

WORKING TITLE

# SolarRemote

Strand, Johan

Svedberg, Pär

johstr@student.chalmers.se

svpar@student.chalmers.se

Åkergren, Oskar

akergren@student.chalmers.se

2015-02-25

## Beteckningar

C	Imperativt programmeringsspråk
CP2102	Enhet från Silicon Labs som omvandlar kommunikation från USB till seriell enligt RS232
GUI	Graphical User Interface, grafiskt användargränssnitt
I/O	Input/Output
PIC32	32-bitars mikrokontroller
Python	Högnivåspråk för programmering
SP3	Parans tredje generationens solpanel
SSH	Secure Shell. Protokoll för säker anslutning mellan datorer
USB	Universal Serial Bus, standard för seriell kommunikation
Enkortsdator	I denna rapport menas enkortsdatorer av typen System on a chip, exempelvis enheter från Raspberry Pi och Beaglebone. En enkortsdator ska enligt rapportens definition klara av att driva operativsystem innehållande Linuxkärnan eller motsvarande.
Mikrokontroller	Här menas enchipdatorer avsedda att programmeras direkt till enhetens programminne, exempelvis enheter från Arduino. Dessa klarar ej av att driva operativsystem innehållande Linuxkärnan eller motsvarande.

# Innehåll

<b>1</b>	<b>Introduktion</b>	<b>4</b>
1.1	Bakgrund . . . . .	4
1.2	Syfte . . . . .	4
1.3	Frågeställning . . . . .	4
1.4	Avgränsning . . . . .	4
<b>2</b>	<b>Metod</b>	<b>5</b>
<b>3</b>	<b>Genomförande</b>	<b>5</b>
3.1	Val av plattform . . . . .	6
<b>4</b>	<b>Resultat</b>	<b>6</b>
<b>5</b>	<b>Diskussion</b>	<b>7</b>
5.1	Hårdvara . . . . .	7
5.2	Mjukvara . . . . .	7
5.3	Framtida bruk . . . . .	7
	<b>Referenser</b>	<b>8</b>

# 1 Introduktion

## 1.1 Bakgrund

Parans har utvecklat en produkt som via optiska fibrer levererar naturligt solljus. Som ett av få bolag i världen levererar de system globalt och deras för närvarande största installationer finns i Malaysia och Los Angeles.

Med hjälp av linser fokuseras solljus in i optiska fibrer och panelen styrs med hjälp av två stegmotorer. Styrningen sker på input dels från en algoritm som, baserat på position (longitud, latitud) och tid, ger en solposition i grader och dels från en solsensor med fotocell som ger data för en finstyrning av panelens positionering då solen är framme. Detta för att alltid maximera solljusets fokusering in i fibern.

Själva panelen körs på 12V och dess systemdesign bygger på en PIC32; koden är skriven i C. Parans kommunicerar med enheten via USB-port och en terminalemulator.

## 1.2 Syfte

Idag styrs panelen till rätt position via en terminalemulator, vilket är en tröskel för Parans kunder vid installation och felsökning. Exempelvis har alla inte vana av att jobba i terminaler och det kan vara krångligt att konfigurera datorns USB-portar så att kommunikation kan ske med panelen.

Parans vill därför utveckla en styrdosa/box med tryckknappar, lysdioder och eventuellt en display som minskar problemen för kunderna. Denna box kan vara i form av ett befintligt kort som t.ex. Raspberry Pi, Arduino eller liknande men skulle också kunna vara en app för Android/iOS som kan köras på en kundens mobila enhet.

## 1.3 Frågeställning

Rapporten ämnar att besvara följande frågeställningar:

- Vad styr valet av plattform för styrdosan?
- Vilken plattform blir enklast för kunderna att använda?
- Påverkar valet av plattform huruvida styrdosan blir kompatibel med framtida versioner av solpanelen?
- Vilket programmeringsspråk lämpar sig bäst för styrdosan?

## 1.4 Avgränsning

Vi ser att detta projekt kommer kunna skapas med existerande hårdvara i form av mikrokontrollerkort, telefoner eller enkortsdatorer. Detta ger att vi kommer att begränsa projektet till dessa former och inte utveckla ett eget mönsterkort.

Dagens paneler kan kommunicera med externa enheter via en USB-port men saknar övriga kommunikationsmöjligheter i dagsläget. Detta gör att projektet begränsas till kommunikation via en ansluten USB-kabel och inte via någon trådlös kommunikation.

## 2 Metod

Vi har för detta projekt valt jobba utifrån undersökningsmetoden beskriven i ”A design science research methodology for information systems research” [1].

Metoden Design Science Research (DSR) beskriver efterforskning uppdelad i fem faser; *Problem Analysis & Motivation*, *Design & Development*, *Demonstration*, *Evaluation* and *Communication*. I den första fasen identifieras problemområdet tillsammans med Parans och vilken lösning som efterfrågas. I faserna *Design & Development*, *Demonstration* and *Evaluation* tar vi fram en prototyp som i varje iteration utvärderas för att se hur den väl den uppfyller de krav som sattes upp. Resultat presenteras slutligen i fasen *Communication* som dels en muntlig presentation och dels en skriven rapport.

Vi kunde istället för DSR ha valt exempelvis ’Action Research’ (AR) vilket också är en iterativ metod och väldigt lik DSR i sitt utförande [2]. Båda metoderna skapar kunskap om specifika situationer och problem men anledningen till att vi valde DSR framför AR var att DSR strävar mot att designa och skapa artefakter vilket går väl ihop med projektet om att utveckla en handhållen fysisk enhet.

## 3 Genomförande

Linuxsystemet på den enkorts dator som tillhandahölls av bolaget, ett utvecklingskort från Olimex, behövdes initialt konfigureras för att möjliggöra SSH-anslutning via nätverk. För detta användes en USB-till-seriell-kabel kopplad mellan utvecklingskortets seriella pinkontakter och en persondators USB-port. Sedermera anslöts enkorts datorn via en nätverksbrygga till bolagets trådlösa nätverk och SSH-anslutningen verifierades. Ett första test för att säkerställa att kommunikation mellan enkorts datorn och SP3 kunde upprättas utfördes genom att koppla samman enheterna med en USB-kabel. Utvecklingskortets linuxsystem identifierade automatiskt anslutning av SP3s CP2102 och denna monterades automatiskt i linux som `/dev/ttyUSB0`. Med hjälp av det i linux medföljande terminalprogrammet Miniterm kunde kommandon skickas till SP3, som i sin tur utförde dessa och svarade korrekt.

### 3.1 Val av plattform

För att konstruera den typ av fjärrkontroll som möter projektets krav måste först ett beslut tas om vilken teknisk plattform som ska användas. De alternativ som diskuteras var androidbaserade enheter, arduinosystem och enkorts datorer, främst Raspberry Pi.

#### Androidenhet

##### Fördelar

- + Pekskärm medför stor valfrihet i utförande av användargränssnitt
- + Etablerat OS
- + Stor skärmyta

##### Nackdelar

- Otydligt vilka enheter som stöder USB-host
- Relativt dyr
- Mer prestanda än nödvändigt

#### Raspberry Pi

##### Fördelar

- + Stor tillgång till information
- + God tillgänglighet till utbyggnadsmoduler
- + Lågt pris

##### Nackdelar

- Mer prestanda än nödvändigt
- Saknar skärm och knappar i grundutförande
- Kan bli otymplig vid användande av många tilläggsmoduler

#### Arduinosystem

##### Fördelar

- + Ofta använt systemet till mindre elektronikprojekt
- + God tillgänglighet till utbyggnadsmoduler
- + Modulär uppbyggnad
- + Tillgång till mycket information
- + Låg energiförbrukning

##### Nackdelar

- Saknar skärm och knappar i grundutförande
- Kan bli otymplig vid användande av många tilläggsmoduler

## 4 Resultat

Här kommer vi att presentera resultatet.

## 5 Diskussion

### 5.1 Hårdvara

Som nämnt på sidan 2 så utgår vi ifrån begreppet enkorts dator för ett kretskort som är kapabel till att driva en Linuxkärna, till skillnad från en mikrokontroller där en svagare krets avses.

Den lösning som vi har tagit fram är fullt fungerande och relativt enkel att reproducera, i förhållande till att utveckla en likartad konstruktion med en mikrokontroller. Det som gör vår lösning enklare är fram för allt att en enkorts dator har de drivrutiner som krävs för att upprätta den seriella kommunikationen, så till vida att den har en Linuxkärna senare än version 3.0 [3].

Nackdelar som vi ser med att använda en enkorts dator är bland andra att dessa generellt har ett större energibehov än en mikrokontroller [4], [5]. Antalet I/O portar är oftast färre på en enkorts dator och den fysiska storleken är större jämfört med de mikrokontrollerkort som hade varit lämpliga för projektet.

### 5.2 Mjukvara

Den mjukvara som har utvecklats, har skrivits i programmeringsspråket **Python**. Språket beror delvis på att personal inom företaget har erfarenhet inom språket vilket underlättar för framtida utveckling och underhåll av projektets produkt och dels valdes språket för dess enkla utveckling av grafiska gränssnitt och bra stöd i den seriella kommunikationen som krävdes i projektet.

Andra språk som hade varit möjliga är till exempel **C** eller **Java** då projektgruppen har erfarenhet av de båda språken. **C** valdes bort då utveckling av grafiska gränssnitt i detta språk kräver externa bibliotek och minskar därför portabiliteten och ökar komplexiteten. **Java** är en lämplig kandidat för projektet, men valdes bort då den grafiska utvecklingen i **Python** är enklare och applikationen som vi utvecklade är såpass simpel att **Java** skulle medföra stor andel så kallad 'overhead' i programmeringskoden. Nackdelen med **Python** jämfört med **Java** är att språket inte är lika effektivt i sina beräkningar, men då applikationen vi skrivit inte utför några tyngre beräkningar så berörs inte användarupplevelsen av detta. Kommer applikationen att vidareutvecklas till något mer än vad projektet skapat, är det fullt rimligt att översätta logiken till **Java**, något som det finns gott om stöd för.[6]

### 5.3 Framtida bruk

Att projektet genomfördes grundar sig i SP3s bristande stöd för kommunikationsstandarder och att dagens kommunikationsgränssnitt inte är användarvänligt, vilket leder till stora underhållskostnader för företaget då det krävs tid och resurser att stötta underhållspersonal. Detta projekt svarar upp på de förväntningar som bolaget hade på oss, men vi ser att projektets produkt är överflödig i nyare revisioner av panelen, där styrkortet kan ha tillgång till fler kommunikationsstandarder och kan komma att styras på distans.

## Referenser

- [1] K. Peffers, T. Tuunanen, M. A. Rothenberger och S. Chatterjee, "A design science research methodology for information systems research", *Management Information System*, vol. 24, s. 45–78, 3 2007. URL: [http://wise.vub.ac.be/thesis\\_info/Design\\_Science\\_Research\\_Methodology\\_2008.pdf](http://wise.vub.ac.be/thesis_info/Design_Science_Research_Methodology_2008.pdf) (hämtad 2015-02-04).
- [2] R. Baskerville, J. Pries-Heje och J. Venable, "Soft design science methodology", *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology*, 2009. URL: <http://dl.acm.org/citation.cfm?id=1555631> (hämtad 2015-02-04).
- [3] Silicon Labs. (2015). Cp210x usb to uart bridge vcp drivers, URL: <https://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx> (hämtad 2015-02-04).
- [4] Igor. (2013). Arduino power consumption normal & sleep, URL: <http://gadgetmakersblog.com/arduino-power-consumption/> (hämtad 2015-02-04).
- [5] Raspberry Pi Foundation. (2015). Power supply, URL: <http://www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md> (hämtad 2015-02-04).
- [6] Jython. (2014). General information, URL: <https://wiki.python.org/jython/JythonFaq/GeneralInfo> (hämtad 2015-02-25).