

Projektbeskrivning

SolarRemote

Strand, Johan

Svedberg, Pär

`johstr@student.chalmers.se`

`svpar@student.chalmers.se`

Åkergren, Oskar

`akergren@student.chalmers.se`

2015-01-22

Bakgrund

Parans har en unik teknologi där vi levererar naturligt solljus via optiska fibrer. Som ett av endast två bolag i världen levererar vi system över hela jorden och har just nu våra två största installationer 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 (lon, lat) och tid ger en sol-position i grader, och dels från en sol-sensor med fotocell som - då solen är framme - ger data för en finstyrning av panelens positionering. Detta för att alltid maximera solljusets fokusering in i fibern.

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

Problembeskrivning

Idag styrs panelen till rätt position via en terminal/putTy 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.

Ställda krav

Kommandon som ska ingå:

- run u
- run d
- run l
- run r
- run stop
- run auto
- setup
- restart

Funktioner som helst ska ingå:

- diod som indikerar "SP3-power"
- diod som indikerar "remote-power"
- "Get time" – date (visas på display)
- "Get pos" – lon och lat (visas på display)

'setup' och 'restart' bör vara två olika knappar då man vid vissa tillfällen endast vill använda enbart 'setup'.

'Get pos' kan vara en knapp så länge displayen kan visa "lon xx.xxxx" och "lat yy.yyyy" samtidigt, annars två olika.

Önskvärda funktioner steg 1

- "Verify position" eller liknande där GPS ger referenstid och lon/lat och som ger signal/diod/utskrift på displayen som OK eller liknande.

Önskvärda funktioner steg 2

- "Set pos" och "Set time" som ställer in "lon xx.xxxxx", "lat yy.yyyyyy" och "date 2015-01-20 13:50:32" baserat på värden från GPS:en

- Diod eller knapp som skriver ut "OK" eller "not verified" om det är så att sensorn har tagit kommando över beräknad position. Detta kan ses genom att utläsa specifika värden i den raden som loggen skriver varje sekund.

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 diskuterades var androidbaserade enheter, arduinosystem och enkorts datorer, främst Raspberry Pi.

Androidenhet

Fördelar

- + stor skärmyta
- + etablerat OS
- + Touchscreen ger stor valfrihet i utförande av interface

Nackdelar

- otydligt vilka enheter som stöder USB-host
- relativt dyr
- mer prestanda än nödvändigt

Raspberry Pi

Fördelar

- + stor community med tillgång till mycket information
- + tillgänglighet till utbyggnadsmoduler
- + lågt pris

Nackdelar

- mer prestanda än nödvändigt
- saknar display och knappar i grundutförande
- kan bli otymplig vid användande av många tilläggsmoduler

Arduinosystem

Fördelar

- + stor community med tillgång till mycket information
- + stor tillgänglighet till utbyggnadsmoduler
- + modulär uppbyggnad
- + mest använda systemet till små elektronikprojekt
- + låg energiförbrukning
- + open source

Nackdelar

- saknar display och knappar i grundutförande
- kan bli otymplig vid användande av många tilläggsmoduler

Efter en mindre diskussion enades projektgruppen och uppdragsgivaren om att utgå ifrån ett arduinosystem. Beslutet grundade sig på att arduino är väl lämpat för mindre elektronikprojekt, då det har tillgång till en stor mängd tilläggsmoduler och ej innehåller fler komponenter än nödvändigt. Att plattformen är open source lämnar också möjligheten öppen för att i framtiden kunna anpassa lösningen till en egen tilläggsmodul.

Tidsplan

Preliminär tidsplan för vårt projekt, där vi lägger ungefär 20 timmar per person och vecka på projektet för att nå upp till den arbetsbelastning som förväntas enligt kursen.

Läsvecka Mål

- lv 1 Upprättande av arbetsstruktur, uppstartsmöten med handledare, införskaffande av materiel.
- lv 2 Inläsning av dokumentation, utveckling av initial kod.
- lv 3 Programutveckling
- lv 4 Halvtidsredovisning, prototyp bör kunna uppvisas
- lv 5 Programutveckling
- lv 6 Undersöka vilka extra funktionaliteter som kan inkluderas
- lv 7 Slutgiltig programutveckling
- lv 8 Förberedelser för slutredovisning