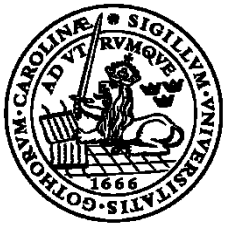
Cactus Eye Testmiljö

Testmiljö för ett SCADA-system



LUNDS UNIVERSITET
Campus Helsingborg

LTH Ingenjörshögskolan vid Campus Helsingborg
Institutionen för Elektro- och Automationsteknik

Examensarbete:
Safaa Zaki
Hamed Rahim Zadeh

LTH Ingenjörshögskolan vid Campus Helsingborg
Lunds universitet
Box 882
251 08 Helsingborg

LTH School of Engineering
Lund University
Box 882
SE-251 08 Helsingborg
Sweden

Tryckt i Sverige
Lunds universitet
Lund 2017

Sammanfattning

Syftet med examensarbetet var att bygga en testmiljö som skall användas av företaget Cactus Uniview. Testmiljön skall vara ett hjälpmedel till testarna på utvecklingsavdelningen, med testmiljön ska man kunna säkerställa kvaliteten på verktyget Cactus Eye. Mjukvaran Cactus Eye är kopplad till PLCer och kan användas för att styra PLCerna.

Cactus Uniview använder en virtuell testmiljö (helt baserad på mjukvara) där de kan utveckla en programkod som ersätter mätvärden från riktiga givare ute hos deras kunder. Nackdelarna med den nuvarande virtuella testmiljön är att den inte kan testa kommunikationen mellan styrsystem (PLC) och Cactus Eye. Den kan inte heller testa hanteringen av mätvärden.

Cactus Uniview vill utveckla kommunikationen mellan PLCer och Cactus Eye. De vill även utveckla hanteringen av mätvärden. Företaget vill utveckla den nuvarande virtuella testmiljön till en fysisk testmiljö där testerna körs med hjälp av ett antal givare som monteras i en rack tillsammans med PLC. Testerna blir mer realistiska, dvs. mätvärdena kommer från riktiga givare.

För att avhjälpa dessa brister har en ny testmiljö utvecklats under examensarbetet. Den består av en server, en PLC, ett antal givare, en lampa och en HMI-panel.

Cactus Eye har en funktion som heter DCS-hantering (Distributed control systems). Denna funktion gör det möjligt att programmera i Cactus Eye. DCS-hantering finns inte i den virtuella testmiljön och därför ville företaget ha den funktionen integrerad i den fysiska testmiljön. Syftet är att hitta fel som uppstår vid programmering av styrsystem.

Med hjälp av den nya testmiljön kan kommunikation mellan styrsystem och Cactus Eye testas. Det går även att skriva kod till PLCerna genom DCS-hantering vilket ger möjligheten att utveckla virtuella signaler om det behövs.

Nyckelord: PLC, HMI, TwinCAT, CACTUS EYE, MELSOFT GT DESGINER 3, DCS-FUNKTONER, SCADA.

Abstract

The aim of the thesis was to build a test environment to be used by the company Cactus Uniview. The test environment should be a tool (software) for testers in the development department, the test environment should be able to ensure the quality of the tool Cactus Eye. Cactus Eye is connected to a PLCs and can be used to control many PLCs.

Cactus Uniview uses a virtual test environment (entirely based on software) where they can develop a program that replaces the measurements obtained from sensors stationed at their clients. The disadvantage of the current virtual test environment is that it is incapable of testing the communication between controllers (PLC) and Cactus Eye. The system is also unable to test whether the data gathered by the sensors, when sent to Cactus Eye, is received or not.

Cactus Uniview wants to develop communication between PLCs and Cactus Eye. They also want to develop the management/handling of the data collected by the sensors. The company also aims to develop the current virtual test environment into a physical test environment. The tests would then be run with the help of a number of sensors that are mounted in a rack with the PLC. The tests will be more realistic, i.e. readings from actual sensors.

To rectify these shortcomings, a new test environment has been developed during the thesis. It consists of a server, a PLC, a number of sensors, a lamp and an HMI panel.

Cactus Eye has a feature called DCS management (Distributed Control Systems). This feature makes it possible to program using Cactus Eye. DCS is not integrated in the old virtual test environment which is why the company wanted to have that feature integrated in the physical test environment. The aim is to find errors which arise during programming of PLCs.

The communication between the controllers and the Cactus Eye system can now be tested using the new test environment. It is now possible to write code to the PLCs by using the DCS function to develop virtual signals if needed.

Keywords: PLC, HMI, TwinCAT, CACTUS EYE, MELSOFT GT DESGINER 3, DCS-FUNKTONER, SCADA.

Förord

Vi tackar Cactus Uniview för möjligheten att få göra vårt examensarbete hos dem. Tack till alla arbetare som verkade och hjälpte oss under arbetets gång, det var en väldigt lärorik period som fick oss att bli mer intresserade av automation. Vi vill även tacka Mohammed Kasem, som hjälpte oss få kontakt med Christoffer Willenfort, utan honom hade vi inte fått göra vårt examensarbete på Cactus Uniview.

Stort tack till vår handledare Mats Lilja och vår examinator Christan Nyberg som har gett oss ledning och feedback under projektets gång.

Terminologi

PLC	Programmable Logic Controller, programmerbar logisk styrenhet
PT-100	Pt-100 (RTD) är en temperaturgivare som utför mätningar baserade på resistans.
HMI	Human Machine Interface, maskin-människa gränssnitt för styrning och övervakning
I/O	Input/Output, Ingång/Utgång
VDC	Volt Direct Current, likspänning
DCS	Distributed control systems, distribuerade styrsystem
SPRS	Språk för reglering och styrning av processer
TwinCAT	Mjukvara för programmering och konfigurering av Beckhoff PLC
VA	Vatten och avlopp
Ebus	Baserad på LVDS (Low Voltage Differential Signaling), standard specificerad i ANSI/TIA/EIA-644-1995
SCADA	Ett datasystem som används inom automation för att övervaka och styra små och stora anläggningar

Innehållsförteckning

Innehåll

1 Inledning

Detta kapitel kommer att beskriva bakgrund, syfte, problemformulering och avgränsningar av examensarbetet.

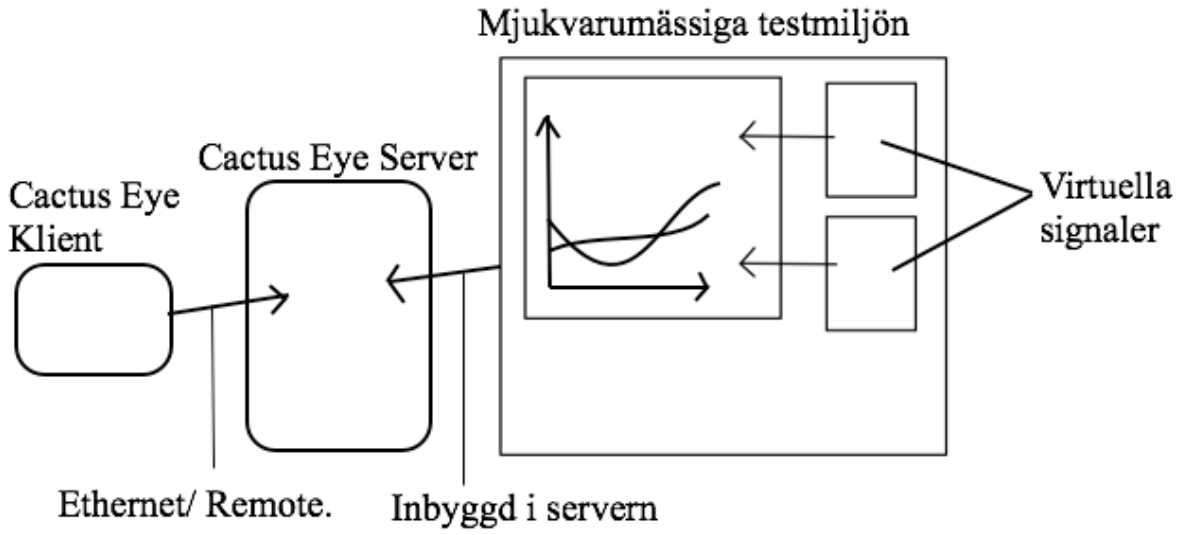
1.1 Bakgrund

Cactus Uniview är ett företag som erbjuder styr- och övervakningssystem inom VA och järnvägbranschen. Huvudkontoret för Cactus Uniview ligger i Göteborg. Där jobbar de med järnvägbranschen för det mesta men även med mindre projekt som har med styr- och övervakningssystem att göra. I Helsingborg jobbar företaget huvudsakligen med VA-branschen. Cactus Uniview levererade styrsystemen till Ringsjöverket som var det första vattenverket som började med datoriserad styrning och övervakning. [14]

1.2 Syfte

Företaget använder ett verktyg som kallas Cactus Eye, vilket de själva har utvecklat. Cactus Eye är ett driftdatorsystem (SCADA-system) där man kan övervaka olika anläggningar, prioritera händelser, sätta olika larm med olika prioritet, hantera realtidsinformation, övervaka och manövrera olika stationer etc.

Figur 1 visar en översikt på den gamla virtuella testmiljön. Den gamla testmiljön var installerad på en server (Cactus Eye Server). Klienter kan anslutas mot den gamla testmiljön lokalt via Ethernet eller över Internet.



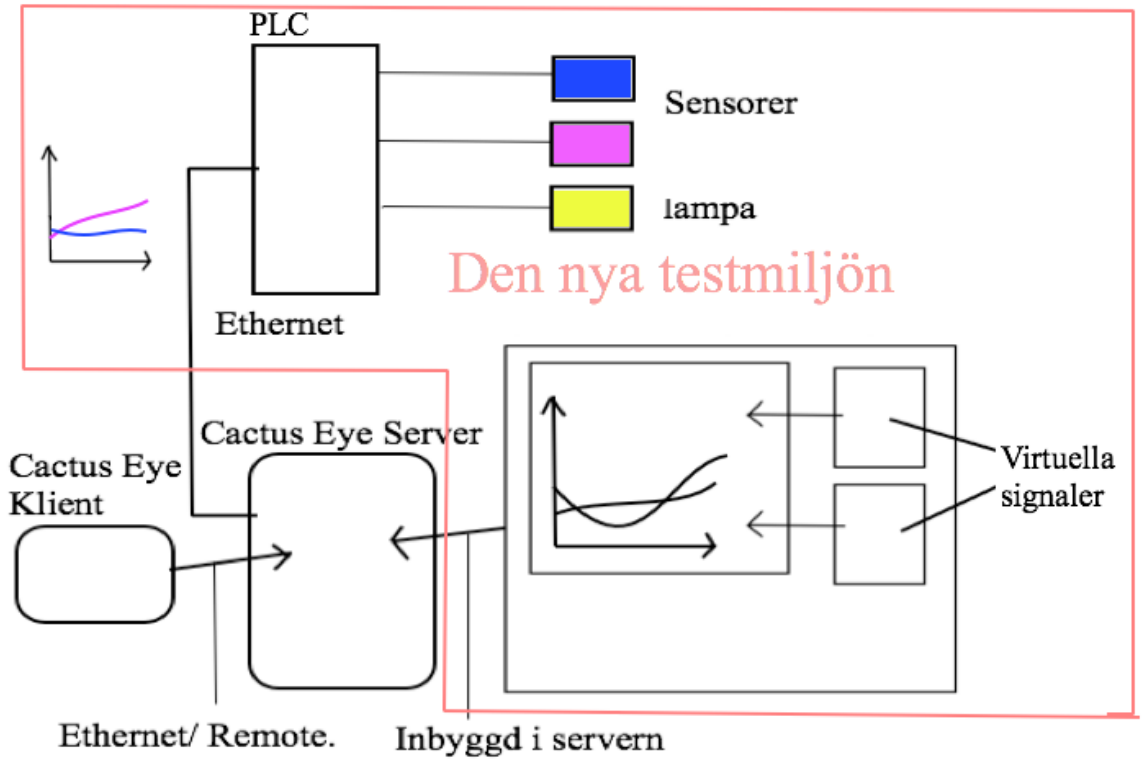
Figur 1 - Översikt över den gamla virtuella testmiljön.

Cactus Uniview behöver säkerställa kvaliteten på Cactus Eye, genom att man kör olika tester. I nuläget körs testerna på en virtuell miljö som en testare kan ansluta sig mot. Den gamla virtuella testmiljön har inga hårdvaror utan bara mjukvara. Testerna som utfördes av företaget testade kommunikationen mellan olika styrsystem t.ex. PLC-Server eller PLC-givare etc. Det var även tester på olika givare som skickade tillbaka mätvärden till Cactus Eye server.

Cactus Uniview vill utveckla sin virtuella miljö och göra den mer verklighetstrogen. Med detta menas att man vill använda ett rack där det kopplas in en eller flera PLCer med olika

sensorer. På detta sätt kan man testa de olika PLCerna innan leverans till kunder. Anledningen till att man vill ersätta den virtuella miljön med en mer realistisk är att det var svårt att upptäcka fel med hjälp av den gamla testmiljön.

Examensarbetets syfte var att skapa en fysisk testmiljö där testerna skulle vara verklighetstrogna dvs. man använder sig av mätvärden från riktiga sensorer till skillnad ifrån en virtuell miljö. I den måste ingå minst en PLC och ett antal sensorer som sedan kan styras och övervakas. Den fysiska testmiljön gör det möjligt för Cactus Uniview att upptäcka fel innan de levererar till sina kunder. Figur 2 visar en översikt över den nya testmiljön.



Figur 2 - Översikt över den nya fysiska testmiljön.

1.3 Problemformulering

Examensarbetet går ut på att skapa en fysisk testmiljö till Cactus Eye som används för att övervaka olika styrsystem inom VA- och järnvägbranschen. För att uppnå detta behöver följande frågor besvaras:

I. Vilka sensorer och givare behövs för att få en testmiljö som är så realistisk som möjligt?

II. Vilken hårdvara ska väljas för servern för att den skall klara Cactus Eye Server - Klient mjukvara?

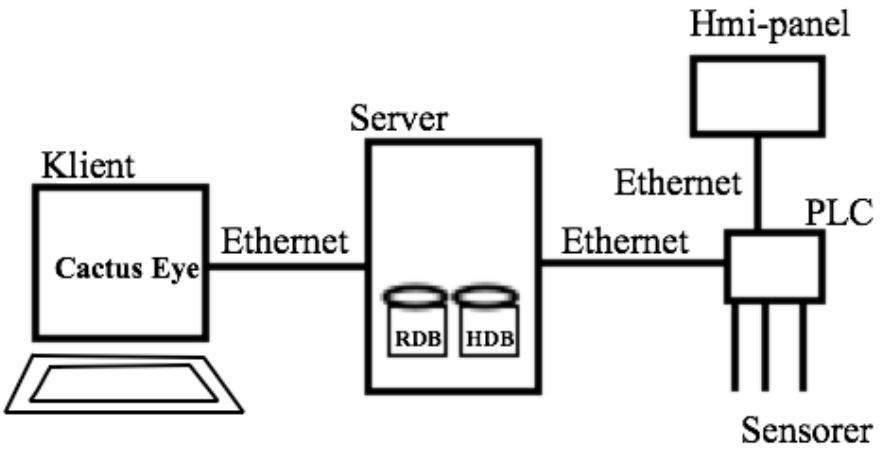
III. Vad är enklaste sättet att återställa systemet till känt läge efter man testat ett projekt?

IV. Hur ska testerna som utförs av Cactus Eye dokumenteras för att underlätta för andra testare att förstå och följa upp?

V. Är det möjligt att integrera någon form av HMI i testmiljön?

1.4 Begränsningar

Examensarbetet begränsades till att utveckla en fysisk testmiljö enligt figur 3.



Figur 3 - Schema över den planerade testmiljön.

2 Teknisk bakgrund

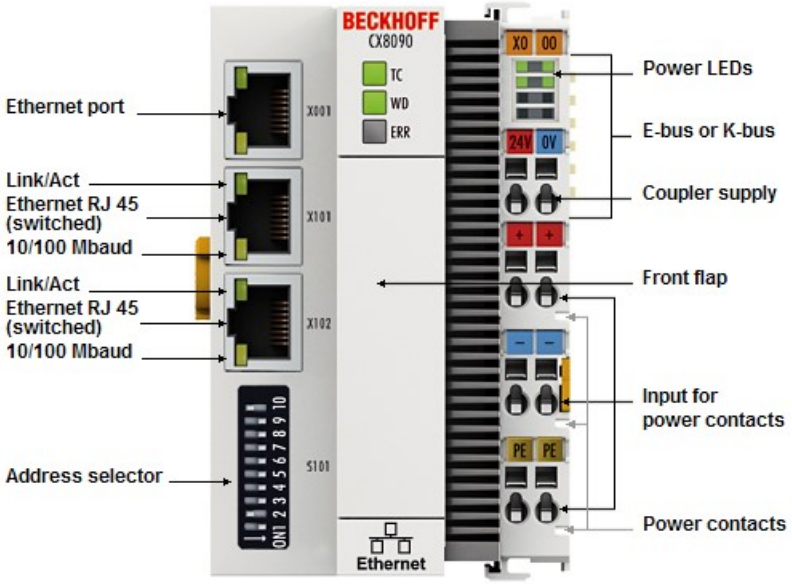
I detta kapitel beskrivs kortfattat vilka komponenter som kommer att användas i det nya systemet samt Cactus Eye-systemets programmeringsspråk. För att inte bryta mot företagets sekretess och policy kommer vissa saker att förklaras kortfattad.

2.1 Beckhoff CX8090 PLC (Cactus C10) [6]

Själva begreppet PLC (*Programmable Logic Controller)* används inom området automation och det är ett styrsystem som kan programmeras för att styra objekt, istället för att styra dem manuellt vilket blir betydligt mer komplicerat speciellt för stora system som t.ex. olika fabriker inom livsmedelsindustrin. [13]

Beckhoff CX8090 (som av Cactus säljs under namnet Cactus C10) är en PLC som Cactus Uniview säljer till sina kunder. Dessa PLCer köps från Beckhoff och sedan modifieras de så att de blir kompatibla med Cactus-systemet. Efter modifieringen kallas PLCerna Cactus C10.

En av de största fördelarna med Beckhoffs styrsystem är att det kan kopplas in ett eller flera I/O – kort. I/O – korten har inbyggda ingångar och outputs och de kan väljas separat beroende på behov. Detta hjälper ifall utvecklare (den person som programmerar eller installerar PLCn) inte vill ha för många ingångar respektive outputs som inte används. Till denna testmiljö har man valt att använda två stycken I/O – kort, ett till temperaturgivarna och ett till lampan. Dessa kort beskrivs lite längre ner i underkapitlet 2.1.1.2 – I/O-kort. Figur 4 är en bild på Cactus C10 PLC (Beckhoff CX8090).



Figur 4 - Cactus C10 PLC. [6]

Cactus Uniview bestämde sig att de skulle använda denna PLC i det nya testsystemet eftersom den var lagom stor för vårt projekt men även att den var kraftfull och billig i jämförelse med andra PLCer som de sålde.

2.1.1 Konfiguration

Mjukvaran som används för att konfigurera PLCn heter TwinCAT. Denna mjukvara hämtas från Beckhoffs hemsida, dock behöver man vara registrerad hos Beckhoff.

I/O-korten behövs identifieras första gången de kopplas in i PLCn, detta görs genom att köra en så kallad device scan i TwinCat. Detta gör att alla I/O-kort detekteras. Efter att den kört denna scan så kommer ett fönster att visas i TwinCat som visar vilka kort som identifierats i PLCn.

Detta är allt som behövs för att konfigurera styrsystemet.

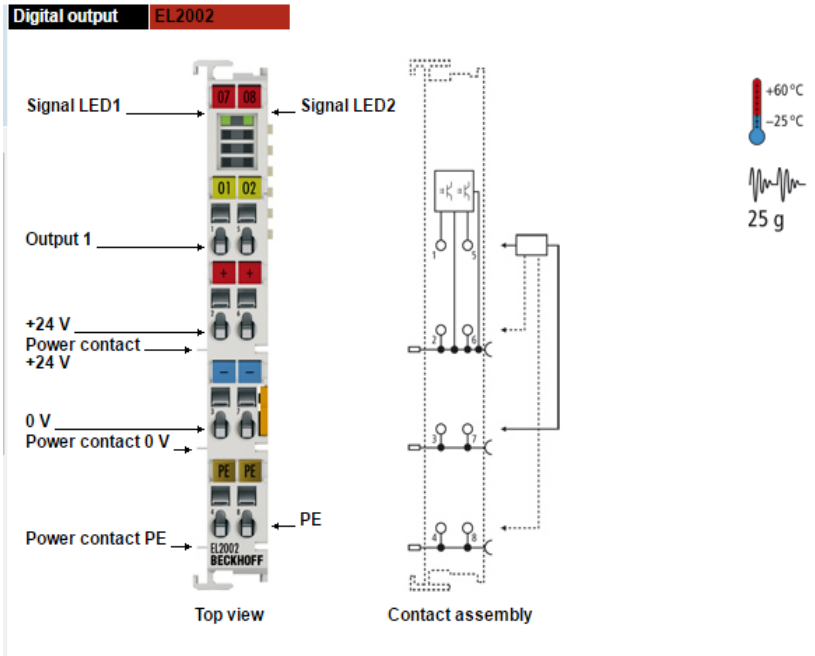
2.1.2 I/O – kort. [7] och [8]

Till testmiljön behövs det två olika I/O-kort, ett till analoga ingångar och ett till digitala utgångar. Det analoga ingångskortet är för givaren; givaren mäter en temperatur som registreras i PLCn genom PLCns ingång (via analoga ingångskortet). Det digitala utgångskortet är för att styra lampan. Lampan styrs med hjälp av en digitalsignal som skickas ut från PLCn via det digitala utgångskortet. Digitala ingångar och analoga utgångar behövs inte i detta projekt eftersom det inte kommer användas.

Lampan däremot är någonting som kan manövreras dvs. lampan kan tändas och släckas. Manövreringen sker med hjälp av en digital utsignal som skickas till en switch där lampan är monterad. Därför behövs det digitala utgångskortet.

Det har använts två olika kort för den nya testmiljön. En **EL2002** och en **EL3202**. Dessa kort är Beckhoff EtherCat terminaler, EtherCat-kort använder E-buss på styrsystemet. E-buss är en standard som använder LVDS. LVDS är en teknisk standard som anger elektriska egenskaperna hos en differential, seriell kommunikationsprotokoll. [17] E-bussen är den som spänningssätter en PLCes I/O-kort. För en närmare beskrivning av E-bussen se [15].

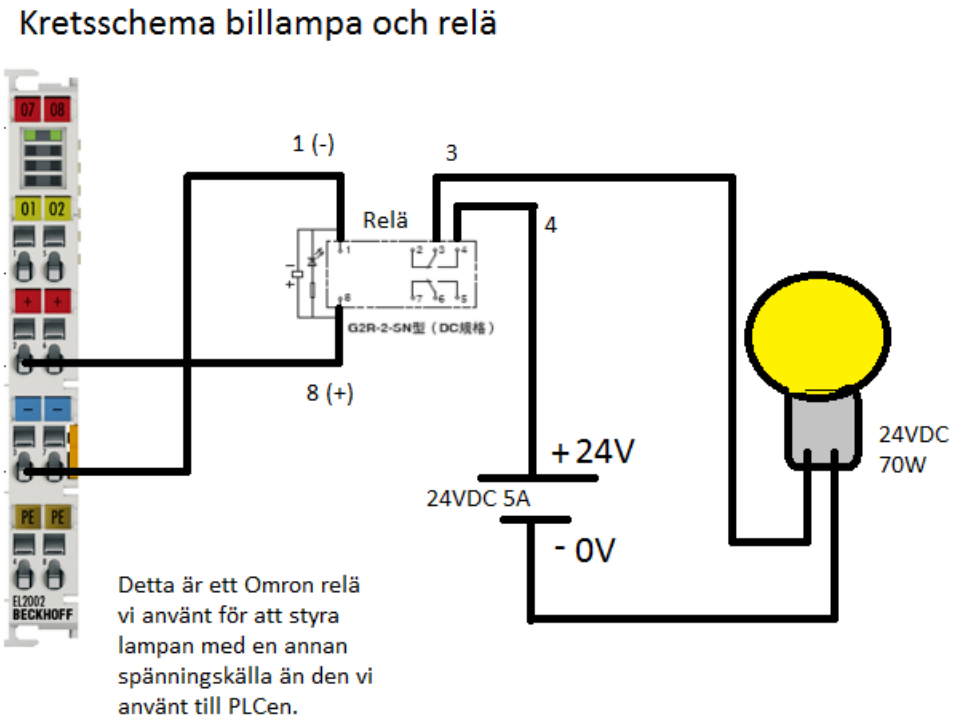
EL2002 är en 2-kanalers digitalt utgångskort. Se figur 5.



Figur 5 - El2002, digital utgångskort. [8]

Detta kort får sin spänning från E-bussen. Den har spänningsintervallet 0-24V. Det som är bra med sådana kort är att de får inmatningsspänning från E-bussen.

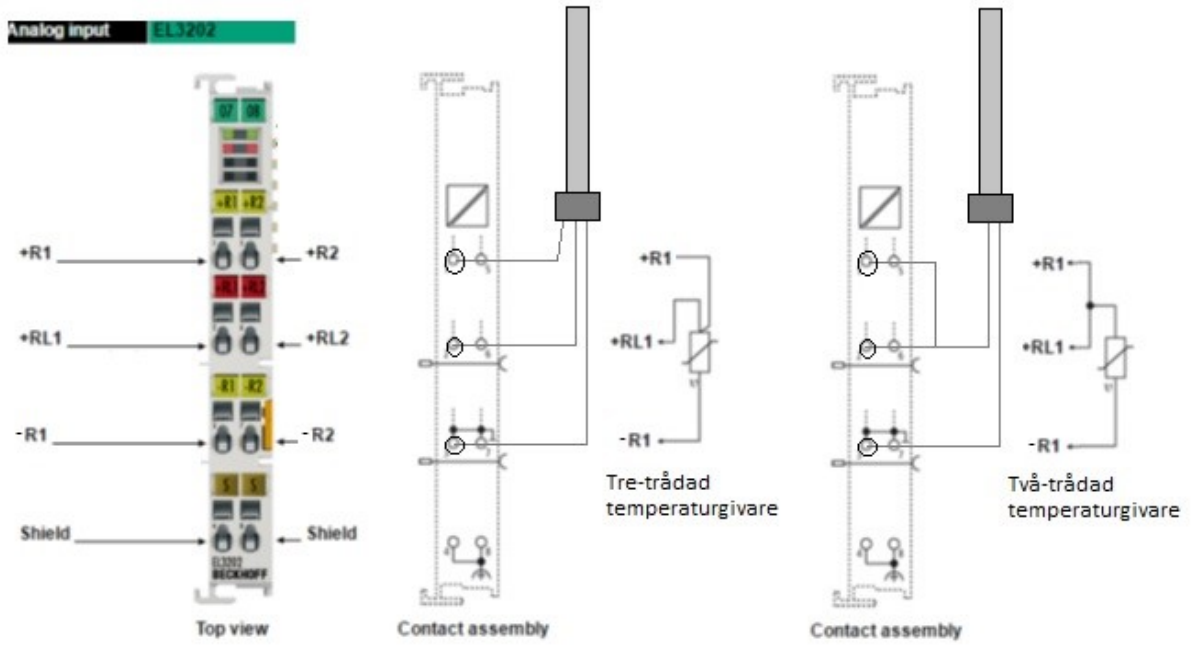
Figur 5 ovan visar att det finns 8 utgångar. De viktigaste utgångarna här är 1:an och 3:an. Utgång 1 har spänningen 24VDC och utgång 3 har 0VDC. Lampan och relä kopplas i 1:an och 3:an. När utgången aktiveras så skickas en signal till relä som tänder lampan.



Figur 6 - Kretsschema över digital ut, lampa och relä

Figur 6 visar hur de olika komponenterna är kopplade i digitala utgångskortet.

EL3202 är en 2-kanalers analog ingångskort (se Figur 7).



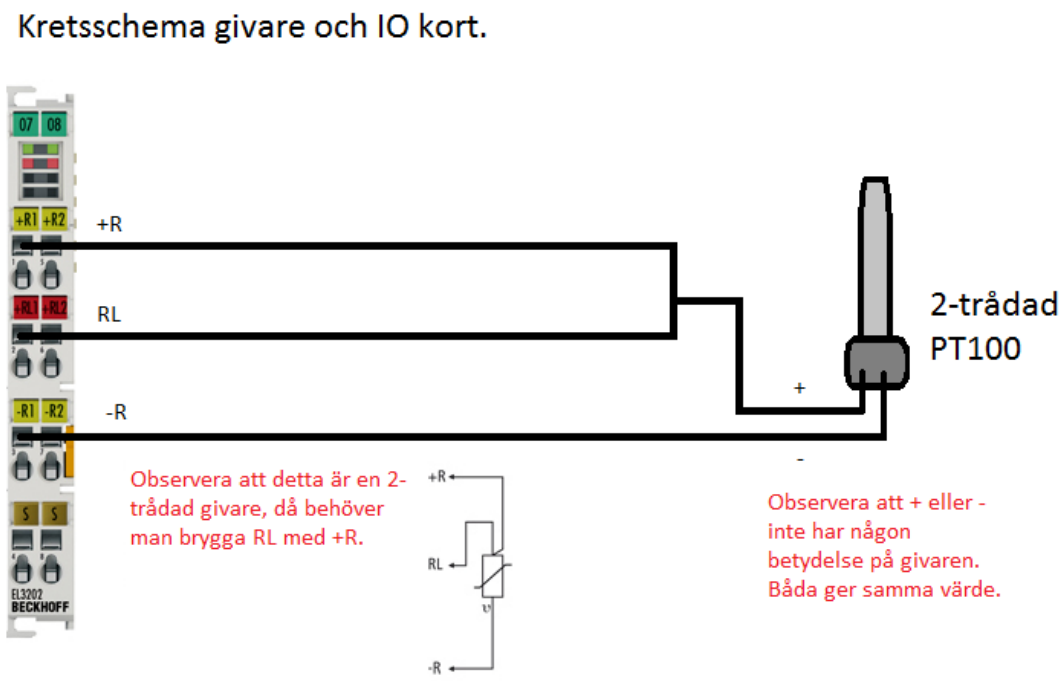
Figur 7 - EL3202, analog ingångskort som är anpassad för PT100 resistiv temperaturgivare. [7]

Inmatningsspänningen för detta kort är samma som för det digitala utgångskortet, dvs. genom E-bussens matningsspänning.

Figur 7 visar att man har olika ingångar (8st) på ett analog ingångskort. Temperaturgivaren kan vara två- eller tre-trådad och därför kopplas den på olika sätt. En tre-trådad givare kan kopplas in på ingångarna +R1, +RL1 och –R1 dvs. varje tråd kopplas mot en ingång. En 2-trådad givare kopplas på ett annat sätt. Ingång +R1 ansluts med +RL1 med hjälp av en brygga (kabel) och sedan kopplas den ena tråden från temperaturgivaren till bryggan (dvs. med +R1 och +RL1) och den andra tråden kopplas med –R1. I figur 7 visas även +R2, +RL2 och –R2; dessa används vid ett tillfälle där två stycken temperaturgivare används.

Se **Contact assembly** i figur 7.

Figur 8 visar en illustration på hur en tvåtrådad-temperaturgivare kopplas till e analog ingångskort.

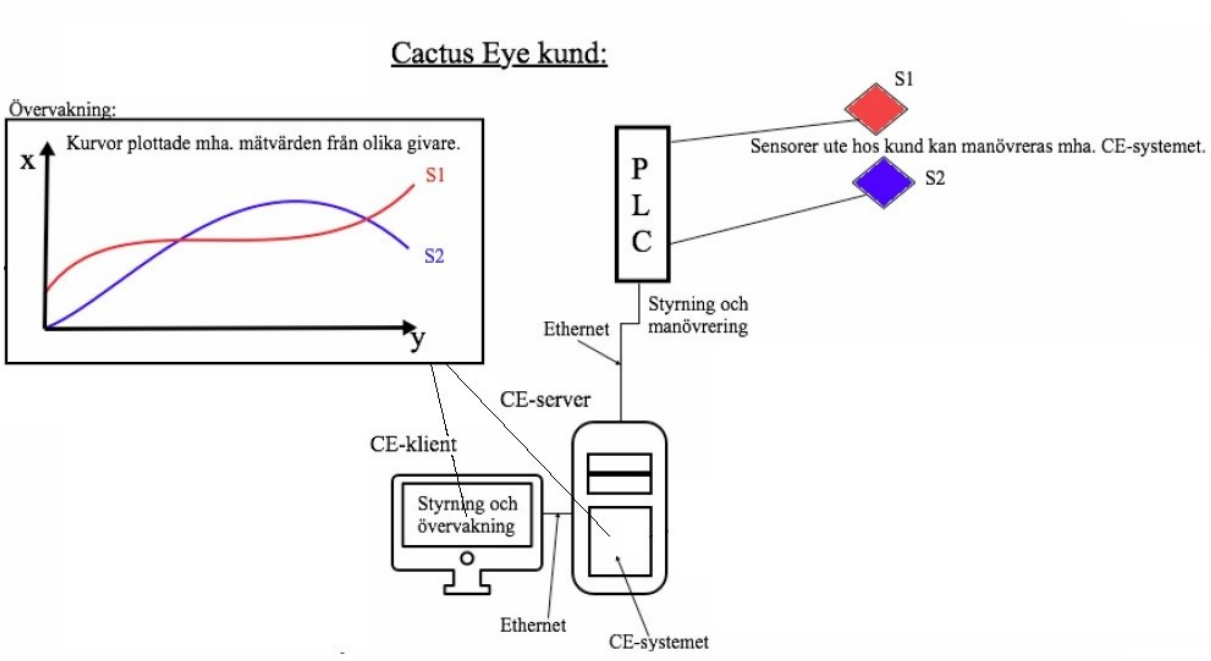


Figur 8 - Kretsschema över analog in.

2.2 Cactus Eye

I denna del av kapitlet ges övergripande information om Cactus Eye. Cactus Eye är en mjukvara som är kopplad till PLCer och kan användas för att styra PLCerna. Cactus Eye är ett mycket omfattande system och det faller utanför ramen för detta examensarbete att ge en fullständig beskrivning av det.

Cactus Eye används av företagets kunder. Det är ett Server-Klient system dvs. Cactus Eye-servern kan hanteras med hjälp av Cactus Eye-klient. I Cactus Eye-servern installeras hela datorsystemet och i klienten kan processbilder, kurvbilder, rapporter etc. visas och hanteras. Systemet hjälper kunderna med att övervaka och hantera stora processer som till exempel reningsverk. Figur 9 visar hur det kan se ut ute hos Cactus Eye kunder.



Figur 9 - Exempel på Cactus Eye server-klient ute hos kund.

Detta är ett exempel på hur Cactus Eye kan vara installerad hos en kund som har en tank med vatten i sin anläggning. När vattennivån sjunker och blir väldigt låg får kunden ett larm som säger att vattennivån är låg. Då kan kunden åtgärda problemet genom att styra vattenpumpen som fyller tanken med vatten. Cactus Eye gör att det blir enklare och effektivare för kunden; kunden hade möjlighet att övervaka samt styra sitt system. Kunden kan även ställa in olika funktioner som går per automatik. T.ex. kan vattenpumpen fylla tanken med vatten så fort Cactus Eye visar varningen för låg nivå.

Cactus Eye-kunder måste logga in i Cactus Eye-servern för att komma åt alla funktioner i Cactus Eye. Kunder kan ha olika behörighet beroende på vad de har beställt för tjänster. Kunder får alltså bara använda de funktioner de beställt och inte annat.

Cactus Eye har ett larmhanteringssystem som ger möjlighet till att välja prioritera olika larm med A, B eller C. A är högst prioriterad och C är lägst prioriterad. Hanteringen av dessa larm sker med både digitala och analoga signaler. Det finns larmlistor där det visas upp aktiva larm som måste kvitteras.

Användarna (kunden) kan rita processbilder. Processbilderna visar styrsystemets analoga och digitala signaler med symboler, staplar, knappar, kurvor, numeriska värden och texter. Objekten visas i en schematisk bild som beskriver hela anläggningar.

Det finns möjlighet att lägga till vyer som har olika funktioner sparade. Dessa vyer underlättar för användare att komma åt funktioner som används ofta av användarna.

Det finns även möjlighet att spara värden för analoga och digitala signaler, dessa värden sparas i en historikdatabas. Värden kan hämtas ut för att användas till analys för processförlopp, förbrukningar, flöden etc. Det finns ett antal olika funktioner i samband med hanteringen av värden, till exempel matematiska funktioner som det fyra räknesätten. Värdena kan exporteras respektive skrivas ut i form av en rapport.

Driften sparas i en så kallad driftloggbok, där det visas en lista med noteringar som gjorts i systemet.

Cactus Eye användare har möjlighet att hantera och påverka tabellerna inne i ett styrsystem (PLC), det utförs med hjälp av DCS-hantering. Det går att skriva kod till PLCerna direkt genom att öppna fönstret för DCS-hantering inne i Cactus Eye systemet.

Det finns olika DCS-funktioner som uppladdning av cykeltider, uppladdning av tabellkonfiguration och uppladdning av SPRS-kod till PLCer direkt från servern. SPRS-funktioner kommer att beskrivas ännu mer i avsnitt 2.4.3 Funktioner. [16]

2.3 Cactus Eye In- och Utgångar – typer av signaler

Det finns olika typer av signaler som kan skapas i Cactus Eye, dessa signaler kan vara analoga, digitala eller logiska. Digitala signaler representerar sekvenser av tal eller symboler. Logiska signaler är digitala signaler som endast antar värdena 0 eller 1, oftast tolkat som från eller till alternativt falsk eller sann. Dessa typer behövs för att skilja olika signaler som används vid mätningar och beräkningar i Cactus Eye.

I tabell 1 visas olika signaltyper som kan användas i Cactus Eye.

LS	Digitala insignaler, dessa är digitala ingångar till styrsystem, förändringar skickas till CactusEye vid standard händelseavfrågning.
LI	Digitala larm skapade av styrsystemet, förändringarna av dessa skickas som LS.
LU	Logiska manövreringssignaler.
LX	Logiska hjälpsignaler.
UT	Digitala utgångar.
MS	Analoga ingångarna i A/D-enheter.
VI	Analoga helhetshjälpsignaler.
VU	Heltalsparametrar.
VX	Heltalsignaler.
MU	Analoga utgångar i A/D-enheter.
RM	Analoga ingångar, skalade.
RI	Analoga korttidsmedelvärden, skalade.
RU	Analoga utgångar, skalade.
RX	Analoga flyttalhjälpsignaler.
HA	Händelseavfrågning.

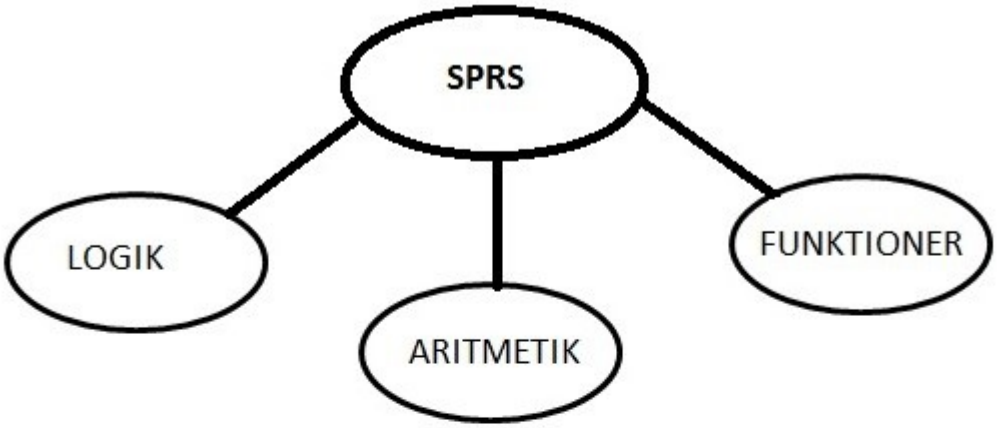
Tabell 1 - Signaltyper i Cactus Eye

2.4 SPRS – Språk för process, reglering och styrning

Sspråk för **P**rocess, **R**eglering och **S**tyrning, så kallad **SPRS.** Det är ett speciell utvecklat språk som används för processdatormiljö. SPRS utvecklades av Cactus Uniview och har använts sedan företaget startades.

Det är enkelt att lära sig detta programmeringsspråk eftersom det baseras på klartextbeskrivningar av styr- och reglerförloppen i processen. Utvecklingen i språket görs centralt på Cactus Eye. På servern finns det tillgång till texteditor.

SPRS-språket innehåller två språkelement - operatorer och funktioner. Operatorer är av två slag - logiska och aritmetiska, se figur 10.



LOGISKA OPERATIONER	Och, eller, icke, skild från (exklusivt eller)
ARITMETISKA OPERATIONER	Addition, subtrktion, multipikation, division
FUNKTIONER	Exekveringinterval, sekvenslogik, gränsvärdefunktioner, pulsräknare, regulatorer, pulsutgångar, drifttids-räknare, konvertering mellan heltal och flyttal. (timers)

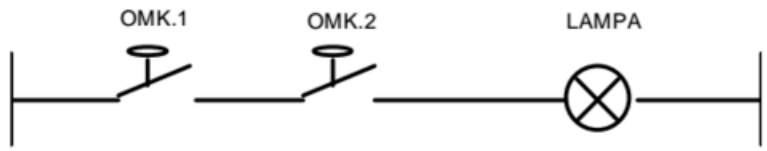
Figur 10 - SPRS består av tre element.

2.4.1 Logiska operatorer

Följande logiska operatorer finns tillgängliga:

+	Eller
*	Och
=	Tilldela, lagra
#	Exklusivt eller
-	Icke
( )	Parenteser bestämmer beräkningsordning

Exempel 1.
Två seriekopplade omkopplare i serie med lampa, se figur 11.



Figur 11 - Två omkopplare och en lampa i serie.

I SPRS skrivs denna koppling så här:

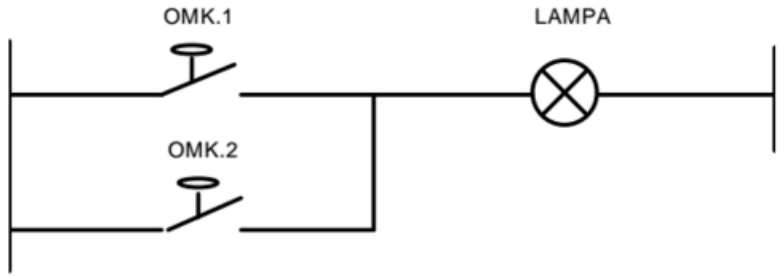
```
Lampa = OMK.1*OMK.2
```

Omk.1 och omk.2 är två variabler, de har tillstånden Från och Till.

Tolkningen till föregående SPRS kod är följande:

Om ingångarna (variablerna) är spänningssatta dvs. om de är båda till så ska variabeln till lampan tilldelas värdet "till" dvs. utgången aktiveras och tänds lampan.

Exempel 2.
Två parallellkopplade omkopplare i serie med en lampa, se figur 12.



Figur 12 - Parallellkopplade omkopplare i serie med lampa.

I SPRS skrivs denna koppling så här:

```
Lampa = OMK.1+OMK.2
```

Omk.1 och omk.2 är två variabler, de har tillstånden Från och Till.

Tolkningen till föregående SPRS kod är följande:

Om minst en av ingångarna (variablerna) är spänningssatt dvs. om de är en av dem är till så ska variabeln till lampan tilldelas värdet "till" dvs. lampan tänds.

2.4.2 Aritmetiska operatorer

Följande **operatorer** finns:

+	Addera
*	Multiplicera
=	Tilldela, lagra
/	Dividera
-	Subtrahera

Operatorerna ovan är våra vanliga fyra räknesätt som i matematiken. Beräkningarna utförs alltid från vänster till höger. T.ex. är A=B+C*D, då innebär det att A tilldelas summan B+C multiplicerad med D.

Exempel på olika aritmetiska operatorer:

```
RÅVARA = TOTVIKT - TANKVIKT
VOLYM = FLÖDE*TID
DOS = MÄNGD/VOLYM
```

Andra restriktioner för aritmetiska ekvationer är följande:
Det finns ett antal restriktioner för aritmetiska uttryck som är enligt följande:

- Parenteser fungerar inte.
- Ingående variabler i uttrycket måste vara av samma typ (det finns olika typer som heltal, flyttal, decimaltal etc.).
- Beräkningar med decimaltal (flyttalsberäkningar) kan utföras endast på centralserver och i vissa typer av styrsystem (PLC).

2.4.3 Funktioner

I Cactus Eye systemet finns det en datafil, FUNKTIONER. I denna fil finns alla fördefinierade programfunktioner definierade, varje funktion har ett symboliskt namn som används i styrprogrammet.

Exempel på olika funktioner är:

- Exekveringsintervall
- Sekvenslogik
- Gränsvärden
- Pulsfunktion
- Pulsräknare
- Drifttidsregistrering

2.4.4 Konstanter i SPRS

SPRS har ett antal konstanter som kan användas godtyckligt, det finns logiska, flyttalskonstanter och heltalskonstanter.
Logiska konstanter deklareras med "¤" som konstantdefinition för logiska värden 1 och 0.

`TILL=¤1`	Innebär att variabel TILL sätts till 1
`FRÅN=¤0`	Innebär att variabel FRÅN sätts till 0

Heltalskonstanter deklareras med “£”.

```
POS1=IN*£1024          Variabel POS1 antar värdet för variabel IN * 1024
POS2=UT*£-512          Variabel POS2 antar värdet för variabel UT * -512
```

Flyttalskonstanter deklareras med “**§**”för flyttalsvärde “n”.

```
BRÄDD1=IN*§3.14    Variabel BRÄDD1 antar värdet för variabel IN * 3.14
BRÄDD2=UT*§-7.0    Variabel BRÄDD2 antar värdet för variabel UT * -7.0
```

2.4.5 Macro i SPRS

Det finns oftast flera olika objekt som styrs och övervakas på samma logiska sätt. Sådana objekt är larmövervakning av ventiler och motorer. För sådana objekt kan en styrkod utnyttjas, så kallad **Macro**. Denna teknik eller **styrkod** gör att flera objekt styrs och har samma programlogik för att styras eller övervakas. Macro definieras med ett unikt namn och anropas med ett antal parametrar. Macro används inte i detta examensarbete och förklaras därför inte mer.

2.4.6 Specialtecken i SPRS

Dessa specialtecken ingår i SPRS:

```
$I      Identifiering av styrsystem.
$G      Styrsystemsgruppering, om det används samma kod för flera styrsystem.
$T      Projektidentifikation, skapar identifikation för projektet.
$S      Rubrik för programmodul (källkod).
$M      Anrop av macro.
```

Specialtecken skrivs alltid överst i en programkod.

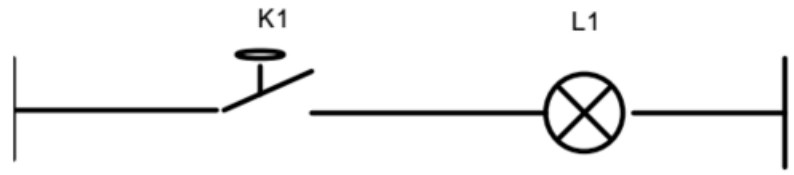
2.4.7 Ytterligare några begränsningar i SPRS är:

- Mellanslag får inte användas förrän på slutet av raden.
- Semikolon (;) används för att separera programtext från kommentarstext.
- Maximal radlängd är 78 tecken. Om ett uttryck blir mer än 78 tecken, måste man utnyttja hjälpvariabel för att lagra “mellanresultatet”.
- Rader tolkas som ett aritmetiskt eller logiskt uttryck beroende på om vänsterledet utgörs av en aritmetisk eller logisk variabel alternativt funktion.

2.4.8 Ett exempel på ett enkelt SPRS-program

I detta exempel visas en enkel sprs-kod med två logiska variabler K1 och L1.

Följande är el-schemat:



Figur 13 - Slutande kontakt och lampa.

När kontakten K1 sluts så tänds lampan, se figur 13.

Programmet som beskriver detta skrivs så här:

1. Formulera problemet i SPRS-språket:

   ```
   L1=K1
   ```

2. Kontrollera om variablerna finns i styrsystemet.
3. Skapa en källkodsfil med texteditorn enligt:

```
$I PLC_ID

$S Testprogram

EXEC1; Programmet genomlöps varje sekund

L1=K1

;Tänd lampa
;Slut
```

Observera att PLCn definieras med raden $I Understation_beteckning, om det testprogrammet skall köras lokalt i dator. Tecknet (;) skrivs före kommentarer.

4. Kompilera källkodsfilen, Cactus Eye har en SPRS-kompilator som översätter källkoden till körbart program i styrsystemet. Om inga fel så installeras källkoden i PLCn.

2.5 Övriga komponenter

2.5.1 Billampa

Billampan har matningsspänningen 24 VDC med effekten 70 W. Billampan har högre temperatur än vanliga glödlampor. Meningen är att mäta denna temperatur med hjälp av temperaturgivarna. För en närmare beskrivning av billampan se [9].

Se bilaga 2 för en bild av billampan.

2.5.2 Temperaturgivare PT-100

Temperaturgivarens kapsling är tillverkad av rostfritt stål. Det är en kabeltemperaturgivare som har en 2-ledaranslutning. Den mäter temperaturer mellan -50 och 180 °C. Denna sorts givare används främst för miljöer där mätning sker på gaser och vätskor men i examensarbetet användes de till att mäta temperaturen hos en billampa. För en närmare beskrivning av givaren se [10].

Se bilaga 3.

3 Metod

Detta kapitel kommer att beskriva projektets tillvägagångssätt, planering och faktainsamling. Det kommer även att beskriva vilka metoder som använts för att uppnå projektets mål och syften samt litteratursökningen och insamlingen av data.

3.1 Tillvägagångssätt

Projektet påbörjades med en introduktion från Cactus Uniview om Cactus Eye-systemet. Introduktionen gav en inblick i hur systemet fungerar. Efter introduktionen påbörjades en datainsamling av grundläggande information om Cactus Eye. Tillsammans med handledaren och testchefen bestämdes vilken utrustning som behövdes till testmiljön. Utrustningen valdes utifrån budgeten och arbetsplatsen som var avsedd för testmiljön. Utrustningen som behövdes till detta projekt var en rack med minst en PLC, två temperaturgivare och en billampa. Cactus Uniview i Helsingborg hade inte möjlighet att erbjuda oss handledning till projektet, utan vi hänvisades till ingenjörer från huvudkontoret i Göteborg som var tillgänglig via telefonkontakt och mail.

3.2 Planering

Planeringen baserade sig på ett så kallat Gantt-schema vilket är ett flödesschema som används i projektledning för att beskriva ett projekts olika faser. Examensarbetet delades på olika faser. Vissa aktiviteter kunde köras parallellt som faktainsamling/ behovsbedömning och utvärdering av tekniska system. Se figur 14.

Figur 14 - Tidsplanering enligt Gantt-schema.

För faktainsamling och behovsbedömning behövdes tre heltidsveckor totalt men de veckorna delades på halvtid pga. tidsbrist från företaget och examensarbetarna. Den fasen (faktainsamling/ behovsbedömning) gick åt att samla information om Cactus Eye systemet och om vilka utrustning som ska köpas in för projektet.

Resultatet från utvärderingen och faktainsamlingen kunde användas vid utförandet och konstruktionen. Andra steg som kördes parallellt är implementeringen av den valda lösningen samt styrning och uppföljning. Fel som uppstod kunde följas upp och korrigeras.

Den resterande tiden, dvs. 12 veckor har gått åt till olika saker så som:

- Läsa och skaffa en övergripande förståelse av Cactus Eye.
- Uppsättning av testmiljö (sätta ihop testmiljöns olika delar).
- Implementation av lösning.
- Testning och uppföljning.

Rapporten skrevs under hela projektets gång.

3.3 Faktainsamling

Examensarbetet är tidsbegränsat och därför är det väldigt viktigt att använda korrekt information, information som ger nyttig kunskap för detta examensarbete. Därför behövs pålitliga faktakällor eftersom det kan kosta tid om fel information används. Fakta för detta examensarbete insamlades ur manualer och annan dokumentation från företaget Cactus Uniview.

Tillgängligt var också en biblioteksliknande hemsida med möjlighet att söka efter tips, manualer och länkar om Cactus Uniview.

3.4 Litteratursökning

Ett antal presentationer, webbsidor, böcker och kompendier har använts som underlag till detta arbete. Den största delen av litteraturen var interna källor tillhandahållna av Cactus Uniview, som kan betraktas som en pålitlig källa. Dessa material är inte tillgängliga för allmänheten utan enbart för anställda i Cactus Uniview. Litteratursökningen utfördes parallellt med ett antal utbildningar som anordnades av företaget Cactus.

3.5 Insamling av data

All data som har använts för testerna är hämtade från riktiga temperaturgivare som monterats i den nya testmiljön.

3.6 Utbildning

Cactus Uniview anordnade ett antal utbildningar som gav en grundläggande överblick över systemet Cactus Eye. Cactus Eye används inte av medarbetarna i Helsingborg. Därför var det viktigt att examensarbetarna fick en introduktionsutbildning för Cactus Eye för att kunna hantera eventuella problem som kunde uppstå vid projektets arbetsgång.

3.7 Arbetsplats och arbetsbeskrivning

Hela arbetet utfördes i företaget i Helsingborg. Arbetet började i början av februari 2016 och slutade i slutet av maj 2016. Två veckor gick åt att få en inblick över SCADA-systemet genom att läsa olika dokument.

Arbetet började först med att programmera en PLC som företaget använde för sin utbildning. Arbetet med testmiljön påbörjades inte förrän flera veckor senare då mycket tid gick åt att beställa utrustning till testmiljön.

Under arbetets gång hade examensarbetarna en kontinuerlig kontakt med handledaren, vilket var till stor hjälp då detta kunde lösa eventuella funderingar kring examensarbetet i tid, vilket undvek onödig uppkomst av problem. Ett schema gjordes i början av arbetets gång för att kunna planera den avsedda tiden.

3.8 Källkritik

Det mesta av materialet, som nämnts ovan är från Cactus Uniview. Inga källor som Wikipedia har använts i detta examensarbete, därför anses källorna som använts i denna rapport tillförlitliga.

Källorna [2], [3], [6]- [12] är pålitliga källor eftersom dessa källor är dels från komponenternas tillverkare och dels från leverantörerna av komponenterna. [5] och [14] är

också pålitliga eftersom de kommer från Cactus Univiews hemsida som alla kan komma åt. [1] är pålitlig eftersom Centino är ett företag som har HMI som ett av deras fyra expertområden. [4] och [13] är pålitliga eftersom de använder källor från företag som är världsledande inom denna bransch (Automation).

4 Analys

I denna del av rapporten beskrivs SCADA-systemet Cactus Eye, den gamla testmiljön och den nya testmiljön.

4.1 Cactus Eye

Cactus Eye är ett driftdatorsystem (SCADA-system) där man kan övervaka anläggningar, prioritera händelser, sätta larm med olika prioritet, hantera realtidsinformation, övervaka och manövrera olika stationer etc. [4] och [5]

Cactus Uniview tycker det är viktigt att de testar de olika lösningarna som företaget vill sälja till sina kunder för att kunna minska risken för fel. Lösningarna skall köras och testas i Cactus Uniview i förebyggande syfte.

4.2 Den gamla testmiljön

Cactus Uniview utförde tester på Cactus Eye med hjälp av en mjukvarumässig testmiljö (den gamla testmiljön). Testarna kopplade upp sig till testmiljön via Remote Desktop eftersom testmiljön var installerad på en virtuell maskin med särskild IP-adress.

Den gamla testmiljön baserade sig på mätvärden som genereras med hjälp av mjukvara. Alla mätvärden var konstanta och fördefinierade. Nackdelen med konstanta mätvärden var att de inte kunde ändras. En nackdel till var att kurvorna som plottades på vissa processbilder var exakt lika.

Den gamla testmiljön var inte så effektiv då det gäller kommunikation. En av de största nackdelarna med den gamla testmiljön var att kommunikationen mellan PLC och Cactus Eye inte kunde testas. Det gäller även för kommunikationen mellan PLC och givare. Kommunikationen testas för första gången hos Cactus Univiews kunder.

4.3 Den nya testmiljön

Uppgiften var att bygga en fysisk testmiljö där testerna körs med hjälp av ett antal givare som monteras i en rack tillsammans med en PLC. Testerna blir mer realistiska, dvs. mätvärdena kommer från riktiga givare. Test-avdelningen på Cactus Uniview vill ha möjligheten att uppgradera miljön efter hand, det viktigaste är att bygga en fungerande miljö med få komponenter där tester kan utföras

Testmiljön skulle vara uppbyggd utifrån kravspecifikationer som planerats av examensarbetarna och företaget. Kravspecifikationerna utformades med hänsyn till de brister som fanns i den gamla testmiljön.

Med hjälp av den nya testmiljön kan kommunikation mellan styrsystem och Cactus Eye testas. Det går även att skriva kod till PLCerna genom DCS-hantering vilket ger möjligheten att utveckla virtuella signaler om det behövs.

Cactus Eye har en funktion som heter DCS-hantering (Distributed control systems). Denna funktion gör det möjligt att programmera PLCer i Cactus Eye. DCS-hantering finns inte i den virtuella testmiljön och därför ville företaget ha den funktionen integrerad i den fysiska testmiljön. Syftet är att hitta fel som uppstår vid programmering av styrsystem.

Den nya testmiljön byggdes utifrån följande kravspecifikationer:

- Miljön ska vara fysisk, dvs. komponenter som PLC och givare ska användas.
- I den fysiska testmiljön ska minst en PLC vara levererad av Beckhoff, för den säljs till Cactus Univiews kunder.
- Den fysiska testmiljön skall vara liten från början men enkel att utvidga.
- Testmiljön ska kunna återställas till ett specifikt läge.
- Systemdokumentation för den nya testmiljön ska tas fram.

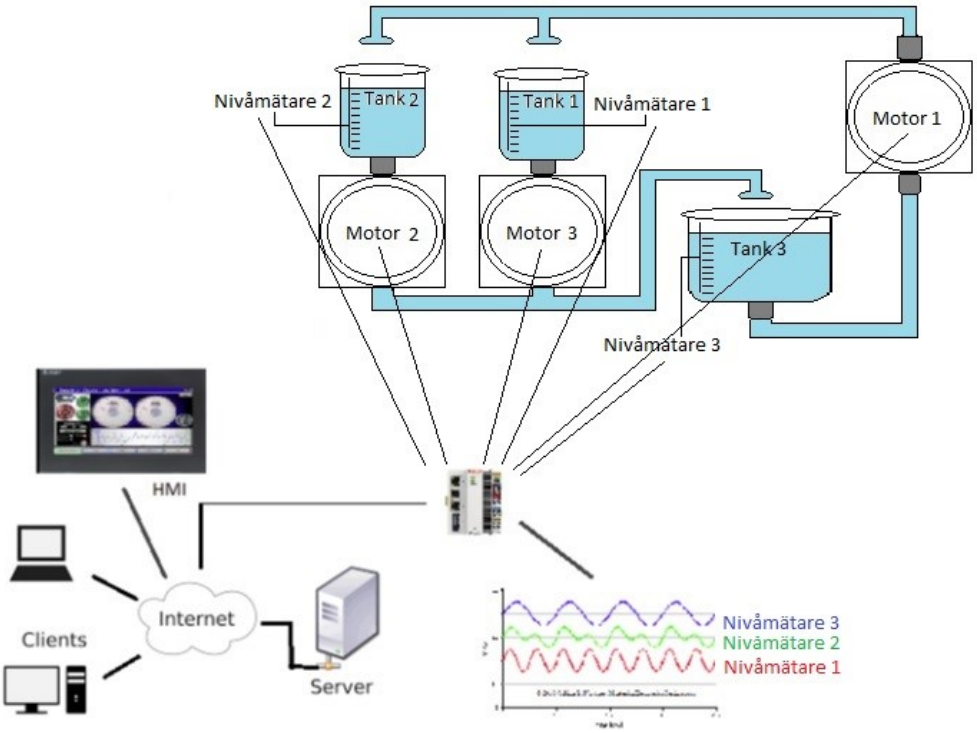
Planeringen av lösningen till den nya testmiljön utformades av examensarbetarna. Slutsatsen av denna planering gav tre olika alternativ. Se kapitel 4.4 ”Analys av den nya testmiljön”.

4.4 Analys av den nya testmiljön

Den här delen kommer att beskriva samt analysera de olika alternativen som examensarbetarna kommit fram till.

4.3.1 Alternativ 1: Styrning och reglering av vattentankar

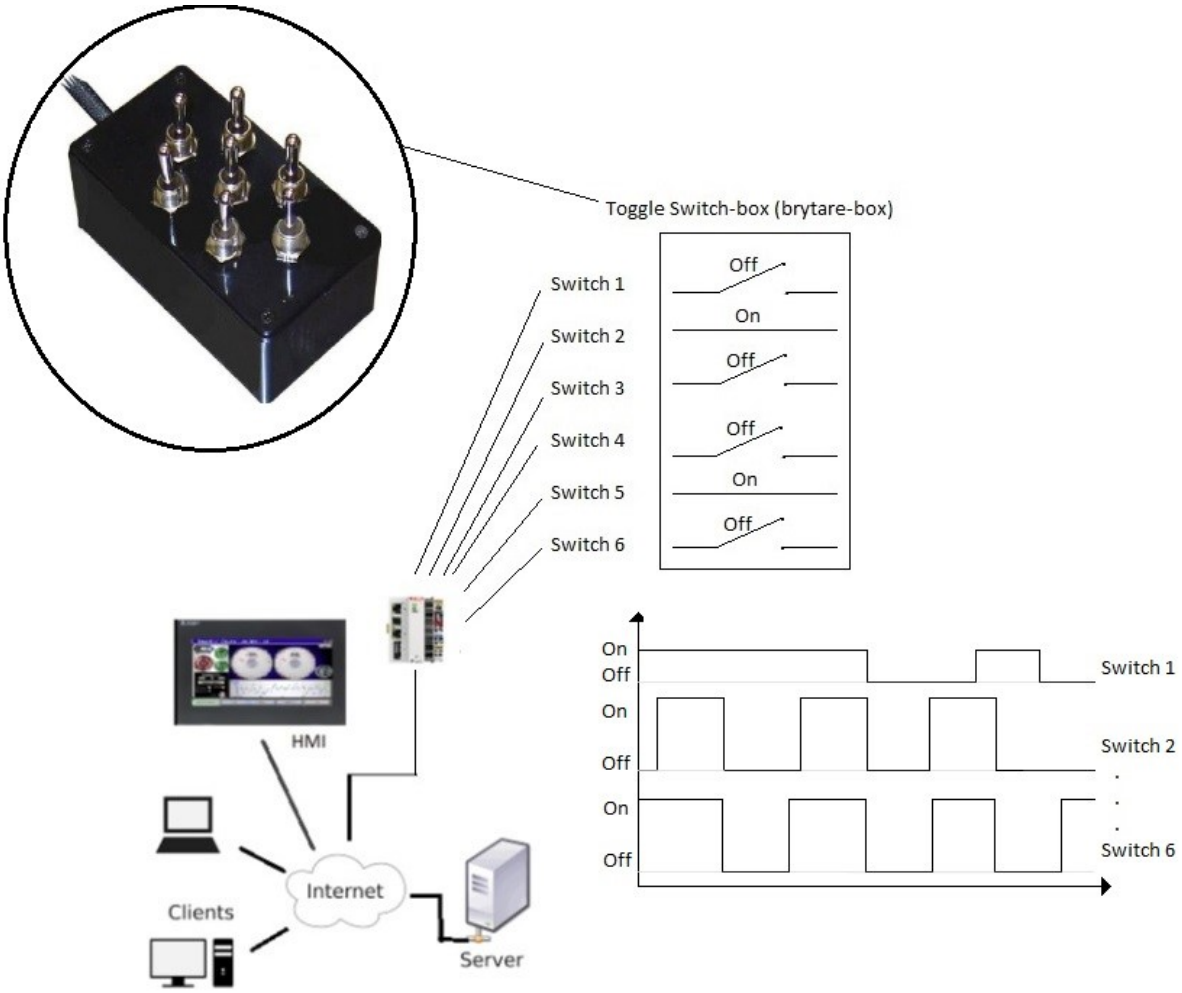
Alternativ 1 handlade om att styra och reglera olika vattentankar. Dels att kontrollera nivån på tankarna och dels tömma respektive fylla tankarna vid behov. Nivån på vattentankarna plottas i ett diagram. Styrning och reglering ska även ske via en HMI-panel. Se figur 15.



Figur 15 - Styrning och reglering av vattentankar.

4.3.2 Alternativ 2: Styrning och reglering med switchknappar

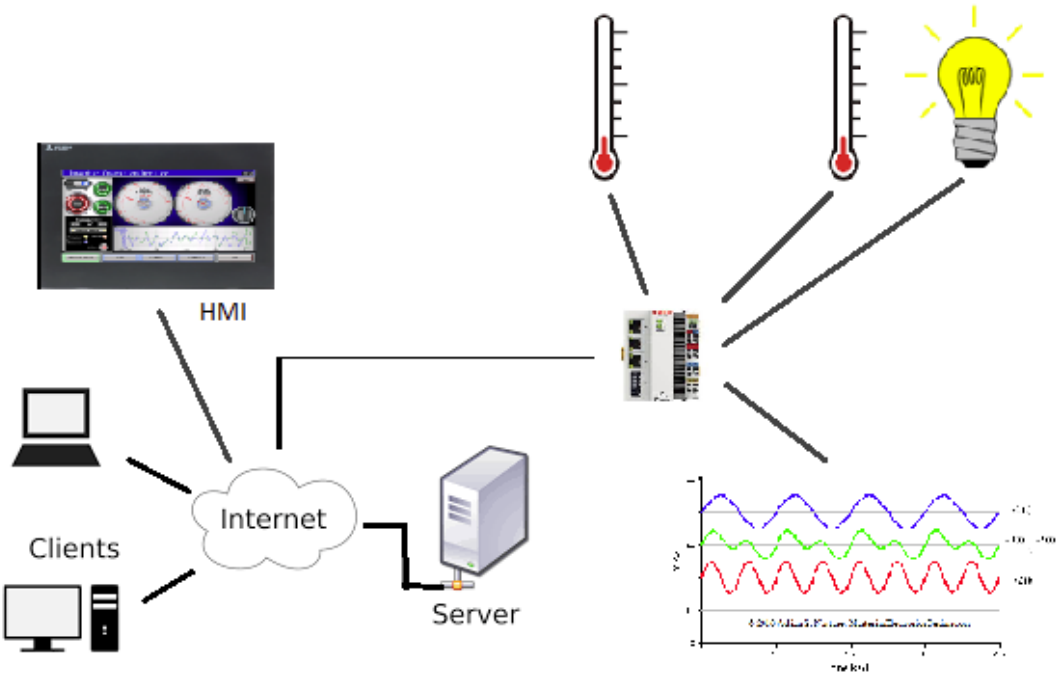
Alternativ 2 var nästan samma som alternativ 1 dvs. styra och reglera olika vattentankar men inte med verkliga objekt (vattentankar, motorer, mätare etc.) utan med switchknappar (brytare). Det ska finnas många knappar som ska ersätta olika lägen; t.ex. om en tank blir full då kan switchknapp ”Tank1Full” slås till. Ett annat exempel är ifall man vill slå på en motor genom att slå till en switchknapp ”Motor1Start” etc. Se figur 16.



Figur 16 – Styrning och reglering med brytare.

4.3.3 Alternativ 3: Temperaturmätning

Alternativ 3 handlar om att mäta olika temperaturer. Två temperaturmätare ska användas för att mäta temperaturer dels i ett rum och dels vid en billampa som slås på och av. Temperaturerna som mäts ska plottas i ett diagram som användare kan komma åt både via HMI-panel och via servern. Det ska även gå att styra billampan (slå på eller av) via HMI-panelen. Se figur 17.



Figur 17 - Den implementerade lösningen.

4.3.4 Analys av de olika alternativen

Alternativ 1 är en lösning som har komponenter som motorer, nivåmätare och tankar. Dessa komponenter kan styras och regleras. Testerna som kan utföras i denna lösning är fysiska. Utvecklare kan t.ex. starta motorerna, mäta vattennivån med hjälp av nivåmätarna etc. Därför anses denna lösning som ett bra alternativ till den nya testmiljön.

Alternativ 2 har inte många komponenter som t.ex. alternativ 1 men den har en switch-box (brytare-box) som kan användas. Brytarna ska föreställa olika lägen i ett system.

Alternativ 3 är en lösning som har ett antal komponenter som t.ex. temperaturgivare, billampa, likriktare etc. Dessa komponenter kan styras och regleras. Antalet komponenter i alternativ 2 är mindre än antalet komponenter i alternativ 1.

4.3.4.1 Det valda alternativet

Lösningen till testmiljön har tre faktorer enligt följande:

1. Kostnaden för testmiljön.
2. Testmiljön ska vara enkel men lätt att utvidga.
3. Verklighetstrogna tester.

Cactus Uniview har inte satt någon gräns på kostnader direkt men testmiljön ska inte vara dyr utan den ska ha ett rimligt pris.

Testmiljön ska vara enkel från början men lätt att utvidga. Den ska inte ha många komponenter som t.ex. motorer, givare och lampor etc.

Cactus Uniview vill att testerna som ska utföras ska vara verklighetsbaserade dvs. utvecklare ska kunna styra och reglera olika saker i testmiljön.

Examensarbetarna valde att redovisa de olika alternativen enligt tabell 2. Examensarbetarna och handledaren från Cactus Uniview har diskuterat resultatet i tabell 2 tillsammans.

	Alternativ 1	**Alternativ 2**	**Alternativ 3**
Kostnad	Dyr	Billig	Medel
Komponenter	Många	Få	Medel
Verklighetstrogna tester	Verkliga	Mindre verkliga	Verkliga

Tabell 2 - Jämförelse mellan olika alternativ.

Den valda lösningen för testmiljön är alternativ 3 eftersom kostnaden var medel jämfört med de andra alternativen. Den är även enkel från början och kan lätt utvidgas eftersom den har medel i komponenter i jämförelse med de andra alternativen. Testerna som ska köras i alternativ 3 anses vara verkliga i jämförelse med alternativ 2.

4.5 Den valda lösningen

Den valda lösningen undviker bristerna i den gamla testmiljön.

Som det nämnts ovan så valdes det att implementera en lösning till Cactus Eyes framtida testmiljö. Den här lösningen visas i figur 17. Den består av en server, en klient, en HMI-panel, en PLC, två stycken Pt-100 givare, en 70W halogen billampa och ett antal virtuella signaler.

Cactus Eye mjukvaran består av två delar, klient- och serverdel. Klient- och serverdelen installerades på servern (fysisk server). Klienten kan styra och övervaka de olika delarna i testmiljön genom Cactus Eye; dvs. från- och tillkoppling av lampan samt övervakning av de olika temperaturgivarna. Den ena temperaturgivaren mäter rumstemperaturen och den andra mäter temperaturen hos lampan.

4.6 Analys av hårdvara för servern

Cactus Uniview hade riktlinjer för hur kraftfull server man ska välja. Trots detta fick examensarbetarna uppgift att jämföra olika hårdvara som är mest lämpligast för servern så att den klarar Cactus Eye-systemet (server- och klientprogrammet).

Examensarbetarna löste uppgiften genom att presentera tre olika alternativ för Cactus Uniview. Cactus Uniview gav inga speciella krav för hur kraftfull servern ska vara och därför var det lite svårt att bestämma vilka alternativ som examensarbetarna skulle välja. Alternativen jämfördes beroende på kostnad och funktion. Specifikationen för de olika alternativen sammanställdes i tre olika tabeller. Cactus Uniview köpte sina IT-relaterade produkter från Dustin.se och därför valde examensarbetarna att köpa från samma leverantör.

4.6.1 Alternativ 1

Alternativ 1 var den kraftfullaste servern bland de som jämfördes. Servern är tillverkad av HP. Processorn är tillverkad av Intel och har 8-kärnor med frekvensen 2,1 – 3.0 GHz. Mer om specifikationen finns i tabell 3.

Modellnamn	Processormodell	Minnesstorlek (RAM)	Max. RAM-minne	Total Hårddiskkapacitet	Pris ekl. moms
HPE ProLiant ML350 Gen9	Intel E5-2620V4	16 GB	Upp till 1536 GB	600 GB	17995 Kr

Tabell 3 - Alternativ 1. [18]

4.6.2 Alternativ 2

Alternativ 2 var den näst kraftfullaste servern bland de som jämfördes. Servern är tillverkad av HP. Processorn är tillverkad av Intel och har 6-kärnor med frekvensen 1,9 GHz. Mer om specifikationen finns i tabell 4.

Modellnamn	Processormodell	Minnesstorlek (RAM)	Max. RAM-minne	Total Hårddiskkapacitet	Pris ekl. moms
HPE ProLiant DL180 Gen9	Intel E5-2609V3	8 GB	Upp till 256 GB	1024 GB	14995 Kr

Tabell 4 - Alternativ 2. [19]

4.6.3 Alternativ 3

Alternativ 3 var den minst kraftfulla servern bland de som jämfördes. Servern är tillverkad av HP. Processorn är tillverkad av Intel och har 2-kärnor med frekvensen 3.5 GHz. Mer om specifikationen finns i tabell 5.

Modellnamn	Processormodell	Minnesstorlek (RAM)	Max. RAM-minne	Total Hårddiskkapacitet	Pris ekl. moms
HPE ProLiant ML310e Gen8 v2	Intel I3-4150	4 GB	Upp till 256 GB	1024 GB	6995 Kr

Tabell 5 - Alternativ 3. [20]

4.6.4 Det valda alternativet

Cactus Uniview behövde en server som kunde klara av Cactus Eye-systemet. Systemet kan bli resurskrävande när det ska implementeras nya lösningar till nya projekt. Cactus Uniview krävde att RAM-minnet skulle minst vara 4 GB samt att det kan uppgraderas vid behov. De krävde även att hårddisken ska vara minst 512 GB.

Examensarbetarna valde att presentera alternativ 2 då den var kraftfull och uppfyllde alla behov. Alternativ 2 hade ett RAM-minne på 8 GB och en hårddisk på 1024 GB dvs. två gånger så stor som vad Cactus Uniview krävde. Servern kan uppgraderas vad gäller RAM-minnet och hårddisken. Denna server är tillverkad av HP. Den har en processor som är tillverkad av Intel med modellen E5-2609V3. Denna processor har nästan samma specifikation som i alternativ 1. Priset för denna server är 14995 kr (exkl. moms).

5 Utförande

I detta kapitel genomgås tillvägagångssättet steg för steg för den valda lösningen, hur arbetet planerades och vilken strategi som valts att användas.

På grund av säkerhet och sekretess kommer det inte finnas en förklaring på hur styrsystemet konfigurerades, då denna information endast får stanna inom Cactus Uniview eftersom den kan vara känslig.

5.1 Material till den nya testmiljön (fysiska testmiljön)

Cactus Eye fysiska testmiljö består utav ett par olika komponenter, se följande:

1. Stationär dator (Server) x 1st
2. Styrsystem (PLC) x 1st
3. I/O kort för Analoga ingångar x 1st
4. I/O kort för Digitala utgångar x 1st
5. Temperaturgivare (PT-100 av typen RTD) x 2st [10]
6. Billampa (24VDC, 70W) x 1 st
7. Relä (230V, 24V Spole) x 1 st
8. HMI-panel x 1st

5.2 Den nya testmiljön

I denna del av kapitel kommer flera olika moment av den nya testmiljön att beskrivas. Delarna är:

- Installation, konfiguration och montering av PLC.
- Vilka virtuella signaler som ska genereras.
- Återställning av testmiljö till känt läge.
- Systemdokumentation.
- Val av HMI.
- Överlämning av testmiljö.

5.2.1 Installation av Cactus Eye

Cactus Eye är ett SCADA-system som används för övervakning och manövrering av olika PLCer och deras givare. Detta system är av typen Klient-Server, dvs. en eller flera klienter kan ansluta mot en server för att kunna använda dess Cactus Eye och dess funktioner. [4]

Systemet installerades på en stationär dator (oftast skåpserver). Denna bör vara placerad på en skyddad plats, som t.ex. i ett serverrum. Mjukvaran för Cactus Eye, som var delad i två delar installerades på servern. Den ena delen är till servern och den andra till klienten. Cactus Uniview förberedde mjukvaran och dess manualer för examensarbetarna tillsammans med en beskrivning av installationen i detalj.

5.2.2 Konfiguration av PLC

PLCn är av typen Beckhoff CX8090 som det nämnts tidigare, den är modifierad av Cactus Uniview och efter modifieringen kallas den Cactus C10. En fördel med denna PLC är att utvecklare/ kund själv kan välja antalet in- och utgångar, de finns i form av I/O – kort. [6]

Det valdes två olika I/O-kort till testmiljön, ett I/O-kort till analoga ingångar och ett annat till digitala utgångar. För att konfigurera dessa kort behöver de först monteras i PLCn (se kapitel 2.1.1.2 angående I/O-kort).

PLCn kopplades mot Cactus Eye (server) via Ethernet, därför behövdes en IP-adress till PLCn. IP-adressen angavs i PLCns operativsystem. Det finns ett program för detta, Certhost. Detta program används för modifierade PLCer som tillhör Cactus Uniview. Certhost gör att utvecklare kan konfigurera PLCn och ange olika inställningar, bland annat nätverksinställningar.

5.2.3 Kommunikation mellan PLC och Cactus Eye

En mall finns för konfigurationen av denna kommunikation i manualer som tillhandahållits av företaget. Dessa manualer har en beskrivning på de olika parametrarna som behövs ställas in för att få upp kopplingen. Manualerna är sekretessbelagda och kan därför inte bifogas i denna rapport. Mallen följdes till punktligt och kommunikationen mellan PLCn och Cactus Eye lyckades.

5.2.4 Montering av komponenterna

De komponenter som monterades är nedanstående:

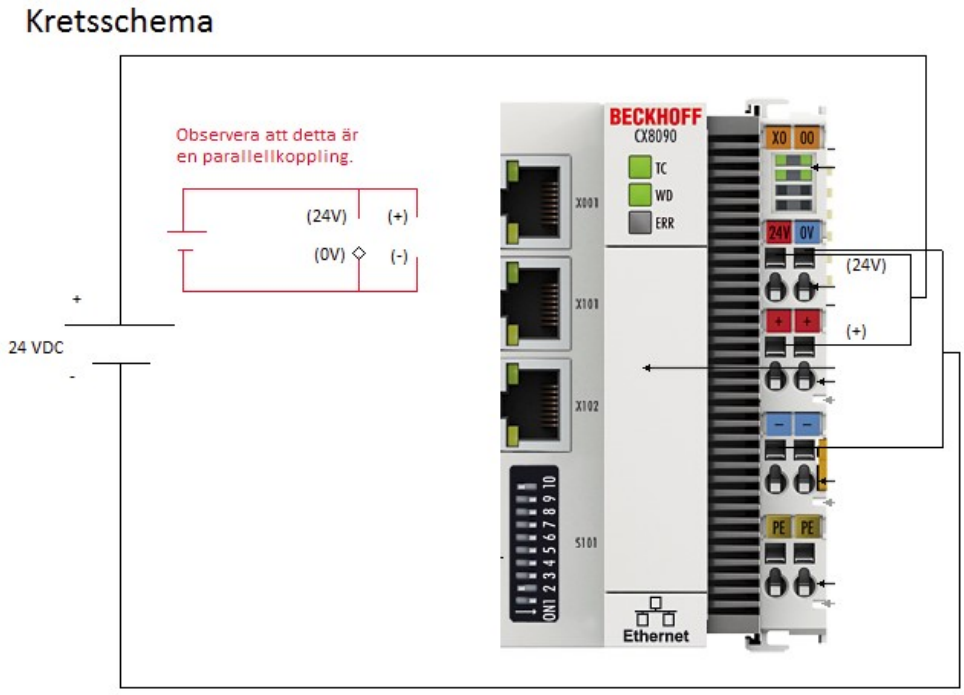
- 2 x likriktare.
- 1 x PLC.
- 1 x I/O – kort, analog ingång.
- 1 x I/O – kort, digital utgång.
- 2 x temperaturgivare.
- 1 x billampa.
- 1 x relä.
- 1 x HMI-panel.

HMI-panelen är avsedd att användas för manövrering av billampan och övervakning av temperaturerna i form av kurvor. Se bilaga 6 och 7.

Komponenterna monterades i en flyttbar rack med hjul, för den var mer användbar och gav möjligheten att flytta till olika platser där det var tänkt att användas. Den fysiska miljön är lätt att utvidga då det finns plats för flera komponenter på racket.

Först monterades likriktarna, PLCn, temperaturgivarna, billampan och relä på racken. PLCn kopplades till en likriktare som ger ut 24VDC med max 5A ut, se kretsschema i figur 16. [1]

Se bilaga 4 för att se likriktaren.

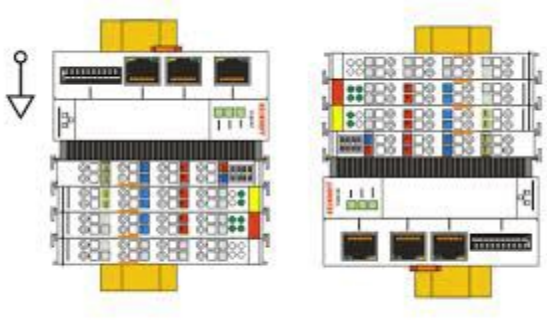


Figur 18 - Inkopplingsschema över styrsystemet.

Kretsschemat i figur 18 visar hur PLCn får sin spänning genom likriktaren. Plintarna (+) och (–) från likriktaren mot PLCn ingångar. Ingångarna i PLCn (24V ingången) och E-bussen (+) spänningssätts parallellt från likriktaren. 24V-ingången matar PLCn och (+)-ingången matar I/O-korten som monteras på PLCn.

Vidare till I/O – korten, de monteras i PLCn. Det finns en fackliknande hållare på varje PLC, se figur 19.

Figur 19 - Beckhoff CX-8090 (Cactus C10).



Figur 20 - PLC med I/O-kort monterade.

Figur 20 visar ett exempel på hur PLCn ser ut med ett antal I/O-kort monterade. Det finns en särskild instruktion som måste följas ifall det ska monteras fler än två I/O-kort. I/O-korten måste kopplas ihop sorterade i en bestämd ordning. Sorteringen behövde inte genomföras då testmiljön inte krävde mer än två I/O kort, dvs. korten kan placeras hur som helst.

5.2.5 Virtuella signaler

Med virtuella signaler menas att signaler som genereras (utvecklas) med hjälp av SPRS programmering i Cactus Eye-systemet. Sådana signaler kan användas i samband med olika tester som utförs av utvecklarna i Cactus Uniview. Fiktiva signaler utvecklas som ska efterlikna verkliga mätvärden från givare. Signalerna kan användas när utvecklare inte har tillräckligt med sensorer i den fysiska testmiljön.

En del av arbetet var att generera/ utveckla några olika signaler, sågtandssignal, triangelvåg och en fyrkantsvåg. Examensarbetarna lyckades utveckla dessa signaler.

Programmeringen av dessa signaler skede med hjälp av SPRS-språket, vilket skiljer sig från andra programmeringsspråk som t.ex. Java eller C. Se avsnitt ”2.4 SPRS- språk för process, reglering och styrning” för en närmare beskrivning av SPRS-språket.

För att se signalernas utseende se bilaga 10, 11 och 12 i appendix.

5.3 Återställning

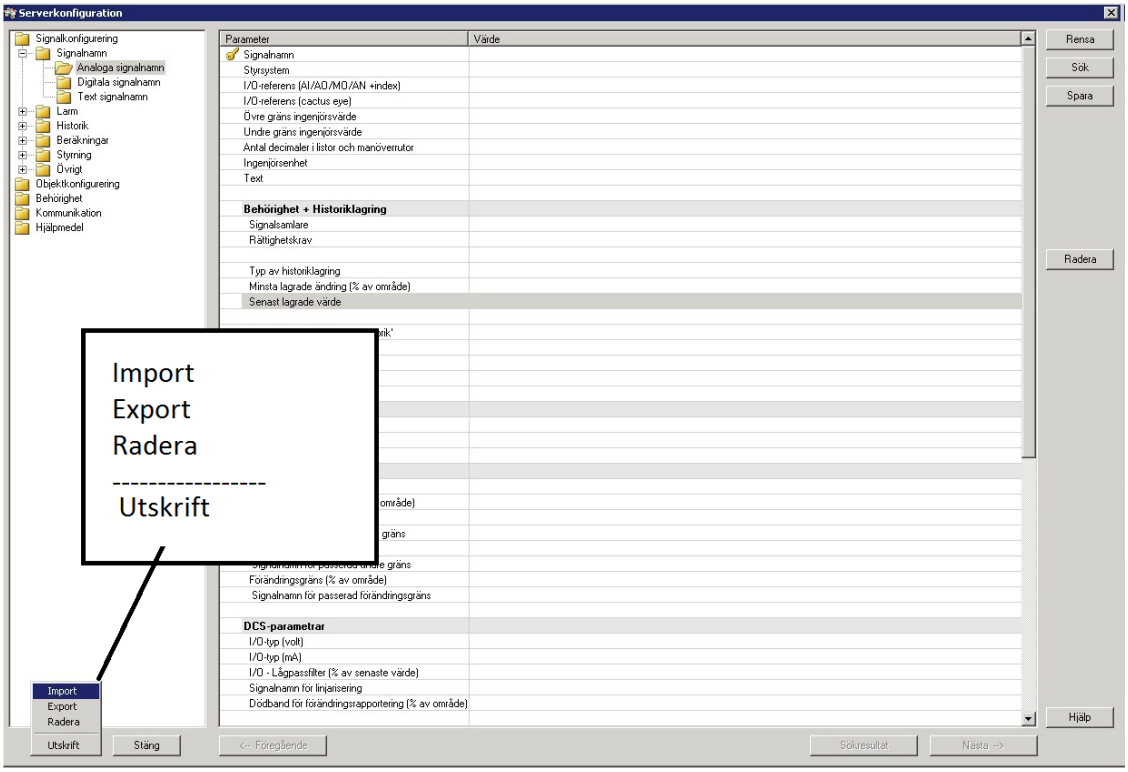
Utvecklare ska kunna återställa testmiljön till ett känt läge. Då rensar utvecklaren bort all data som använts till det slutförda projektet, bland annat signaler, objekt, larm, nycklar etc.

Handledaren på Cactus Uniview krävde att denna lösning skulle utformas på följande sätt. Återställningen för systemet ska ske via en knapptryckning. Handledaren menade att så fort en utvecklare trycker på knappen för återställning så ska data säkerhetskopieras och sparas i en speciell disk som är endast för Backup. Efter säkerhetskopiering slutförts ska systemet återställas till ett känt läge. Examensarbetarna tänkte lösa detta från början men sist efter att de pratat med andra ingenjörer så upptäckte examensarbetarna att detta inte kunde genomföras eftersom examensarbetarna inte fick ändra något i systemets gränssnitt.

Återställningen kunde lösas på olika sätt, t.ex. genom att köra en Backup på hela servern vilket gör att utvecklaren har en kopia på allt i systemet. Sedan kan utvecklaren använda backupen för att återställa servern till det läge man vill ha. Nackdelen med detta är att utvecklaren kommer ta kopia på mycket ”onödiga” filer så som t.ex. Windows OS, program (som inte tillhör testmiljön) och annat. För att lösa detta på ett effektivare sätt så finns det en

backupfunktion i Cactus Eye som utnyttjades. Backupfunktionen användes för att exportera databaser innehållande data om det aktuella projektet. Exempel på sådana data är signaler, konfiguration av kommunikationsdata, bildprocesser, behörighet osv. Examensarbetarna skapade en mapp på datorn och namngav den ”Backup Testmiljö”. Till denna mapp exporterades allt nödvändigt från Cactus Eye vid en återställningspunkt. Inne i denna mapp kan utvecklare ha undermappar med olika namn beroende på vad utvecklaren vill exportera.

Cactus Eyes backupfunktion exporterar till Excel-filer, varje fil kommer att ha många blad fyllda med data. Utvecklare ska inte skriva eller ändra något på dessa filer då det finns risk att de förlorar viktiga saker. Filerna kan användas ifall utvecklare vill återställa Cactus Eye systemet, genom att importera dessa filer direkt i Cactus Eye. Dessa backupfiler ska skrivskyddas och sparas på ett säkert sätt. Se figur 21. Examensarbetarna testade återställningen genom att gå in i Cactus Eye installationsmapp och radera databaserna för projektet manuellt. Efter att databaserna hade raderats så importerades databaserna (Excel-filer som togs backup på) genom Cactus Eye funktionen. Export och import av databaser (Excel-filer) lyckades och var en bra lösning enligt Cactus Uniview.



Figur 21 - Import resp. export av data.

5.4 Systemdokumentation

En annan viktig del av detta examensarbete var att skriva systemdokumentation för testmiljön, dessa dokument skall användas av andra utvecklare när de t.ex. vill sätta upp nya PLCer.

Det skrevs ett antal systemdokument för detta projekt vilka beskriver:

- PLC-konfiguration och mappningar för I/O-kort.
- Montering av PLC, objekt och givare.
- Kommunikation mellan styrsystemet och Cactus Eye.
- Signaler, analoga och digitala.
- DCS och SPRS, programmering av PLC med Cactus Eye.
- Signaltyper, de signaler/ tabeller som stöds av Cactus Eye.
- Virtuella signaler, kurvor man skapar med hjälp av SPRS-programmering.
- Backup och återställning.

Dessa systemdokument skrevs i samband med arbetet som utfördes. Det skrevs som en instruktion på hur arbetet bör utföras. Detta kommer underlätta för personen som läser dokumentet att utföra samma arbete.

Dessa systemdokument kommer inte bifogas i appendix då det kan vara känsligt för Cactus Uniview.

5.5 Val av HMI

HMI står för Human Machine Interface, det är ett gränssnitt mellan människa och maskin. HMI används oftast inom industri- och automationssammanhang. HMI kan utvecklas och installeras på paneler, mobiler, datorer, plattor etc. HMI kan övervaka och manövrera maskiner man kopplat den mot. [1]

Testmiljön består av ett antal komponenter som kan övervakas och styras. Det valdes en HMI-panel som skulle användas till att övervaka temperaturgivarna och styra lampan. Genom denna panel ska examensarbetarna ha koll på testmiljöns olika komponenter.

Examensarbetarna hade inte tillräckligt med tid för att göra en fullständig utredning av vilken HMI-panel som ska väljas till testmiljön. Men det gjordes en jämförelse av olika alternativ som examensarbetarna ansåg vara lämpliga. Denna jämförelse hade tre olika faktorer som påverkade valet av HMI-panelen. Faktorerna är enligt följande:

1. Kostnaden för HMI.
2. Funktioner som används i testmiljön.
3. Lättanvänd.

Cactus Uniview har inte satt någon gräns på kostnaden. Men panelen ska inte vara dyr, den ska ha ett rimligt pris.

Panelen ska vara lätt att använda och ha funktioner som vara anpassad till vår testmiljö.

Cactus Uniview vill att panelen ska vara tillverkad av Mitsubishi. Anledningen till detta är att Cactus Uniview har stor erfarenhet av att använda dessa paneler.

Examensarbetarna valde att redovisa de olika alternativen enligt tabell 6. Examensarbetarna och handledaren från Cactus Uniview har diskuterat resultatet i tabell 6.

	GOT 2000 Series	GOT 1000 Series	GOT Simple
Kostnad	Dyr	Medel	Billig
Funktioner som används i testmiljön	För många funktioner	För många funktioner	Tillräckligt för testmiljön
Lättanvänd	Lätt	Lätt	Lätt

Tabell 6 - Jämförelse mellan olika HMI-paneler.

Examensarbetarna valde tillsammans med handledaren från Cactus Uniview valde den billigaste modellen som var GOT Simple för att den hade alla de funktioner som behövs för testmiljön och den var lätt att använda. En fördel med denna panel är att den var prisvärd i jämförelse med andra paneler från samma tillverkare. Se [11].

Se bilaga 5 för att se bild över HMI-panelen.

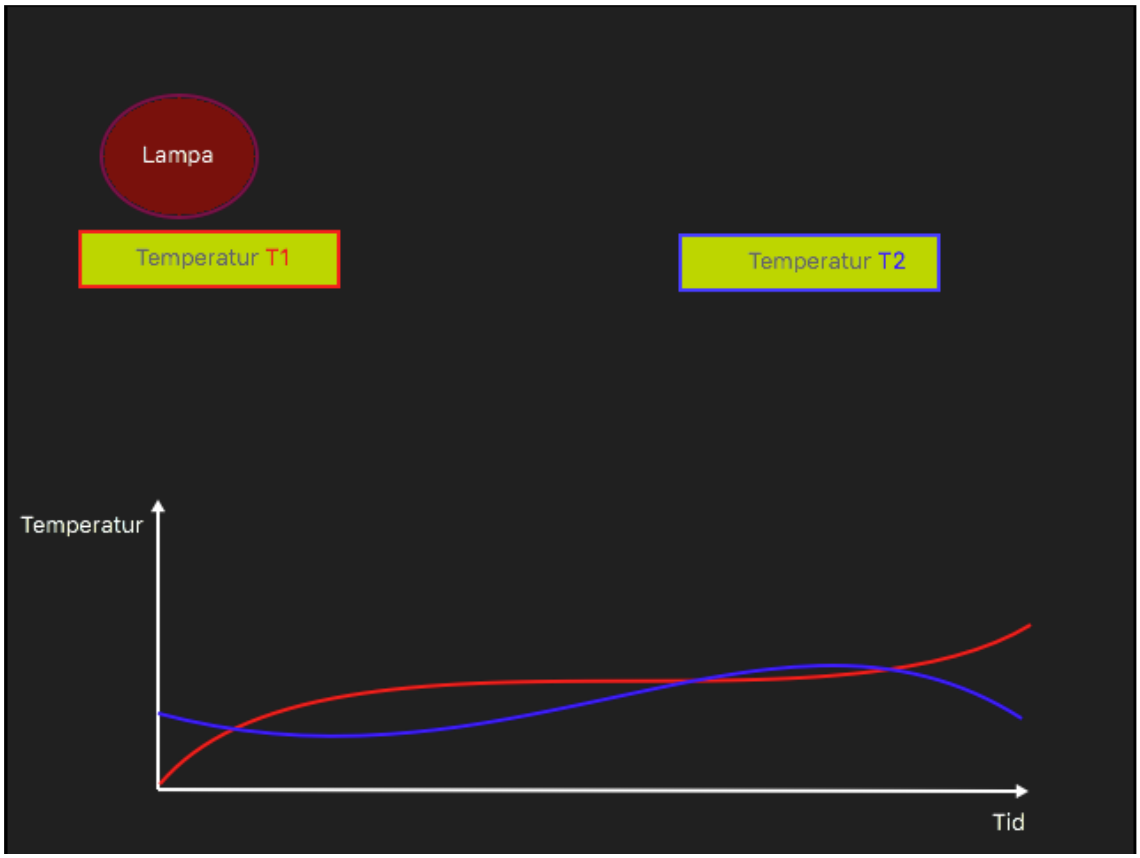
5.5.1 Konfiguration

Denna panel konfigurerades med Melsoft GT Designer 3, installationsfilerna hämtades från mitsubishielectric.se men det krävdes en licens för att använda programmet. Melsoft GT Designer 3 är ett program som används för att skapa/ rita professionella bilder som används inom processer. Se [2] och [3].

Konfigurationen sker under ”skapa ett nytt projekt” inne i programmet Melsoft GT Designer 3. Det dyker upp ett antal rutor där det behöver ställas in information och parametrar om PLCn. Hur detta går till förklaras i [2].

5.5.2 Processbild

Examensarbetarna utvecklade en processbild till testmiljön, denna bild har ett antal objekt som mätare-displayer, knappar och diagram. Två displayer användes för avläsning av temperaturen. Den ena displayen för att visa temperaturen i rummet och den andra för att visa om lampan är tänd. En knapp fungerade som switch för att slå på och av lampan. Diagrammen är till de olika temperaturerna som upplästs, de kommer att plottas med hjälp av historikdatabasen. Se figur 22.



Figur 22 - Skiss på processbild till HMI-panelen.

5.5.3 Problem

Examensarbetarna fick ett problem under denna del av arbetet. Problemet var att Cactus C10 PLCn hanterade bara paneler från tillverkaren Beijer Electronics eftersom den hade drivrutiner till denna. Mitsubishis panel krävde egna drivrutiner som behövdes skrivas till Cactus C10.

Problemet orsakades av ett missförstånd mellan handledaren och examensarbetarna. Handledaren på Cactus Uniview krävde att HMI-panelen skulle vara av tillverkaren Mitsubishi och därför ansåg examensarbetarna att denna HMI-panel kunde hanteras av Cactus C10 PLCer. Men när examensarbetarna försökte upprätta kommunikationen mellan HMI-panelen och PLCen så misslyckades detta. Cactus C10 PLCn krävde drivrutiner för HMI-panelen.

Utvecklingen av drivrutinerna ligger utanför ramen till detta examensarbete och därför lyckades examensarbetarna inte med denna del av projekt.

5.6 Testning och överlämning av den fysiska testmiljön

Examensarbetarna utförde tester på testmiljön innan överlämningen till företaget. Testerna som utfördes var följande delar i testmiljön:

- PLCn och temperaturgivarna.
- Återställning av PLCn resp. Cactus Eye.

PLCn och temperaturgivarna testades genom att registrera temperaturer i Cactus Eye under en längre period. Det ena temperaturgivaren ställdes i ett rum och den andra ställdes vid en

billampa där den skulle mäta temperaturen över hettan från billampan när den är tänd. De två temperaturerna ritades i form av kurvor. Kurvorna plottades med hjälp av de olika mätvärdena som registrerats under den långa perioden.

Återställning av styrsystemet utfördes genom att reseta konfigurationen och sedan installerades den på nytt med hjälp av systemdokumentationen för styrsystemet enligt dokumentet ”PLC konfiguration och I/O-kort mappningar”.

Cactus Eye systemet avinstallerades från servern och installerades om på nytt med hjälp av systemdokumentationen för Cactus Eye enligt dokumentet ”Kommunikation mellan styrsystemet och Cactus Eye” som examensarbetarna skrivit under delkapitel 5.4 Systemdokumentation.

Testningen av testmiljön lyckades och då överlämnades denna miljö vidare till företagets systemtestare/ utvecklare.

Se bilaga 8 och 9 för en bild över hela testmiljön.

6 Resultat

I detta kapitel kommer resultatet av examensarbetet att redovisas. Det kommer vara uppdelat i olika delar, delarna kommer att besvara frågorna i problemformuleringen för detta arbete.

6.1 Översikt av den färdiga testmiljön

Examensarbetarna lyckades med att utveckla den önskade testmiljön. Den nya testmiljön kan hantera både tester där insignalerna genereras av både fysisk hårdvara och av mjukvara med hjälp av Cactus Eye DCS-hantering. DCS-hanteringen gör det möjligt att skriva program som styr PLCn genom Cactus Eye. Cactus Eye överför programmen till PLCn efter att de har kompilerats utan felmeddelanden. Testmiljön visas i figur 23.

Figur 23 - Resultat av examensarbete.

6.2 Vilka komponenter behövs för att få en testmiljö som är så realistisk som möjligt?

I en testmiljö kan olika komponenter väljas beroende på vad projektet är tänkt att utföra. Det är viktigt att välja rätt komponenter som passar bra till testerna som är tänkta att utföras under projektets gång. Testerna som utförs av testarna på Cactus Uniview fokuserade på hantering av mätvärden och kommunikationen mellan styrsystemet och Cactus Eye. En diskussion om vilken typ av komponenter som är lämpliga att använda i testmiljön finns i avsnitt 4.6 Analys av hårdvara för servern.

PLCn som valdes till detta arbete är tillverkad av Beckhoff med PLC-modell CX8090. Efter modifiering kallas den Cactus C10. Denna typ av PLC säljs till kunder med små och medelstora anläggningar. Anledningen till att denna PLC valdes var att Cactus Uniview rekommenderade denna typ till sina kunder.

Det var tänkt att mäta temperaturer från två olika källor i den nya testmiljön och därför valdes det två temperaturgivare till detta. Därför valdes det temperaturgivare som kunde mäta temperatur från olika källor bland annat rumstemperatur och värme från billampa. Anledningen till att billampa valdes är eftersom den gav snabb mätvärdesändring på grund av värmet som strålades ut ifrån lampan.

Temperaturgivare finns i olika sorter och typer. Valet mellan dessa är beroende av behovet, det beror på vilken miljö de är tänkta att användas i. Två olika mätningar utfördes i ett rum, den ena skall mäta rummets temperatur och den andra mäter upp när en billampa sätts på intill givaren. Givarna som valdes är av typen PT-100, dessa valdes eftersom de kunde monteras i det I/O – kort till PLCn som examensarbetarna planerat köpa. En fördel med det I/O-kortet var att det inte behövdes en omvandlare för detta. Den hade inbyggda ingångar för temperaturgivarna. [10]

Lampan som valdes till testmiljön har 24 VDC och 70W. Den effekten på lampan ger högre värme till skillnad från andra lampor med mindre effekt.

Det behövs även en likriktare som ska spänningsmata PLC, lampan och temperaturgivarna. Likriktarna valdes med hänsyn till PLCns spänning dvs. de har 230VAC in och 24VDC ut, riktarna likriktade växelspänningen 230V till en likspänning med 24V och max 5A. Det behövdes två likriktare, den ena skulle mata endast PLCn och E-bussen (I/O-korten) och den andra skulle mata lampan och HMI-panelen. [12]

6.3 Vilken hårdvara är lämplig för servern för att den skall klara Cactus Eye Server - Klient mjukvara?

Cactus Eye är ett dator-klient system. Detta SCADA-system installeras på en dator, oftast servrar. Den installeras på en server eftersom det kan finnas flera operatörer som går in och använder systemet samtidigt. Därför valdes det en kraftfull server som klarade av de krav som Cactus Eye ställer på en sådan server.

Cactus Uniview hade riktlinjer för hur kraftfull server man ska välja. Trots detta fick examensarbetarna i uppgift att jämföra olika servrar som uppfyller kraven för Cactus Eye-systemet.

Examensarbetarna löste uppgiften genom att presentera tre olika alternativ för Cactus Uniview. Cactus Uniview gav inga speciella krav för hur kraftfull servern ska vara och därför var det svårt att bestämma vilka alternativ som examensarbetarna skulle välja. Alternativen jämfördes beroende på kostnad och funktion. Specifikationen för de olika alternativen sammanställdes i tre olika tabeller som sedan presenterades för Cactus Uniview. Mer om jämförelsen finns i avsnitt 4.6.

Cactus Uniview tyckte om jämförelsen men de valde att beställa en annan specifikation enligt deras riktlinjer.

Specifikationen kan man se i bilaga 1.

6.4 Vad är lämpligaste sättet att återställa systemet till känt läge efter man testat ett projekt?

Cactus Eye är ett system som kan innehålla olika projekt med olika anläggningar, PLCer, signaler och mycket annat som användare (utvecklare/ kunder) vill övervaka eller styra.

Till den nya testmiljön utvecklades ett återställningssystem som kunde återställa Cactus Eye. Med att återställa menas att all data som inte används av operatören/ testaren rensas bort från systemet.

När utvecklarna skapar ett projekt i Cactus Eye så skapas det ett antal olika databaser. Dessa databaser kommer att fyllas med information beroende på vad utvecklarna valt att utveckla. En viss information som kommunikationsdata och behörighet behöver sparas för att de är en del av konfigurationen. Konfigurationen av Cactus Eye är samma för alla projekt. Men det kan finnas annan information som kan vara onödig ifall utvecklare avslutat ett projekt. Till exempel kan det finnas kvar data (objekt-parametrar) från ett avslutat projekt (testfall) kvar i systemet.

Onödiga data kan ta mycket utrymme och därför bör de rensas bort. Detta problem löstes med hjälp av en funktion som var inbyggd i Cactus Eye-systemet. Funktionen användes till att exportera respektive importera viktigt data som var sparad i Cactus Eye. Examensarbetarna skapade en mapp och namngav ”Backup Testmiljö”. I denna mapp sparades det olika datafiler från Cactus Eye systemet. Datafilerna var viktig information som konfigurationsinställningar, behörighet och andra parametrar som tillhör testmiljön till exempel komponentparametrar.

HMI-panelens och PLCns konfigurationsfiler sparades i en separat mapp.

Det skapades backupfiler för följande i testmiljön,

- Behörighet i Cactus Eye.
- Kommunikation i Cactus Eye.
- Signaler i Cactus Eye.
- Signalgrupper i Cactus Eye.
- Styrsystem i Cactus Eye.
- Styrsystem i TwinCAT.
- HMI-panel i Melsoft GT Designer 3.
- Återställning av system.

6.5 Hur ska testerna dokumenteras för att underlätta för andra testare att förstå och följa upp?

En stor del av detta arbete handlade om att skriva systemdokumentation som kunde underlätta för testarna i Cactus Uniview att ha övergripande information om olika delar i testmiljön. Sådana dokument kan till exempel innehålla fakta, datablad, instruktioner, kopplingsscheman etc.

Det skapades ett systemdokument om hur man sätter upp en kommunikationslinje trots att det fanns en manual till detta. I manualen fanns en mycket detaljerad beskrivning som inte behövde utföras. I dokumentet valdes endast de punkter som var relevanta för den nya testmiljön. Detta medförde att filen blev betydligt mindre än manualen.

Examensarbetarna skapade även en systemdokumentation till styrsystemet, vilka parametrar som behöver fyllas i Cactus Eye server. Alla styrsystem har inte samma konfiguration, men de är ganska lika förutom att de behöver t.ex. ändra antalet I/O in och utgångar, ändra IP-adresser etc.

Det skapades fler systemdokumentationer vad som nämnts ovan, dessa skapades i samband med arbetets gång. Följande systemdokumentation skapades:

- PLC konfiguration och I/O-kort mappningar.
- Montering av PLC, lampa och givare.
- Kommunikationslinje för styrsystemet och Cactus Eye.
- Signaler, analoga och digitala.
- DCS och SPRS, programmering av PLC med Cactus Eye.
- Signaltyper, de signaler/ tabeller som stöds av Cactus Eye.
- Virtuella signaler, kurvor som utvecklats med hjälp av SPRS programmering.

6.6 Är det möjligt att integrera någon form av HMI i testmiljön?

Ett HMI kunde vara lämplig till denna testmiljö. Det valdes en operatörspanel med möjlighet för styrning och övervakning av testmiljön. Anledningen till att en panel valdes är att den var lämplig till testmiljön. Utvecklare kan t.ex. styra en lampa eller övervaka temperaturer genom denna panel. Andra typer av HMIer kan inte göra mycket i vår testmiljö, en panel däremot kunde göra en del.

Det valdes en HMI av typen Mitsubishi GOT Simple GS2107-WTBD. [11] Denna panel var ganska enkel, kraftfull och prisvärd. Den kan hantera ett antal olika tillverkare av (inte bara Mitsubishi) PLCer. Fördelar med denna panel är funktionen, priset och enkelheten att skapa

bra bilder. Med funktionen menas att den kan utföra professionella processbilder på ett enkelt sätt.

Mjukvara som använts till att konfigurera panelen och skapat bilden heter Melsoft GT Designer 3. Installationsfilerna hittades på tillverkarens hemsida, de är gratis att ladda ner men krävde en licens som tillhandahölls av Cactus Uniview.

Ett problem uppstod när examensarbetarna försökte sätta upp panelen mot PLCn. Problemet var att PLCn inte kunde hantera denna panel eftersom den inte hade drivrutinerna till denna panel. Drivrutinerna för paneler skapades av utvecklare som jobbade i Cactus Uniview Göteborg. Examensarbetarna kunde inte själva utveckla dessa eftersom det krävs en mer kvalificerad utbildning till detta. Detta problem kunde lösas genom att köpa paneler som stöds av de nuvarande drivrutinerna i PLCn. Men examensarbetarna hade inte tillräcklig tid för detta och bestämde därför att överlämna det jobbet till utvecklarna som tar emot projektet.

7 Slutsats

Tanken med detta examensarbete var att skapa en helt fysisk testmiljö till ett verktyg som används av företaget Cactus Uniview. Verktyget heter Cactus Eye, det är ett SCADA-system som kan styra och övervaka olika processer för industri företag och reningsverk. Testmiljön skulle vara uppbyggd utifrån kravspecifikationer som planerats av examensarbetarna och företaget. Kravspecifikationerna utformades med hänsyn till de bristerna som var i den gamla testmiljön.

Problemformuleringen som sattes upp på inledningen besvarades separat och har uppfyllts, vissa till en viss del p.g.a felbeställning av komponenter. Dessa problem var sorterade med hänsyn till prioritet, den första uppgiften har högst prioritet och den sista har lägst prioritering.

Ett antal olika komponenter har valts till den nya testmiljön med hänsyn till de bristerna i den gamla testmiljön. Hårdvara för servern har valts med tanke på kunder. Det har även skapats en lösning för en enkel återställning. Det skapades ett antal systemdokumentationer till testmiljön som kan användas av andra utvecklare/ testare.

Den nya testmiljön skulle vara fysisk där testerna körs med hjälp av ett antal givare som monteras i en rack tillsammans med PLC. Testerna blir mer realistiska, dvs. mätvärdena kommer från riktiga givare. Den nya testmiljön skall även vara liten från början men enkelt att utvidga.

Testerna som skulle utföras är sådana som testar olika saker som t.ex. hantering av mätvärden samt kommunikation mellan ett styrsystem och Cactus Eye. För testning av mätvärde hantering valdes temperaturgivare som kunde mäta upp värden från olika källor och som kunde registreras i Cactus Eye.

Kommunikationen testades mellan Cactus Eye och en PLC. Detta gjordes p.g.a att företaget tidigare hade brister i uppsättningen av kommunikationen, när en lösning skulle sättas upp hos företagets kunder.

Det är viktigt att man har tillräckligt med kunskap och fakta för att kunna påbörja ett jobb som aldrig gjorts innan. Sådana jobb kan vara till exempel uppsättning av en PLC. I skolan har eleverna nästan alltid fått PLCn färdig monterad där eleverna inte behöver tänka på något annat än t.ex. programmeringskoden. När vi fick PLCn så visste vi inte hur vi skulle sätta upp den, därför behövde vi lägga mycket tid på att läsa datablad och andra artiklar om styrsystemet. Men det var inte bara styrsystemet som var nytt utan även andra delar i detta projekt, allt från montering av objekt till mjukvara. Vi visste heller inte hur man använde Cactus Eye (SCADA-systemet) förre än vi läste en tjock pärm, som var en härledning till oss.

7.1 Slutsats

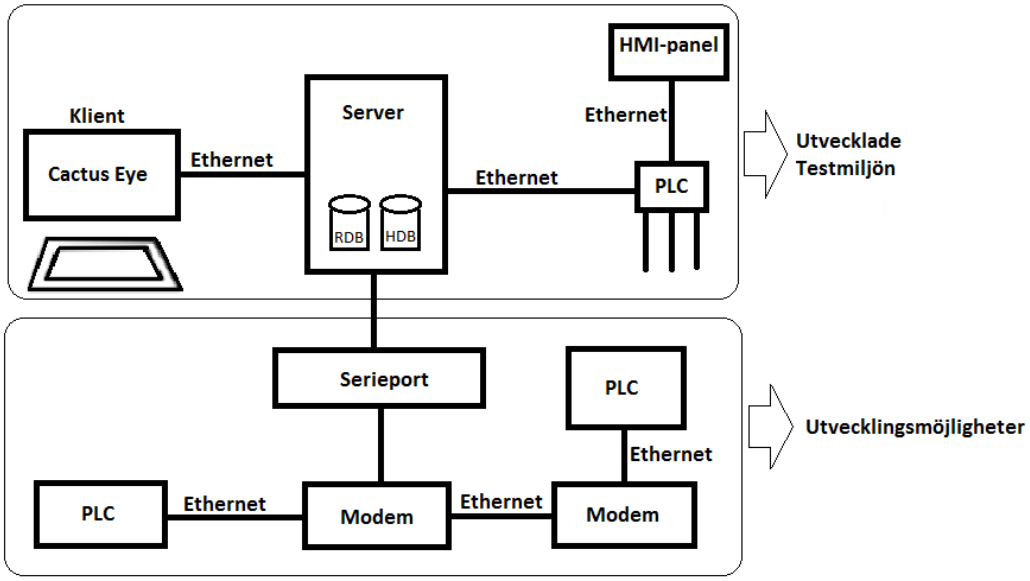
Slutsatsen av detta examensarbete är att den nya testmiljön som utvecklats har kunnat uppfylla alla kravspecifikationer och lösa bristerna i den gamla testmiljön som företaget tidigare haft.

7.2 Utvecklingsmöjligheter

En möjlig utveckling i framtiden är att kunna testa flera styrsystem samtidigt. De kan till exempel kommunicera med Cactus Eye och med varandra.

Man kan även testa olika kommunikationsprotokoll än det som använts till denna testmiljö, testarna kan testa nya protokoll som de skapat innan de lanserar ut dem till kunderna. Cactus Uniview har tidigare inte testat på fall där flera PLCer kommunicerar med varandra. Det hade varit bra om detta kan testas då detta anses vara nödvändigt. Cactus Uniview har inte kört testfall med serieport som kommunikation.

Figur 24 visar vad man ytterligare kunnat utföra för att få en mer utvecklad testmiljö.



Figur 24 - Schema över ett utvecklat testmiljö.

8 Referenser

I detta kapitel finns det olika källor som vi använt under projektet. Det finns interna respektive externa källor.

Interna källor är inte inkluderade eftersom som de endast befinner sig i företagets nät. VPN-anslutning till företagets kommer inte åt dem utanför om man inte har VPN.

Externa källor:

[1] Vad är en HMI?
http://www.cenito.se/sv/hmi/ (Maj 2016).

[2] Mitsubishi GOT2000 Quick Start.
http://dl.mitsubishielectric.com/dl/fa/document/catalog/got/l08311eng/l08311enga.pdf (Juni 2016).

[3] Melsoft GT Works 3, Designer GT3 är en del av GT Works 3.
http://sg.mitsubishielectric.com/fa/en/download_files/hmi/got/MELSOFT_GT_Works3.pdf (Juni 2016).

[4] Vad är ett SCADA system?
https://inductiveautomation.com/what-is-scada (Juni 2016).

[5] Cactus Eye – driftsystem.
http://www.cactus.se/vatten_cactus-eye-driftsystem.php (Juni 2016).

[6] Beckhoff CX8090 PLC.
http://www.beckhoff.si/CX8090/ (Juni 2016).

[7] Analog input I/O kort.
http://www.beckhoff.si/english/ethercat/el3202.htm (Juni 2016).

[8] Digital output I/O kort.
http://www.beckhoff.si/english/ethercat/el2002.htm (Juni 2016).

[9] Billampa 24V 70W.
http://www.biltema.se/sv/Bil---MC/Bilreservdelar/Belysning/Halogenlampor/H7-24V70W-STANDARD-ECE-R37-2000032128/ (Juni 2016).

[10] Temperaturgivare Pt100.
https://www.elfa.se/sv/kabeltemperaturgivare-ledaranslutning-50-180-htf50-pt100-si-10m-thermasgard-regeltechnik-gmbh-htf50-pt100-silikon-10m/p/30000924?q=temperaturgivare&filter_InStock=2&page=92&origPos=259&origPageSize=50&simi=97.03 (Februari 2016).

[11] HMI – panelen, Mitsubishi GS2107-wtbd.
https://se3a.mitsubishielectric.com/fa/sv/products/hmi/got/items/got_simple/#pageUnit01 (Februari 2016).

[12] Nätaggregat till testmiljön, 230VAC till 24VDC max 5A ut.
https://www.elfa.se/sv/switchat-naetaggregat-24-vdc-120-mean-well-dr-120-24/p/16977078?q=16977078&page=1&origPos=1&origPageSize=50&simi=98.0 (April 2016).

[13] Hur funkar en PLC?
http://www.plcdev.com/how_plcs_work (Augusti, 2016).

[14] Om Cactus Uniview.
http://cactus.se/omoss.php (Augusti, 2016).

[15] Om E-buss.
https://download.beckhoff.com/download/document/io/ethercat-development-products/ethercat_esc_datasheet_sec1_technology_2i3.pdf
(Mars, 2017)

[16] Om DCS-funktioner.
http://www.cactusrail.se/cactus-tms/integration-och-kommunikation/
(Mars, 2017).

[17] Vad LVDS är.
http://www.analog.com/media/en/technical-documentation/application-notes/AN-1177.pdf (Mars, 2017).

[18] HPE ProLiant ML350 Gen9.
https://www.dustin.se/product/5010976111/proliant-ml350-gen9
(Mars, 2017).

[19] HPE ProLiant Dl180 Gen9.
https://www.dustin.se/product/5010824306/proliant-dl180-gen9
(Mars, 2017).

[20] HPE ProLiant ML310e Gen8 v2.
https://www.dustin.se/product/5010795686/proliant-ml310e-gen8-v2
(Mars, 2017).

9 Appendix

9.1 Specifikation, Server.

TD nr	Tillv nr	Tillverkare/Beskrivning	Pris/st SEK	Miljöa..	Lager	Onlin.. Time	Antal	Totalt exkl moms	Valuta
3672087	K8P38A*K1	HEWLETT PACKARD ENTERPRISE K/HPDL380 Gen9 E5-2620V3 SP7997TV EU Svr	23 529,63	--	7	13:28	1	23 529,63	SEK
2392216	652564-B21	HPE Enterprise HPE Enterprise - Hårddisk - 300 GB - hot-swap - 2.5" SFF - SAS 6Gb/s - 10000 rpm - med HP SmartDrive-bärvåg	1 570,31	--	262	13:28	2	3 140,62	SEK
2392221	652605-B21	HPE Enterprise HPE Enterprise - Hårddisk - 146 GB - hot-swap - 2.5" SFF - SAS 6Gb/s - 15000 rpm - med HP SmartDrive-bärvåg	1 997,54	--	114	13:28	1	1 997,54	SEK
2392222	652611-B21	HPE Enterprise HPE Enterprise - Hårddisk - 300 GB - hot-swap - 2.5" SFF - SAS 6Gb/s - 15000 rpm - med HP SmartDrive-bärvåg	3 056,24	--	44	13:28	4	12 224,96	SEK

Bilaga 1 - Serverspecifikation.

9.2 Objekt som använts till testmiljön.

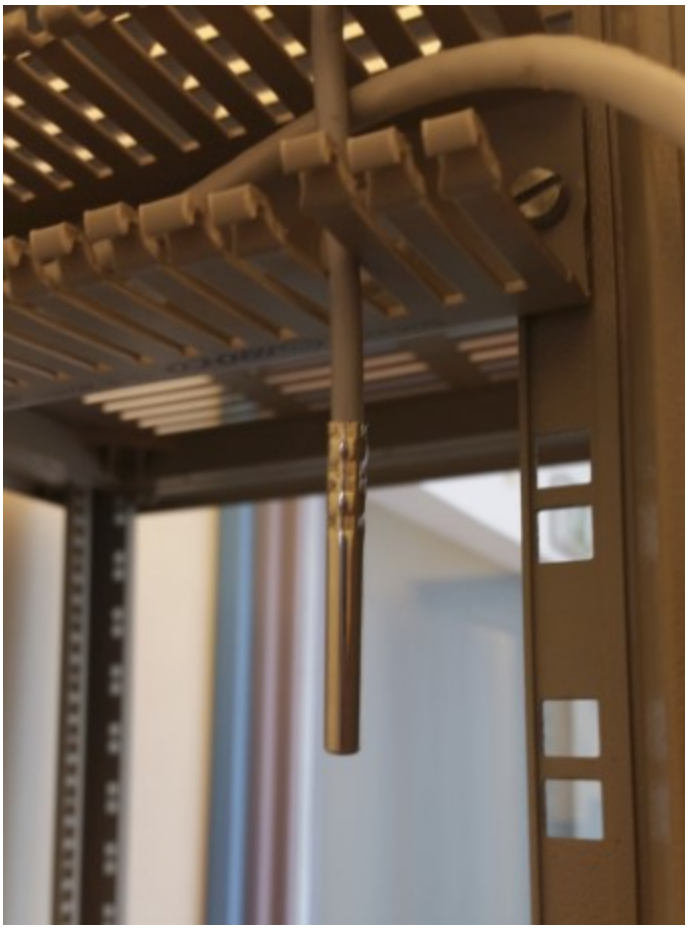
Bilaga 2 - Billampa 24V 70W. [9]

Bilaga 3 - PT-100 temperaturgivare, 2-trådad.

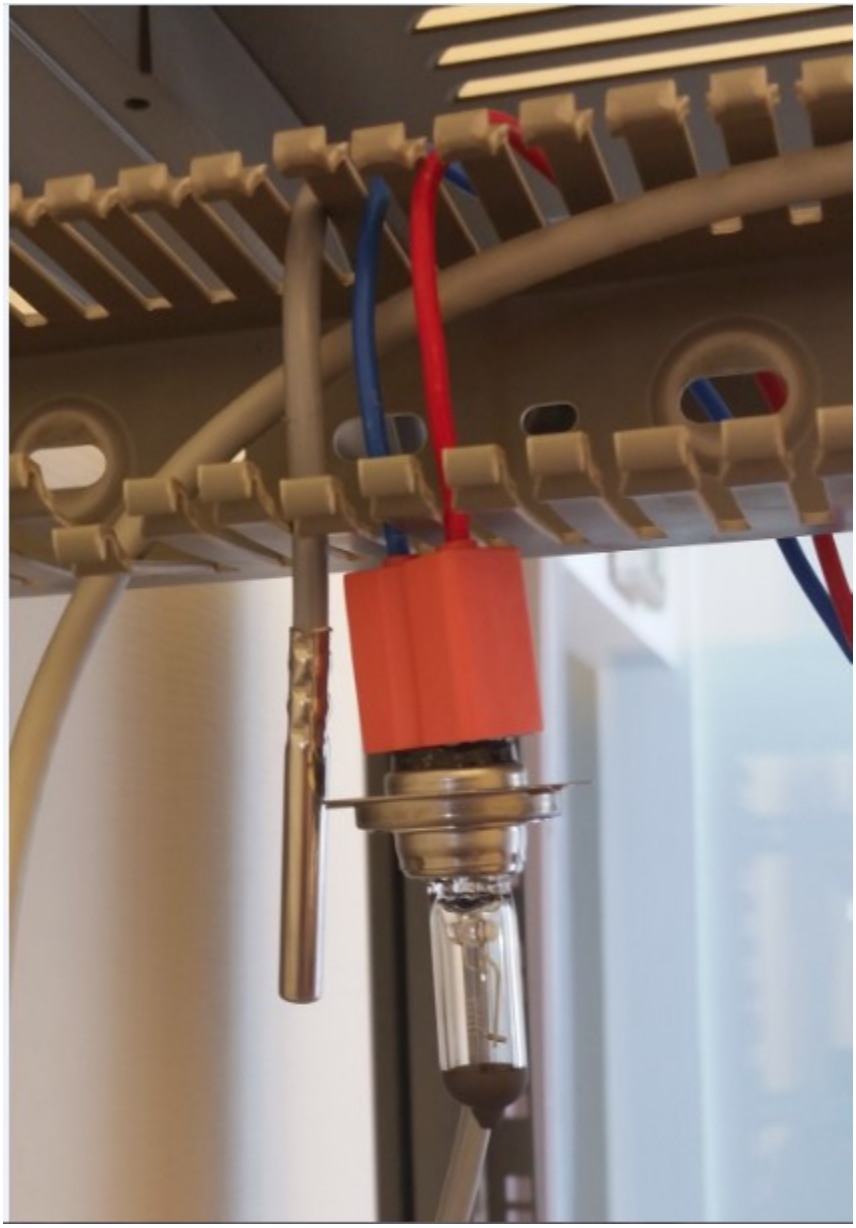
Bilaga 4 - Likriktare 24VDC max 5A ut. [12]

Bilaga 5 - HMI-panel Mitsubishi GS2107-WTBD. [11]

9.3 Resultat



Bilaga 6 - Mäta temperatur i rum.

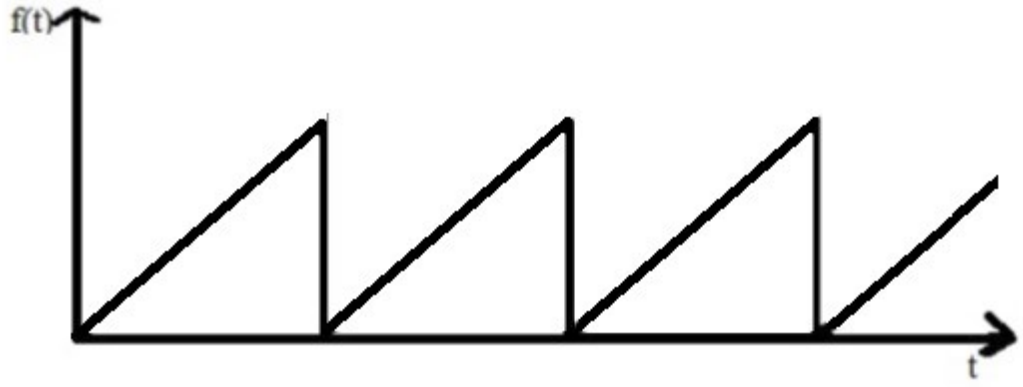


Bilaga 7 - Mäta temperatur över lampa.

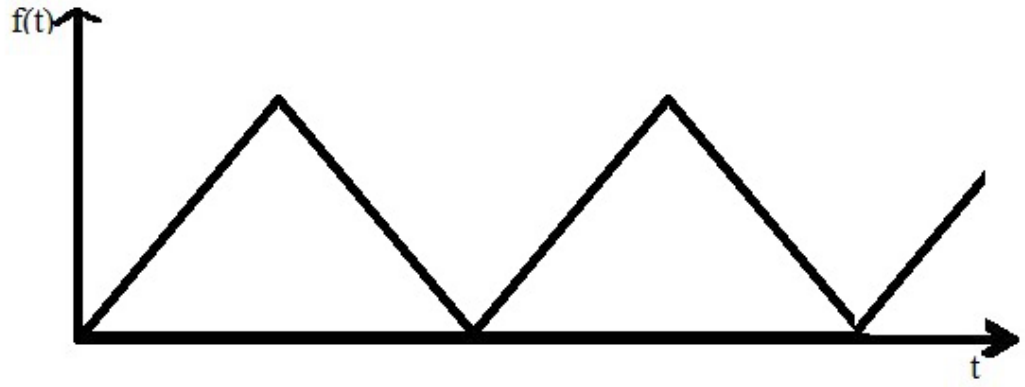
Bilaga 8 - Testmiljö med PLC, lampa, temperaturgivare, HMI-panel och likriktare.

Bilaga 9 - Testmiljö med flyttbar rack/ ställ.

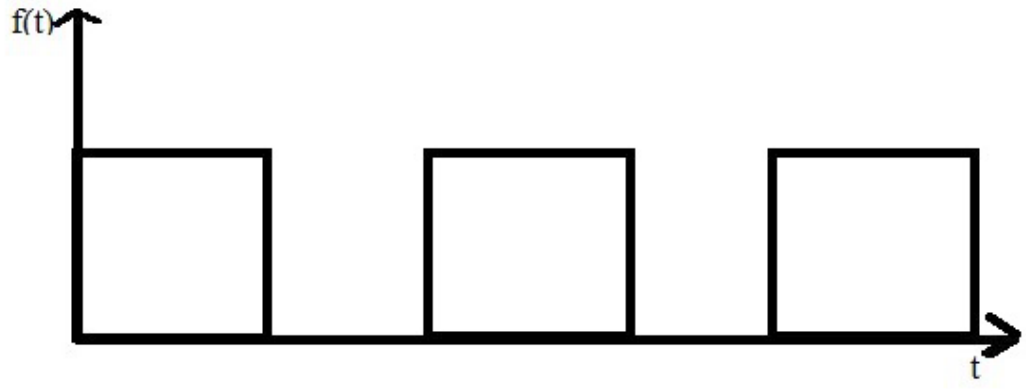
9.4 Virtuella signaler



Bilaga 10 - Sågtand signal.



Bilaga 11 - Triangelvåg signal.



Bilaga 12 - Fyrkant signal.