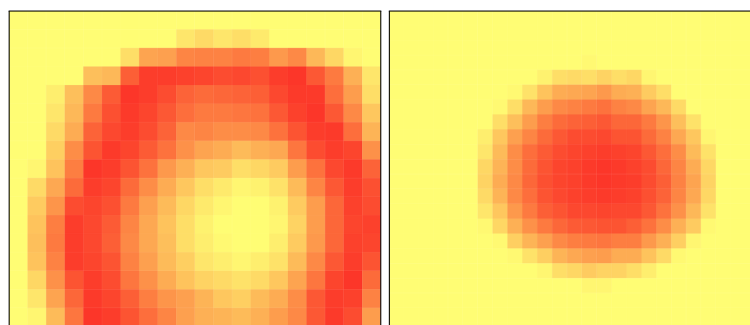
Innan algoritmen testades på solpanelen gjordes en undersökning om panelens fokuspunkt överensstämmer med projektets antagande, figur 4.1. En enkel genomsökning av solintaget genomfördes, där panelens justeringsvärden för ljussensorn stegades över ett kvadratisk fält samtidigt som ljusstyrkan avlästes och fördes in i en tvådimensionell matris. Resultatet av denna sökning gav olika bilder beroende på vilken luxmätare

Tabell 1: Minskning av antal söksteg mellan algoritmversioner

Från	Till version 2	Till version 3
Version 1	14 %	51 %
Version 2	-	43 %

Baserat på medianvärden, se bilaga A





(a) Luxmätare från Adafruit (b) Luxmätare från Yoctopuce

Figur 4.5: Uppmätt ljusstyrka

som användes. Den ena luxmätarens bild stämde inte överens med antagandet medan bilden från den andra stämde precis, vilket visas i figur 4.5. Hos Parans har tidigare efterforskningar aldrig resulterat i något liknande figur 4.5a och avvikelsen antas bero på luxmätaren Adafruit TSL2591. Att förlita sig till denna mätning skulle ge ett lägre ljusflöde från panelen då dess maximipunkt ligger i utkanten av äkta maximum. Projektet beslutade att endast använda den luxmätare som genererade förväntad fokuspunkt, Yoctopuce Yocto-Light-V3. Fullständig data från sökningen finns bifogad i bilaga F.

#### 4.3.4 Applikation

Efter att ha implementerat en algoritm som i simuleringar uppfyllde önskvärd funktion påbörjades utveckling av en applikation där den kan tillämpas. Förutom själva kalibreringsalgoritmen skulle applikationen ha egenskaper som kommunikation med panelen, ett användarvänligt grafiskt gränssnitt och inhämtning av värden från luxmätare. Programmeringsspråket som valdes för utveckling av applikationen var **Python** och valet hade stöd av flera motiveringar. Den första anledningen till att språket valdes var att företaget har kompetens och erfarenhet att utveckla i detta språk, då den nya panelen SP4 kommer att drivas av en källkod skriven i just **Python**. Att kompetens och kunskap finns inom företaget medför att applikationen kan underhållas av företaget själv och modifieras vid framtida behov. Vidare har företaget sedan tidigare produkter utvecklade i detta programmeringsspråk, så att utveckla applikationen i samma språk underlättar för en eventuell implementering av mjukvaran i de existerande produkterna. Mer generella fördelar med att utveckla i **Python** är att språket är plattformsoberoende och är enkelt att utveckla grafiska gränssnitt i. Nackdelar med språket är att det är långsamt i förhållande till andra språk såsom **C** eller **Java**. Då applikationen kommer att kommunicera med mekanik, vilket leder till flertalet fördröjningar för rotation av panelen och inhämtning och sändning av information, anser projektet att språkets åverkan på applikationens snabbhet är försumbar [16].

För att kommunicera med panelen och den Arduino som mottar värden från luxmätaren Adafruit TSL2591 behövdes stöd för seriell kommunikation. Ett bibliotek som möjliggjorde seriell kommunikation med enheter anslutna till en dator var pySerial. Eftersom pySerial har stöd för operativsystem baserade på Windows, Linux och BSD är biblioteket i praktiken plattformsoberoende [17]. Kodbibliotek till luxmätaren Yocto-Light-V3, som anslöts direkt till den dator som kör applikationen, fanns att tillgå från tillverkaren Yoctopuce.

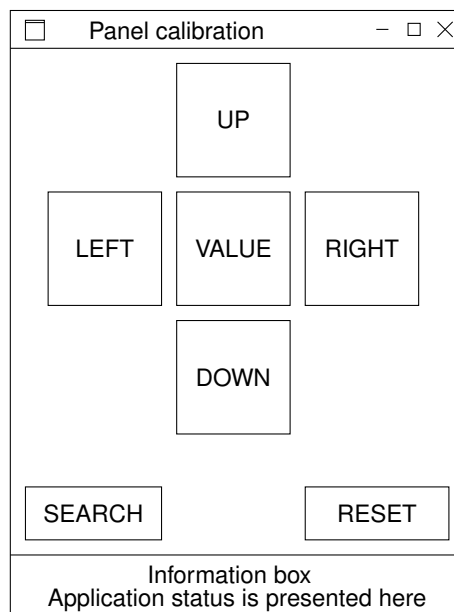
Till det grafiska gränssnittet fanns flera tillgängliga ramverk som var utvecklade för och anpassade till Python. Det som ansågs mest lämpligt för applikation var TkInter, där en anledning var att Parans sedan tidigare hade kringutrustning som använder sig av detta [18]. Hänsyn togs till att för bolaget underlätta både framtida hantering av koden och en eventuell integrering av tidigare produkter med den av projektet utvecklade applikationen. En annan anledning var att ramverket distribueras med standardinstallationer av Python till Windows och Mac OS X [19]. På Linux-system behövs TkInter installeras separat och det finns tillgängligt i flera stora Linuxdistributioners pakethanterare. Behovet av extra installationer för ett ramverk kunde minimeras och det grafiska gränssnittet kunde göras plattformsoberoende.

För att underlätta framtida hantering av den producerade koden användes en objekt-orienterad design av applikationen, där hög kohesion och låg koppling mellan klasserna eftersträvades [20]. Inledningsvis isolerades den framtagna algoritmen i en egen klass, **Search**. Anslutningarna till panelen och de olika luxmätarna implementerades i separata klasser med specifik kod för att hantera varje enskild enhet. För att hantera de olika anslutningarna skapades klassen **SerialHandler** som ett mellanliggande gränssnitt till applikationens övriga komponenter. För att kompensera för de fluktuerande värden som beskrevs i avsnitt 4.3.2 hämtar **SerialHandler** fyra luxvärden från en ansluten mätare och medelvärdet av de två mellersta rapporteras vidare för att få en bättre uppskattning vid kalibreringen. Det grafiska gränssnittet hanteras av klassen **GUI** och där skapades tryckknappar för att aktivera applikationens funktioner och ett fält för återkoppling och presentation av information. De tre klasserna **Search**, **SerialHandler** och **GUI** fick tillsammans utgöra grundstommen i applikationen och utformades för att vara plattformsoberoende. Genom att sökalgoritmen och det grafiska gränssnittet endast använder sig av det mellanliggande gränssnittet i **SerialHandler** och ej de underliggande anslutningarna uppnåddes en mer modulär design och ett oberoende av de implementationsspecifika delarna av applikationen. En översikt av klasserna finns i bilaga C.

Då applikationen är beroende av anslutning till externa enheter lades funktionalitet till för att underlätta anslutningen. Applikationen kan användas på olika plattformar och olika förfaranden framställdes för olika operativsystem. Stöd för automatisk identifiering av ansluten panel och Arduino implementerades till Mac OS X och Linux. Förutsättningen för detta var att inga fler externa enheter av samma typ är anslutna till den aktuella datorn. För Windowssystem utvecklades en dialogruta

som vid applikationsstart frågar användaren efter de aktuella COM-portarna för respektive enhet. Om ingen Arduino är tillkopplad försöker applikationen använda sig av Yocto-Lux-V3.

Det grafiska gränssnittet gavs en enkel design och baserades på det utseende som används i företagets kringutrustning [18]. Nämnade kringutrustning var utformad för att kunna användas med pekskärm och således beaktades detta även här. Fyra knappar i form av ett styrkorset lades till för att manuellt ändra panelens justeringsvärden för ljussensorn i x- och y-led. I mitten av styrkorset placerades en knapp för manuell avläsning av den anslutna luxmätaren. Nedanför styrkorset placerades en knapp som startar automatisk kalibrering och en knapp som återställer panelens justeringsvärden till de som var aktuella vid applikationens starttillfälle. Ett fält som kan förmedla aktuell information placerades i applikationens underkant. Det grafiska gränssnittet är avbildat i figur 4.6.



Figur 4.6: Skiss av det grafiska gränssnittet

#### 4.4 Fas 4

För detta projekt innebar fas fyra att skriva denna rapport, vilket förtydligar och sammanfattar resultaten som har uppnåtts under de genomförda iterationerna. Vidare hålls en presentation av resultatet inom ramen för den kurs som genomförs, vilket även är en del av metoden.



## 5 Resultat

### 5.1 Algoritm

För att möjliggöra automatisk kalibrering av panelens ljussensor har projektet utvecklat en sökalgoritm med positionsregistrering och stegreducering. Den utvecklade algoritmen gav svar på den första av projektets tre frågeställningar, "Vilken algoritm kan anses vara lämplig för kalibreringen?". Algoritmen uppsöker ett lokalt maximum i ljusstyrka, beskrivet i avsnitt 4.3.3, och kan anses vara en variant av algoritmtypen *hill climbing* [15].

Algoritmen söker stegvis efter det maximala inlästa värdet tills inga kringliggande större värden påträffas. I varje söksteg justeras panelens korrigeringsvärde för ljussensorn, vilket får panelen att vrida sig till den position som ger solljusets fokus i ljussensors korrigerade mittpunkt. Sökning sker i fyra riktningar, representerade av väderstrecksuttryck motsvarande den koordinatsystemsrepresentation panelen har för korrigeringsvärden där positiva x och y är öst respektive nord, och sker medurs med utgångsriktning österut. Om ett lika stort eller större värde avläses efter en vridning av panelen så kommer nästkommande undersökta position vara i samma riktning som den senast utförda, då avlästa värden antas vara kontinuerligt fallande från maximipunkten. Detta i enlighet med initialt antagande, representerat i figur 4.1, och senare undersökning, enligt figur 4.5b. Algoritmen registrerar besökta positioner så att samma position ej undersöks upprepade gånger. Den utvecklade algoritmen fanns lämplig för ändamålet och anses vara passande då sökområdet förväntas vara konvext [15]. Flödesschema för algoritmen finnes i bilaga B.

Tillgången till kontinuerligt solljus är en förutsättning för kalibrering av enheter tagna i bruk då variationer i molnighet markant påverkar ljusintensiteten och således det avlästa värdet. Då ljussensorn deaktiveras vid molnighet så implementerades funktionalitet i algoritmen för att hantera uppkomna störningar.

### 5.2 Optisk kommunikation

För att svara upp mot målet att ta fram en lösning för "kommunikation mellan en luxmätare inne i byggnaden och en panel som befinner sig på taket" besvaras i efterföljande frågeställningar för förutsättningar för kommunikation och hur tillförlitliga dessa lösningar kan anses vara.

#### 5.2.1 Förutsättningar

Frågeställningen "[v]ilka förutsättningar för kommunikation finns mellan solpanelen och det upplysta rummet" har resulterat i en undersökning som visade att trådlös kommunikation inte är att anse som lämplig, utan den redan dragna optiska fiberkabeln är det kommunikationsmedia som bör nyttjas. Projektet föreslog två metoder för att nyttja fibern som databärare, där en metod innebar att skicka asynkron seriell

data och en annan metod att två optiska fibrer kopplas samman för att returnera ljusintaget genom panelens linser och där uppmäta ljusstyrkan.

Den första lösningen som projektet föreslog var att en mikrokontroller inne i det upplysta rummet omvandlar utdata från en luxmätare till en optisk signal som sedan sänds seriellt upp till panelen, för att där avkodas av en andra mottagande mikrokontroller. När signalerna skickas via fibrerna strålas ljuset ut ur solpanelens linser, vilket mottagaren då kan analysera. Mottagaren är monterad på panelen och har en ljuskänslig sensor som omvandlar de optiska signalerna till digitala. De digitala signalerna skickas från mottagaren vidare till den enhet som utför den algoritm som är avsedd att kalibrera ljussensorn.

Den andra lösningen var att, istället för att mäta upp ljusstyrkan i rummet, koppla ihop två stycken optiska fibrer från samma panel i det rum de är avsedda att upplysa, vilket då skickar ljusintaget tillbaka upp till panelen. Genom att täcka över de linser som förser den ena fiberkabeln med ljus kommer den andra fiberkabeln att skicka ut sitt ljusintag mot de nu täckta linserna. I detta förslag kan en luxmätare placeras i övertäckningsanordningen och där, via omvägen till det upplysta rummet och tillbaka, mäta upp hur mycket ljus panelen tar emot. Då luxmätaren nu befinner sig på panelen kan den direkt skicka sin data till den enhet som förväntas utföra algoritmen.

### 5.2.2 Tillförlitlighet

För att nå ett svar på frågan "[h]ur tillförlitligt är det valda kommunikationssättet?" utreddes flera alternativ för att sända data. Genom att använda de fördragna fiber-optiska kablarna säkerställs att det ljus som skickas från rummet upp till panelen alltid kommer att levereras, förutsatt att ljuskällan är tillräckligt stark. Detta i skarp kontrast mot en trådlös lösning där flera lager betong mellan rummet och panelen inte är en osannolik företeelse, vilket då skulle resultera att signalerna under normala förutsättningar aldrig når panelen.

Huruvida de två föreslagna lösningarna för ljussändning är tillförlitliga beror på hur ljuset läses av uppe vid panelen. I förslaget med två mikrokontroller, där data sänds seriellt, är tillförlitligheten lägre. Lyckade försök har genomförts i labbmiljö, men lösningen är känslig för störningar från bakgrundsljus så en tillförlitlig fästningsanordning för panelen behöver tillverkas. Varje fiberkabel består av sex stycken fibrer där varje fiber är kopplad till en egen lins för att fokusera in solljuset. När data nu skickas nedifrån och upp kommer linsen att agera omvänt genom att omvandla fiberns fokuserade ljus till parallella strålar. Då ljuset nu är parallellt istället för fokuserat finns en risk att skillnaden i ljusintensitet mellan hög och låg inte är tillräckligt stor för att registreras av fotosensorn eller att motståndet inte ändras tillräckligt mycket. För att ha möjlighet att registrera förändringarna av ljusstyrka hade det varit idealt att ha en fokuseringslins till fotoresistorn och att resistorn hade varit innesluten i någon form av behållare som fästs över de sex linserna.

Det andra förslaget, där två fiberkablar kopplas samman för att skapa rundgång, är tillförlitligheten högre. Ingen behandlad data behöver kommuniceras utan endast rådata i form av ljusstyrka skickas ut genom linserna. Luxmätaren är inte beroende av snabba ändringar i ljuset, utan ljuset är konstant vilket leder till en högre tillförlitlighet. Ljusstyrkan kommer vara lägre när den kommer upp till panelen jämfört med om den skulle stråla ut i rummet, då den behöver färdas dubbelt så långt i fiberkabeln, men mätningen av ljusstyrkan är inte beroende av ett korrekt absolut värde. När panelen kalibreras är det istället av intresse att finna det högsta relativa värdet, det värde som uppmäts när panelen tar in mest ljus, då det är till det högsta värdet som panelen ska vara kalibrerad. Detta medför också att denna metod är mindre känslig för bakgrundsljus, så länge bakgrundsljuset är konstant, eftersom skillnaden i det utstrålade ljuset ändå kan registreras.

För att kontrollera vilket luxvärde panelen faktiskt levererar till rummet behöver mer kvalificerad utrustning användas, utrustning som är kalibrerad och granskad för att mäta luxvärden i inomhusmiljö. Det är apparatur som företaget har tillgängligt men som ligger utanför detta projekt.

### 5.3 Applikation

Den framtagna applikationen gav en förutsättning att uppfylla målet ”att ta fram en produkt som justerar fokuspunkten på ljussensorn”. En automatisk kalibrering av panelens korrigeringsvärde för ljussensorn kan utföras genom att applikationen har kopplat samman sökalgoritmen med solpanelen och en luxmätare.

Sökalgoritmen implementerades i form av en Pythonapplikation med grafiskt gränssnitt och applikationen stödjer inhämtning av ljusvärden från en luxmätare, där värden förmedlas genom antingen direkt anslutning till datorn eller seriell kommunikation från en annan enhet. Den luxmätare som användes vid implementationen av direkt anslutning var Yocto-Light-V3 medan Adafruit TSL2591 användes för avläsning som överfördes seriellt via en Arduino Uno. För seriell kommunikation använder applikationen sig av pySerial, ett bibliotek som kan hantera seriell kommunikation på de flesta vanligt förekommande operativsystem [17].

Hos Parans fanns sedan tidigare kringutrustning som använder en Pythonapplikation med grafiskt gränssnitt anpassat till en pekskärm på en Raspberry Pi, där det grafiska gränssnittet är implementerat med ramverket TkInter [18]. Samma ramverk och grafiska formgivning har använts till den applikation som utvecklats av projektet. Detta upplägg är tänkt att underlätta framtida hantering och utveckling av applikationerna och möjliggör en eventuell framtida integrering av de båda.

Applikationen är utvecklad enligt en objektorienterad utvecklingsmodell och nya avläsningsmetoder kan implementeras utan större ingrepp i befintlig kod. För en översikt av källkodens struktur se UML diagrammen i bilaga C. Mjukvaran till Parans kommande solpanel, SP4, var ej färdigställd under projektets gång och således är applikationen riktad till SP3. Implementeringen av sökalgoritmen kan återanvändas till kommande versioner av solpaneler men vissa anpassningar kan behövas.



## 6 Slutsats

### 6.1 Sammanfattning

Detta projekt har kommit fram till att en automatiserad metod för att kalibrera Parans solpanel SP3 är möjlig och levererar en mjukvara som utför just denna uppgift. Krav för att kalibreringsmetoden ska fungera är att rätt hårdvara finns att tillgå, där kalibreringen i sig kräver en luxmätare för avläsning och intag av belysningsgrad och en kommunikation till panelen SP3.

För att kommunicera mellan panelen och det upplysta rummet har projektet lämnat två huvudsakliga förslag, dessa är dock inte lika långt utvecklade såsom algoritmen. Mer arbete behövs på den fysiska delen av kommunikationen där framför allt en fästordning på panelen för att fästa en ljuskänslig mottagare behöver utvecklas. Även sändare och mottagare behöver designas, då dessa är i prototypstadiet med kopplingar på kopplingsbrädor. Teorin bakom kommunikationen är dock utredd och kommunikation har upprättats i laborationsmiljö via företagets eget kablage, så vidare utveckling av denna kommunikationsprincip är möjlig.

### 6.2 Diskussion

Projektet kan anses ha två huvudsakliga syften, där det första är att ”möjliggöra en helt automatisk process som kan kalibrera fotosensorn i Parans solpaneler” och att ”minska tidsåtgången och höja precisionen jämfört med dagens manuella metod”. Den metod som bolaget tidigare använt sig av var dels baserad på manuell inmatning av värden, vilket tar tid och kan leda till fel på grund av den mänskliga faktorn, och dels på en manuell uppskattning av ljusstyrkan vilket också kan leda till en felaktig kalibrering. Med hjälp av den algoritm som projektet har utvecklat och redovisat, i samverkan med den tekniska implementationen, anser författarna att detta syfte är uppnått. Processen kan skötas helt automatiskt under förutsättning att ljusflödet ut från panelen kan uppmätas. Denna automatiserade kalibrering är att anse som tidsbesparande då inga värden behöver anges manuellt, särskilt då skillnaden mellan sensors ursprungliga värde och det optimala inställningsvärdet och är stort så att många kalibreringssteg behöver göras. Den framtagna algoritmen är generell i det avseende att den inte är begränsad till användning för SP3-panelen utan bör även kunna användas för andra liknande produkter, såsom bolagets efterföljande modell SP4, eller tillämpningar där datastrukturer likt de illustrerade i figur 4.1 kan påträffas.

Anledningen till att projektet inte fortsatte vidare i utvecklingen av algoritmen efter den tredje iterationen är förutsättningen att paneler satta tagna i bruk bör vara relativt kalibrerad från fabrik. Det vill säga att maxpunkten förväntas befinna sig nära nuvarande inställning, där en sökning av ett område större än  $20 \times 20$  söksteg ses som osannolik. Detta baserat både på de fokuspunkter som uppmätts i bilaga F och på uppgift från utvecklare inom företaget. Denna grovkalibrering innebär att fortsatt förbättringsarbete inte skulle ge ett stort utslag på tidsbesparing, men skulle denna algoritm användas till helt okalibrerade sensorer kan vidare utveckling vara nödvändig. Exempel på vidare optimering kan vara större söksteg först, för att grovt söka igenom ljusbilden, innan en finare kalibrering vidtar.

Gällande resultat från körningar av algoritmen saknas utförlig sådan. Författarna är medvetna om att resultat gällande bland annat söktider hade varit lämpliga att redovisa i rapporten, men på grund av de väderförhållanden som rått under projekttiden har inga sådana resultat kunnat framställas. De enstaka dagar med ihållande solsken har ägnats åt att framställa de resultat som visas i kapitel 5 och att utföra enstaka tester som visade på att algoritmen fungerar som tänkt.

Gällande bestämningen av ljusintensiteten finns det både för- och nackdelar med att göra en uppmätning av ljusstyrkan och en mänsklig uppskattning. Fördelarna med en automatiserad inläsning är att kalibreringen blir standardiserad och inte behöver bero på den person som utför kalibreringen. När författarna deltog i en manuell kalibrering av installerade paneler ute i produktion upplevde vi att ljusintensiteten varierar väldigt mycket, från bländande till i princip helt släckt och med tanke på att det mänskliga ögats anpassning till olika ljusintensiteter varierar beroende på om intensiteten ökar eller minskar kan kalibreringen tappa i precision vid en manuell bedömning [21, s. 273]. Det är svårt att jämföra hur två inställningar förhåller sig till varandra, vilken som är starkare eller svagare, om ljuskällan blivit väldigt mörk mellan de båda tillfällena.

Ett problem som kan uppstå är ifall projektets luxmätare är mer känslig för andra frekvenser än de frekvenser som det mänskliga ögat är känsligt för och om panelen vid felkalibrering tar in ett högre antal av de för luxmätarens känsliga frekvenser på grund av brytning vid linsen, så skulle mätaren registrera ett högre belysningsvärde än vad en människa skulle anse. Denna spektrumförskjutning kan vara en delförklaring till den avvikande fokuspunkten i figur 4.5a, där luxmätaren till synes ger större utslag för vissa frekvenser som ligger strax utanför fokuspunkten. Risken att luxmätarens utslag beror på aningen felaktiga ljusfrekvenser är något som projektet har accepterat, med stöd av litteratur. Litteraturen beskriver att ljusmiljö är komplext att bedöma och kräver personer med erfarenhet för att bedömas korrekt, personer som företaget inte har att tillgå vid paneler redan satta i drift [21, s. 278]. De kalibreringar som har genomförts i testmiljön har inte kunnat visa på att fel frekvenser skulle leda till en lägre upplevd belysningsgrad. Oavsett den mån teknikern skulle kunna bedöma belysningsgraden så finns vanligtvis inte möjlighet för denna att befinna sig i rummet

dit ljuset leder då teknikern befinner sig vid panelen för att sköta kalibreringen. Detta leder då antingen till att teknikern behöver gå emellan panelen och det upplysta rummet, något som är väldigt tidsödande, eller att det krävs två personer för att utföra kalibreringen, en som sköter inmatningen till panelen och en som rapporterar ljusstyrkan. Sammantaget är vår bedömning att en uppmätning av ljusstyrkan är en lämplig metod då det sparar tid vid kalibreringen och resultatet blir oberoende av operatörers erfarenhet gällande bedömning av ljusintensitet.

Det bör dock påpekas att den ljusuppmätning som detta projekt utför endast är till för att hitta det maximala ljuset ut från panelen och ska inte tas för en ljusmätning som kan representera belysningen i rummet. För att mäta upp belysningen av rummet behövs mer avancerade mätanordningar som kan ge korrekta värden.

Det andra huvudsakliga syftet till projektet är att ”möjliggöra kommunikation mellan panelen och en luxmätare inne i byggnaden” där vi har undersökt två huvudsakliga metoder, trådlöst eller via de fiberoptiska kablarna som redan är dragna. Kommunikation med standardiserade trådlösa metoder för persondatorer utesluts tidigt i arbetet, i och med att hinder i form av betongkonstruktioner såsom golv, väggar och tak befinner sig emellan panelen och luxmätaren, vilket hindrar spridningen av radiovågor och överföringen av data.

För optisk kommunikation lämnar författarna dels ett förslag för att lösa detta delproblem så som det är formulerat, genom att upprätta en datakommunikation mellan rummet och panelen, men även ett annat förslag lämnas som möjliggör att flytta luxmätaren från rummet upp till panelen. Det andra förslaget ointetgör då behovet av att skicka datan från luxmätaren, eftersom den finns tillgänglig för den beräkningsenhet som utför algoritmen.

Detta syfte är delvis uppnått då information kring hur dessa två förslag kan realiserats har redovisats, dock saknas viss materiel för att kunna testa förslagen. Fotoresistorn som nyttjas vid mottagande av data behöver en infästning till panelen som är ljusskyddad samt eventuellt en lins för att fokuserar det utsända ljuset till resistorn. Denna infästning behöver tillverkas, men med avseende på projektets avgränsningar så anser vi att denna tillverkning ligger utanför projektets ramar. Är det möjligt att skicka data optiskt i det kablage som företaget använder i produktion? Ja det har bevisats möjligt, men har inte kunnat testas på deras produkt utan endast via kablaget.

Ur ett hållbarhetsperspektiv har projektet arbetat med att förbättra en produkt, vars syfte är att höja ljus kvaliteten inomhus genom att leda in fullspektrumljus från solen till platser i byggnader där solen inte kan stråla in [22]. Ytterligare en aspekt av produkten är att energiförbrukningen för upplysning sänks [23], vilket leder till lägre kostnader och minskad klimatåverkan från de platser där panelen är installerad.

### 6.3 Arbetsgång

Den tidsplan som togs fram innan projektets start har i huvudsak följts och finns bifogad i bilaga G. Vad vi i efterhand kunde se var att inledande utveckling av algoritmen påbörjades en vecka innan planerad utvecklingsstart, om än i väldigt skissartad form. Detta berodde dels på ett behov av att fördjupa sig i programmeringsspråket `Python`, av vilket tidigare erfarenhet var begränsad, och dels på att formulera tidiga tankar och idéer i kod. Det kan också nämnas att fokus skiftade till rapportskrivning och utvärdering en vecka tidigare än planerat. Vid detta tillfälle hade kalibreringsapplikationen visat önskvärd funktion och projektets huvudmål hade nåtts. Då detta påskyndade dokumentationen gavs ytterligare tid till utvärdering och uppföljning.

Författarna anser att den metod som projektet har nyttjat sig av har fungerat väl. Att fokusera på ett itererande sätt, där en artefakt kunde förbättras genom att utvärdera den efter varje förbättring, var gynnsamt och ledde till en naturlig utveckling av produkten. När det kommer till utvecklingen av kommunikationen och prioriteringen av vilken artefakt som borde ha utvecklats först kan beslutet kritiseras. Projektet valde att inrikta sig på seriell kommunikation för att undersöka om det fungerade via optisk överföring, när en frekvensmodulerad standard hade kunnat vara mer lämplig. Detta berodde delvis på idéer från företaget gällande vilken lösning de såg som intressant att testa men även på bristande förkunskap från projektet. Förslaget att testa frekvensmodulerad överföring kom under en pågående utvecklingsiteration av den seriella kommunikationen och när nästa utvecklingssteg skulle genomföras hade inköp och leverans av komponenter tagit för lång tid samt medfört en extra kostnad, vilket gjorde att den lösningen inte utvecklades. Istället för att försöka skicka data framfördes förslaget om att skapa en rundgång i systemet, vilket då nästa utvecklingssteg fokuserade på, på grund av lägre komplexitet och materialkostnad.

### 6.4 Vidareutveckling

Det område som framförallt behöver vidareutvecklas från detta projekt är kommunikationsmöjligheten över optisk fiber. Författarna rekommenderar en undersökning i hur en fästianordning till solpanelen kan utformas på lämpligast sätt för att minska bakgrundsljus, samt hur mottagaren och sändaren kan utformas för att lämna prototypstadiet. Dessa förslag är att anses som utanför institutionen för data- och informationsteknik och bör vara mer lämpligt för ett designinriktat examensarbete.

Vidare är en undersökning av modulerad sändning av ljus i det synliga spektret av intresse. Detta projekt har kännedom om grunderna men det har inte undersökts närmre på grund av avsaknad av materiel och tidsutrymme.

## Referenser

- [1] A. Dresch, D. P. Lacerda och J. A. V. Antunes. *Design science research: a method for science and technology advancement*. Springer Verlag, 2015. ISBN: 3319073737; 9783319073736.
- [2] C. Sims och H. L. Johnson. *The elements of Scrum*. Foster City, Calif.: Dymaxis, 2011. ISBN: 0982866917; 9780982866917.
- [3] *Parans Installation manual SP3*. 2013/06. Parans Solar Lighting AB. 2013. URL: <http://parans.com/eng/customerservice/documents/ParansInstallationManual.pdf> (hämtad 2015-03-27).
- [4] D. Ibrahim. "Chapter 2 - PIC32 Microcontroller Series". I: *Designing Embedded Systems with 32-bit PIC Microcontrollers and MikroC*. Utg. av D. Ibrahim. Oxford: Newnes, 2014, s. 41–90. ISBN: 978-0-08-097786-7. DOI: 10.1016/B978-0-08-097786-7.00002-6. URL: <http://www.sciencedirect.com/science/article/pii/B9780080977867000026>.
- [5] *Product Specifications*. Parans Solar Lighting AB. 2015. URL: [http://www.parans.com/eng/customerservice/documents/ParansProductSpecifications\\_web.pdf](http://www.parans.com/eng/customerservice/documents/ParansProductSpecifications_web.pdf) (hämtad 2015-03-30).
- [6] *Two-dimensional PSD*. Hamamatsu Photonics K.K. 2013. URL: [http://www.hamamatsu.com/resources/pdf/ssd/s5990-01\\_etc\\_kpsd1010e.pdf](http://www.hamamatsu.com/resources/pdf/ssd/s5990-01_etc_kpsd1010e.pdf) (hämtad 2015-03-31).
- [7] *TSL2591 Datasheet Apr. 2013 ams163.5*. ams AG. 2015. URL: [http://www.adafruit.com/datasheets/TSL25911\\_Datasheet\\_EN\\_v1.pdf](http://www.adafruit.com/datasheets/TSL25911_Datasheet_EN_v1.pdf) (hämtad 2015-05-12).
- [8] *Yocto-Light-V3, User's guide*. Yoctopuce. 2015. URL: <http://www.yoctopuce.com/projects/yoctolightV3/LIGHTMK3.usermanual-EN.pdf> (hämtad 2015-05-12).
- [9] D. Wheat. *Arduino Internals*. Apress, 2011. Kap. 1. ISBN: 9781430238829; 1430238828.
- [10] J. Purdum. *Beginning C for Arduino: learn C programming for the Arduino and compatible microcontrollers*. Apress, 2012. Kap. 2. ISBN: 9781430247777; 9781430247760.
- [11] *Arduino Build Process*. Arduino. 2015. URL: <http://www.arduino.cc/en/Hacking/BuildProcess> (hämtad 2015-05-11).

- [12] S. Mizushina och A. Adachi. "A technique for wireless LAN connection through building concrete wall at 2.4GHz". I: *Microwave Symposium Digest (MTT), 2010 IEEE MTT-S International*. Maj 2010, s. 260–263. DOI: 10.1109/MWSYM.2010.5514751.
- [13] *Arduino Uno*. Arduino. 2015. URL: <http://www.arduino.cc/en/Main/ArduinoBoardUno> (hämtad 2015-05-05).
- [14] *Fritzing*. Version 0.9.2b. Interaction Design Lab Potsdam. URL: <http://fritzing.org> (hämtad 2015-06-08).
- [15] S. S. Skiena. *The Algorithm Design Manual*. New York; London: Springer, 2008, s. 251–253. ISBN: 9781848000698; 9781848000704. DOI: 10.1007/978-1-84800-070-4.
- [16] R. Power och A. Rubinsteyn. "How fast can we make interpreted Python?" I: *CoRR* abs/1306.6047 (2013). URL: <http://arxiv.org/abs/1306.6047v2> (hämtad 2015-05-12).
- [17] C. Liechti. *pySerial*. 2015. URL: <http://pyserial.sourceforge.net/pyserial.html> (hämtad 2015-05-13).
- [18] J. Strand, P. Svedberg och O. Åkergren. *Styrdosa till Parans solpanel*. 2015. URL: <https://github.com/MrAkergren/SolarRemote/blob/master/doc/thesis.pdf> (hämtad 2015-05-13).
- [19] *How to install Tkinter*. Tkinter Wiki. 2015. URL: [http://tkinter.unpythonic.net/wiki/How\\_to\\_install\\_Tkinter](http://tkinter.unpythonic.net/wiki/How_to_install_Tkinter) (hämtad 2015-05-19).
- [20] D. Barnes och M. Kölling. *Objects first with Java: a practical introduction using BlueJ*. Boston, [Mass.]; London: Pearson, 2012, s. 200. ISBN: 0132835541; 9780132835541.
- [21] M. Bohgard. *Arbete och teknik på människans villkor*. Upplaga 2. Stockholm: Prevent, 2011. ISBN: 9173651109; 9789173651103.
- [22] L. Edwards och P. Torcellini. *Literature Review of the Effects of Natural Light on Building Occupants*. Tekn. rapport. 2002, s. 3–4.
- [23] *Kräver Paranssystemet elektricitet?* Parans Solar Lighting AB. URL: <http://parans.com/swe/customerservice/faq.cfm> (hämtad 2015-06-10).

## Bilagor

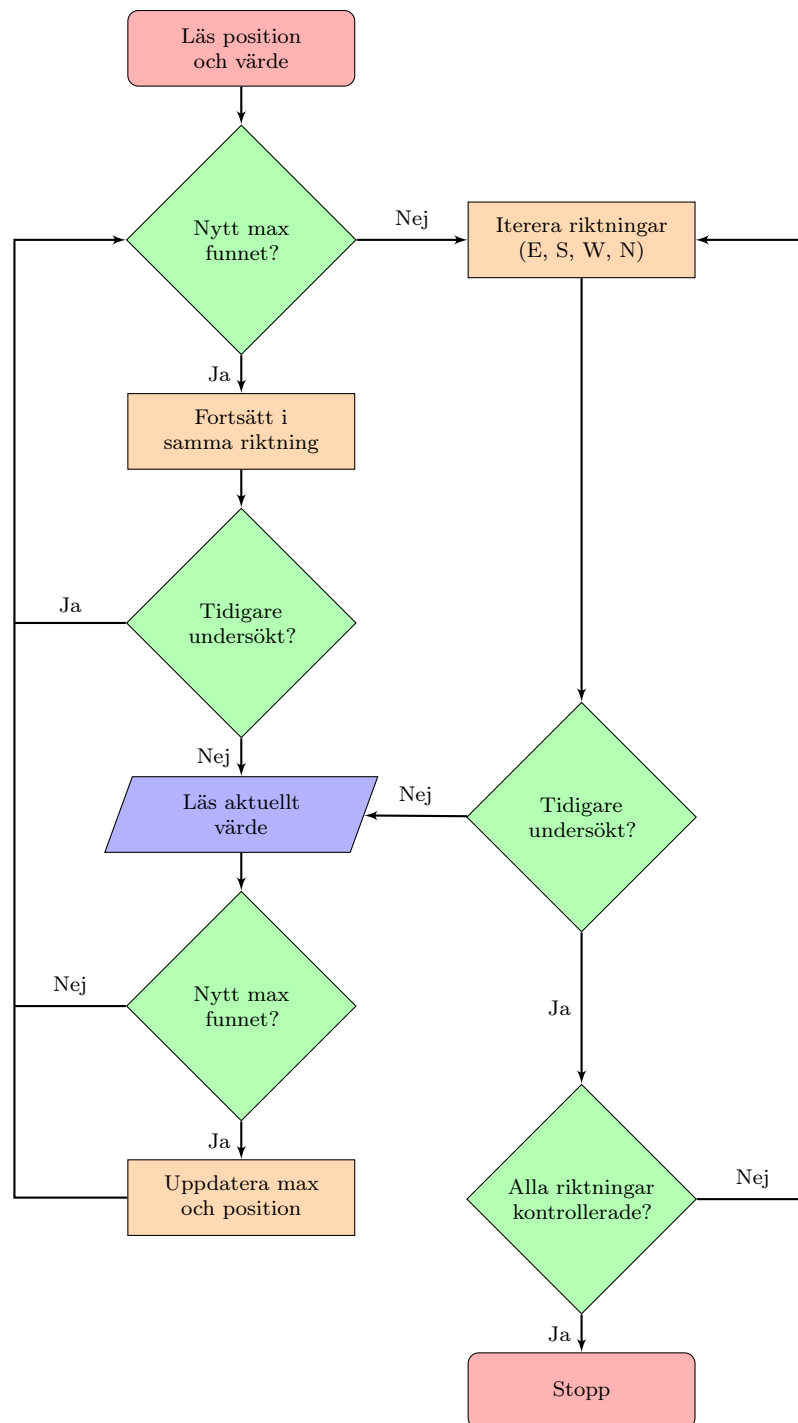
### A Simuleringsresultat av kalibreringsalgoritm

Steg	$\mathcal{A}$ : 8 riktningar			$\mathcal{B}$ : 4 riktningar		
	v.1	v.2	v3	v.1	v.2	v3
Medel	255,7	134,2	78,8	160,2	130,5	72,0
Median	215	124	76	140	120	69
Max	1065	450	195	621	448	186
Min	1	7	9	1	4	5
$\sigma$	176,3	73,5	33,1	101,1	73,4	32,9

Varje version implementerar ny funktionalitet:

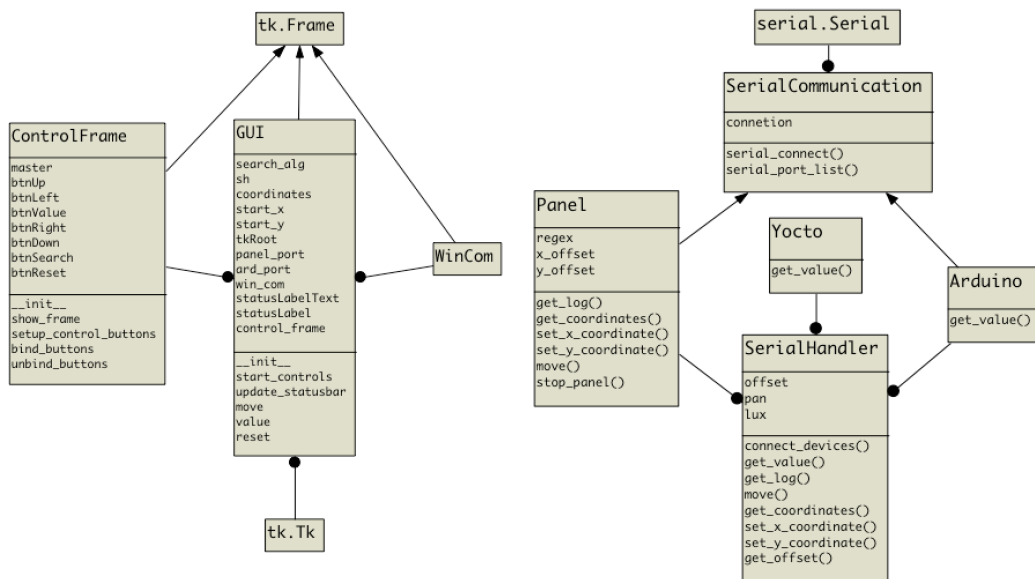
1. Söker medurs i åtta alternativt fyra riktningar.
2. Testa ej redan kontrollerade koordinater.
3. Testa först den riktning som senast lyckad förflyttning.

## B Flödesschema av kalibreringsalgoritmen



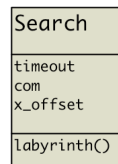


## C UML-Diagram



(a) Översikt av det grafiska gränssnittet

(b) Översikt av den seriella kommunikationen



(c) Klassen som håller kalibreringsalgoritmen. Ärver inte av andra

```

# Main.py
from search import Search
from serial_handler import SerialHandler
from gui.gui import GUI
import sys

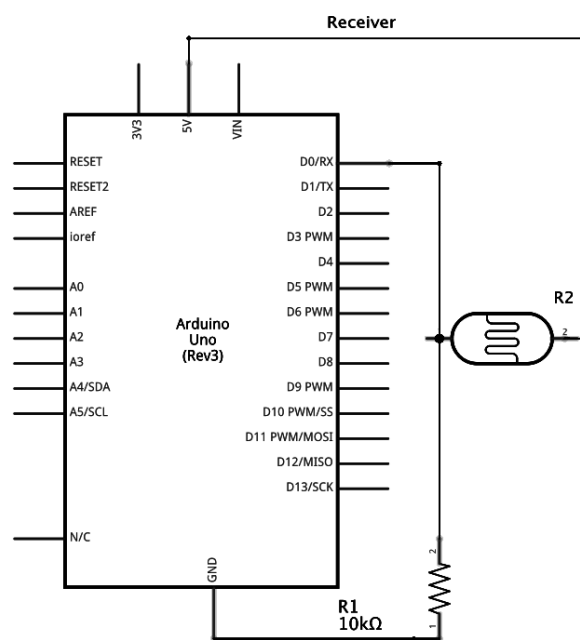
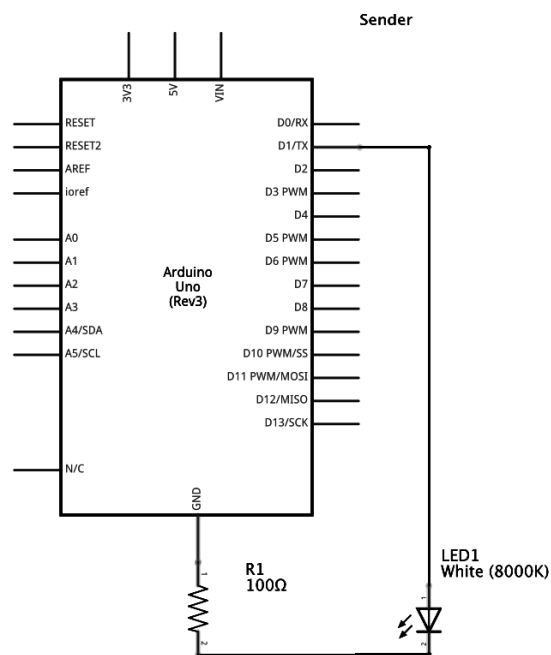
def is_windows():
    return sys.platform.startswith('win')

sh = SerialHandler()
search = Search(sh)
GUI(sh, search, windows=is_windows())
  
```

(d) Skriptet som initierar applikationen.

Fullständig källkod finns att tillgå på: [github.com/MrAkergrén/calibration](https://github.com/MrAkergrén/calibration)

## D Kopplingsschema



Figurer skapade med hjälp av verktyget Fritzing [14]

## E Specifikationer

### E.1 Arduino

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

### E.2 Sändare

Mikrokontroller	Arduino Uno
Lysdiod	Vit, 21 000 mcd, 3,4 V, 15°
Resistor	100 $\Omega$

### E.3 Mottagare

Mikrokontroller	Arduino Uno
Fotoresistor	18 - 50 k $\Omega$
Resistor	10 k $\Omega$

## F Uppmätt fokuspunkt från panel

Tabeller över normerade värden uppmätta från Parans solpanel SP3. Värdena utgör procent från det maximalt uppmätta värdet, avrundat till närmsta heltal. Anledningen till normeringen är att visa på storleksändring mellan talen, då de faktiska värdena är intetsägande då de inte direkt kan representera lux, på grund av felmarginal av luxmätaren och uppmättningsmetoden.

2	2	3	3	5	6	7	10	11	12	14	16	16	17	17	16	15	13	11	10	7	5	4	3	3
2	3	4	5	6	8	9	13	17	18	20	21	22	27	25	26	21	19	17	15	12	8	6	5	3
3	4	5	7	9	12	15	19	23	25	30	33	33	38	36	38	35	29	24	19	17	13	10	7	5
4	5	6	8	12	15	19	26	28	36	39	45	45	49	47	47	42	38	34	28	22	18	14	9	6
4	6	8	11	15	20	25	35	39	46	51	56	57	56	59	56	58	50	44	38	30	25	17	13	8
5	7	10	15	20	26	33	43	49	55	63	69	70	71	74	69	68	61	56	47	38	28	24	17	11
6	8	12	17	23	31	40	49	56	66	73	78	80	82	84	80	78	71	66	57	47	37	30	21	13
7	10	14	20	27	35	47	56	64	73	81	86	88	89	91	87	87	82	75	64	55	45	35	26	18
8	11	16	22	33	42	52	62	72	79	86	91	94	94	95	92	92	86	80	70	62	50	40	29	20
8	12	17	25	34	44	54	65	77	85	90	94	96	97	98	97	96	91	85	77	65	55	43	32	22
8	13	18	27	34	45	58	67	77	86	92	96	97	99	100	99	97	94	87	79	69	58	46	34	23
8	13	18	27	36	46	59	69	78	87	93	96	97	99	100	99	98	94	89	80	71	59	47	34	24
8	13	17	26	36	45	57	67	78	85	92	95	97	98	99	98	97	93	88	79	70	58	46	34	23
8	11	16	25	32	42	54	62	74	82	89	92	95	95	96	95	95	91	84	76	66	55	43	32	22
6	11	14	22	29	37	49	58	68	76	82	87	90	91	92	92	89	84	77	68	61	49	38	28	18
5	9	12	19	24	32	39	49	60	67	72	78	81	82	83	81	81	74	68	60	50	41	32	23	15
3	7	10	14	21	25	34	39	49	55	60	67	70	72	73	69	69	63	57	49	42	34	24	17	11
2	6	8	11	16	21	28	32	35	44	50	56	58	61	59	59	56	52	46	39	33	26	19	13	8
2	5	7	9	12	17	23	28	30	38	41	46	48	50	50	48	44	43	38	32	26	20	15	10	6
2	4	6	7	10	13	18	23	26	30	33	37	40	43	39	38	36	34	30	25	20	14	11	8	5
2	3	5	6	8	11	13	18	22	24	27	32	32	34	32	32	28	26	24	20	15	10	9	6	4
1	3	3	4	5	7	9	10	13	15	16	19	21	20	22	20	18	18	16	13	10	7	5	4	3
1	2	2	3	4	4	6	7	8	8	11	12	12	14	13	12	10	10	9	7	6	4	3	3	2
1	1	2	2	3	3	4	5	5	6	7	8	8	8	8	7	6	6	6	5	4	3	2	2	1
1	1	2	2	3	3	4	5	5	6	6	7	7	7	7	6	6	6	6	4	4	3	2	1	1

(a) Yoctopuce luxmätare, panel nummer 1

3	4	5	6	7	8	9	9	9	11	10	7	10	9	9	9	8	7	9	6
5	6	8	8	10	10	13	12	13	14	14	11	15	14	13	12	11	10	13	9
6	8	10	11	13	14	16	17	18	19	21	16	24	21	19	18	16	14	17	11
8	10	13	15	18	19	23	24	26	30	32	23	36	30	28	26	22	19	22	14
11	13	17	20	25	27	31	34	40	43	44	36	52	42	36	38	31	27	29	19
14	17	23	27	35	39	45	49	53	62	61	54	68	57	51	49	41	33	36	25
17	23	30	36	44	51	58	67	71	79	78	74	71	72	66	61	51	43	46	33
22	28	37	45	56	62	70	77	83	86	89	85	72	83	77	72	61	52	52	43
25	30	38	47	55	65	71	78	84	86	89	89	74	84	83	74	67	57	51	50
26	32	39	48	58	67	72	79	86	87	91	90	76	85	84	76	68	58	52	52
27	31	39	49	59	68	74	80	87	89	92	91	79	86	85	78	70	59	53	53
28	32	41	50	61	69	76	82	89	91	94	93	86	87	86	79	71	61	55	52
28	34	44	52	63	73	80	86	92	94	96	95	90	91	87	82	73	63	58	55
28	36	46	55	66	77	83	90	95	97	98	97	93	93	91	84	76	65	59	57
30	37	47	57	67	76	85	91	96	99	100	99	94	94	91	85	76	67	59	57
30	37	46	57	67	76	84	91	97	98	100	99	95	95	91	85	78	66	58	56
29	35	45	56	66	76	82	90	96	97	99	98	95	93	89	83	76	64	54	53
28	33	43	52	64	73	79	87	92	95	96	95	94	91	86	80	72	61	52	49
27	32	41	50	59	71	77	84	88	93	94	91	93	88	82	75	68	59	49	47
26	31	40	49	59	71	76	84	87	92	93	93	93	88	80	75	68	58	49	46

(b) Yoctopuce luxmätare, panel nummer 2

5	6	7	8	10	11	11	13	13	14	16	15	15	14	13	12	10	9	9	7
6	7	9	10	13	14	16	18	18	20	20	20	21	20	17	16	14	11	12	9
8	9	13	13	17	19	21	24	24	27	29	29	27	26	24	23	20	17	15	13
10	13	16	18	21	24	26	31	32	37	39	38	41	38	32	31	25	22	21	18
13	16	19	22	29	33	38	45	47	54	57	54	56	53	46	47	39	34	31	26
16	20	25	30	43	45	56	71	72	79	80	80	82	81	74	78	66	58	50	42
19	27	36	43	64	66	89	98	98	97	96	94	95	93	98	100	91	82	68	54
28	40	52	64	78	90	100	98	94	88	86	87	88	89	95	99	99	85	73	56
29	42	54	66	82	93	99	96	91	85	82	82	83	84	91	97	100	91	78	62
31	47	57	69	88	95	98	94	88	78	79	77	77	78	84	92	98	99	85	67
32	46	60	77	92	99	96	90	84	76	71	69	65	68	70	83	94	98	96	76
35	50	63	81	100	98	92	84	74	68	62	61	56	58	61	73	84	95	99	88
38	57	70	86	98	96	86	77	69	62	56	53	50	51	55	65	79	91	98	93
39	61	75	94	96	93	82	72	65	58	53	50	46	46	51	61	73	87	97	94
39	63	81	98	95	90	79	70	63	56	51	48	46	46	50	59	73	87	96	97
39	64	82	98	95	89	78	70	64	56	52	49	47	48	52	61	74	87	97	96
35	63	80	98	96	92	80	72	65	60	54	52	50	52	55	65	78	91	98	92
28	58	76	92	98	94	83	76	69	63	59	57	55	57	62	71	83	96	100	86
28	54	71	87	99	96	89	81	75	72	67	62	60	65	69	77	91	98	97	80
30	50	65	83	100	98	92	85	79	72	68	67	65	68	73	78	91	99	94	73

(c) Adafruit luxmätare, panel nummer 2

## G Tidsplan

Innan projektets start upprättades en tidsplan enligt följande.

Projektarbetet skulle inledas vecka 13 och målet var att, om inget oförutsett inträffade, den skriftliga rapporten skulle slutföras under vecka 23. Resultatet skulle sedan muntligen presenteras under vecka 24 eller 25. De första 2-3 veckorna planerades att mestadels ägnas åt planering, litteraturstudier och problemanalys. Därefter beräknades tiden ägnas åt utveckling, kompletterande informationsinhämtning och kontinuerlig utvärdering. Parallellt med detta skulle löpande rapportskrivning att ske. I projektets avslutande 1-2 veckor skulle större fokus ligga på att färdigställa den skriftliga rapporten och presentationen.

Vecka (2015)	Projekt- vecka	Mål
13	1	Planering och upprättande av arbetsstruktur, uppstartsmöten
14	2	Litteraturstudier, planering och problemanalys
15	–	Påskuppehåll
16	3	Litteraturstudier, problemanalys och utveckling
17	4	Utveckling
18	5	Utveckling och utvärdering
19	6	Utveckling och utvärdering
20	7	Utveckling och utvärdering
21	8	Utveckling och utvärdering
22	9	Utveckling, utvärdering och fokus på skriftlig rapport
23	10	Fokus på skriftlig rapport och presentation

