

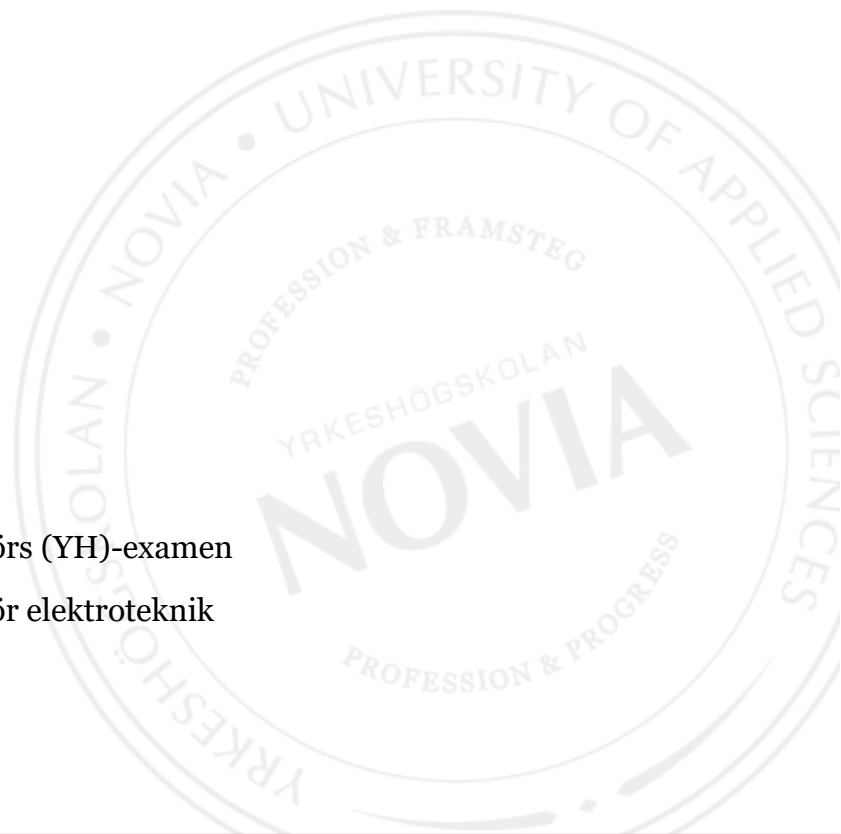
Driftövervakning vid rörlinje UNI-2

Joel Bexar

Examensarbete för ingenjör (YH)-examen

Utbildningsprogrammet för elektroteknik

Vasa 2016



EXAMENSARBETE

Författare: Joel Bexar
Utbildningsprogram och ort: Elektroteknik, Vasa
Inriktningsalternativ: Elkraftsteknik
Handledare: Erik Englund

Titel: Driftövervakning vid rörlinje UNI-2

Datum 31.5.2016 Sidantal 61 Bilagor 3 (17 sidor)

Abstrakt

Detta examensarbete omfattar förverkligandet av ett övervaknings- och styrsystem. Övervaknings- och styrsystemets främsta uppgift är att förhindra en viss typ av processfel att inträffa vid en maskinlinje med tillverkning av processrör. Examensarbetets uppdragsgivare var företaget OSTP i Jakobstad, Finland.

Syftet med examensarbetet var att planera, programmera och installera ett fungerande driftövervakningssystem till en maskinlinje UNI-2. Arbetet innebar att undersöka maskinlinjens tillverkningsprocess och hur man kan förhindra att processfelet inträffar.

Resultatet av projektet blev ett separat elskåp vid maskinlinjen innefattandes en PLC, HMI-panel och ett ljusstorn som utgör ett övervaknings- och styrsystem. Detta system förser maskinoperatören med mätdata från givare och frekvensomriktare som förverkligar önskade styråtgärder för automatiskt styrning av maskinlinjens linjestopp.

Språk: svenska
Nyckelord: övervaknings- och styrsystem, PLC, HMI, pulsgivare, IEC 61131

BACHELOR'S THESIS

Author: Joel Bexar
Degree Programme: Electrical Engineering, Vaasa
Specialization: Electrical Power Engineering
Supervisors: Erik Englund

Title: Operational Monitoring at the Machine Line UNI-2

Date 31.5.2016 Number of pages 61 Appendices 3 (17 sides)

Abstract

This thesis covers the realization of a supervisory control system. The main task of the supervisory control system is to prevent a certain type of process error to occur at a machine line with manufacturing of process pipes. The commissioner of the thesis was the company OSTP in Jakobstad, Finland.

The purpose of the thesis was to plan, program and install a functional operational monitoring system for a machine line UNI-2. This task involved examining the manufacturing process of the machine line and how to prevent the process error to occur.

The result was a separate electrical cabinet at the machine line consisting of a PLC, HMI panel and a Light tower which forms a supervisory control system. This system provides the machine operator with data from the sensors and frequency converters which realize desired control actions for automatic control of the machine line's execution of line stopping.

Language: swedish
Key words: supervisory control system, PLC, HMI, encoder, IEC 61131

Innehållsförteckning

| | | |
|-------|---|----|
| 1 | Inledning..... | 1 |
| 1.1 | Bakgrund | 1 |
| 1.2 | Mål och syfte..... | 2 |
| 1.3 | Avgränsningar | 2 |
| 1.4 | Uppdragsgivare..... | 3 |
| 2 | Teori | 4 |
| 2.1 | Övervaknings- och styrsystem..... | 4 |
| 2.2 | Programmerbara styrenheter – PLC..... | 7 |
| 2.2.1 | Uppbyggnad..... | 7 |
| 2.2.2 | Funktionsbeskrivning..... | 9 |
| 2.3 | Människa-maskin gränssnitt och operatörspaneler..... | 10 |
| 2.3.1 | Riktlinjer vid programmering av användargränssnitt..... | 12 |
| 2.4 | Pulsgivare..... | 14 |
| 2.4.1 | Inkrementella och absoluta pulsgivare..... | 14 |
| 2.4.2 | Magnetiska pulsgivare | 16 |
| 2.4.3 | Utsignal och resolution..... | 17 |
| 2.5 | Programmeringsspråk | 19 |
| 2.5.1 | IEC 61131-standardens omfattning..... | 19 |
| 2.5.2 | IL – Instruction List..... | 20 |
| 2.5.3 | ST – Structured Text..... | 21 |
| 2.5.4 | LD – Ladder Diagram | 22 |
| 2.5.5 | FBD – Function Block Diagram..... | 23 |
| 2.5.6 | SFC – Sequential Function Chart..... | 24 |
| 3 | Maskinlinje UNI-2..... | 26 |
| 3.1 | Maskinlinjens beståndsdelar | 27 |
| 3.2 | Tillverkningsprocess vid maskinlinje UNI-2 | 30 |
| 3.3 | Projektets problemformulering kring processfelet..... | 31 |
| 3.4 | Funktionsspecifikationen..... | 32 |
| 4 | Planering..... | 34 |
| 4.1 | Processöversikt och genomgång av nuvarande uppföljningssystem | 35 |
| 4.2 | Val av maskinvara och komponenter för systemet..... | 36 |
| 4.2.1 | PLC – Siemens S7-1200..... | 37 |

| | | |
|-------|--|----|
| 4.2.2 | HMI-panel – Siemens KTP900 Basic PN | 38 |
| 4.2.3 | Strömkälla och relämoduler | 39 |
| 4.2.4 | Elskåpets komponenter och konstruktion | 40 |
| 5 | Programmering med Siemens TIA-Portal V13 | 41 |
| 5.1 | Skapande av ett projekt | 42 |
| 5.1.1 | Lägga till maskinvara | 43 |
| 5.1.2 | Kommunikationsinställningar | 44 |
| 5.2 | Programmering av styrprogrammet för projektet..... | 46 |
| 5.2.1 | Definiering av PLC- och HMI-taggar | 47 |
| 5.2.2 | Val av programmeringsspråk | 48 |
| 5.2.3 | Konfigurering av HSC och funktionsblock | 49 |
| 5.2.4 | Konfigurering av funktionsblock för driftlägesindikering | 51 |
| 5.2.5 | Assignment list | 52 |
| 5.3 | Programmering och konfigurering av HMI-gränssnittet..... | 53 |
| 5.3.1 | Konfigurering av Template | 53 |
| 5.3.2 | Konfigurering av I/O-fält | 54 |
| 5.3.3 | Ihopkoppling av HMI- och PLC-taggar | 55 |
| 5.3.4 | Konfigurering av dynamiska objekt | 56 |
| 5.3.5 | Loggning av data med HMI-panelen..... | 57 |
| 6 | Resultat och utvärdering..... | 58 |
| 7 | Diskussion | 60 |
| | Källförteckning | 61 |

Bilageförteckning

| | |
|----------|---------------------|
| Bilaga A | Ritningar |
| Bilaga B | Datablad |
| Bilaga C | Installationsbilder |

Ordlista

Förklaringar på ord som förekommer i texten:

| | |
|----------|--|
| Bockar | Är roterande valsar för tillformning eller riktning vid rörtillverkning. |
| FBD | Förkortning av Function Block Diagram, som är ett programmeringsspråk i IEC 61131-3-standarderna. |
| IL | Förkortning av Instruction List, som är ett programmeringsspråk i IEC 61131-3-standarderna. |
| LD | Förkortning av Ladder Diagram, som är ett programmeringsspråk i IEC 61131-3-standarderna. |
| SFC | Förkortning av Sequential Function Chart, som är ett programmeringsspråk i IEC 61131-3-standarderna. |
| ST | Förkortning av Structured Text, som är ett programmeringsspråk i IEC 61131-3-standarderna. |
| UNI-2 | Namnet på den maskinlinje/rörinje som övervaknings- och styrsystemet i examensarbetet utvecklas för. |
| Variabel | Är ett namngivet objekt som används för att representera ett okänt värde. |

1 Inledning

I detta examensarbete handlar det om att planera, programmera och installera ett övervaknings- och styrsystem. Projektet är ett arbete som innefattar många olika moment för att förverkliga systemet. I ingenjörsarbetet framkommer också de kunskaper som jag har fått och använt mig av i detta arbete.

Under sommarjobbet 2015 frågade jag vid företaget OSTP Oy om ett lämpligt examensarbete skulle vara möjligt. Jag erbjöds att utföra och leda ett pilotprojekt som omfattar realisering av ett driftövervakningssystem för rörlinje UNI-2 vid produktionsanläggningen i Jakobstad som ett examensarbete.

Jag visste att det skulle innebära en omfattande undersökning och analysering av rörlinjens tillverkningsprocess för att få en bättre förståelse för hur ett övervaknings- och styrsystem kan anpassas till den.

Behovet av ett övervaknings- och styrsystem för maskinlinjen blev aktuellt av den anledningen att man ville utföra ett pilotprojekt. Projektet handlar om att ta reda på om man har möjligheten att förhindra ett tidigare uppdagat förekommande processfel vid maskinlinjen med hjälp av ett driftövervakningssystem.

1.1 Bakgrund

OSTP har haft maskinlinjen som konstruerades på 1970-talet i produktionen i många år. Maskinlinjen har moderniserats med jämna mellanrum fram till dagsläget, men man har ändå inte kunnat undvika att vissa typer av processfel tidvis inträffar.

Processfelet som examensarbetet kretsar kring är också av sådan art att det finns pengar att spara om man kan förhindra dess uppkomst. Det är en orsak till att OSTP gett mig detta intressanta projekt att förverkliga ett driftövervakningssystem till maskinlinjen.

Skribenten har tidigare varit anställd hos företaget och arbetat på underhållsavdelningen som elautomationsmontör. Arbetsuppgifterna har omfattat bl.a. elunderhåll på företagets maskinlinjer i produktionen samt även med nyinstallationer och modernisering av maskinlinjer. Vilket har medfört att projektet i fråga lämpade sig mycket väl som ett examensarbete.

1.2 Mål och syfte

Huvudsyftet med examensarbetet är att planera, programmera och installera ett övervaknings- och styrsystem för en maskinlinje UNI-2. Första delmålet är att analysera maskinlinjen för att samla processkunskap som underlag för val av angreppsmetod för projektet. Andra delmålet innefattar utvecklandet av ett fungerande övervaknings- och styrsystem.

1.3 Avgränsningar

Projektet som jag tilldelades av OSTP fick jag i början på hösten 2015. Vid den tidpunkten kunde jag börja arbeta och undersöka hur tillverkningsprocessen fungerar vid maskinlinjen 1–2 dagar i veckan vid sidan om skolan. Arbetet med att förverkliga ett övervaknings- och styrsystem för maskinlinjen är gjord från OSTP:s behov och kan möjligtvis inte direkt användas för något annat ändamål.

Den första avgränsningen var det analytiska arbetet med att utreda vilken angreppsmetod som skulle väljas för att kunna förebygga processfelet hos maskinlinjen. Det tog mycket tid, men det gav också en viss processkunskap som underlag för hur jag skulle gå tillväga.

Andra avgränsningen var att jag skulle utföra planering, programmering, installation och ibruktagning av det förverkligade övervaknings- och styrsystemet. Det innebar att övervaknings- och styrsystemet skulle utgöras av ett separat elskåp vid maskinlinjen som innefattar en PLC, HMI-panel och ett ljusstorn. När de fysiska arrangemangen hade utförts skulle maskinvaran för systemet programmeras i programmeringsverktyget Siemens TIA-Portal V13 SP1.

Det som inte inkluderas i examensarbetet var utredning av maskinlinjens befintliga styrsystem för rörtillverkningsprocessen samt skolning åt maskinoperatörerna vid maskinlinje UNI-2 för hur övervaknings- och styrsystemet används.

1.4 Uppdragsgivare

Uppdragsgivaren till projektet är metallindustriföretaget OSTP Oy. OSTP är en förkortning som står för *Outokumpu Stainless Tubular Products*. Företaget är en global koncern som har produktionsanläggningar i Jakobstad – Finland, Örnsköldsvik – Sverige och Riyadh – Saudi Arabien.

Vid produktionsanläggningen i Jakobstad tillverkas rostfria och syrafasta produkter för processindustrin. Råmaterialet för tillverkningsprodukterna dvs. plåtrullarna köps i huvudsak från företaget Outokumpu i Torneå, Finland. I produktsortimentet ingår b.l.a. Processrör, Rörkrökar, T-stycken, Flänsar och Kransar som kan ses i figur 1 nedan.

Företaget har i dagsläget ca 230 anställda vid produktionsanläggningen i Jakobstad och koncernens omsättning var för verksamhetsåret 2014 ca 122 miljoner Euro.



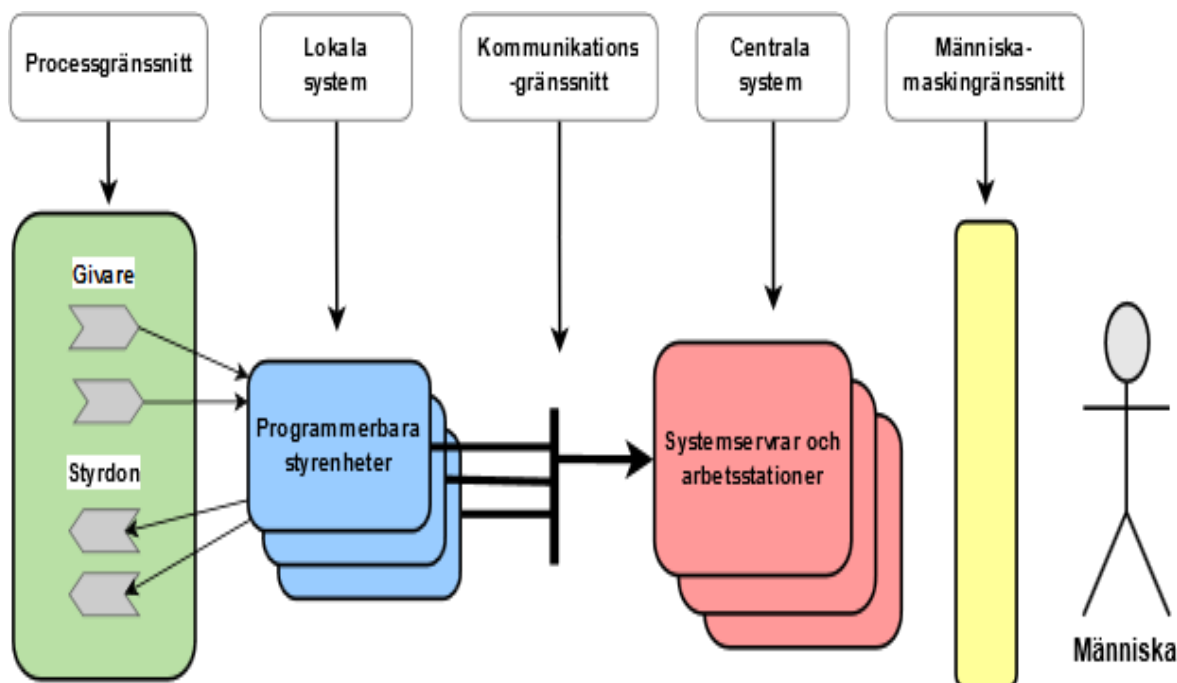
Figur 1. Produkter som tillverkas vid OSTP:s produktionsanläggning i Jakobstad.

2 Teori

I detta kapitel behandlas teknisk och teoretisk referenskunskap som ligger till grund för hur examensarbetet utförts. Teoridelen omfattar bl.a. strukturen och innebörden av övervaknings- och styrsystem, funktion och uppbyggnad hos programmerbara styrenheter samt riktlinjer och programmering av människa-maskingränssnitt. Även teori kring mätmetoder samt funktion för olika typer av pulsgivare tas upp och avslutningsvis korta beskrivningar kring de fem enskilda programmeringsspråken i IEC 61131-3-standarden.

2.1 Övervaknings- och styrsystem

Övervaknings- och styrsystem definieras av fem olika byggblock där alla eller flera byggblock kan finnas inuti en och samma fysiska komponent. Exempelvis styrsystem som är utvecklade i en industriell PC. Dock kan inte själva processgränssnittet finnas med i en PC i och med att det byggblocket består av givare, styrdon och fysiska signaler från själva processen som kräver ett I/O-gränssnitt. I figur 2 nedan beskrivs strukturen av de fem olika byggblocken och därefter följer en närmare beskrivning på vad som menas med de enskilda blocken.



Figur 2. Styrsystemets olika byggblock. (Cegrell, 1994, s.144)

Processgränssnitt

Information från processen samlas in med hjälp av sensorer som ingår i olika typer av givare. Givarnas uppgift är omvandla de fysikaliska storheterna i processen till signaler som kan förmedlas till styrsystemet för att behandlas. Styrsystemet skickar i sin tur styrsignaler till processen som via olika hjälpkretsar skickar styrsignalerna till verkställande manöverdon.

Manöverdonen styr sedan exempelvis motorer, pumpar och ventiler. Kommunikations- eller signalgränssnitt används mer och mer hos moderna processobjekt för att styrsystem av olika slag direkt kan hämta information och styra objektet utan mellansteg. (Cegrell, 1994, s. 144–145).

Lokala system

Är den del av systemet som befinner sig närmast processen och som ska samla in respektive styra alla in- och utsignaler till givare och den verkställande styrutrustningen. Av prestandaskäl kan vanligtvis inte sådana system förläggas långt ifrån själva processen.

En lokalenhet eller ett lokalsystem är på så vis allmänt förbundet med en eller flera överordnade centrala enheter. Styrsystemets uppgifter och funktionella innehåll bestämmer hur samspelet mellan centrala enheter och lokala system anpassas. Sådana lokala enheter som endast samlar in information från processen med s.k. distribuerade I/O-enheter vilka fungerar som slavar *slave* till den centrala enheten *Master*.

En lokalenhet kan även utformas så att den klarar av att fortsätta att samla in information från processen, reglera processobjekt eller styra processobjekt med sekvensstyrprogram vid ett avbrott i kommunikationen med centralsystemet. Och när kommunikationen återupptas överförs den insamlade informationen från den lokala enheten så att den centrala enheten i sin tur uppdateras. (Cegrell, 1994, s. 145).

Kommunikationsgränssnitt

Information som innehåller data, händelser, styr- och regler kommandon o.s.v. överförs med hjälp av datakommunikation mellan lokala- och centrala enheter. Datakommunikation baseras på att två enheter ska överföra data till- och från varandra på ett sådant sätt att båda enheterna förstår varandra. (Cegrell, 1994, s. 145–146).

Det krävs gemensamma regler för hur kommunikationen ska gå till, vilket utgörs av olika kommunikationsprotokoll. Exempel på två allmänt förekommande moderna kommunikationsprotokoll är ethernetbaserade *PROFINET* och fältbussbaserade *PROFIBUS* automationsstandarderna. (PROFIBUS, 2008) (PROFINET, 2014)

Centrala system

För många olika systemlösningar finns det alltid ett behov av ett eller flera system där information från processen kan lagras och funktioner realiseras. Centrala enheter är exempel på sådana enheter som kan ta hand om insamling av processinformation från flera olika delar av processen för att sedan användas av någon funktion som kräver stor beräkningskapacitet.

Av säkerhetskäl är det naturligt att en central enhet lagrar all processinformation eller program som används i styrsystemet. I mindre styrsystem förläggs vanligtvis alla funktioner till en enda central enhet.

Det centrala systemets uppgifter kan variera i stor grad, från att vara en resurs för ett styrsystems alla funktioner till att tillhandahålla lagring av data eller att vara länken mellan processen och människa-maskingränssnittet. (Cegrell, 1994, s. 146–147).

Människa-maskingränssnitt

Människa-maskingränssnittet omfattar utrustning för människans interaktion med processen. Som exempelvis kan vara utmatning av processinformation till människan och utrustning som människan kan använda för inmatning av data till styrsystemet för att styra processen.

För att styrsystemet ska kunna presentera processinformationen åt människan används bildskärmar, väggtavlor och manöverpulpeter med eventuellt kompletterande utrustning såsom skrivare och ljud- eller ljusindikeringsutrustning. (Cegrell, 1994, s. 147).

2.2 Programmerbara styrenheter – PLC

PLC (Programmable Logic Controller) är en unik form av programmerbar styrenhet som baserar sig på en mikroprocessor med ett programmerbart minne. I mikroprocessorns programmerbara minne sparas instruktioner för att genomföra olika funktioner som bl.a. sekvensstyrning, tidsstyrning, räknefunktioner, aritmetiska funktioner och logiska algoritmer för att styra olika processer och maskiner.

En PLC är utvecklad för att kunna programmeras med relativt okomplicerade programmeringsspråk så att de kan användas och programmeras av ingenjörer med kunskaper om tekniken kring datorer och olika programmeringsspråk. (Bolton, 2006, s. 3).

2.2.1 Uppbyggnad

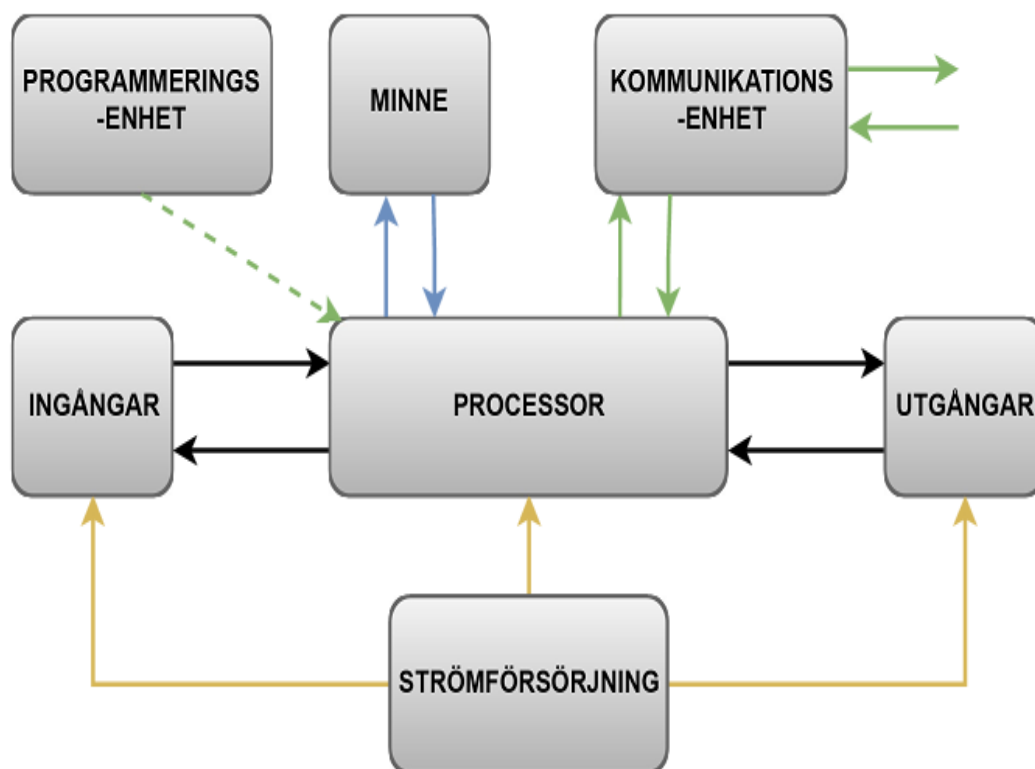
Processorn, CPU (Central Processing Unit), är den enhet som består av en mikroprocessor som läser av ingångssignalerna och utför styråtgärderna för utsignalerna enligt det programmerade styrprogrammet i dess minne.

Strömkällan är den enhet som sköter strömförsörjningen till processorn och I/O-enheterna som vanligtvis är en likströmsmatning på 24 V DC.

Med programmeringsenheten programmeras styrprogrammet till PLC:ns minne med hjälp av ett kompatibelt programmeringsverktyg installerat för den PLC-typ som styrprogrammet görs för. Minnet kan antingen vara internt eller externt eller en kombination av båda för en PLC. I minnet sparas främst styrprogrammet samt indata från in- och utsignalernas värden för att sedan bearbetas av processorn.

Från ingångarna och utgångarna eller de s.k. I/O-enheterna läser och skriver processorn värden från givare och till styrdon hos den verkliga processen.

Med kommunikationsenheten kommunicerar PLC:n med externa enheter som kan vara exempelvis andra PLC:n, användargränssnitt eller databaser m.m. Se beskrivning av en PLC:s uppbyggnad i figur 3 på följande sida. (Bolton, 2006, s. 4).

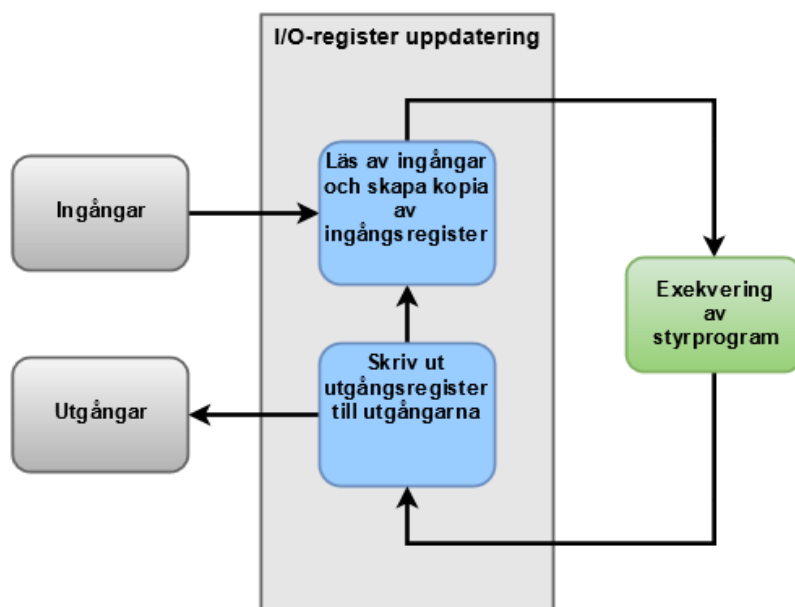


Figur 3. En PLC:s uppbyggnad. (Bolton, 2006, s. 4).

2.2.2 Funktionsbeskrivning

Man kan beskriva hur en PLC arbetar genom att dela upp tillvägagångssätten i två olika steg. Se beskrivning av cyklisk programexekvering för en PLC i figur 4 nedan.

1. Först uppdateras en kopia av ingångsregistret så att det överensstämmer med nuvarande värden för insignalerna. För utsignalerna är processen däremot den motsatta och utsignalerna ställs då enligt kopian för utsignalsregistren.
2. Sedan exekveras programmet så att processorn endast arbetar med kopiorna av I/O-registren. Kopian av insignalsregistret förblir därmed oförändrat medan kopian för utsignalsregistret kompletteras med nya värden vartefter programkoden körs.



Figur 4. En PLC:s programcykel. (Hackworth, 2004, kap. 2, s. 15).

När följande programcykel startas så upprepas steg ett och först i detta skede får utsignalerna uppdaterade värden. Tidsåtgången mellan två cykler kan vara endera förbestämd till specifika värden eller så fortsätter helt enkelt nästa cykel efter att föregående är klar. I och med att uppdaterade värden på ingångarna inte kommer fram till processorn före steg ett har upprepats, så innebär det att responstiden kraftigt begränsas.

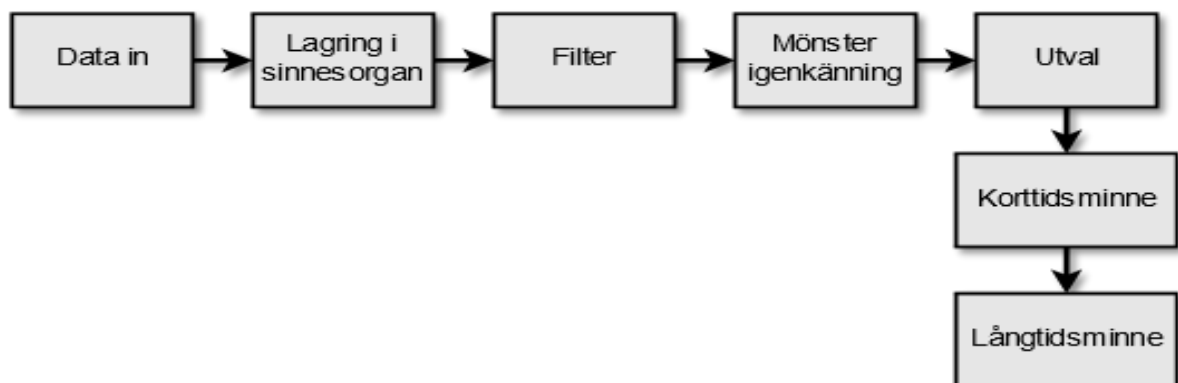
I vissa system är det inte alltid acceptabelt med en långsam responstid och då kan avbrottsrutiner användas i programmet för att minska på responstiden. När en konfigurerad avbrottsrutin inträffar under en pågående cykel så avbryter processorn denna cykel för att istället direkt hantera avbrottsrutinen. (Bolton, 2006, s. 76).

2.3 Människa-maskningränssnitt och operatörspaneler

Människa-maskningränssnittet utformas enligt människans sätt att behandla och tolka information. Det primära intresset är själva människans arbetsmoment i processen. Det vill säga om ett system är undermåligt anpassat till människan skapar det både en dålig arbetsmiljö och ett ogynnsamt resultat samt att det även kan vara direkt farligt i vissa processer. Uppmärksamheten koncentreras på systemet istället för arbetsuppgiften i och med att ett bristfälligt utformat system bidrar till att hindra användaren att utföra sitt arbete.

Människans förhållande och samverkan med omgivningen kan beskrivas med tre stycken nyckelord: perception, kognition och motorik. Med perception menas hur människans mottagande och tolkning av yttre stimulans bearbetas och tolkas i hjärnan med hjälp av de fem sinnen. Kognition är människans förmåga att samla in, lagra och hantera information. Med den information som insamlats tolkar och bearbetar människan för att sedan omvandla till användbar kunskap. Denna kunskap fungerar sedan som underlag för beslutsfattande och handling. Slutligen ska människan reagera, vidta någon åtgärd och utföra praktiska uppgifter med händerna genom att utnyttja kroppens motorik. (Cegrell, 1994, s. 124).

Av människans fem sinnen så är det egentligen bara synen och hörseln som används i samverkan med människa-maskningränssnittet där synen är den absolut viktigaste. I figur 5 nedan presenteras en modell på hur människan behandlar information. (Cegrell, 1994, s. 125).



Figur 5. En förenklad modell av mänsklig informationsbehandling. (Cegrell, 1994, s. 129).

En operatörspanel eller s.k. HMI-panel används allmänt när ett industriellt styrsystem behöver ett användargränssnitt för att övervaka, styra- och eller reglera en process. Moderna HMI-paneler har många olika egenskaper som liknar de hos en PC i och med att de har egna inbyggda mikroprocessorer som exempelvis kan samla in processdatan som styrsystemet tillhandahåller. Det som gjort detta möjligt är att minneskapaciteten hos HMI-paneler har signifikant blivit större under årens lopp och fortsätter att växa i takt med utveckling av tekniken.

HMI-paneler har också i dagsläget blivit mer regel än undantag för olika processer i och med att nutidens maskinoperatörer och användare mer eller mindre kräver att en översikt av processens tillstånd ska kunna visualiseras. Detta har även lett till att människans arbetssätt har förändrats i takt med att tekniken utvecklats. Det ger också en säkrare och bättre arbetsmiljö när processen kan regleras och visualiseras på ett tillförlitligt sätt. I figur 6 nedan kan man se olika varianter av stationära och portabla operatörspaneler samt även s.k. *Programmable Keypads* från tillverkaren Siemens. (Control Engineering, 2012).



Figur 6. Några användargränssnitt ur tillverkaren Siemens produktsortiment.

2.3.1 Riktlinjer vid programmering av användargränssnitt

I fråga om programmering och design av användargränssnitt så finns det ett antal olika faktorer som man behöver beakta för att skapa ett så effektivt och användarvänligt gränssnitt som möjligt.

I och med att alla människor är olika så varierar förmågan att lära, tänka och lösa problem. Av den anledningen så behöver man vid skapandet av ett användargränssnitt beakta vilken målgrupp som är den avsedda användaren av systemet och utveckla därefter.

Processen att lära känna användarna av gränssnittet är en oändlig process på grund av att det finns så mycket information att ta reda på samtidigt som målgruppen vanligen byts ut eller förändras. Med mera information om vilken målgrupp man utvecklar gränssnittet för kommer man hela tiden ett steg närmare mot ett framgångsrikt och funktionsdugligt användargränssnitt.

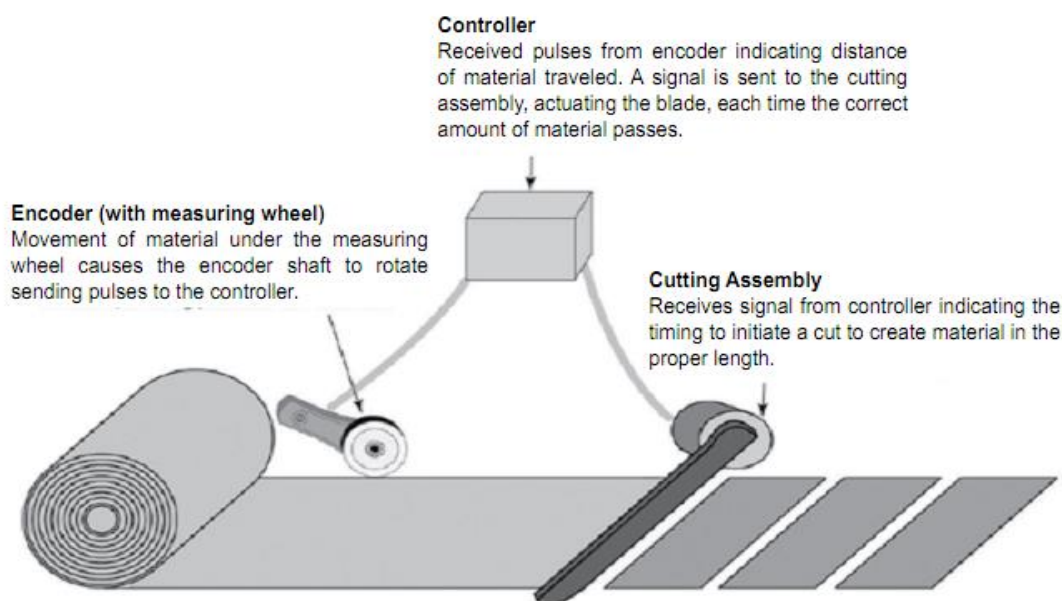
Man brukar nämna de åtta gyllene reglerna som kan tillämpas vid skapandet av ett användargränssnitt. Just denna lista med regler kan ju inte på något sätt vara fullständig och beakta alla aspekter men listan fungerar bra att använda som riktlinjer vid skapandet. Principen hos de här riktlinjerna måste ändå beaktas beroende på till vilken typ av system som användargränssnittet skapas för. Reglerna är:

1. **Sträva efter konsistens** – Sträva efter att använda liknande terminologi och utformning genom hela gränssnittet, t.ex. tillämpning av samma layout, färgsättning och textstorlek m.m.
2. **Tillgodose mångsidig användbarhet** – Ta i beaktande att det finns olika användare med varierande förmåga att använda gränssnittet. Av den orsaken kan man överväga att införa olika användarnivåer, t.ex. där nybörjare kan få ytterligare förklaringar på olika saker medan erfarna användare kan få genvägar och rättigheter till att utföra ändringar.
3. **Erbjuda informativ respons** – För varje åtgärd som en användare utför ska respons ges efter varje utförd åtgärd så att användaren får klart för sig vilken inverkan det har på systemet. Exempelvis kan man vid knapptryckning för start och stopp av en motor använda objekt eller färgändring som respons efter utförd åtgärd medan vid enklare och mera självklara åtgärder kan man vara aningen diskretare.

4. **Utforma dialoger med tydliga avslut** – Om användaren ges tydlig och informativ respons vid avslutandet av en sekvens eller åtgärd så ger det en känsla av lättnad och förnöjelse efter en utförd prestation. Till exempel att en dialogruta dyker upp som visar att, ”pumpning av vatten till bassängen är färdig”, vilken sedan avslutas manuellt eller med ett visst tidsintervall.
5. **Förhindra fel** – Man ska utforma användargränssnittet så att sannolikheten för att göra något allvarligt misstag är så liten som möjligt. Man kan exempelvis förhindra att inmatning av bokstäver i ett sifferinmatningsfält inte kan utföras eller måla knappar som inte ska tryckas på i grå färg. Ifall felaktig data matas in ska systemet kunna självständigt upptäcka problemet och lämna värdet oförändrat. Samtidigt ska systemet erbjuda användaren att korrigera inmatningen istället för att förkasta hela den felaktiga inmatningen.
6. **Tillåta enkel återkallning av åtgärder** – Man ska så långt som möjligt utforma systemet så att man kan ångra utförda åtgärder. När en användare känner till att en åtgärd enkelt kan återkallas ger det en känsla av lugn istället för oro och uppmuntrar denne istället till att utforska och undersöka andra nya intressanta egenskaper hos systemet.
7. **Främja en upplevelse av kontroll** – Erfarna användare vill känna att de har full kontroll över gränssnittet och att det utför de åtgärder som deras rutinerade handlingar medför. De blir missnöjda med plötsliga förändringar i bekanta sekvenser i systemet, svårigheter med att skaffa nödvändig information och att önskvärda åtgärder inte går att utföra för att kunna få de resultat som eftersträvas.
8. **Minska användarens minnesbelastning** – Människans begränsade förmåga att hantera information i korttidsminnet kräver att skärmvisningar hålls enkla, visning på flera bildskärmar sammanslås, sidbytesfrekvensen reduceras och att tillräckligt med tid erbjuds för inmatningar och sekvenser för olika åtgärder. (Schneidermann, 2009, s. 88–89).

2.4 Pulsgivare

Pulsgivare är sensorer som genererar digitala signaler som återkoppling från någon mekanisk rörelse. Det finns roterande pulsgivare och linjära pulsgivare som ger återkoppling för roterande rörelser respektive linjära rörelser. Roterande pulsgivare används vanligen också tillsammans med mekaniska delar som t.ex. drev, axlar och mätjul för att omvandla linjära-, hastighets- eller positionsrörelser till mätbara storheter, se figur 7 nedan med exempel på en applikation där en pulsgivare används.



Figur 7. Återkoppling med pulsgivare för styrning av en sax-kap.
(Encoder Products Company, 2012)

2.4.1 Inkrementella och absoluta pulsgivare

Det finns två huvudtyper för hur en pulsgivare genererar en utsignal. Nämligen inkrementella pulsgivare och absoluta pulsgivare. Inkrementella pulsgivare genererar pulser vartefter de rör sig och dessa pulser kan exempelvis användas för att mäta hastighet eller utgående från någon referenspunkt räkna antalet pulser för att bestämma positionen.

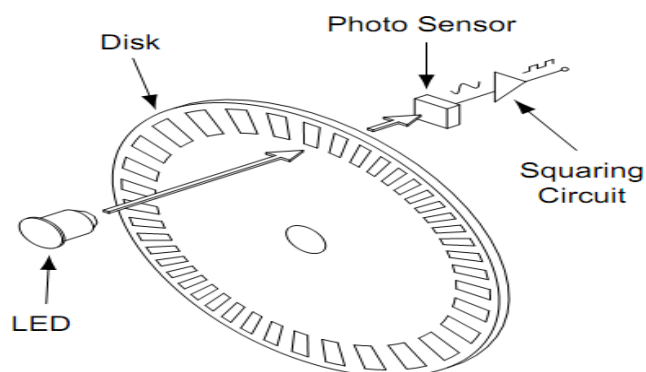
Absoluta pulsgivare genererar däremot en unik binär kod för varje rörelse vilket specifikt indikerar positionen som blir noggrannare gentemot inkrementella pulsgivare. Exempel på användningsområden för pulsgivare kan vara följande: för att styra hastigheten hos motorer, robotstyrningar, mäta olika avstånd och positionering av X/Y-axel register. (Danaher, 2003, s.2)

De allmänt förekommande sensorteknikerna som påträffas i industrin för pulsgivare är optiska- eller magnetiska pulsgivare. Optiska pulsgivare ger allmänt den noggrannaste återkopplingen från mekaniska rörelser. Men de måste också vara bättre skyddade mot smuts, vibrationer och andra faktorer som varierar för olika tillämpningar i industriella miljöer. I sådana miljöer där smuts, vibrationer, ånga och andra faktorer är vanligt förekommande lämpar sig användning av magnetiska pulsgivare bättre. Dock kan man inte med magnetiska pulsgivare uppnå samma noggrannhet och upplösning som hos optiska pulsgivare. Optiska pulsgivare använder allmänt en glaskiva med olika linjemönster eller spår på sig. Hos roterande pulsgivare förekommer även plast- eller metallskivor med spår på men däremot har linjära pulsgivare i allmänhet ett glas- eller metallband med spår på, se figur 8.



Figur 8. Optiska pulsgivares olika tekniker för indikering av rörelse. (Danaher).

Tekniken fungerar så att en LED-ljuskälla belyser skivan eller bandet på ena sidan medan en eller flera fotodetektorer registrerar det ljus som kommer genom och genererar således pulsgivarens utsignal, se figur 9. En inkrementell pulsgivare har ett visst antal spår medan en absolut pulsgivare har lika många spår som den har kodbitar. (Danaher, 2003, s. 2).



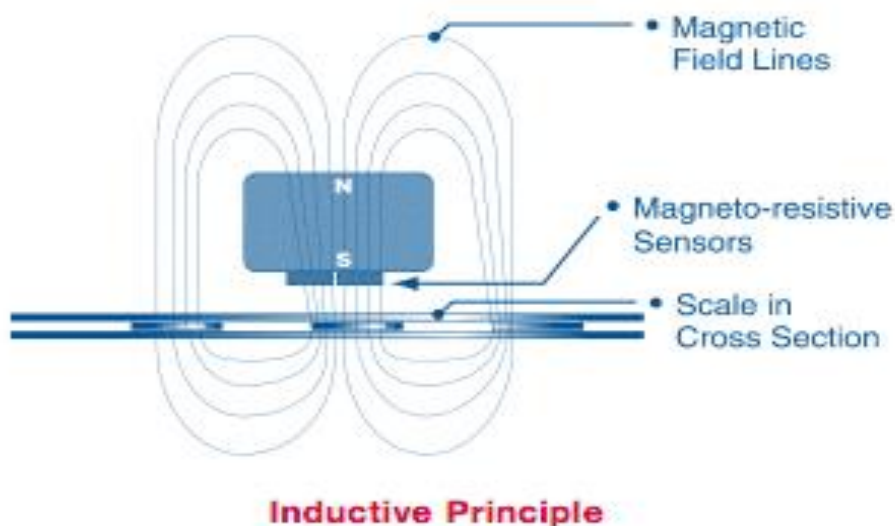
Figur 9. Optiska pulsgivares allmänna uppbyggnad. (Encoder Products Company, 2012).

2.4.2 Magnetiska pulsgivare

Magnetiska pulsgivare finns i flertalet olika utföranden gällande givartekniken. Den ena kategorin är typen *Variable Reluctance sensors* som registrerar det magnetiska fältets förändringar som uppstår när ferromagnetiska föremål kommer i dess närhet. Den enklaste konstruktionen av en sådan kallas allmänt för *Magnetic pick-up sensor*, se figur 10 nedan.

En sådan består av en spole som är lindad runt en permanentmagnet. Det är en robust, förmånlig och driftsäker konstruktion som lämpar sig bra vid hastighetsmätning. Exempelvis när ett roterande drev med kuggar kommer i dess närhet genereras en spänningspuls för varje kugg som registreras. Hastigheten på det förbipasserande föremålet som registreras måste vara högre än 457 cm per sekund.

En annan typ av magnetiska pulsgivare är sådana som använder sig av en permanentmagnet och en magnetresistiv apparat för att generera en förändring i endera spänning eller resistans när ett ferromagnetsikt föremål kommer i dess närhet. Denna metod kallas även för ”Hall effekt” och kan användas för både roterande- och linjära pulsgivare. Den här typen av magnetiska givare kan även fungera vid så låga hastigheter att det registrerade föremålet nästan är stillastående. (Danaher, 2003, s. 3).



Figur 10. Magnetiska pulsgivares allmänna uppbyggnad. (Danaher, 2003, s. 3)

2.4.3 Utsignal och resolution

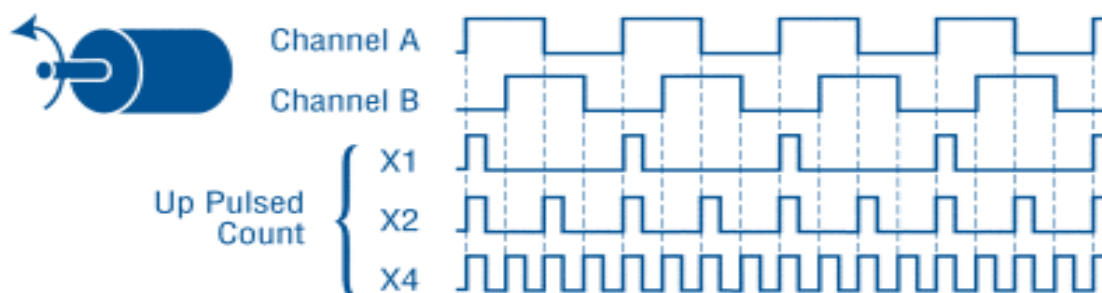
Inkrementella pulsgivares utsignal består av ett specifikt antal pulser med lika stort mellanrum mellan pulserna under ett helt varv för roterande pulsgivare. Detta anges som PPR (Pulses Per Revolution) som även kallas för en pulsgivares upplösning, och anger antalet pulser som registreras när pulsgivaren roterar ett helt varv.

För linjära optiska givare anges antalet pulser däremot som *Pulses per mm/inch*. För användningsområden där man inte är intresserad av att känna till rotations- eller rörelseriktningen är enkanaliga pulsgivare tillräckligt ändamålsenliga.

Däremot när vetskapen om riktningen är betydelsefull används pulsgivare med s.k. *Quadrature output* som betyder att pulsgivaren är två-kanalig där de enskilda kanalernas respektive utsignal är 90-grader elektriskt förskjutna gentemot varandra.

Den här typen kan även användas när man vill öka noggrannheten, till exempel vid avståndsmätning som bara löper i en riktning. Detta görs genom att räkna stig- respektive falltid hos en utav kanalernas utsignal och på så sätt dubblas antalet pulser per varv, se figur 11 nedan.

Vissa inkrementella pulsgivare har även en tredje kanal som kallas för Z-kanal eller referensmarkör. En sådan kanal genererar en enda puls som utsignal när pulsgivaren roterat 360-grader och kan användas för att t.ex. lokalisera en nollpunkt för en rörlig kapsåg. (Danaher, 2003, s. 3).



Figur 11. Inkrementell två-kanalig pulsgivares utsignal. (Danaher, 2003, s. 3).

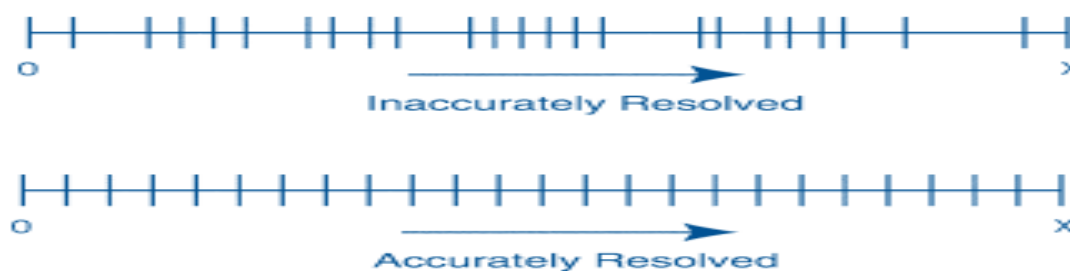
En absolut pulsgivare genererar en unik digital binär kod i datatypen *Word* som representerar pulsgivarens specifika position, hastighet och rörelseriktning. Med den här tekniken påverkas inte pulsgivaren av eventuella strömavbrott i och med att den binära koden är unik för varje position och någon referenspunkt behövs inte heller.

Upplösningen för en absolut pulsgivare definieras på basis av antalet bitar som dess utsignal i *Word* har. Utsignalen kan vara antingen rent binär eller i *Gray-code* där den senare endast genererar en enkel *bit* för varje förändring för att reducera eventuella fel.

Det finns även enkel- och multiroterande absoluta pulsgivare. En enkelroterande pulsgivares utsignalers kodning repeteras för varje rotationsvarv, vilket betyder att det inte finns någon indikering på att pulsgivaren har roterat ett eller flera varv. För en multiroterande pulsgivare är däremot utsignalerna för varje rotationsvarv unikt ända upp till 4096 rotationsvarv. (Danaher, 2003, s. 4).

Resolution eller upplösning är definitionen på antalet mätbara segment för en pulsgivare under ett rotationsvarv. Roterande pulsgivare finns med en upplösning på upp till 10 000 PPR, eller 40 000 PPR med A och B kanalens beräkning av stig- och falltid, medan linjära pulsgivare anges i antalet pulser per mikrometer (1 meter = 1000000 mikrometer).

När pulsgivare ska väljas till en process så finns en tumregel som säger att resolutionen hos pulsgivaren ska vara lika med eller större än vad processen kräver. Man brukar skilja på noggrannhet och resolution eftersom de är så olika och oberoende av varandra. För att förklara skillnaden mellan dem beskrivs fördelningen av pulserna för ett rotationsvarv med en linje där X står för 360-grader, se figur 12 nedan.



Figur 12. Skillnaden mellan resolution och noggrannhet för pulsgivare. (Danaher, 2003, s. 4).

Figuren beskriver sträckan X uppdelad i 24 delar vilket betyder att ett rotationsvarv ger 24-pulser. Den övre linjen visar en ojämn fördelning av de 24 delarna och en sådan konfiguration kan inte användas för att bestämma position, hastighet eller acceleration för någon rörelse. Den nedre linjen med jämn fördelning av de 24 pulserna har däremot en hög noggrannhet och resolution i och med att varje puls representerar $1/24$ av ett rotationsvarv. Exempelvis en process där man endast räknar om ett hjul har roterat ett helt varv så kommer de båda alternativen ge samma noggrannhet. (Danaher, 2003, s. 4).

2.5 Programmeringsspråk

När man för första gången kommer i kontakt med PLC-programmering lägger man ganska snabbt märke till att det finns ett antal olika programmeringsspråk till förfogande att välja mellan och de har alla sina för- och nackdelar. De flesta PLC-tillverkare håller sig till de standardiseringar som definieras för de olika PLC-programmeringsspråken i IEC 61131-3-standarden, vilket är en stor fördel när det kommer till hur väl man kan konvertera en programkod skriven i ett programmeringsspråk till ett annat. De fem olika programmeringsspråken består av följande: IL (Instruction List), ST (Structured Text), LD (Ladder Diagram), FBD (Function Block Diagram) och SFC (Sequential Function Chart).

2.5.1 IEC 61131-standardens omfattning

IEC (International Electrotechnical Commission) är en världsledande organisation som grundades år 1906. Organisationen publicerar internationella standarder för alla elektriska, elektroniska och tillhörande teknik som allmänt benämns elektroteknik. Man har från tidigare lagt ner mycket tid och arbete på att fastställa och utveckla en standard för teknologin kring PLC-programmering och IEC 61131-standarden blev den första som fick internationellt och industriellt godkännande. IEC 61131-standarden grundar sig på utveckling av tidigare publicerade standarder som utkom mellan år 1977–1985 och de första delarna av IEC 61131-standarden publicerades år 1990. I dagsläget består IEC 61131-standarden av nio olika delar som sammanfattar de krav och regler som ställs på moderna PLC-system, vilka gäller både för utveckling av PLC-maskinvaran och programmeringssystemet. De olika delarna som ingår i den aktuella IEC 61131-standarden är listade i tabellen nedan enligt innehåll och publiceringsår. (IEC, u.å.)

Tabell 1. Definiering av de nio delarna i IEC 61131-standarden. (IEC, u.å.)

| Standard | Publicering | Innehåll |
|----------|-------------|--|
| 61131-1 | 2003 | General information |
| 61131-2 | 2007 | Equipment requirements and tests |
| 61131-3 | 2013 | Programming languages |
| 61131-4 | 2004 | User guidelines |
| 61131-5 | 2000 | Communications |
| 61131-6 | 2012 | Functional safety |
| 61131-7 | 2000 | Fuzzy control programming |
| 61131-8 | 2003 | Guidelines for the application and implementation of programming languages |
| 61131-9 | 2013 | Single-drop digital communication interface for small sensors and actuators (SDCI) |

2.5.2 IL – Instruction List

Programmeringsspråket IL är ett relativt simpelt och begripligt lågnivåspråk som har en struktur som påminner om assembler och är ett programmeringsspråk som omvandlar maskinkod dvs. ettor och nollor till läsbar kod. Arbetsgruppen för IEC 61131–3-standarden har tagit fram IL som ett standardiserat PLC-programmeringsspråk genom att ha forskat kring strukturen av de patentskyddade IL-lågnivåspråken som flera olika PLC-tillverkare erbjuder. De har sedan tagit gemensamma kännetecken hos flera av dem som standard för utvecklandet av IEC-standardens IL. Programmeringsspråket IL baseras på serier av instruktioner där varje instruktion tilldelas en egen rad, vilket kan ses för exemplet i figur 13 nedan.

| <u>Label</u> | <u>Operator</u> | <u>Operand</u> | <u>Comment</u> |
|--------------|-----------------|----------------|------------------------|
| | LD | Speed | (* Load Speed and *) |
| | GT | 1000 | (* Test if > 1000 *) |
| | JMPCN | VOLTS_OK | (* Jump if not *) |
| | LD | Volts | (* Load Volts and *) |
| | ADD | 20 | (* Increase by 20 *) |
| | ST | Volts | (* Store Volts *) |
| VOLTS_OK: | LD | 1 | (* Load 1 and store *) |
| | ST | %Q57 | (* in output 57 *) |

Figur 13. Exempel på IL-programmeringsspråkets struktur. (Lewis, 1998, s. 171).

För att förklara vad som menas med kodraderna i exemplet i figur 13 ovan måste man använda sig av de engelska termerna i och med att det inte finns några bra exempel på svenska ersättningsord för en del av dem. LD (Load Operator) betyder att en variabls värde hämtas från minnet. GT (Greater Than) är en funktion som jämför t.ex. en variabls värde i förhållande till det som specificeras i nämnaren, *Operand*. JMPCN (Jump if NOT) är en funktion som byter till följande kodrad ifall en ovanstående kodrad inte stämmer överens med ett påstående i dess nämnare. ADD (Addition) betyder att funktionen adderar ett tal med det som specificerats i nämnaren. ST (Store) är en funktion som används för att spara ett angivet värde till en variabel. Vad programmet gör är att det kontrollerar om variabeln *Speed* är större än värdet 1000. Om påståendet stämmer kommer utgången *%Q57* få en logisk 1:a, sant värde *True* och om det inte stämmer tilldelas utgången istället en 0:a falskt värde *False*. (Lewis, 1998, s. 169–176).

2.5.3 ST – Structured Text

ST är ett relativt omfattande högnivåspråk som har en stor mängd funktioner för att t.ex. skriva värden till variabler, anropa funktioner och funktionsblock, skapande av olika uttryck och skapa villkor för utvärdering av valda funktioner för upprepning av exempelvis specifika sektioner i ett program. Se programexempel skriven i ST i figur 14 nedan.

I och med att ST baseras på programmeringsspråket *PASCAL* så påminner deras strukturer lite om varandra. Dock är programmeringsspråket ST speciellt utvecklat för just industriella styrsystem. (Lewis, 1998, s. 117–118).

```

FUNCTION_BLOCK TEST_POWER
  VAR_INPUT
    CURRENT, VOLTS1, VOLTS2, VOLTS3 : REAL;
  END_VAR
  VAR_OUTPUT
    OVERVOLTAGE : BOOL;
  END_VAR

  IF VOLTS1*CURRENT > 90 THEN
    OVERVOLTS := TRUE; RETURN;
  END_IF;
  IF VOLTS2*(CURRENT+20.0) > 90 THEN
    OVERVOLTS := TRUE; RETURN;
  END_IF;
  IF VOLTS3*(CURRENT+30.0) > 90 THEN
    OVERVOLTS := TRUE;
  END_IF;
END_FUNCTION_BLOCK;

```

Figur 14. Exempel på ST-programmeringsspråkets struktur. (Lewis, 1998, s. 130).

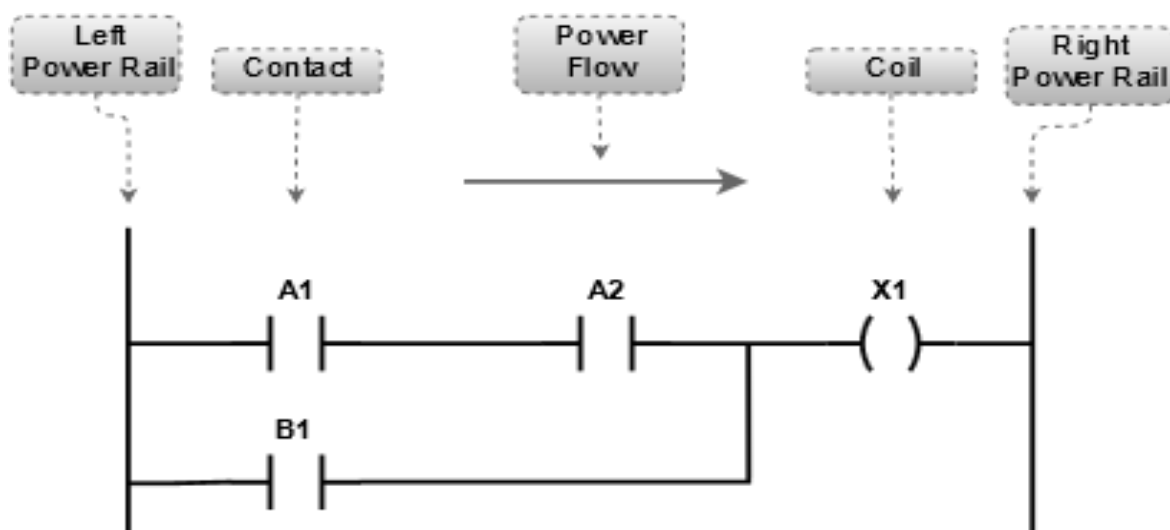
Exemplet ovan i figur 14 utgör ett funktionsblock som heter *TEST_POWER*. Variablerna med invärden till funktionsblocket heter *CURRENT*, *VOLTS1*, *VOLTS2* och *VOLTS3* som alla är av datatypen *REAL*, vilket betyder flyttal. Funktionsblocket kan skriva ett logiskt värde *True* eller *False* till variabeln *OVERVOLTAGE*.

Om någondera av de tre påståendena är större än talet 90 kommer variabeln *OVERVOLTS* att tilldelas det logiska värdet *True*. *RETURN* betyder i kodraden att ifall påståendet, *IF VOLTS1*CURRENT > 90 THEN*, i översta raden är sant kommer funktionsblocket avslutas efter att det gett variabeln *OVERVOLTS* värdet *True*. I och med att programmet inte behöver kontrollera de två andra påståendena kan man med andra ord säga att de hoppas över. (Lewis, 1998, s. 130).

2.5.4 LD – Ladder Diagram

Programmeringsspråket LD är bland de mest allmänt använda PLC-programmeringsspråken i världen. Anledningen till att LD är så populärt är för att det baserar sig på strukturen hos s.k. relälogik och är väldigt lätt att lära sig. Bestämmelserna i IEC-standarden för hur strukturen för LD ska se ut är ganska enkla.

LD läses alltid från den vänstra spalten *Left Power Rail* horisontellt till den högra. Varje kontakt representerar ett logiskt värde som antingen är sant eller falskt när det aktiveras respektive inaktiveras. Genom att använda öppnande och slutande kontakter kan man på så sätt skapa olika logiska funktioner för att styra en eller flera logiska eller fysiska utgångar. Den högra *Power Rail* strömskenan är inte obligatorisk eftersom läsligheten inte påverkas av dess frånvaro. (Lewis, 1998, s. 151–153).



Figur 15. Exempel på LD-programmeringsspråkets struktur. (Lewis, 1998, s. 152).

Logiken för LD nätverket i figur 15 ovan kan beskrivas enligt följande: Om kontakterna A1 och A2 eller B1 har ett värde som är sant dvs. *True* kommer utgången X1 aktiveras. Skulle man skriva om funktionen i LD till programmeringsspråket ST, skulle koden kunna se ut som den beskriven nedan som också ger samma logiska funktion:

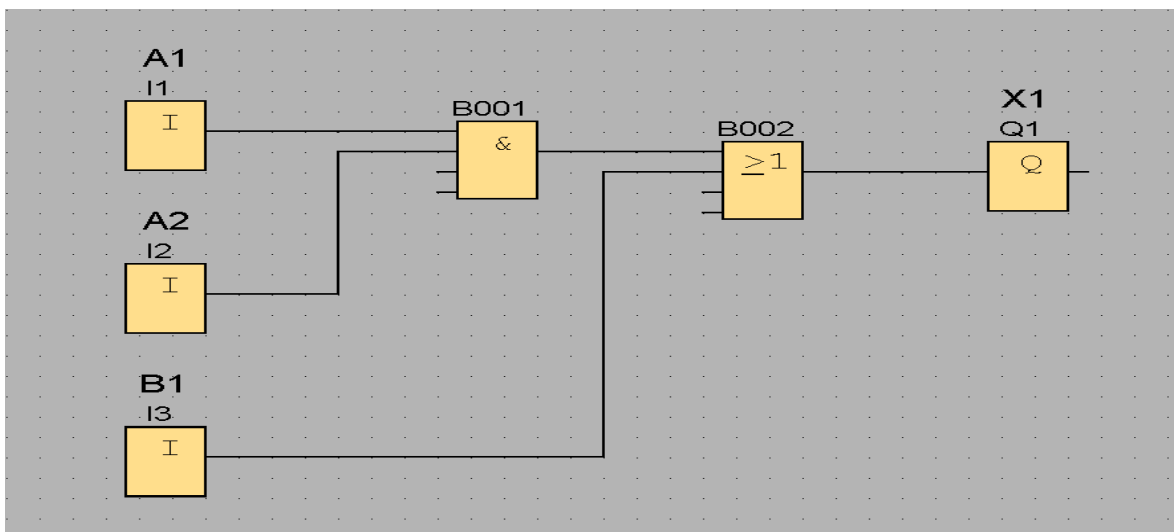
X1 := (A1 AND A2) OR B1;

2.5.5 FBD – Function Block Diagram

FBD är ett grafiskt programmeringsspråk som kan användas för att beskriva beteendet hos olika funktioner, funktionsblock och diverse program som en form av internt förbundna grafiska block. FBD baseras på visning av ett system i form av signalflöden mellan flera block som utför olika funktioner.

Ett nätverk i ett FBD-program har därmed en slående likhet med ett kretsschema i vilket man beskriver elektriska förbindelser mellan komponenter. Signalflödet mellan blocken i ett FBD-program anses gå från utgångarna på blocken till ingångarna på andra funktioner och funktionsblock, se exemplet i figur 16 nedan.

Bestämmelserna i IEC-standarden för hur de olika FBD-blocken och funktionerna ska se ut är uppdelat i semi-grafisk och full-grafisk representation, av vilka den senare används i största grad av de flesta PLC-tillverkare för designens skull. (Lewis, 1998, s. 135–137).



Figur 16. Exempel på FBD-programmeringsspråkets struktur i programmeringsverktyget Siemens LOGO!Soft Comfort V6.0.

FBD kan också användas inuti programmeringsspråket SFC (Sequential Function Chart) för att beskriva funktionen för olika övergångar, sekvenser, steg och åtgärder. Ett program/nätverk som skapas i programmeringsspråket FBD kan vanligtvis omvandlas utan komplikationer till exempelvis ST. Men att omvandla från ST till FBD fungerar inte alltid eftersom att funktioner som exempelvis komplicerade matematiska beräkningar med SIN, COS och SQRT m.m. inte finns att tillgå i FBD. (Lewis, 1998, s. 145–146).

2.5.6 SFC – Sequential Function Chart

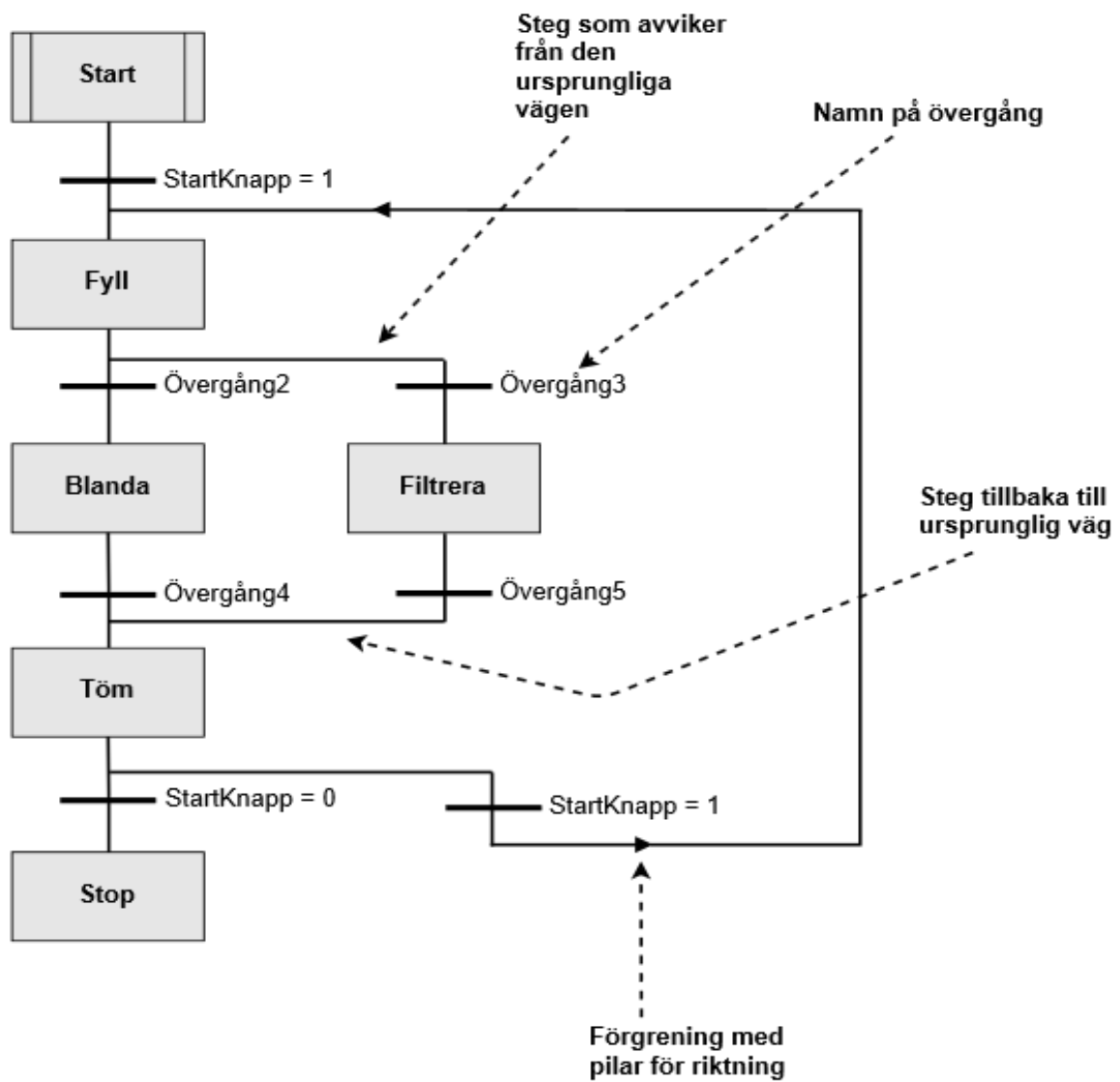
Programmeringsspråket SFC är beskrivet i IEC-standarden som ett grafiskt programmeringsspråk vilket tillåter alla sekvensstyrda funktioner i ett PLC-program att beskrivas i grafisk form.

SFC kan användas för att beskriva en översikt över funktionerna i en process i form av boxar. Det är även mycket användbart för avskiljning när exempelvis ett styrproblem behöver lokaliseras i ett program. Man kan då snabbt gå direkt till det steg i programmet som inte fungerar, vilket gör att man inte behöver gå igenom alla programstegen enskilt vid felsökning.

I SFC ska ett program alltid börja med ett inledande steg, ”Start”, se figur 17 på följande sida. Detta steg definieras av en rektangulär box med två raka vertikala linjer inuti. Det inledande steget är det första som läses av, och programflödesriktningen läses följdaktligen också vanligtvis från topp till botten. Det ska endast finnas ett inledande steg i ett SFC program och det steget förblir aktivt ända tills det följande steget aktiveras efter att villkoren i övergången mellan dem uppfylls.

De andra rektangulära boxarna som representeras med namnen ”Fyll”, ”Blanda”, ”Filtrera” och ”Töm” o.s.v. kallas för steg. Dessa steg har ett eget program inuti sig för att utföra olika åtgärder. Programmen inuti stegen kan skrivas med något av de övriga programmeringsspråken i IEC-standarden nämligen IL, ST, LD och FBD.

Övergångarna före och efter varje steg representeras av horisontella linjer. Dessa övergångar består av logiska villkor som ska uppfyllas för att aktivera respektive avsluta ett steg. (Lewis, 1998, s. 185–190).



Figur 17. Exempel på SFC-programmeringsspråkets struktur. (Lewis, 1998, s. 187).

3 Maskinlinje UNI-2

Vid maskinlinje, eller som den internt kallas rörlinje, UNI-2 tillverkas rostfria syrafasta processrör med ett antal olika standardiserade diameterstorlekar, vägg tjocklekar och längdmått, se tabell i figur 18 nedan. Redan på 1970-talet blev den mångåriga maskinlinjen planerad och tillverkad. Den har sedan dess moderniserats flera gånger fram till dagsläget.

För att få en överblick över maskinlinjens storlek så är dess totala längd på ca 35 meter med alla tillhörande delar inräknade. I och med att maskinlinjen är av egentillverkad konstruktion är den också väldigt unik i sitt slag även om flera liknande maskinlinjer vid företaget delar samma grundkonstruktioner, men skiljer sig åt främst i storlek och andra viktiga egenskaper.

Maskinlinjen är konstruerad för att kunna vara i drift dygnet runt och därför tillämpar man treskiftssystem med en maskinoperatör per arbetsskift. Likt alla maskiner inom industrin slits även denna maskinlinje och man försöker hålla ner driftstoppen till ett minimum vid diverse underhållsåtgärder. Främst under semestertider och vid längre högtider utför man lite mera komplicerade förebyggande underhållsåtgärder på maskinlinjerna inom företaget vid behov.

Produktsortiment för maskinlinje UNI-2:

- Rördiameter Ø: ca 21,3–60,3 mm
- Vägg tjocklek: ca 1,5–3,6 mm
- Rörlängd: standard 6000 mm

| ISO 4200 Nominal diameter DN | SSG 1361 Tab 3-4 OD mm | EN ISO 1127 Outer Diameter mm Serie | | | | | ANSI B36,19M | | Imperial OD | Hygienic OD | Hydraul OD |
|---------------------------------------|---------------------------------|---|---------|---------|------|-------|-----------------|-------|----------------|----------------|---------------|
| | | 1 | 2 | 3 | Old | NPS | mm | mm | Inch | mm | mm |
| | 20 | 21.3 | 19 20 | 18 | 20 | 1/2 | 21.3 | 19.05 | 3/4 | 18/19 | 20 |
| 20 | | | | | | | | | | 22/23 | 22 |
| | 25 | 26.9 | 25 | 22 25.4 | 25 | 3/4 | 26.7 | 25.4 | 1 | 25 | 25 |
| 25 | | | | | | | | | | 28/29 | 28 |
| | 30 | 33.7 | 31.8 32 | 30 | 30 | 1 | 33.4 | 31.75 | 1 1/4 | 32 | 30 |
| 32 | | | | | | | | | | | |
| | | | | | | | | | | 34/35 | 38 |
| | 38 | 42.4 | 38 40 | 35 | 38 | 1 1/4 | 42.2 | 38.1 | 1 1/2 | 40/41 | 38 |
| 40 | 43/44 | 48.3 | | 44.5 | 44.5 | 1 1/2 | 48.3 | 44.5 | 1 3/4 | | 50 |
| 50 | | | 51 | | | | | 50.8 | 2 | 51 | |
| | 53/54 | | 57 | 54 | | | | | | 52/53 | |
| | | 60.3 | | | 57 | 2 | 60.3 | | | | 60 |

Figur 18. Tabellen är tagen från OSTP Product Cataloge 2014 och visar de standardiserade måtten på de rördiametrar som tillverkas vid maskinlinje UNI-2.

3.1 Maskinlinjens beståndsdelar

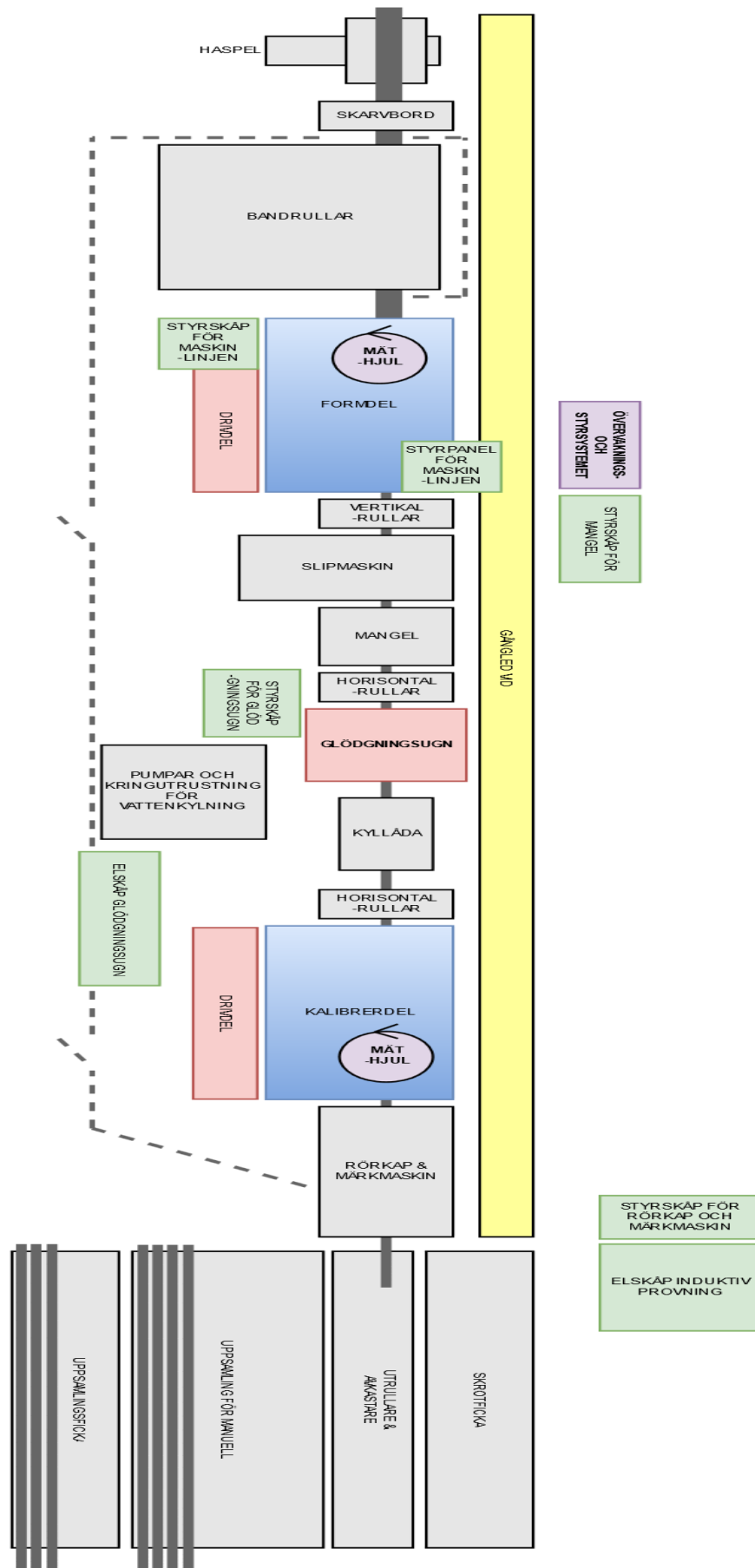
Maskinlinjen är relativt komplext uppbyggd och består därmed också av ett flertal olika egentillverkade maskiner som tillsammans med industriellt utvecklade och standardiserade maskiner bildar ett antal olika maskinhelheter vid maskinlinjen. Därför kommer maskinlinjens huvuddelar som har att göra med tillverkningsprocessen att förklaras närmare i ordningsföljd enligt följande lista:

- **Haspel** – Maskindel som stegvist lindar upp plåtrullen eller den s.k. ”råvaran”.
- **Skarvbord** – Ett stödbord som används när man sammanfogar slutet på pågående plåtrulle med följande genom manuell svetsning.
- **Bandrullar** – Maskindel som gör det möjligt att sammanfoga plåtrullar utan att stanna maskinlinjen.
- **Instyrning** – Maskindel som utför kantfräsning på plåten samtidigt som den matas in till formdelen.
- **Formdelen** – Är en maskinhelhet som består av form- och rullbockar. Inuti formdelen finns även horisontalrullar, svetsbord och kyllåda för svetsfogen.
- **Vertikalrullar (1)** – Är vertikalt roterande stödrullar som varierar i storlek och utförande på maskinlinjen.
- **Slipmaskin** – Är en maskindel som fungerar likt en bandslip. Den har till uppgift att slipa ner svetsfogen på röret för att avlägsna slagg.
- **Vertikalrullar (2)** – Samma beskrivning som för Vertikalrullar (1).
- **Mangel/Vals** – Mangeln är en hydraulstyrd mekanisk maskin som valsar fram- och tillbaka svetsfogen från insidan av röret och sträcker sig mellan formdelen fram till horisontalrullpar 1. Delen som valsar svetsfogen inuti röret kallas för dornet.
- **Horisontalrullar (1)** – Är horisontalt roterande stödrullar som varierar i storlek och utförande på maskinlinjen.
- **Glödgningsugn** – Är en form av induktionsugn som har till uppgift att jämna ut inre spänningar hos röret genom att värma upp det till ca 600–900°C.

- **Kyllåda** – Egentillverkad konstruktion som kyler ner röret efter glödgningsugnen. Det krävs en hel del vatten för kylningen av röret och därmed finns det pumpar och kringutrustning som förser maskinlinjen med kallt vatten.
- **Horisontalrullar** – Samma beskrivning som för horisontalrullar (1).
- **Kalibrerdel/Dragdel** – Är en maskinhet som består av rullbockar som både drar fram röret och delvis plastiskt tillformar röret. Inuti kalibrerdelen finns även utrustning som induktionsprovning. Induktionsprovningen kontrollerar att det inte finns några hål i svetsfogen eller plåten.
- **Turk huvud** – Placerad i slutet av kalibrerdelen och är en typ av riktbock med horisontalt och vertikalt roterande formrullar eller styrhjul som ser till att röret blir så rakt som möjligt. Heter på engelska för *Turk Head Assembly Machine*.
- **Rörkap och märkmaskin** – Rörkapen kapar röret till rätt längd vilket utförs under processens gång och som betyder att sågen förflyttar sig i förhållande till röret samtidigt som märkmaskinen stämplar in produktnummer och annan information på röret.
- **Utrullare och avkastare** – Är en enkel-motorstyrd rullbana som tar tag i det tillverkade röret för att sedan enligt vald styråtgärd med avkastare flytta röret till uppsamlingsbanan eller skrotfickan. Avkastare är luftstyrda cylindrar med mekaniska ”armar”.
- **Skrotficka, uppsamlingsbana och uppsamlingsficka** – Skrotfickan är en uppsamlingsplats för defekta rör. Uppsamlingsbanan används för manuell gradning av ändorna på rören och uppsamlingsfickan är den plats där de färdigt tillverkade rören samlas innan de flyttas bort från maskinlinjen till betning.

Figur 19 på följande sida visar en översiktlig bild av rörlinje UNI-2 med maskindelarna representerade som lådor. De rosa färgade lådorna, övervaknings- och styrsystemet samt de två mät hjulen, representerar de synliga förändringarna som gjorts på maskinlinjen till följd av projektet.

Allt som representeras i figur 19 är dock inte i rätt skala i förhållande till den verkliga maskinlinjen och dess kringutrustning. Men det ger en överskådlig översikt av hur omfattande en rörtillverkningslinje kan vara.



Figur 19. Översiktlig beskrivning av rörlinje UNI-2

3.2 Tillverkningsprocess vid maskinlinje UNI-2

Tillverkningsprocessen börjar från det att man har en rostfri plåt med specifika kvalitetsegenskaper, mängden Nickelhalt (Ni), Kromhalt (Cr) och plåttjocklek som är ihoprullad som en enda stor plåtrulle vilket kan beskrivas som "råvaran". För att sammanfatta tillverkningsprocessen för ett processrör utan att gå in på alltför detaljrika uppgifter om maskinlinjen, kan man förklara den fortlöpande processen uppdelad i 6 olika steg enligt följande:

1. Plåtrullen lindas stegvis ut från en roterande vals kallad haspel och matas sedan in i maskinlinjen där plåten formas kontinuerligt i olika steg för att bilda strukturen av ett cirkulärt rör. Maskindelen där röret tillformas kallas helt enkelt för formdelen.
2. Det formade röret sammanfogas sedan genom svetsning med två olika svetsmetoder för att få den rätta hållfastheten på svetsfogen som bestäms av plåttjockleken. TIG-svetsning används vid plåttjocklekar upp till ca 2,6 mm medan tjockare plåt svetsas med plasmavetsning.
3. Eftersom röret både formas och svetsas medför det ojämnt fördelade spänningar i plåten på röret. Av den anledningen behöver röret genomgå en värmebehandling i en glödgningssugn som åtgärdar dessa och andra negativt inverkanse kvalitetsfaktorer. Direkt efter värmebehandlingen av röret ska det kylas ner för att uppnå vissa specifika hållfasthetsegenskaper.
4. Därefter får röret sin slutgiltiga form i en maskindel kallad kalibreringsdel och där röret både tillformas i viss grad samtidigt som även svetsfogen kontrolleras för eventuella uppkomna håligheter i det tillverkade rörets svetsfog.
5. Efter den slutgiltiga tillformningen av röret ska det sedan kapas till korrekt längd och få sin rätta produktmärkning samt eventuell finputsning av dess kapade ändor.
6. När rören är tillverkade ser de ganska smutsiga ut pga hela tillverkningsprocessens olika skeden varvid röret ska genomgå en s.k. betningsprocess där röret får sin rätta färg och icke önskvärda partiklar och slagg avlägsnas. Därefter granskas rören i olika steg med bl.a. röntgen av svetsfogen och andra viktiga inspektioner för att säkerställa kvaliteten innan de slutligen kan packas och är klara för leverans.

3.3 Projektets problemformulering kring processfelet

Uppdraget kretsar kring ett tidigare känt processfel hos maskinlinjen som kan uppkomma till följd av felaktigt drifttillstånd hos processen. Uppkomsten av processfelet baserar sig främst på att hastighetsvariationerna mellan maskinlinjens formdel och kalibrerdel blir alltför stora i förhållande till varandra. Detta kan leda till att röret som tillverkas antingen kan dras av eller deformeras inuti glödgningsugnen om processfelet får fortgå utan att maskinoperatören ingriper genom att stoppa maskinlinjen. Dessa två händelseförlopp har hänt tidigare och beror på att någondera av drivdelarna inte åtföljt varandra på korrekt sätt.

Vanligen hinner inte processfelet uppträda om maskinoperatören är inom maskinlinjens arbetsområde och övervakar processen, av den orsaken att processfelet är en ganska långsam förändring. Men i och med att maskinlinjen i allmänhet inte stannas under arbetspauser och dylikt så är det samtidigt lätt hänt att något fel uppstår i tillverkningsprocessen under tiden maskinoperatören inte är på plats.

Maskinlinjens styrsystem och maskinhelheter har givetvis flertalet olika givare och stoppkretsar för att förhindra olika fel i tillverkningsprocessen. Men maskinlinjen saknar ändå ett konkret övervaknings- och styrsystem för tillverkningsprocessen som framförallt skulle vara gynnsamt i de fall då maskinlinjen lämnas obevakad under drift. Samtidigt är det viktigt att få någon form av återkoppling från uppkomsten av processfel för att kunna förhindra sådana i framtiden och det är just där vikten av ett övervaknings- och styrsystem kommer in i bilden.

Ofta kan ett befintligt styrsystem omkonfigureras för att göra det mera avancerat. Men i praktiken skulle det innebära att hela styrsystemet och all kringutrustning byts ut i och med att det inte finns någon dokumentation kvar för maskinlinjen på grund av alla omändringar som gjorts under årens lopp. Det här är nog kanske den vanligaste typen av utmaningar som man stöter på ute i industrin när befintliga maskinlinjer saknar dokumentation. När dokumentation saknas försvårar det även felsökning på maskiner och kringutrustning också när de ska hållas i drift.

3.4 Funktionsspecifikationen

Med en funktionsspecifikation definierar man vad övervaknings- och styrsystemet ska kunna utföra och klara av men inte hur. Alla de krav som ställs av processen och användarna måste en funktionsspecifikation innehålla. Skapandet av en funktionsspecifikation går därmed från en övergripande och översiktlig nivå ner till en mera detaljerad nivå.

En överblick över projektet skapas normalt utifrån en verksamhetsanalys i vilken de funktionella kraven framkommer och som baserar sig på kunskap om processen samt användarna av systemet. En funktionsanalys fokuserar på processutrustning, styrsystem och varje arbetsmoment som människan har i processen. Med dessa uppgifter kan man då metodiskt härleda fram och formulera funktionsspecifikationen.

Efter att en funktionsspecifikation är framtagen ska systemet sedan realiseras. I allmänhet utförs detta genom att funktionsspecifikationen erbjuds åt ett antal leverantörer som var och en för sig skapar en systemspecifikation som beskriver hur deras lösning för systemet ser ut. Därefter väljs en lämplig leverantör för systemet. När systemet är färdigt och kontrollerats kan det sedan installeras och därefter tas i bruk för processen. Efter att systemet varit i drift en viss tid behöver systemet i regel ersättas av olika anledningar, bl.a. tekniska och ekonomiska eller båda. Vid den tidpunkten inleds en ny livscykel för följande system.

Exempel på hur en felaktig och en användbar funktionsspecifikation specificeras ges i exemplet nedan:

Antag att en stol funktionellt ska specificeras. Om man specificerar att ”en stol ska vara utrustad med fyra ben med gummibeklädda fötter och ha ett stöd för ryggen” är ingen funktionell beskrivning av en stol. ”Stolen ska vara bekväm, ge stöd för armar och rygg samt inte lämna märken på golvet”, är ett exempel på en bättre specifikation. I den senare specifikationen för stolen överläts åt tillverkaren möjlighet att erbjuda stöd för armar och rygg och att klara av kravet att stolen inte lämnar märken efter sig på golvet. Definitionen på hur bekväm en specifik stol är utvärderas sedan när de tänkbara leverantörerna presenterar sina lösningsförslag till uppdraget. (Cegrell, 1994, s. 71).

Funktionsspecifikation för driftövervakning vid rörlinje UNI-2

Användar- och funktionsbeskrivning:

Systemet ska användas för att förhindra en specifik typ av processfel hos maskinlinjen, se punkt (3.3). Användarna av övervaknings- och styrsystemet är maskinoperatörerna vid maskinlinjen samt underhållsavdelningens elautomationsmontörer som utför eventuell felsökning av systemet. Maskinoperatörerna ska inte behöva ingripa för att övervaka eller styra systemet utan endast ges processinformation i form av lägesindikering av processläget eller kunna se på insamlade värden i ett användargränssnitt. De funktioner som behövs för att realisera detta är hastighetsjämförelse mellan maskinlinjens kalibrerdel och formdel. Via användargränssnittet ska behörig personal kunna ställa in de värden som påverkar regleringen av de styråtgärder som utförs av PLC-programmet.

Övergripande systemkrav:

Övervaknings- och styrsystemet ska installeras inuti ett separat elskåp som ska monteras bredvid eller på maskinlinjen på lämpligt ställe inom maskinoperatörens arbetsområde. Elskåpet ska vara konstruerat så att man enkelt kan demontera och använda övervaknings- och styrsystemet på andra motsvarande applikationer. Elskåpet ska lämpligen innehålla en PLC-logik som styrsystem och en HMI-panel som användargränssnitt från tillverkaren Siemens för att reservdelar och programmeringen ska bli enhetligare hos maskinlinjen. De signaler som styrsystemet ska kunna processera är följande:

Insignaler:

Signaler från två stycken pulsgivare för hastighetsmätning och signaler från två stycken frekvensomriktare för strömmätning samt möjlighet till eventuella tillägg.

Utsignaler:

Signaler för styrning av ljusstorn samt styrning av maskinlinjens linjestopp och möjlighet till eventuella tillägg.

Utöver detta ska även systemet kunna samla in data från processens linjehastighet för form- respektive kalibrerdel samt strömåtgång hos drivdelsmotorerna. Datainsamlingen står som grund för att optimera värden för de olika styråtgärderna och behöver inte sparas i någon databas utan istället sparas på lämpligt sätt lokalt i övervaknings- och styrsystemet.

4 Planering

När man påbörjar planeringen av ett projekt som omfattar ett övervaknings- och styrsystem behöver man ha gjort en utförlig funktionsspecifikation på vad systemet ska utföra och klara av, se punkt (3.4).

Samtidigt genomfördes en grundlig undersökning över hur maskinlinjen fungerar för att få en så bra processöversikt som möjligt. Under planeringens gång fördes även givande diskussioner med maskinlinjens operatörer vilket gav ett bra underlag för utvecklingen av användargränssnittet och kunskap om processen.

Vid planering av ett övervaknings- och styrsystem börjar man vanligtvis med att följa upp och undersöka eventuella gamla dokument och hur det befintliga systemet är uppbyggt. Därefter bestäms det vilka I/O-moduler som behövs för systemet.

När man bestämt antalet in- och utgångar som behövs kan man sedan välja ut den maskinvara som lämpar sig för systemet. Efter att maskinvaran är vald fortsätter man med att rita upp ett kretsschema för systemet och till sist en centrallayout, se bilaga A på sid 2 och 4.

4.1 Processöversikt och genomgång av nuvarande uppföljningssystem

Det nuvarande uppföljningssystemet är ett helt fristående system som samlar in information från processen och som endast ett fåtal personer i företaget får utföra ändringar på. I och med det har inte heller det nuvarande uppföljningssystemet någon stor vikt som behöver beaktas i projektet. Det enda som behövde tas i beaktande var att inte påverka maskinlinjens signal för linjehastighet, som det befintliga uppföljningssystemet använder.

Maskinlinjen är utrustad med två stycken separata mätjul från tidigare, se figur 20 nedan. Det ena mätjulet med en rullomkrets på 500 mm är utrustat med två stycken pulsgivare och placerat vid slutet av kalibrerdelen. Dessa pulsgivare har till uppgift att räkna ut kaplängden för röret samt för uppföljning av linjehastigheten till det befintliga uppföljningssystemet.

Det andra lite mindre mätjulet med en rullomkrets på 200 mm har en pulsgivare. Det mindre mätjulet som befinner sig vid början av formdelen har till uppgift att visa linjehastigheten åt maskinoperatören på en digital display.

Med dessa mätjul konstaterades att man mycket väl kan använda dem för att mäta hastigheten före och efter glödgningsugnen och på så sätt jämföra hastigheterna med varandra för att erhålla information om processförhållandet.



Figur 20. Mätjulet till vänster hör till kalibrerdelen och det högra till formdelen.

4.2 Val av maskinvara och komponenter för systemet

Vid valet av maskinvara för övervaknings- och styrsystemet valdes en Siemens S7-1200 seriens PLC samt en Siemens KTP900 Basic HMI-panel till användargränssnittet. Valet föll på dessa två enheter av den anledningen att de var det förmånligaste alternativet och har alla de nödvändiga funktionerna som behövs för förverkligande av övervaknings- och styrsystemet.

Uppdragsgivaren håller även internt som standard att alla PLC:n som installeras vid företaget ska vara från tillverkaren Siemens. Anledningen till det är att lagerhållning av nödvändiga reservdelar och komponenter görs enklare samt att kunskapen hos montörerna som utför underhåll och felsökning är god och effektiv för Siemens PLC:n.

För styrenheten fanns ett krav på att kunna räkna de signaler som pulsgivarna hos de bägge mät hjulen gav. Ju mindre mät hjul och ju större upplösning man har, desto högre pulsfrekvens. I och med att de inkrementella pulsgivarna har en upplösning på 500 PPR. Kan man med hjälp av formeln nedan räkna ut pulsfrekvensen hos en sådan pulsgivare om processen har en hastighet på exempelvis 8 m/min och det mindre mät hjulets rullomkrets är 200 mm.

Beräkning av pulsfrekvensen x:

a = 500 pulser, b = 60/1 s/min, c = 0,2 m och d = 8 m/min

$$x = \frac{a}{\frac{b * c}{d}} = \frac{500 \text{ pulser}}{\frac{60}{1} \text{ s/min} * 0,2 \text{ m}} = 333,333 \text{ pulser/s}$$

$$\frac{500 \text{ pulser}}{8 \text{ m/min}}$$

4.2.1 PLC – Siemens S7-1200

Som PLC för övervaknings- och styrsystemet valdes en Siemens S7-1215C DC/DC/DC med serienummer (6ES7215-1AG40-0XB0), som kan ses i figur 21 nedan.

Denna PLC som hör till kategorin kompakta PLC:n med inbyggda I/O-moduler som har, två stycken Profinet/Ethernet portar, 14st digitala ingångar 24 V DC, 10st digitala utgångar 24 V DC, två stycken analoga ingångar 0–10 V DC och två stycken analoga utgångar 0–20 mA. Att PLC:n har två stycken ethernet portar ger också möjligheten att kunna programmera både PLC:n och HMI-panelen samtidigt.

Det interna minnet hos en S7-1215C CPU är 125 kB stort och räcker mycket väl till för projektet. Det finns även möjlighet till att utvidga minneskapaciteten genom att köpa till en specifik typ av Siemens SD minneskort.

En orsak till varför denna PLC valdes var att PLC:ns 4-digitala ingångar, DI a.0 - DI a.3, är s.k. *High Speed Counters* som klarar av att läsa pulsfrekvenser på upp till 100 kHz. Med det menas att 100 000 pulssignaler hinner dessa digitala ingångar läsa in under loppet av en sekund vilket är användbart vid räkning av pulser från inkrementella pulsgivare. 100 kHz ger en stor marginal i förhållande till den pulsfrekvens som de pulsgivare som används i projektet ger. För mer info om PLC:n se bilaga B på sid 3.



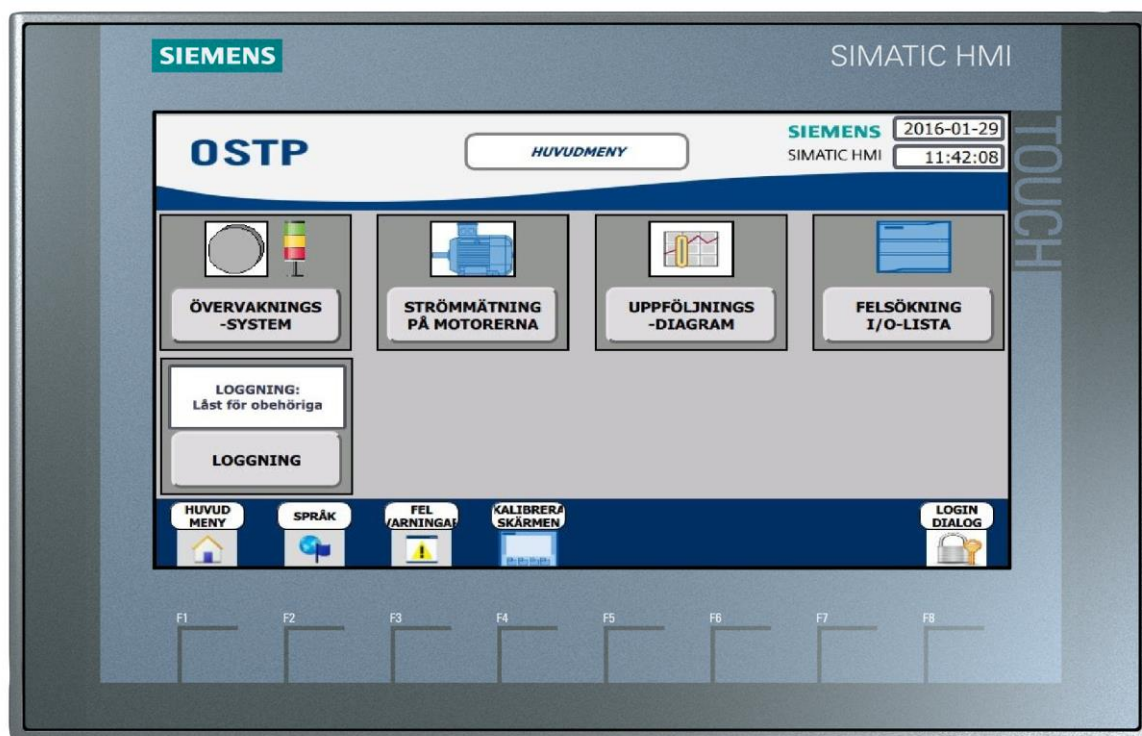
Figur 21. PLC – Siemens S7-1215C DC/DC/DC.

4.2.2 HMI-panel – Siemens KTP900 Basic PN

Till användargränssnittets maskinvara valdes en HMI-panel från tillverkaren Siemens. Modellen heter KTP900 Basic PN med serienummer (6AV2123-2JB03-0AX0), som kan ses i figur 22 nedan tillsammans med användargränssnittets konfigurerade huvudmeny.

KTP900 Basic är en 9”-tums TFT-skärm med 65536 färger som förutom pekskärmen också är utrustad med åtta stycken programmerbara tryckknappar. PN (PROFINET) betyder att panelen använder sig av ett s.k. *Industrial Ethernet* kommunikationsprotokoll för att skicka och ta emot data.

För att kunna logga data i någon större mängd med denna HMI-panel måste man använda ett USB-minne som kopplas till USB-porten på skärmens baksida. Se bilaga B sid 4 för mer detaljerad info om HMI-panelen.



Figur 22. HMI – Siemens KTP900 Basic PN.

4.2.3 Strömkälla och relämoduler

Som 24 V DC matning till PLC:n, HMI:n, relämodulerna och ljusstornet valdes en Siemens S7-1200 Power Module PM1207 med serienummer (6EP1332-1SH71), som kan ses till vänster i figur 23 nedan. PM1207 är en enfastransformator med likströmsriktare på sekundärsidan och ger således ut en likströmsspänning på 24 V DC och kan mata upp till 2,5 A ström. Dess konstruktion och design är främst gjord för att passa in tillsammans med en S7-1200 seriens PLC, lämpligen inuti ett elskåp eftersom dess IP-klass är IP20. För mer information om strömkällan PM1207, se bilaga B sid 5.



Figur 23. Siemes – PM1207 & Phoenix Contact – Relämodul

För att kunna styra och mata tillräckligt med ström till ljusstornets olika lampor och dess siren behöver man använda sig av relän eftersom PLC:ns utgångar endast är gjorda för signalstyrning och inte som matningskretsar. Den relätyp med tillhörande relämodul som använts i projektet är av märket Phoenix Contact med modellbeteckningen (PLC-RSC-24DC21), som kan ses till vänster i figur 23 ovan.

Förutom relämodulen behöver man även själva reläet som ibland säljs separat. Relän för denna typ av kompakta relämoduler heter (REL-MR-24DC/21), som kan ses i figur 24 nedan. Relätypen tillåter en kontinuerlig belastningsström på 6 A på kontaktsidan.



Figur 24. Phoenix Contact – Relä

4.2.4 Elskåpets komponenter och konstruktion

I fråga om själva elskåpets konstruktion så fanns det vissa krav på elskåpet att ta i beaktande vid planeringen utgående från funktionsspecifikationen och vilken maskinvara som valts.

För att uppfylla kravet att elskåpet måste vara synligt på avstånd i fråga om driftlägesindikeringen, togs initiativet till att montera ihop elskåpet tillsammans med en vridbar koppling, bärprofil samt en bottenplatta. Se bilaga C sid 1 för översikt av hela konstruktionen.

För driftlägesindikeringen valdes ett ljusstorn för montering på utsidan av själva elskåpet, se figur 25 nedan. Den blåa LED-lampan i ljusstornet valdes för att indikera att driftövervakningssystemet är aktivt respektive inaktivt. Med det avses att övervaknings- och styrsystemet ska kunna utföra linjestopp på maskinlinjen om kriterierna för stoppläge aktiveras. De tre driftlägena som konfigurerades fick representeras i ljusstornet av de tre följande LED-lamporna nämligen grön-, orange- och röd lampa. Där den gröna lampan betyder normalt läge, orange – varningsläge och röd – stoppläge.

Utöver LED-lamporna i ljusstornet monterades även en siren högst upp på ljusstornet som kan programmeras för att ge olika ljudsignaler på 85 dB för att indikera varnings- respektive stoppläge. Med ljusstornet så kan maskinoperatörerna snabbt få en översikt över drifttillståndet i fråga om hastighetsförhållandet mellan form- och kalibredelen utan att behöva gå fram till HMI-panelen och kontrollera de olika värdena.



Figur 25. Ljusstornet för driftlägesindikeringen.

5 Programmering med Siemens TIA-Portal V13

Programmeringsverktyget som använts i projektet heter *Siemens Simatic Totally Integrated Automation Portal V13 Service Pack 1*. Det är en nyare version av tidigare lanserade TIA-Portal V13, vilket Siemens tillfört nödvändiga uppdateringar och en rad olika förbättringar men som grundar sig på TIA-Portal V13.

Siemens TIA-Portal innehåller PLC-programmeringsverktyget *Step 7 Basic/Professional*, HMI-programmeringsverktyget *WinCC Basic/Comfort/Advanced/Professional* samt HMI-simuleringsverktyget *Simatic WinCC Runtime Advanced*.

Ett annat viktigt program som medföljer är, *TIA-Portal Automation Software Updater*, i vilket man automatiskt kan ladda ner alla uppdateringar till programmeringsverktyget som Siemens publicerar.

För mera utmanande programmering finns även olika tillägsprogram som man kan integrera med TIA-Portal. Exempelvis *Siemens Simotion Scout* som används för exempelvis programmering av servomotorstyrningar.

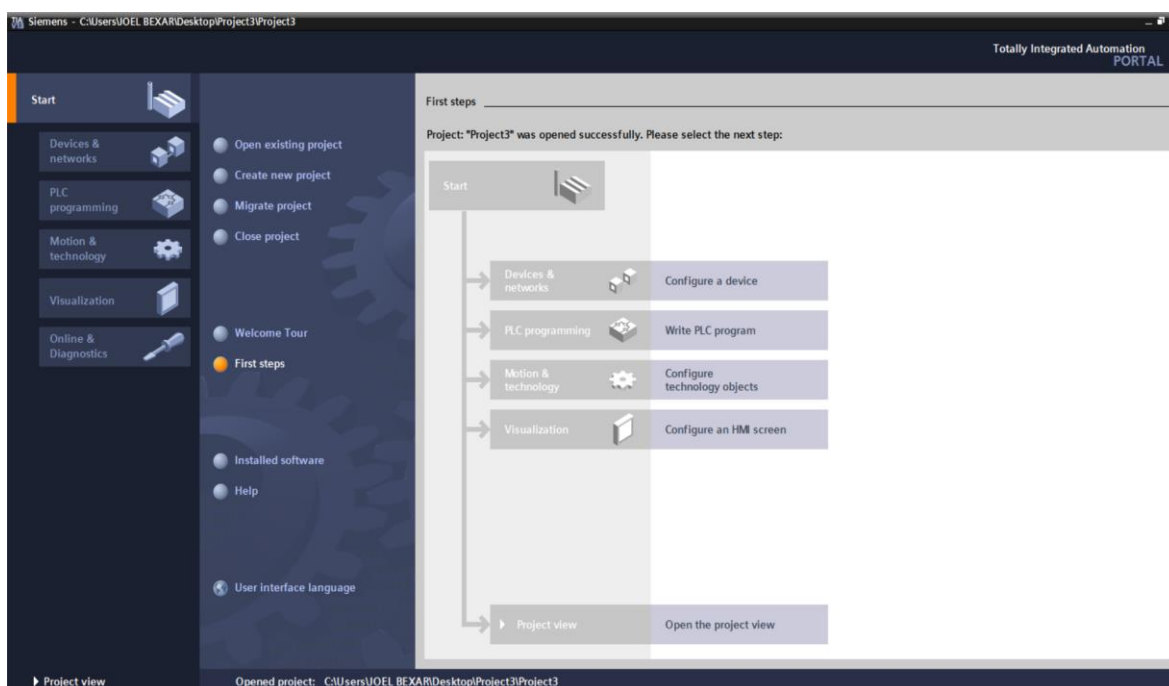
Ett annat tillägsprogram är *Simatic Step 7 Safety Advanced* som behövs vid programmering av s.k. *Fail-Safe Automation*, som hör till en skild kategori i Siemens produktsortiment. Det används vanligtvis för sådana systemkonfigurationer där farliga situationer i processen har en större benägenhet att uppstå, speciellt i sådana situationer där människan är en del i processen som samverkar med automation. I praktiken betyder det att man programmerar funktioner för exempelvis nödstoppskretsar som kräver en snabbare exekvering i PLC-programvaran än övriga funktioner och sekvenser.

5.1 Skapande av ett projekt

Programmering i Siemens TIA-Portal börjar med att man först skapar ett nytt projekt. Programmet är konfigurerat med en tydlig stegvis beskrivning på vad som man vill börja programmera med i projektet under fliken ”Start”, oberoende om man redan har lagt till någon maskinvara eller inte.

Programmet är strukturerat så att man kan välja mellan att ha översikt av projektet med portalvyn eller med projektvyn. Dessa två vyer kan man växla mellan genom att klicka på texten som alltid är längst nere till vänster, vilket kan ses i figur 26 nedan.

Portalvyn är den vy som man främst använder för att lägga till maskinvara i projektet i början och för att tydligare se strukturen av ett projekt. Projektvyn är den vy som man använder när man utför programmeringen och konfigureringen av maskinvaran. Programmet är konfigurerat så att när man t.ex. vill utföra någon ändring på en vald PLC i portalvyn så byts automatiskt portalvyn till projektvyn.



Figur 26. Siemens TIA-Portal V13 – Portalvy – First steps.

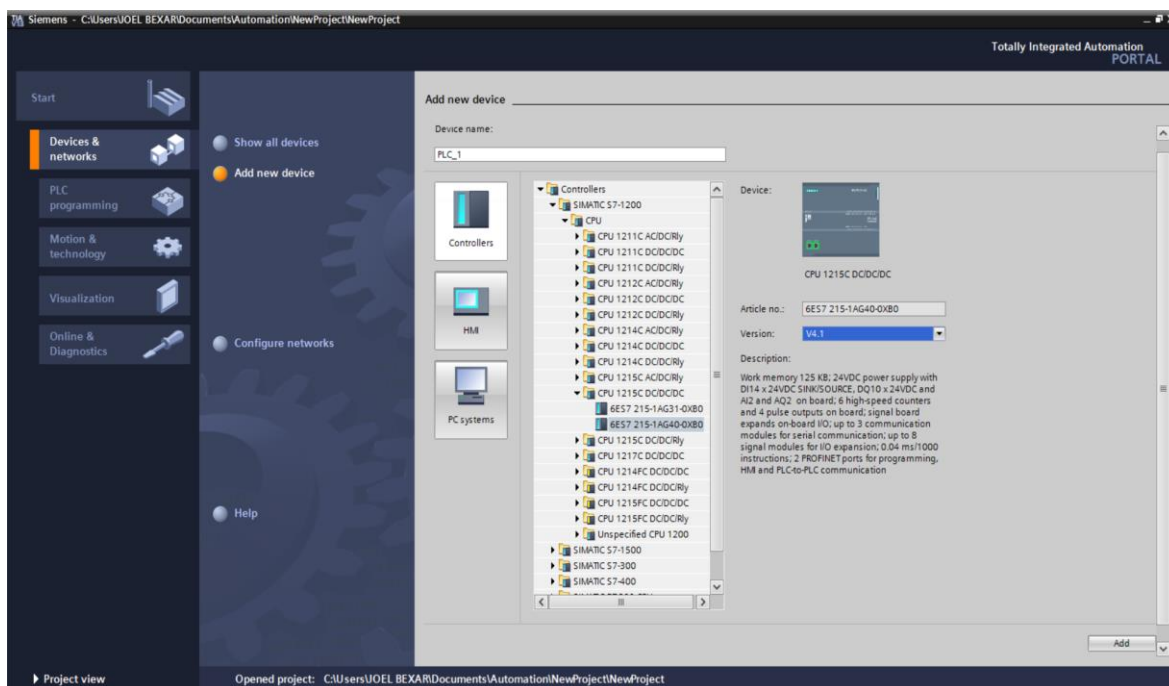
5.1.1 Lägga till maskinvara

När man sedan ska lägga till maskinvaran som man har i det fysiska eller avsedda systemet så görs det med hjälp av en databas som innehåller flera olika produkter, exempelvis maskinvara för styrsystem, användargränssnitt och industriella datorer. Produkterna som man kan välja mellan i databasen utgår ifrån vilka tilläggsprogram samt vilka uppdateringar som installerats i TIA-Portal.

För att säkerställa att man väljer rätt produkt måste man vara noga med att kontrollera vilket serienummer och eventuellt vilken version som den fysiska maskinvaran har, se exempel i figur 27 nedan. Eventuella komplikationer som kan uppstå är exempelvis då man ska testa kommunikationen eller överföra den konfigurerade programvaran till de fysiska enheterna, om den fysiska maskinvaran och den virtuella inte stämmer överens.

I händelse att man har valt fel produkt så kan man enkelt ångra och byta den virtuella maskinvaran i ett senare skede i exempelvis projektvyn genom att högerklicka på figuren för den enhet man vill byta där en dialogruta dyker upp, i vilken man kan välja fliken *Change device*.

En HMI-panel läggs till på samma tillvägagångssätt som för PLC:n. Det som skiljer sig åt är att man erbjuds av programmet i dialogrutan *HMI Device Wizard* att i förväg stegvist konfigurera färdiga funktioner för användargränssnittet.



Figur 27. Siemens TIA-Portal V13 – Portalvy – Add new device.

5.1.2 Kommunikationsinställningar

Innan man går vidare till programmeringen av styrenhetens program eller användargränssnittet är det bra att säkerställa att kommunikationen fungerar som den ska mellan enheterna. För att kunna kontrollera att kommunikationen fungerar förutsätts att man redan har installerat den fysiska maskinvaran.

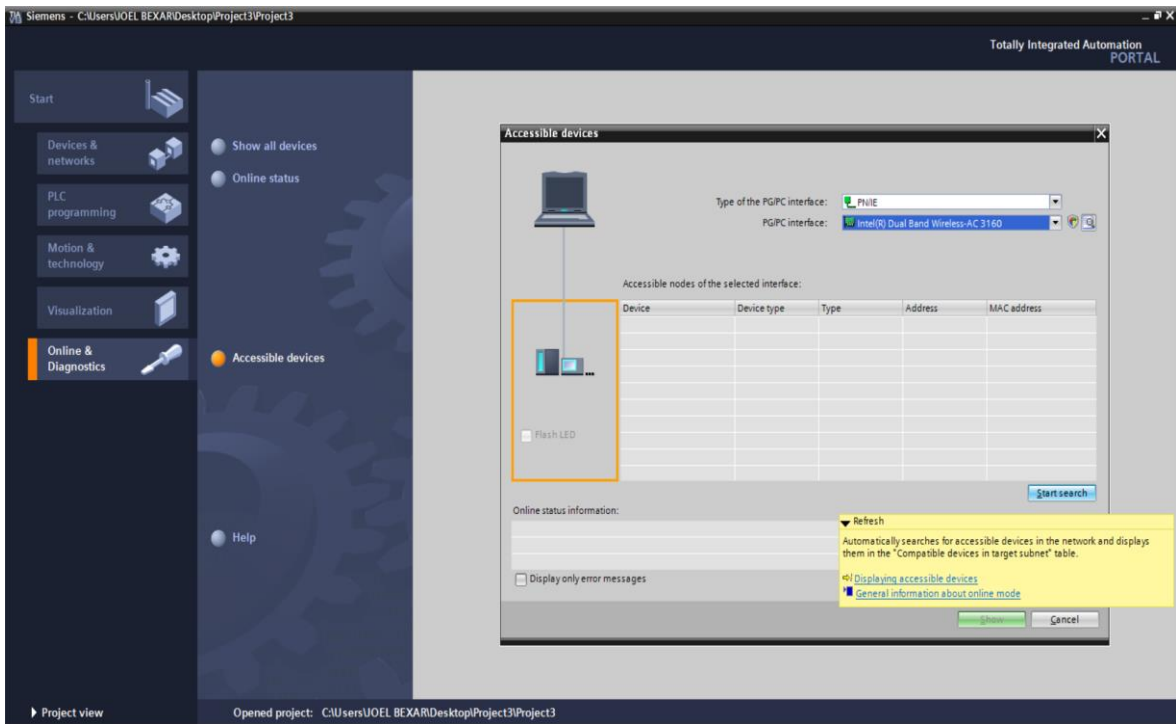
Detta kan utföras genom att klicka på fliken som heter *Online & Diagnostics* och välja punkten *Accessible devices*, i det vänstra fältet i portalvyn som man kan se i figur 28 på följande sida. I portalvyn dyker då en dialogruta upp där man ska välja kommunikationsgränssnitt samt vilket nätverkskort eller COM-port, (Communication Port), som den dator programmeringsverktyget är installerat på använder för att kommunicera med maskinvaran.

I projektvyn kan man även specificera IP-adressen för kommunikationsgränssnittet i de fall då man har ett större nätverk med flera enheter samt olika kommunikationsvägar och vill länka ihop olika enheter med varandra.

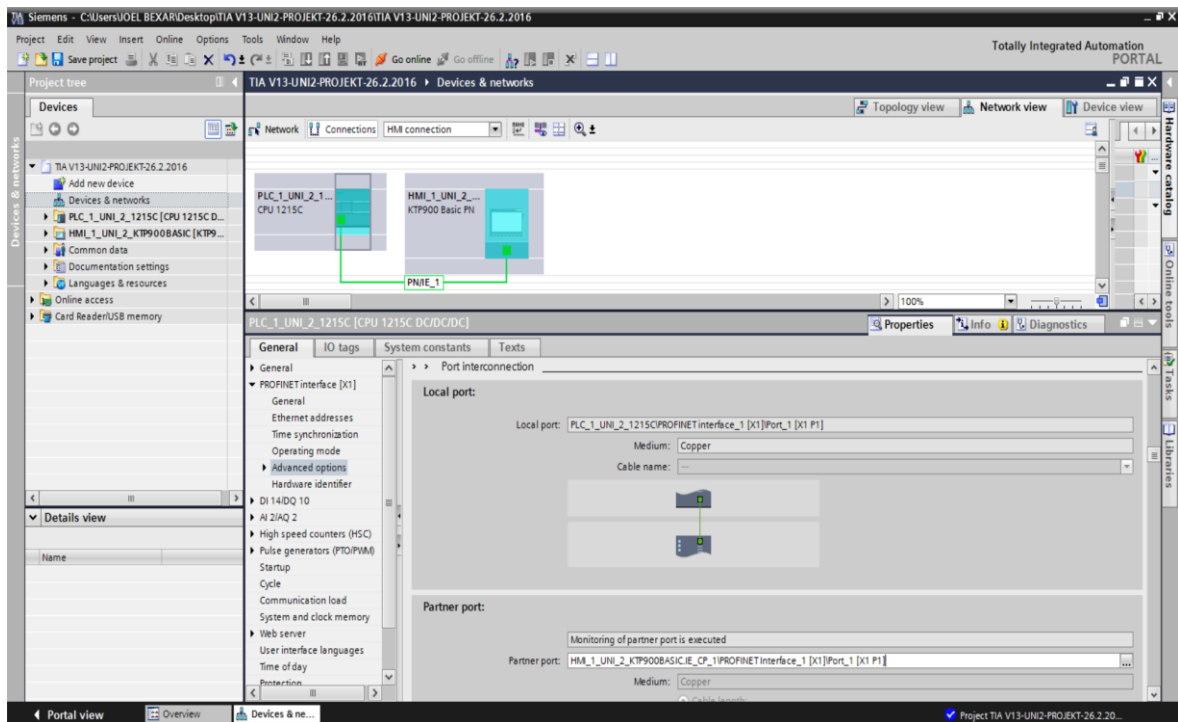
En annan sak som är viktigt att konfigurera är valet av kommunikationsport för länken mellan PLC:n och HMI:n. Detta ställde till med problem för kommunikationen till en början i och med att den PLC som används i övervaknings- och styrsystemet har två *Ethernet/Profinet* portanslutningar.

Anledningen till felet var att port X1 hade automatiskt konfigurerats som huvudport för kommunikationen mellan HMI:n och PLC:n, vilket hade att göra med att kommunikationsförbindelsen hade konfigurerats i den grafiska visningen i projektvyn genom dragning av ett streck mellan dem i konfigurationsmenyn *Devices & networks*. Den fysiska förbindelsen var i sin tur i omvänd ordning så att port X2 användes som kommunikation mellan enheterna vilket var anledningen till att det inte fungerade.

I denna situation kunde man enkelt åtgärda problemet genom att fysiskt byta plats på ethernet-kabelns anslutning vid PLC:n. Däremot om man skulle ha ett större system med långa avstånd mellan de fysiska enheterna, är det nog mera praktiskt att konfigurera kommunikationsgränssnittet direkt i konfigurationsmenyn likt exemplet i figur 29 på följande sida.



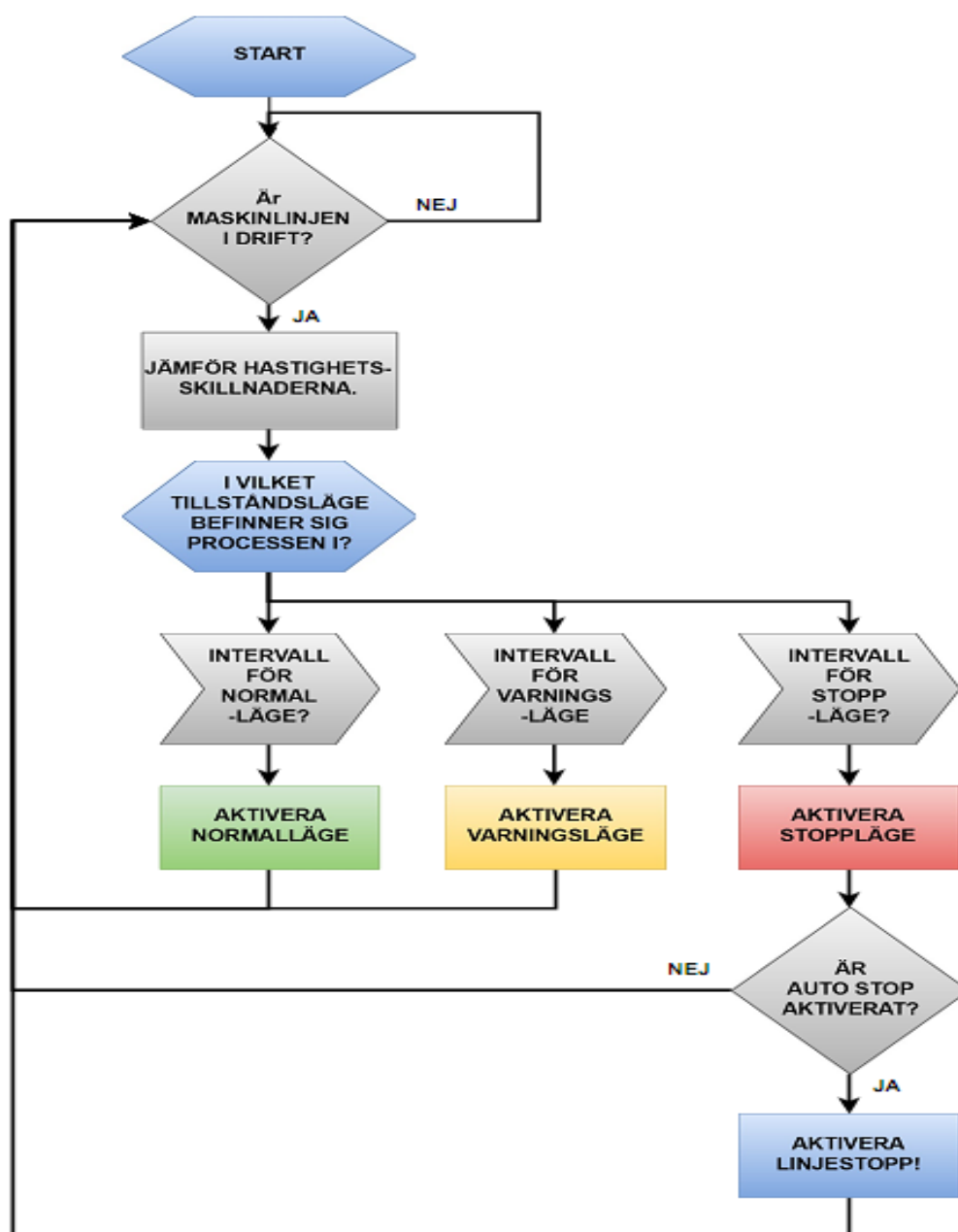
Figur 28. Siemens TIA-Portal V13 – Portalvy – Online & Diagnostics.



Figur 29. Siemens TIA-Portal V13 – Projektvy – Devices & networks portkonfiguration.

5.2 Programmering av styrprogrammet för projektet

För programmeringen av styrenheten gjordes först upp ett flödesschema som skapades utifrån krav och målsättningar i funktionsspecifikationen i punkt (3.4). Ett flödesschema är i stort sett en överskådlig skiss över hur ordningsföljden för programsekvenserna hos ett styrsystem är tänkt att fungera och dess struktur påminner lite om programmeringsspråket SFC. Figur 30 nedan visar det flödesschema som skapades som underlag för PLC-programmeringen till övervaknings- och styrsystemet.



Figur 30. Flödesschema för övervaknings- och styrsystemets styrprogram.

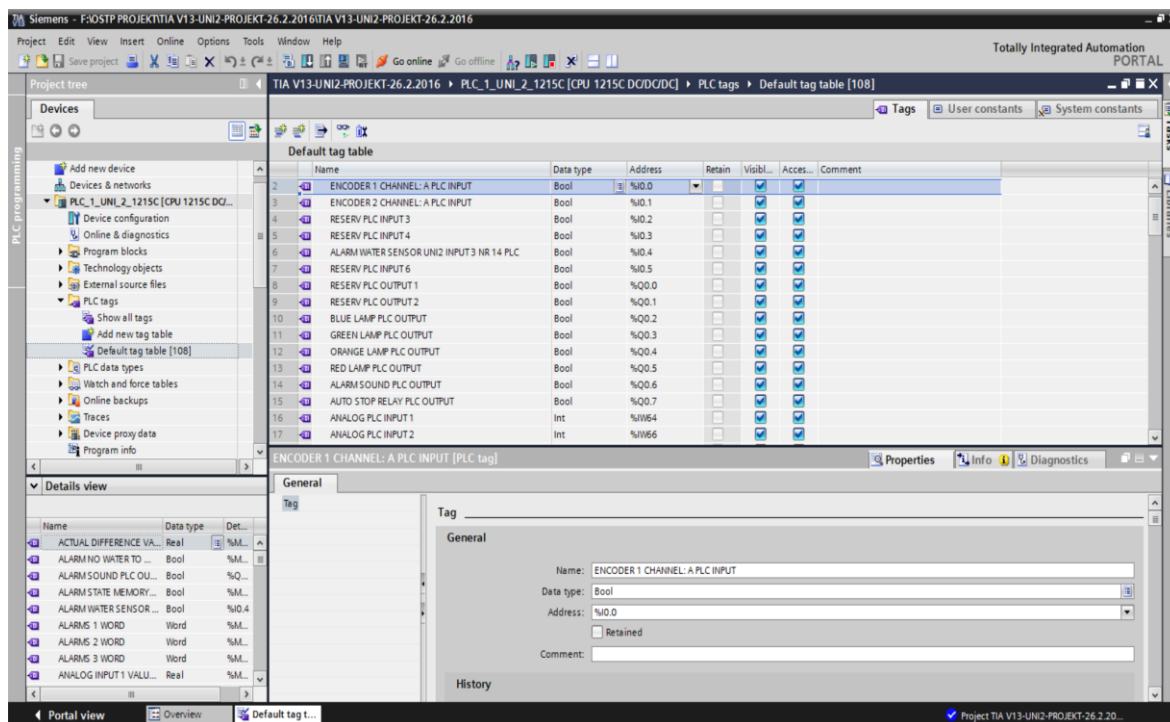
5.2.1 Definiering av PLC- och HMI-taggar

Utöver flödesschemat behöver man även skapa en utförlig lista på styrenhetens alla in- och utgångar en s.k. I/O-lista, se bilaga B sid 1. En sådan lista representeras i programmeringsverktyget i form av *Tags* som är variabler i form av in- och utgångar, minnespositioner m.m. som kan ses i exemplet i figur 31 nedan.

I och med att det bara finns en PLC och en HMI-panel i nätverket för projektet kan man i projektvyn konfigurera de olika taggarna under flikarna *PLC tags* och *HMI tags*. I *PLC tags* konfigurerar man vilka ingångar, utgångar och minnen som programmeras i PLC-programmet. I *HMI tags* konfigureras motsvarande med skillnaden att adresserna antingen är interna eller externa.

I listan med *PLC tags* definieras ett namn på taggen, datatyp och adress till tag:en. I kryssrutan *Retain* konfigurerar man om värdet på adressen ska hållas kvar i minnet vid exempelvis ett strömavbrott, vilket dock bara är möjligt att konfigurera för *Data block* och inte i en tagg-lista och av den orsaken är kolumnen gråmålad.

Kryssrutorna *Visible from HMI* och *Accessible from HMI*, vilka är till för att göra en PLC-tag synlig respektive åtkomlig från programmeringen av HMI-panelen. I Step 7 skrivs alla adresser med ett % före initialerna: %I (Input), %Q (Output) och %M (Memory).



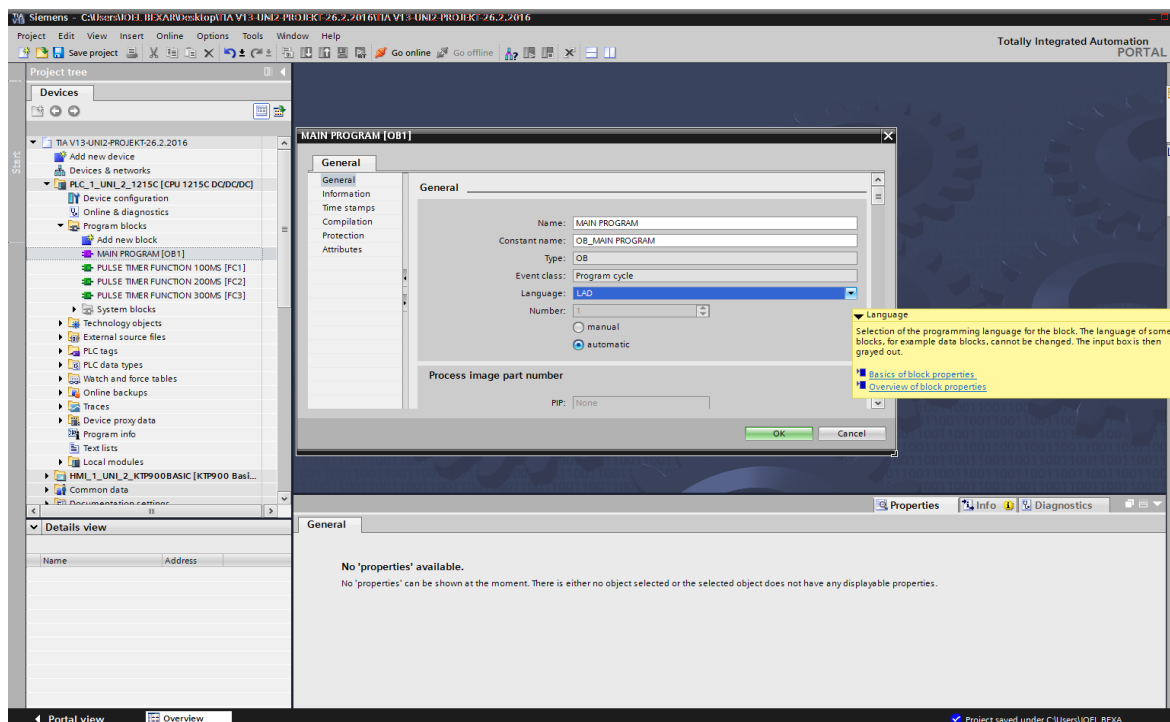
Figur 31. Siemens TIA-Portal V13 – PLC-Tags.

5.2.2 Val av programmeringsspråk

Uppdragsgivaren hade som krav att programmeringsspråket skulle vara Ladder Diagram i och med att företagets elautomationsmontörer som utför underhållet på maskinlinjerna har mest kunskap kring det programmeringsspråket.

För att kontrollera inställningarna för vilket programmeringsspråk som valts görs genom att högerklicka på det skapade organisationsblocket, OB1 (Organization Block), i projektvyn och välja fliken *Properties*. En dialogruta dyker då upp i vilken man uppe i vänstra hörnet kan välja menyn *General*, fliken *Language* och i den välja programmeringsspråket för programblocket, se exempel nedan i figur 32.

Det finns tre programmeringsspråk att välja mellan för en Siemens S7-1200 series PLC. Dessa tre programmeringsspråk är: LAD (Ladder Diagram), FBD (Function Block Diagram) och SCL (Structured Control Language). Dessa är Siemens egna varianter av programmeringsspråken LD, FBD och ST i IEC 61131-3-standarden. Se punkt (2.5).



Figur 32. Siemens TIA-Portal V13 – Val av programmeringsspråk för OB1.

5.2.3 Konfigurering av HSC och funktionsblock

Styrprogrammets kanske viktigaste uppgift för övervaknings- och styrsystemet var att det skulle kunna läsa pulsfrekvenser från respektive pulsgivare. För att kunna räkna en pulsfrekvens ska man använda s.k. HSC (High Speed Counters) i programmet Step 7.

En HSC kan räkna antalet pulser snabbare än vad själva huvudprogrammet klarar av under en programcykel och även uppdatera invärdet under en pågående programexekvering så att inga gamla värden processeras.

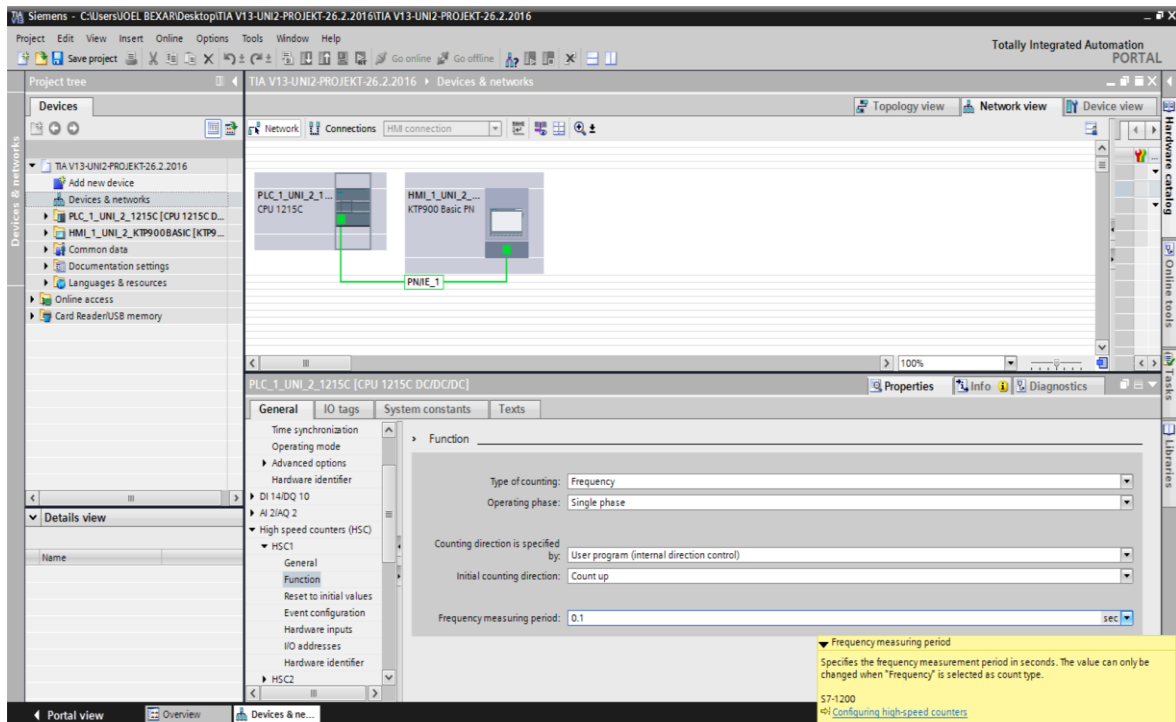
För att beskriva hur en HSC konfigurerats för övervaknings- och styrsystemet kan vi se på exemplet i figur 33 på följande sida där det framkommer hur man konfigurerar en HSC. För att få en digital insignal på PLC:n att fungera som en HSC måste man kryssa i användningen av en sådan i rutan *Enable this high speed counter* under fliken *General* i menyn där inställningarna för en HSC finns.

Det som specificeras i menyn i exemplet i Figur 33 är b.l.a. om pulssignalen som HSC:n räknar ska omvandlas till en pulsfrekvens eller att den helt enkelt ska räkna antalet pulser. Man kan också konfigurera vilken typ av pulståg eller pulstyp som insignalen består av och hur t.ex. räknaren registrerar om pulsgivaren roterar medsols respektive motsols, och som har att göra med hur insignalen ser ut.

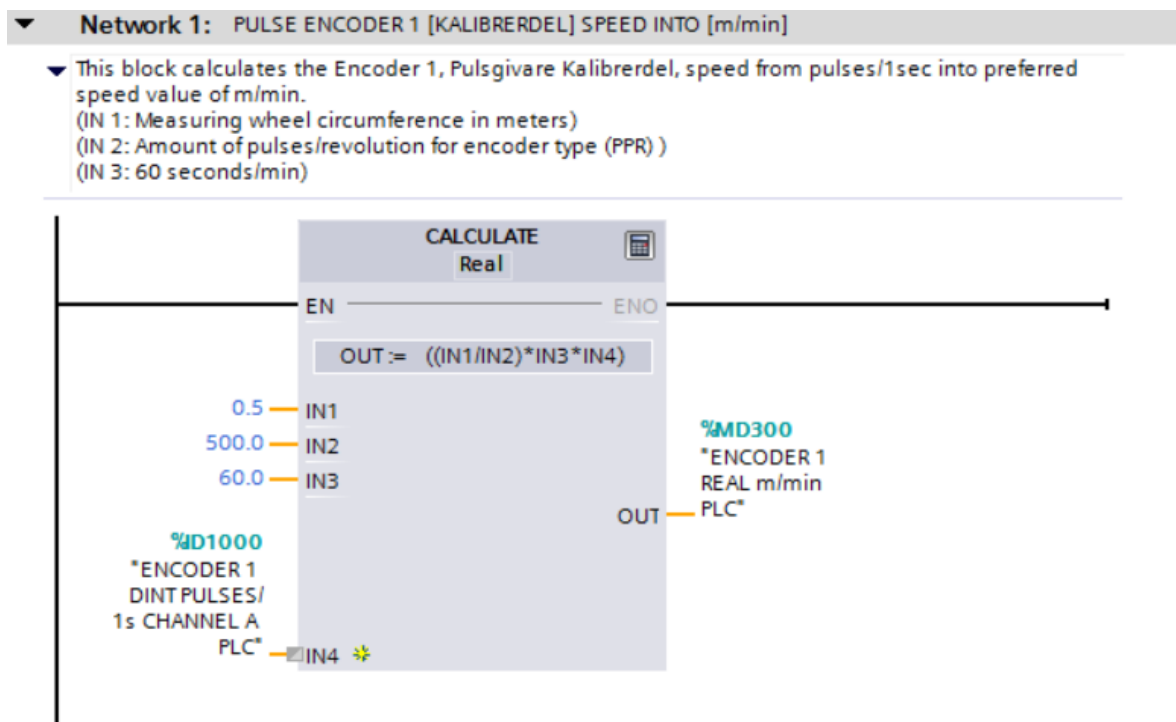
Om man då valt att signalen från pulsgivaren ger ut en enkanalig pulsfrekvens och man vill att HSC:n ska läsa denna pulsfrekvens väljer man räknetypen till frekvens, att räknaren räknar uppåt och slutligen väljer man för vilken tid som antalet pulser räknas för att en pulsfrekvens ska fås.

När man konfigurerat maskinvarans inställningar för HSC:n kan man börja med att skapa olika funktioner och beräkningar för att omvandla pulsfrekvensen från pulsgivaren till en mätbar storhet. I Step 7 och närmare bestämt programmeringsspråket LAD kan man använda ett fördefinierat *CALCULATE* funktionsblock för att utföra beräkningar.

Exemplet i figur 34 på följande sida är configurationen för att omvandla pulsfrekvensen till en hastighet i m/min. För att kunna utföra beräkningar med olika invärden måste de också vara av samma datatyp, vilket i exemplet är ett 32 bitars invärde, ID1000 (Double Integer), från HSC:n och ett 32 bitars minne MD (Double Word/REAL).



Figur 33. Siemens TIA-Portal V13 – Aktivering och konfiguration av en HSC.

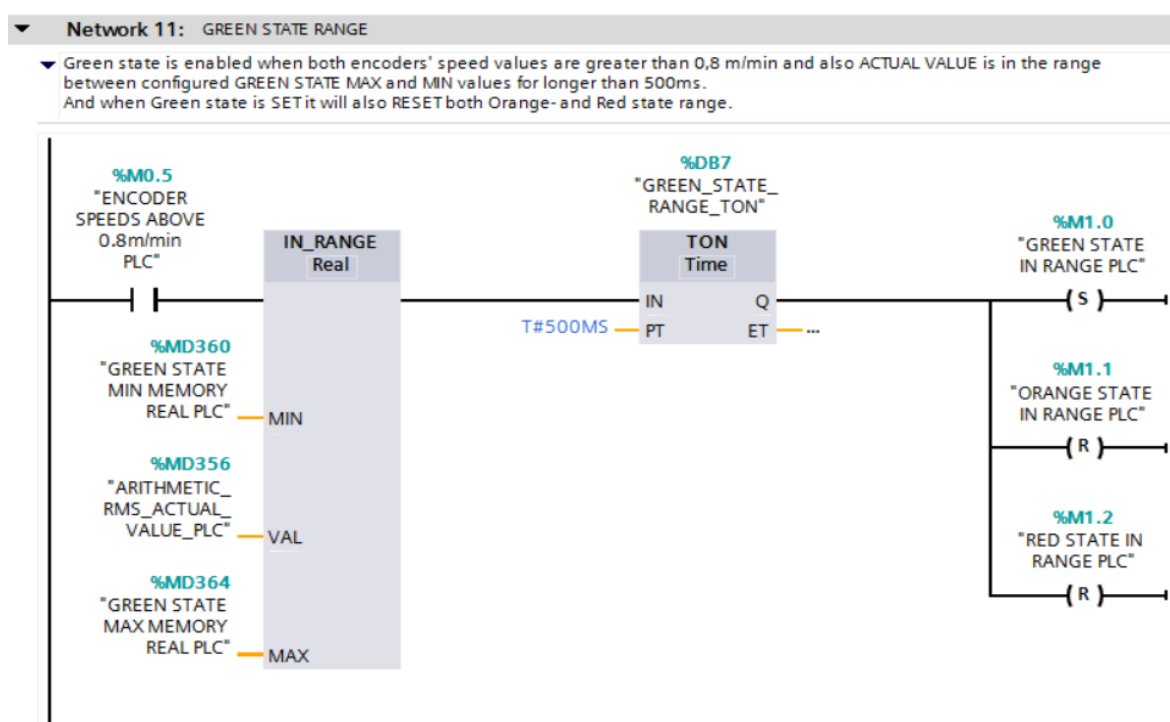


Figur 34. Siemens TIA-Portal V13 – Kalkylator funktionsblock.

5.2.4 Konfigurering av funktionsblock för driftlägesindikering

För att jämföra de två pulsgivarnas beräknade hastighetsvärden med varandra så användes fördefinierade funktionsblock som heter *IN_RANGE*. Sådana funktionsblock kan användas för att jämföra om en variabls värde befinner sig inom ett visst område, se exempel i figur 35 nedan. I exemplet jämförs det beräknade medelvärdet för hastighetsskillnaden mellan kalibrer- och formdelen ifall värdet är inom det specificerade intervallet för normalläge.

Med hjälp av olika typer av fördefinierade funktionsblock slipper man skapa stora och komplicerade logiska funktioner, vilket underlättar programmeringen avsevärt samtidigt som översikten av de olika programnätverken bibehålls.

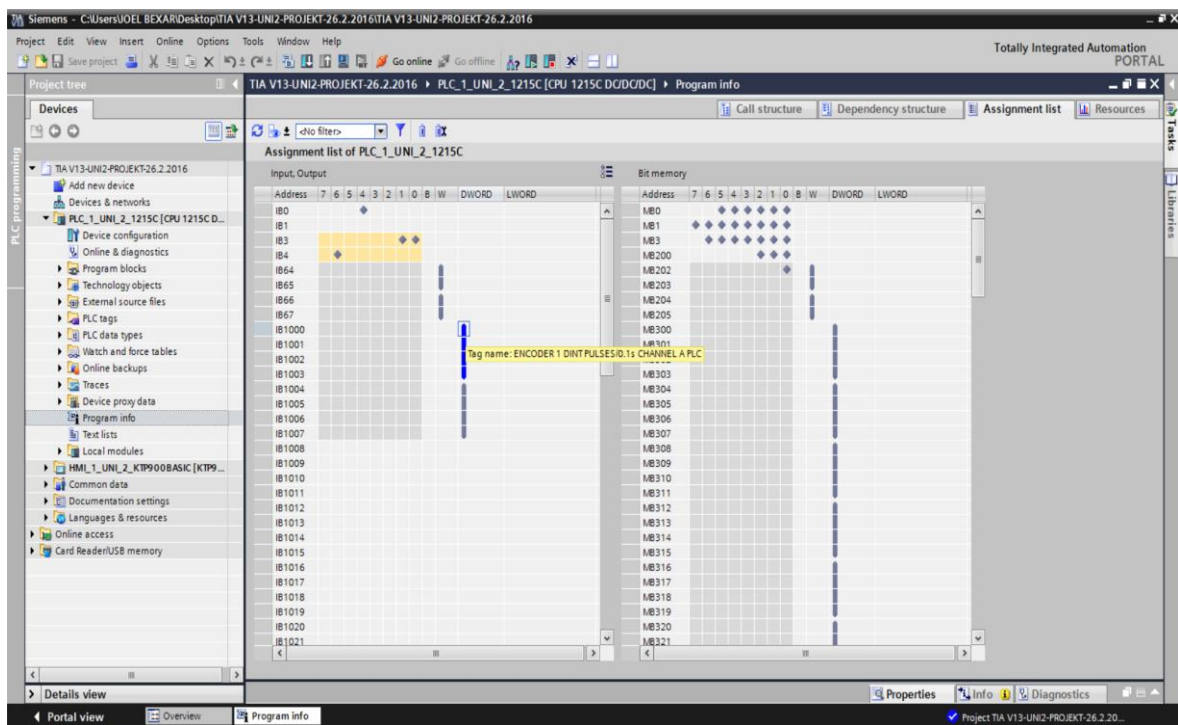


Figur 35. Siemens TIA-Portal V13 – Exempel på *IN_RANGE* funktionsblock.

5.2.5 Assignment list

En annan användbar egenskap hos TIA-Portal V13 Step 7 är att man enkelt kan kontrollera vilka adresser man använt i systemet och till vilken PLC-tag som adressen tilldelats. Man kan även se hur stor minneskapacitet som används av de olika programmen och funktionerna m.m.

För att ta sig till denna visningssida som heter *Assignment List* går man via *Device proxy data* och *Program info* som finns i det vänstra hörnet under fliken med egenskaper för PLC:n, se exempel i figur 36 nedan.



Figur 36. Siemens TIA-Portal V13 – Assignment list.

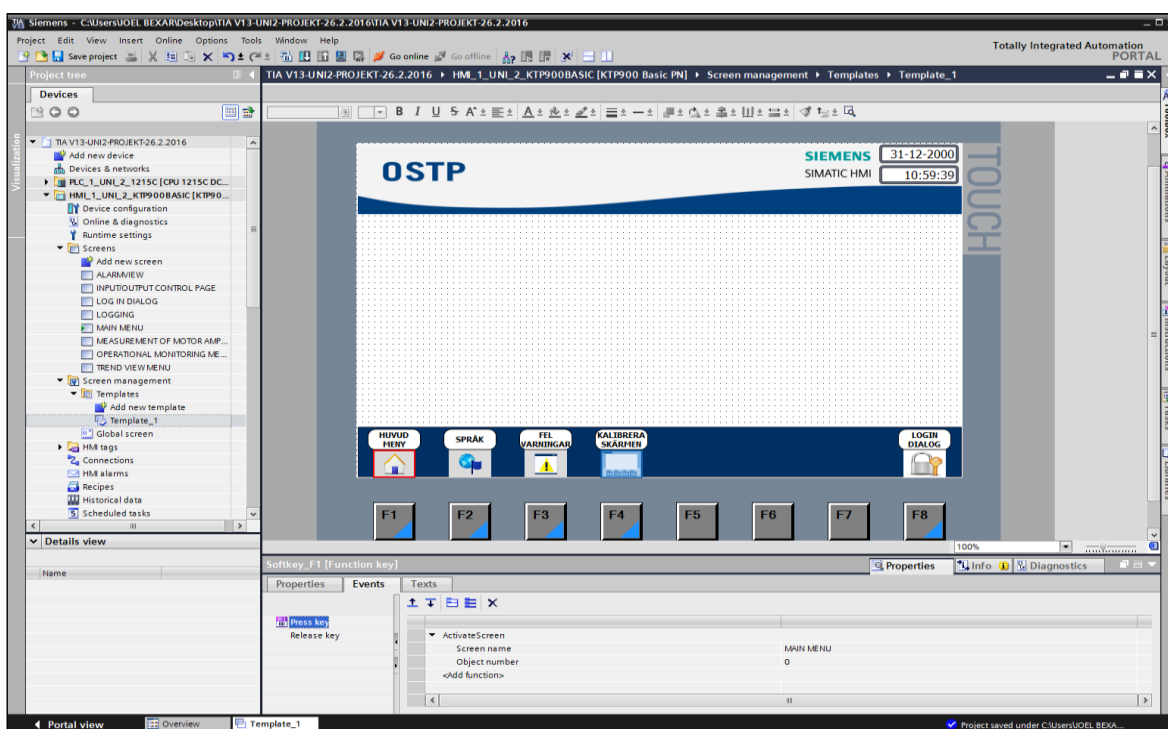
5.3 Programmering och konfigurering av HMI-gränssnittet

Programmering av användargränssnitt i TIA-Portal V13 WinCC för HMI-paneler kan endast utföras i projektvyn där det finns flera flikar och undermenyer som används för olika ändamål när man skapar användargränssnittet.

5.3.1 Konfigurering av Template

Vid programmeringen av användargränssnittet för övervaknings- och styrsystemet valdes att först skapa en botten eller s.k. *Template*. Det är en *Layer* som kommer ovanpå andra sidor och lager i användargränssnittet. Anledningen till det var att försöka skapa en så enhetlig användarmiljö som möjligt. HMI-panelens fysiska knappar F1–F8, valdes att representeras med deras konfigurerade funktioner som inte ska bytas när man växlar mellan de olika sidorna för att göra navigeringen enklare. För att skapa en sådan egenskap är det enklaste tillvägagångssättet att konfigurera det i template:n.

När man skapar en funktion för ett objekt görs det i fliken *Events* som kan ses i figur 37 nedan. Det finns flera olika funktioner som man kan tilldela för knapptryckningar vilket väljs i en *Drop down*-lista som i exemplet för tilldelning av funktionen hos knappen F1. För knapp F1 tilldelades funktionen att byta till sidan med huvudmenyn när man fysiskt trycker på HMI-panelens knapp F1.



Figur 37. Siemens TIA-Portal V13 – Konfigurering av Template.

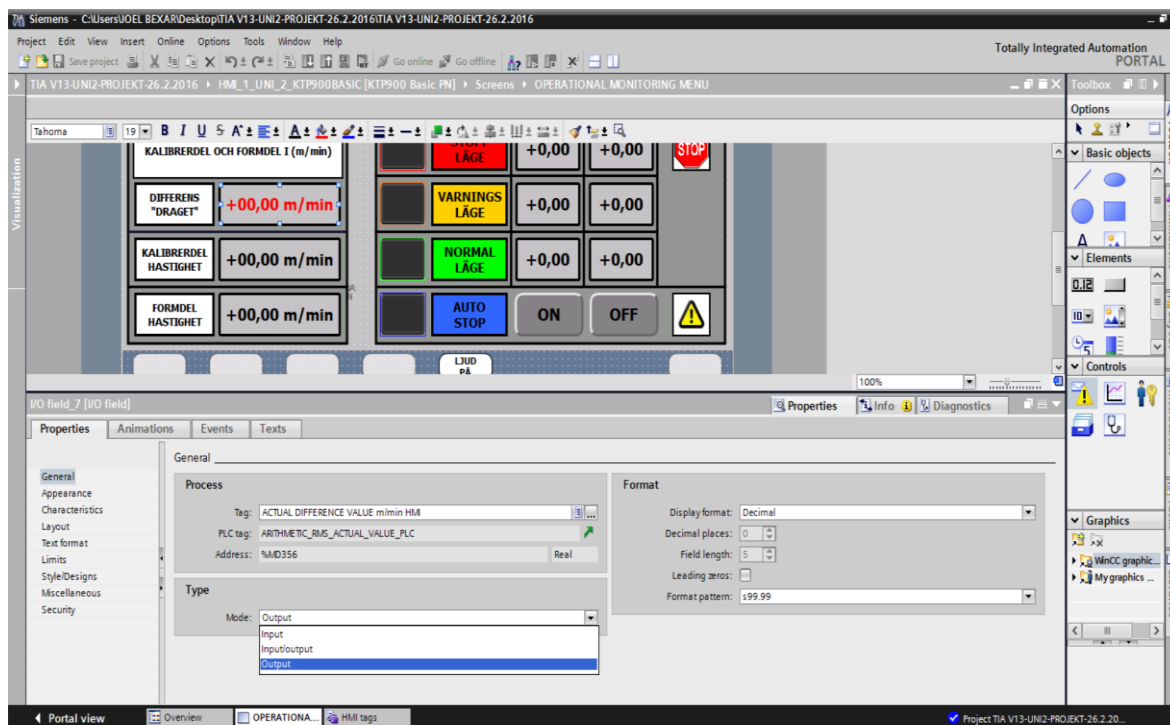
5.3.2 Konfigurering av I/O-fält

För att konfigurera inmatning eller visning av olika värden som systemet samlar in eller skickar till processen använder man I/O-fält som hittas under menyn *Toolbox* och *Elements* vilket kan ses till höger i figur 38 nedan.

I ett I/O-fält måste man konfigurera under fliken inställningar följande saker: vilken HMI-tag som ska länkas ihop med respektive PLC-tag från vilken själva värdet fås från, om I/O-fältet ska vara ett inmatningsfält eller ett visningsfält eller också en kombination av båda och slutligen konfigurering av formatet för hur det värde som PLC-taggen förfogar ska visas i rutan.

Andra inställningar som kan göras för I/O-fältet är exempelvis olika användarrättigheter för tillåtelse att mata in värden. Resterande inställningar är främst konfigurationer för fältets grafiska design och dynamiken i visningen. Alla de inställningarna som man inte kan ändra på gråmålas även här av programmet.

I TIA-Portal har man lyckats bra med informationen kring olika objekt och funktioner som finns i menyn *Help*. Den informationen kommer man åt genom att välja det objekt som man vill ha information om och sedan trycka på tangentbordets knapp F1.



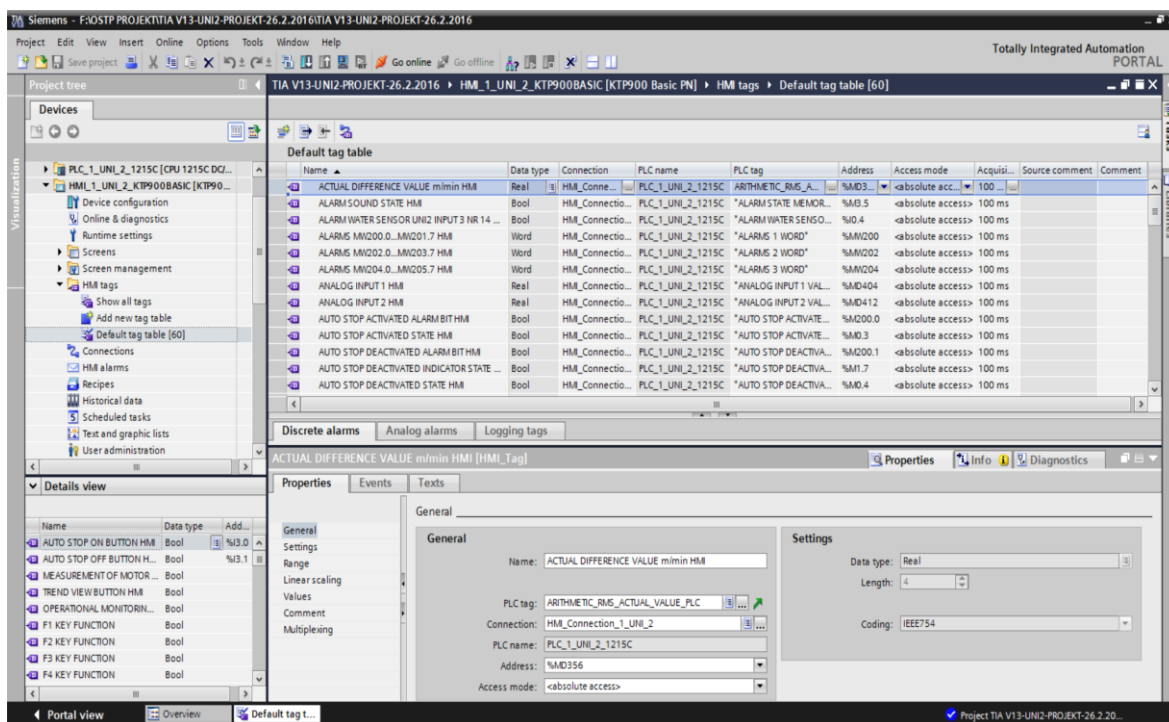
Figur 38. Siemens TIA-Portal V13 – Konfigurering av I/O-fält.

5.3.3 Ihopkoppling av HMI- och PLC-taggar

För att länka ihop en HMI-tag med någon PLC-tag ska man gå via fliken *HMI tags* och välja *Default tag table*. I den konfigurationsmenyn namnger man en HMI-tag som sedan länkas ihop med rätt kommunikationsförbindelse dvs. korrekt PLC och val av PLC-tag.

Om man tidigare vid skapandet av PLC-taggar hade kryssat i rutorna att de skulle vara synliga och åtkomliga från HMI-panelen så kommer man kunna se dem alla när man vill länka ihop dem med en HMI-tag. Interna HMI-taggar för exempelvis knappfunktioner ska man givetvis inte länka ihop med en PLC-tag.

I fliken *Access mode* så måste man välja alternativet *Absolute access* för de HMI-taggar som länkas ihop med en PLC-tag för att kommunikationen med att skicka och ta emot data mellan PLC:n och HMI:n annars inte skulle fungera, se exempel i figur 39 nedan. För att få den snabbaste responsen gällande kommunikationen mellan enheterna ska man också välja *Acquisition cycle* till det snabbaste alternativet på 100 ms.

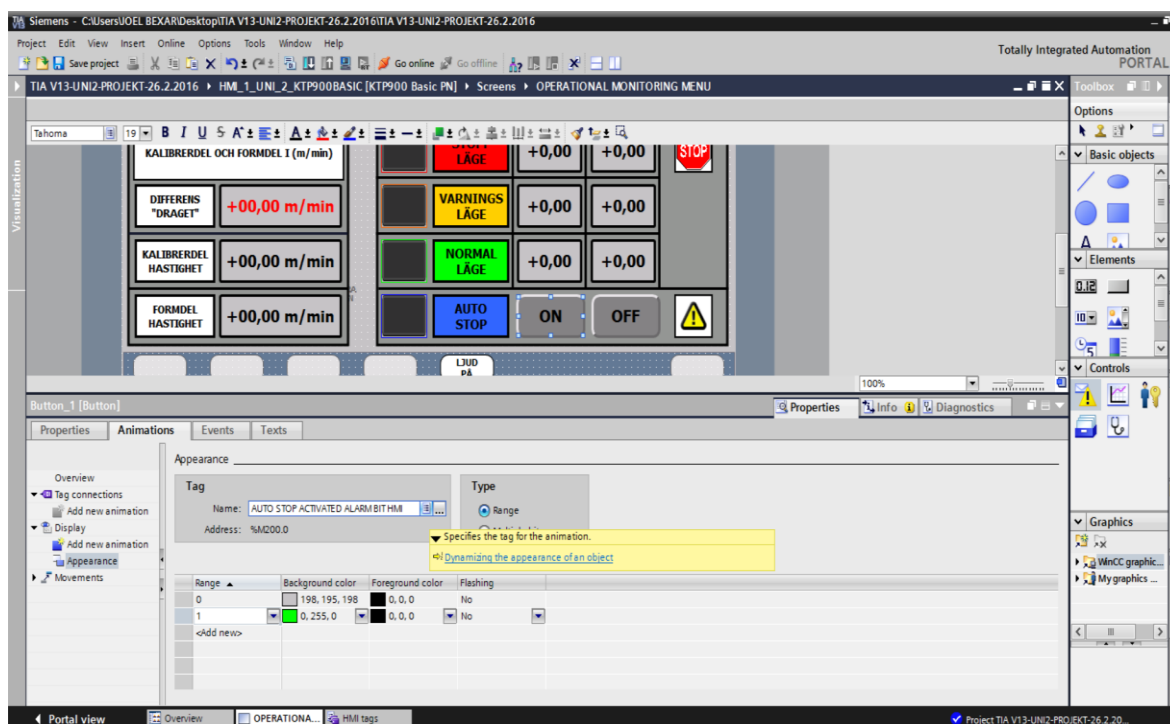


Figur 39. Siemens TIA-Portal V13 – Ihoplänkning av HMI-taggar med PLC-taggar.

5.3.4 Konfigurering av dynamiska objekt

När man vill att en knappfunktion ska indikera på skärmen att funktionen blivit utförd kan man använda sig av verktyget *Appearance* i menyn till vänster i inställningarna för ett objekt. I exemplet i figur 40 nedan kan man se att knappen ON är tänkt att skifta färg från grå färg till grön när styrprogrammet har exekverat programmet för att utföra den valda funktionen.

I exemplet ändras värdet från 0–1 på en viss PLC-tag som är ihoplänkad med en HMI-tag vid namn *AUTO STOP ACTIVATED ALARM BIT HMI*. Detta betyder att den gröna färgen för knappen aktiveras inte förrän HMI-taggen ändrar värde och som i sin tur är en indikering på att åtgärden utförts av styrprogrammet och kommunikationen fungerar.

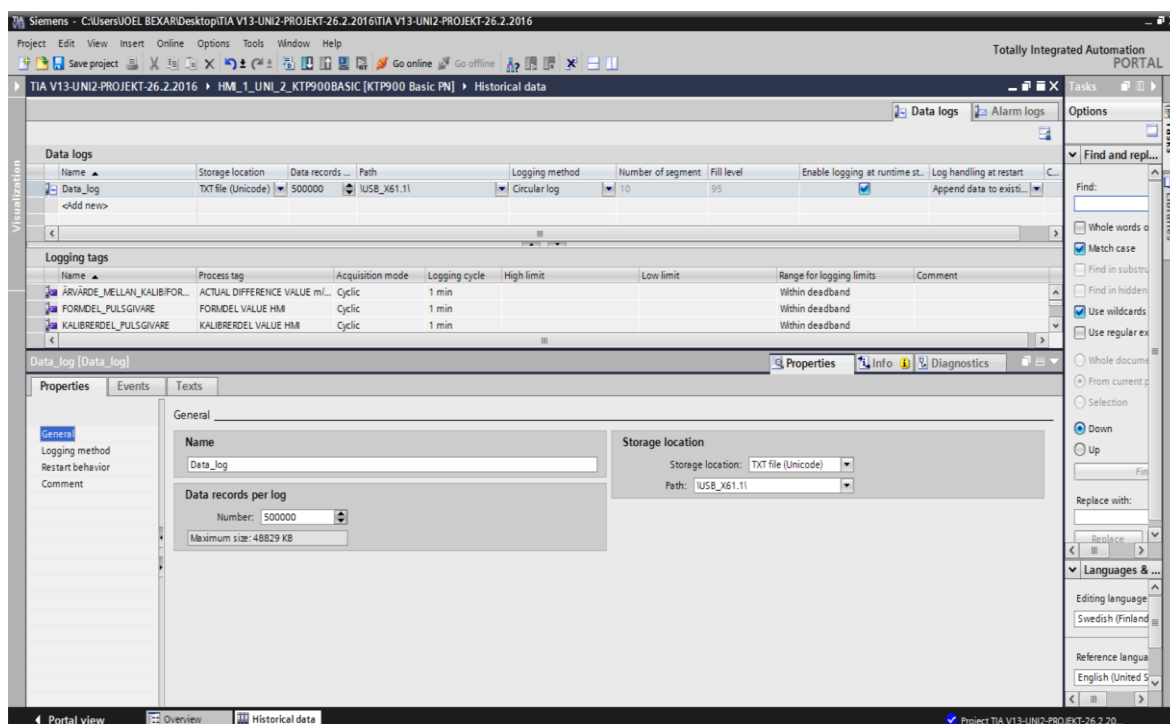


Figur 40. Siemens TIA-Portal V13 – Konfigurering av knappfunktioner.

5.3.5 Loggning av data med HMI-panelen

Den sista konfigurationen som utfördes för övervaknings- och styrsystemet var att logga data. Det visade sig att den enklaste metoden för att konfigurera loggning av vissa specifika värden hos systemet var att använda den färdiga funktionen *Historical data* hos HMI-panelen, se exempel i figur 41 nedan.

De värden som loggas med en KTP900 Basic HMI panel ska konfigureras för att sparas till ett USB-minne som inkopplas på baksidan av panelen till HMI-panelens USB-port. Loggdatan kan endast sparas i TXT-format för en Siemens KTP900 Basic HMI-panel.



Figur 41. Siemens TIA-Portal V13 – Loggning av data med HMI panel.

6 Resultat och utvärdering

Slutresultatet av projektet blev ett fungerande övervaknings- och styrsystem som blivit installerat, testats och tagits i bruk vid maskinlinjen. Systemet uppfyller kraven i funktionsspecifikationen och utför helautomatiskt huvuduppgiften, som var att förhindra att ett specifikt processfel uppstår hos maskinlinjen.

Att övervaknings- och styrsystemet utför huvuduppgiften grundar sig på utförda funktionstester av systemet och processanalyser. Med dessa metoder har man kunnat kartlägga hur processfelet uppstår och hur systemet uppträder i olika situationer.

Det bör påpekas att funktionsspecifikationen varit dynamisk under hela projektets gång vartefter projektet framskridits och sammanställningen av funktionsspecifikationen i kapitel (3.4), är ett resultat av idesamlingar från alla inblandade som projektet påverkat.

Däremot har man i efterhand konstaterat utifrån datainsamlingen att övervaknings- och styrsystemet har ingripit för att stoppa maskinlinjen när en annan typ av processfel inträffat. Dock har övervaknings- och styrsystemet vid de tillfällena fungerat som en tilläggsresurs för att stoppa maskinlinjen eftersom det reagerat kort efter maskinlinjens egna styrsystems stoppkrets.

Övervaknings- och styrsystemet installerades som en separat enhet vid maskinlinjen. Men alla signalkablarna och strömmatningen till övervaknings- och styrsystemet blev givetvis dragna mellan maskinlinjens styrskåp och manöverskåp som således gör systemet till en del av maskinlinjen.

De huvudsakliga ingreppen som utfördes på maskinlinjen är följande: inkoppling av 230 V AC-matning från styrskåpet till systemet, kopplat vidare båda pulsgivarnas signaler gällande pulsfrekvensen, båda frekvensomriktarnas analoga signaler och slutligen inkoppling av ett nytt relä med styrning av en seriekopplad brytspets för maskinlinjens linjestopp.

Ibruktagningstestningen för övervaknings- och styrsystemet utfördes efter att installationen var klar samt att ca två månaders insamling av processdata var utfört. Processdatan som insamlades utvärderades och man kunde följaktligen skapa ett bör- och ärvärdesregister för inställningsvärden till de tre driftlägena för styrprogrammet.

Maskinoperatörerna har också bekantat sig med systemet vartefter och kunnat på egen hand utforska dess fördelar och användbara egenskaper vid manövrering av maskinlinjen i det dagliga arbetet. Systemet har på så vis inte haft någon stor inverkan på hur maskinoperatörerna skött processen från tidigare i och med att systemet är helautomatiserat och behöver därmed inte övervakas.

Strömmätningen av kalibrer- och formdelens elmotorer som via växellådor och rullbockar driver fram röret under processens gång, blev ett resultat som inte riktigt motsvarar maskinoperatörernas förväntningar. En av anledningarna till det var att visningen av den ström som respektive elmotor tar under processens gång inte kunde simuleras som analoga mätare med visare i användargränssnittets sida. Det beror främst på att HMI-panelen har vissa begränsningar och man kunde således inte skapa en sådan funktion i HMI-programmeringsverktyget *WinCC* för Basic HMI-paneler. En annan orsak var att man antog att det skulle kunna gå att jämföra strömvärdena mellan varandra för att tydligare kunna se skillnader vid hastighetsregleringen av drivdelsmotorerna. Det visade sig följaktligen att strömåtgången hos elmotorerna inte är tillräckligt linjärt för att kunna göra det. Resultatet blev därmed en kompromiss som kan ses i bilaga C sid 5.

7 Diskussion

Att studera tillverkningsprocessen vid maskinlinjen var det första som gjordes när detta examensarbete påbörjades. Detta realiserades genom att föra diskussioner med maskinlinjens maskinoperatörer samt att på egen hand analysera tillverkningsprocessen på plats vid maskinlinjen.

Enligt min egen åsikt så blev både styrprogrammet och användargränssnittets struktur tillräckligt användbart för ändamålet, med tanke på att de krav som ställdes av uppdragsgivaren hade en hel del brister i det stora hela, att installationen av systemet hade sina utmaningar när ingen dokumentation för maskinlinjen fanns att tillgå samt att maskinlinjens kabelvägar och elskåp hade stora brister.

I och med att projekttiden blev ganska lång och jag kunde arbeta med projektet högst 1–2 dagar i veckan var det ganska svårt med just PLC- och HMI-programmeringen att varje vecka försöka hitta tillbaka och fortsätta där man slutade veckan innan.

Om projektet skulle genomföras på nytt skulle jag satsa mera tid på att följa upp det befintliga styrsystemet på maskinlinjen för att kunna utvärdera hur tillverkningsprocessens alla olika sekvenser styrs. Man skulle då kunna undersöka bättre om man eventuellt haft möjligheten att integrera övervaknings- och styrsystemet med hjälp av tilläggsmoduler till maskinvaran hos det befintliga styrsystemet istället. Att lägga mera tid på att göra en utförligare funktionsspecifikation skulle också ha en hög prioritet. Hade jag haft bättre förkunskaper om den programmering som gjorts hade projektet sannolikt även gått snabbare och mera effektivt att utföra.

Något jag dock saknat i detta examensarbete är klara målsättningar för projektet från början samt att installationen av övervaknings- och styrsystemet vid maskinlinjen hade kunnat utföras med bättre förutsättningar.

I övrigt tycker jag att både den praktiska och den teoretiska delen i examensarbetet varit synnerligen lärorikt. Det som jag anser varit mest lärorikt är att ha fått vara ansvarig för att sköta projektet från start till slut som har gett mig mera kunskaper i projekthantering.

Källförteckning

Bolton, W., 2006. *Programmable Logic Controllers*. 4th red. Oxford: Elsevier Newnes.

Cegrell, T. & Sandberg, U., 1994. *Industriella Styrssystem*. Borås: Responstryck.

Control Engineering, 2012. *Information systems: The evolution of the HMI*. [Online]
<http://www.controleng.com/single-article/information-systems-the-evolution-of-the-hmi/d643c1d8644f73884df2c0827cb31f38.html>
 [Hämtat: 19.5.2016].

Danaher Industrial Controls, 2003. *Encoder Application Handbook*. [Online]
http://www.dynapar.com/uploadedFiles/Site_Root/Service_and_Support/Danaher_Encoder_Handbook.pdf
 [Hämtat: 17.5.2016].

Encoder Products Company, 2012. *WP-2011: The Basics Of How An Encoder Works*. [Online]
http://encoder.com/core/files/encoder/uploads/files/WP-2011_Basics%20of%20How%20an%20Encoder%20Works.pdf
 [Hämtat: 20.5.2016].

Hackworth, J. R. & Hackworth Jr., F. D., 2004. *Programmable Logic Controllers: Programming Methods and Applications*. New Jersey: Pearson Ltd..

IEC - International Electrotechnical Commission, u.å. *About the IEC: Vision & mission*. [Online]
<http://www.iec.ch/about/>
 [Hämtat: 11.5.2016].

IEC - International Electrotechnical Commission, u.å. *IEC Webstore: Standards of IEC 61131*. [Online]
<https://webstore.iec.ch/searchform&q=IEC%2061131#>
 [Hämtat: 11.5.2016].

Lewis, R. W., 1998. *Programming industrial control systems using IEC 1131-3 Revised edition*. Exeter: Short Run Press Ltd..

PROFIBUS, 2008. *PROFIBUS PA: Teknologi och applikation, PROFIBUS Nutzerorganisation e. V. (PNO)*. [Online]
http://www.profibus.se/download/PDF/PA-System-Description_SE_Aug07.pdf
 [Hämtat: 19.5.2016].

PROFINET, 2014. *PROFINET: Systembeskrivning – Teknologi och applikation, PROFIBUS Nutzerorganisation e. V. (PNO)*. [Online]
http://www.profibus.se/download/PDF/PROFINET_Systembeskrivning_2014.pdf
 [Hämtat: 19.5.2016].

Shneidermann, B. & Plaisant, C., 2009. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 5th red. Boston: Pearson.

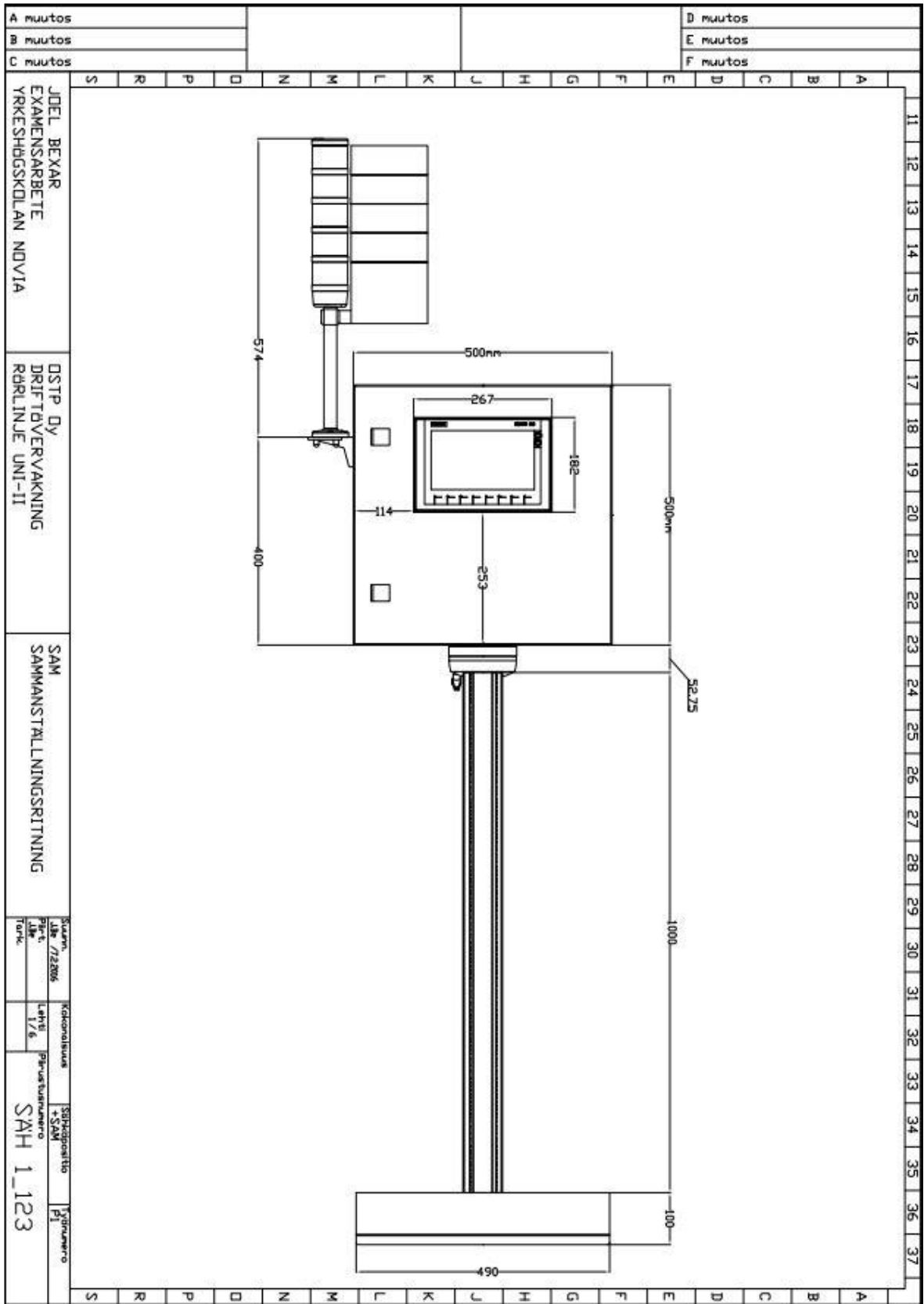
Bilagor

A RITNINGAR

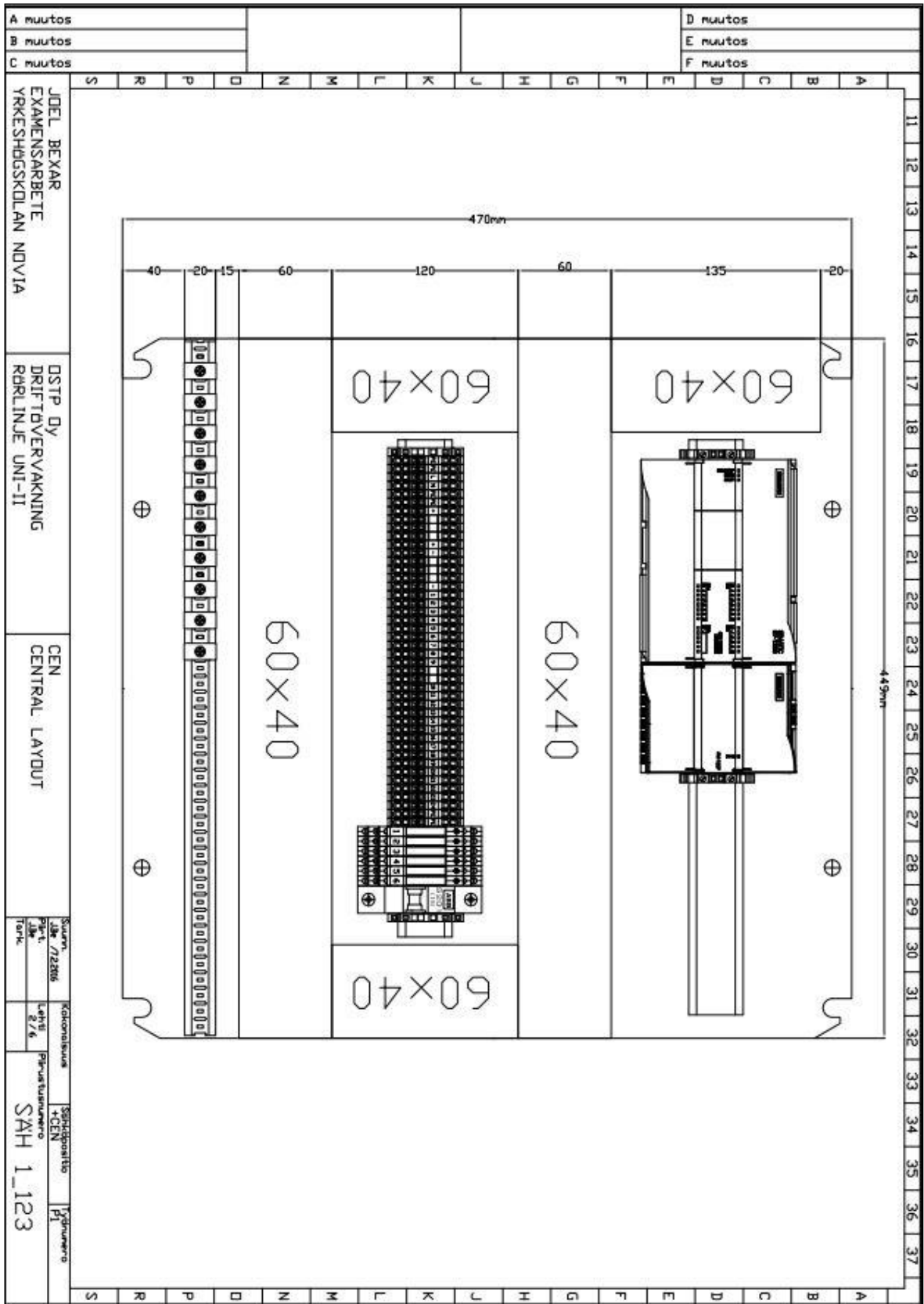
B DATABLAD

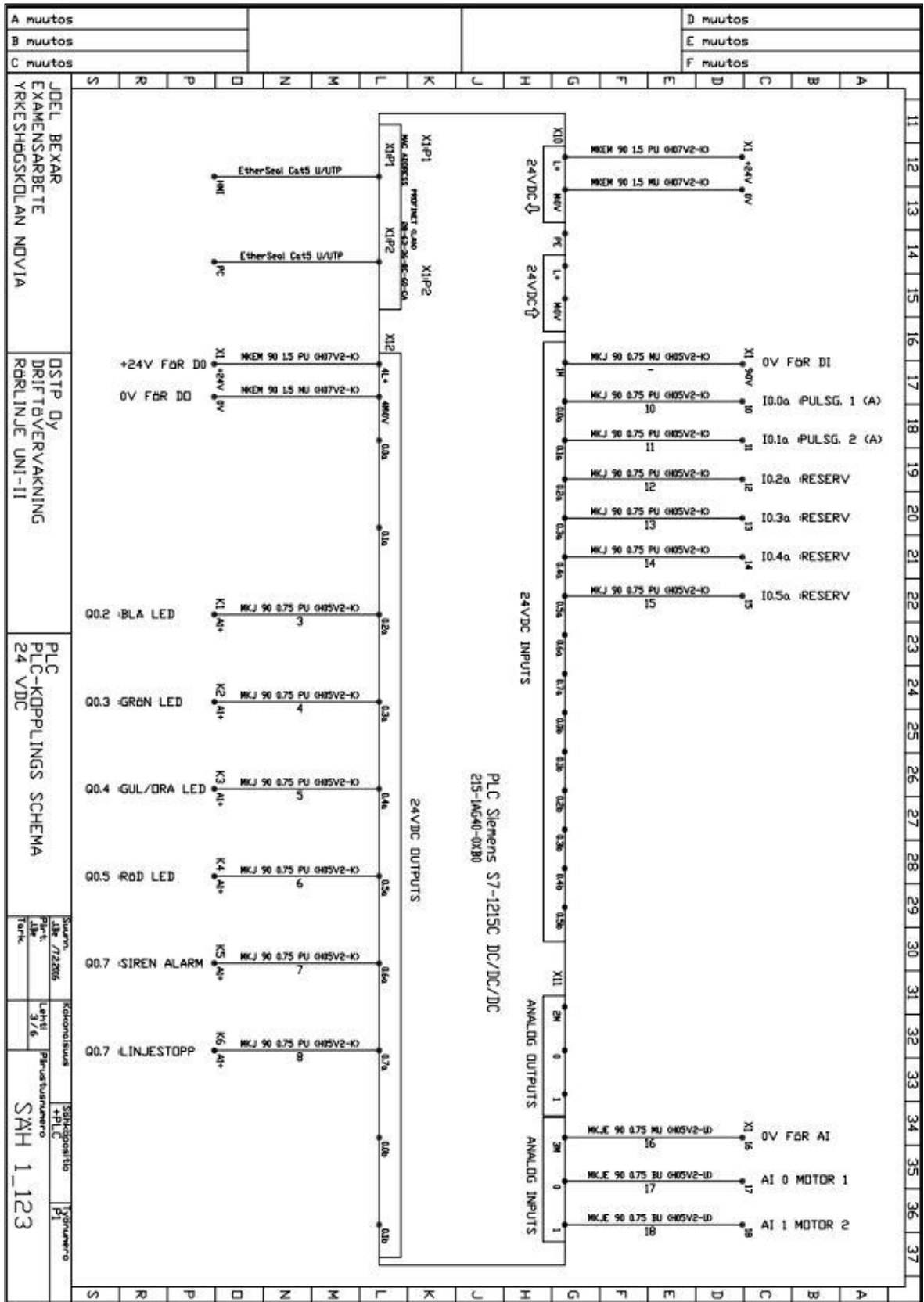
C INSTALLATIONSBILDER

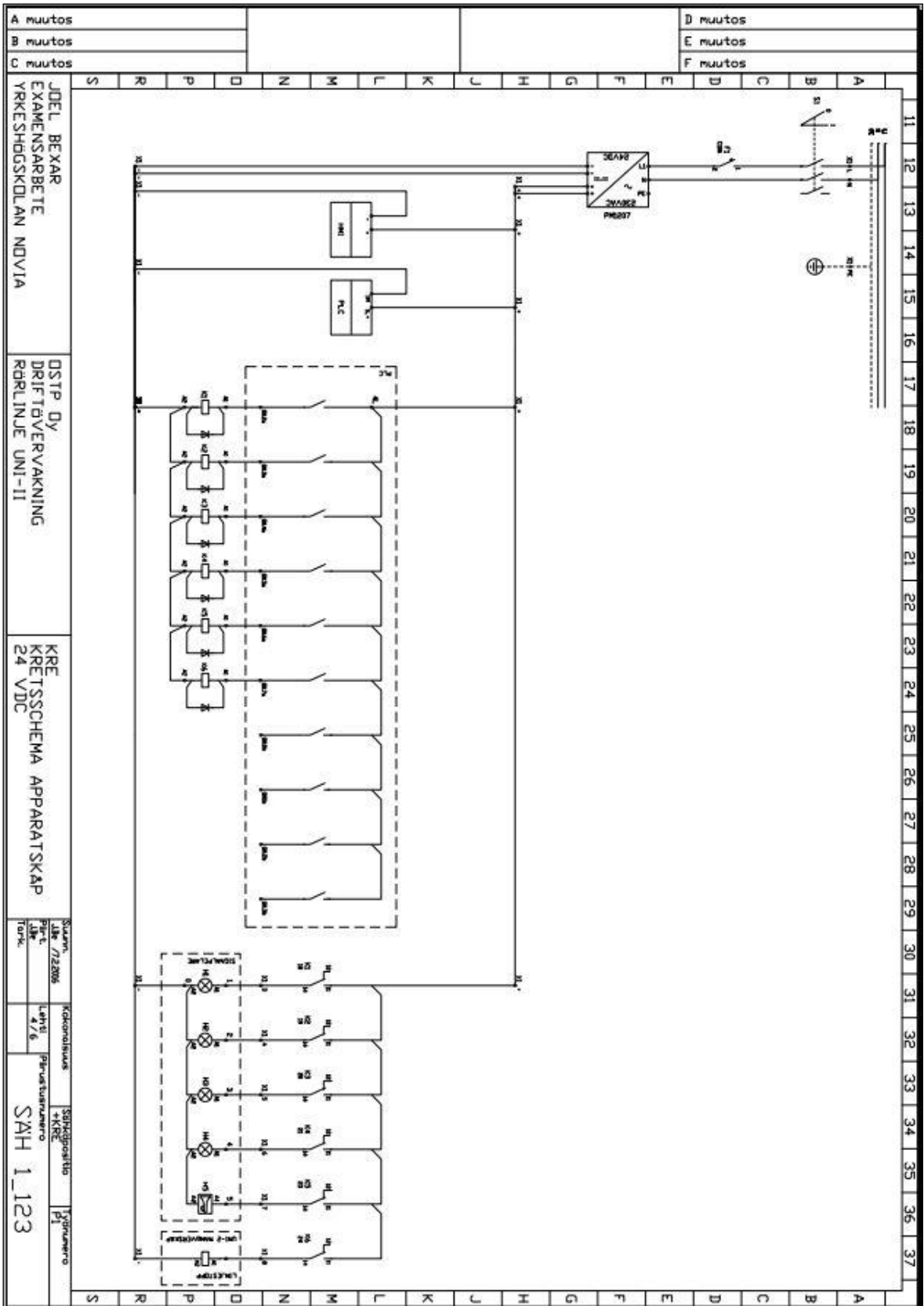
Bilaga A
Ritningar
sid 1

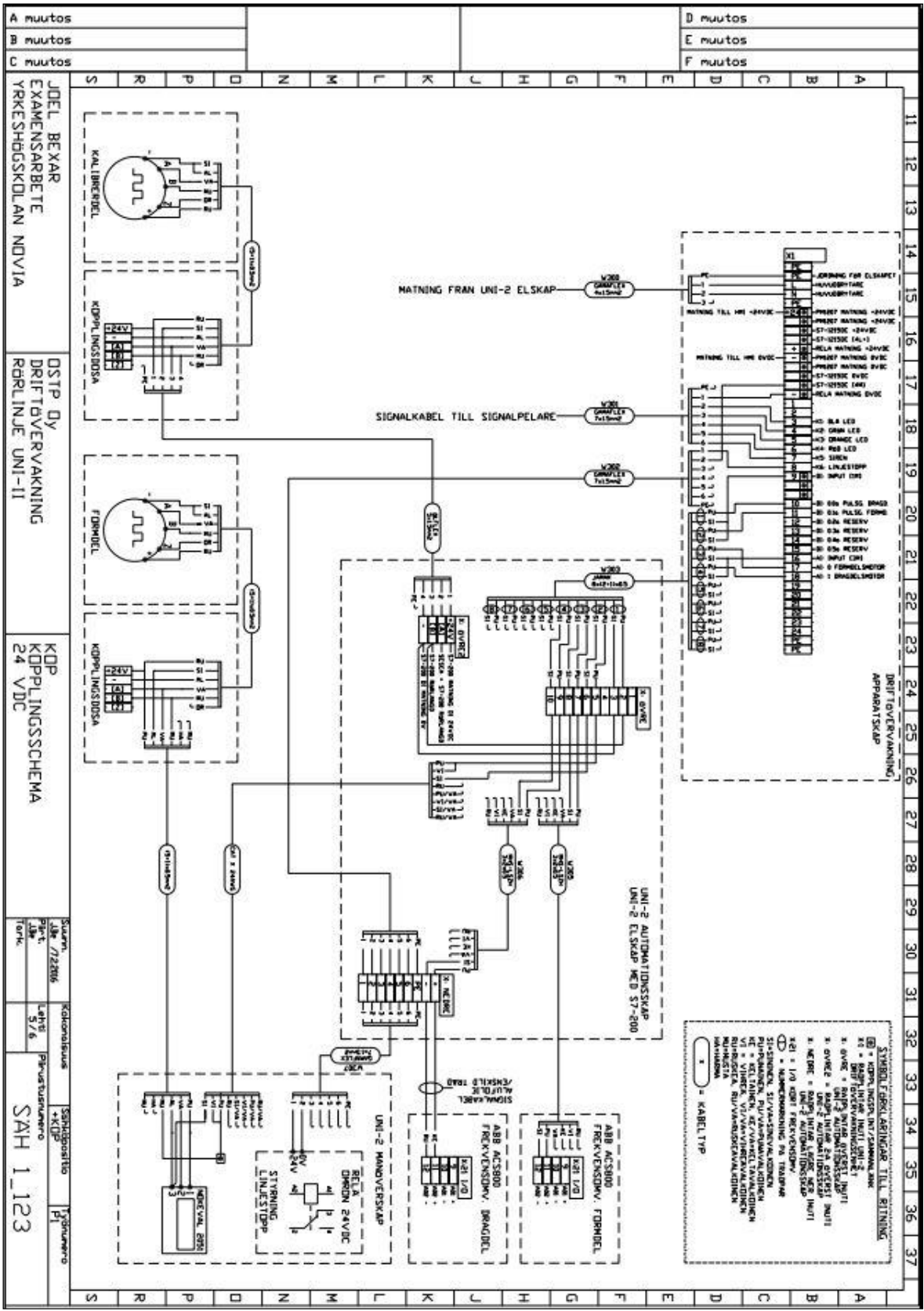


Bilaga A
Ritningar
sid 2









| | |
|---|--------|
| A | muutos |
| B | muutos |
| C | muutos |

| | |
|---|--------|
| D | muutos |
| E | muutos |
| F | muutos |

JOEL BEKAR
EXAMENSARBETE
YRKESHÖGSKOLAN NOVIA

DSTP OY
DRIFTÖVERVAKNING
ROBLINJE UNI-II

KOP
KOPPLINGSSCHEMA
24 VDC

| | |
|---------|--------|
| Säkerh. | 7/2816 |
| Fig-t. | 5/6 |
| Läp-t. | |
| Form. | |

Ködnombus
SSHöjdnus
+KOP
P
SAH 1_123

| I/O-lista | | |
|--------------------|--------|-------------------------------------|
| PLC kopplingspunkt | Adress | Beskrivning |
| DI a.0 | I 0.0 | Pulsgivare 1 kalibrerdel hastighet |
| DI a.1 | I 0.1 | Pulsgivare 2 formdel hastighet |
| DI a.2 | I 0.2 | X1:12 (Reserv) |
| DI a.3 | I 0.3 | X1:13 (Reserv) |
| DI a.4 | I 0.4 | X1:14 (Reserv) |
| DI a.5 | I 0.5 | X1:15 (Reserv) |
| DI a.6 | I 0.6 | Reserv |
| DI a.7 | I 0.7 | Reserv |
| DI b.0 | I 1.0 | Reserv |
| DI b.1 | I 1.1 | Reserv |
| DI b.2 | I 1.2 | Reserv |
| DI b.3 | I 1.3 | Reserv |
| DI b.4 | I 1.4 | Reserv |
| DI b.5 | I 1.5 | Reserv |
| | | |
| AI 0 | IW 64 | Frekvensomriktare formdel ström |
| AI 1 | IW 66 | Frekvensomriktare kalibrerdel ström |
| | | |
| DQ a.0 | Q 0.0 | Reserv |
| DQ a.1 | Q 0.1 | Reserv |
| DQ a.2 | Q 0.2 | Blå lampa (Ljustorn) |
| DQ a.3 | Q 0.3 | Grön lampa (Ljustorn) |
| DQ a.4 | Q 0.4 | Orange lampa (Ljustorn) |
| DQ a.5 | Q 0.5 | Röd lampa (Ljustorn) |
| DQ a.6 | Q 0.6 | Siren (Ljustorn) |
| DQ a.7 | Q 0.7 | Linjestopp |
| DQ b.0 | Q 1.0 | Reserv |
| DQ b.1 | Q 1.1 | Reserv |
| | | |
| AQ 0 | QW 64 | Reserv |
| AQ 1 | QW 66 | Reserv |

Materiallista för driftövervakning vid förhållje UMN-2

| Produkt | Tillverkare | Modell | Modellspecifikation | Typbeckning / Övrig info | Elnummer: SLO | Antal |
|------------------------------|-----------------|-----------------------------|------------------------------------|--------------------------|---------------|-------|
| Elskåp | Rittal | 1050.500 | L500KK500xS210 | 1050500 AE | 34 046 11 | 1 |
| Botenplatta för härprofil | Rittal | CP 60 | CP 6106.200 | - | - | 1 |
| Härprofil för botenplatta | Rittal | CP 120 sluten | CP 6212.100 | - | - | 1 |
| Vridbart fäste för härprofil | Rittal | CP 120 ø 130mm | CP 6212.300 | - | - | 1 |
| Blå Led 24VDC | Rittal | Stack Light | SG3372.040 | - | - | 1 |
| Grön Led 24VDC | Rittal | Stack Light | SG3372.010 | - | - | 1 |
| Gul Led 24VDC | Rittal | Stack Light | SG3372.020 | - | - | 1 |
| Röd Led 24VDC | Rittal | Stack Light | SG3372.000 | - | - | 1 |
| Siren 24VDC | Rittal | Stack Light | SG3376.000 | - | - | 1 |
| Anslutningskomponent | Rittal | Stack Light | SG3368.010 | - | - | 1 |
| Fot-rörmontage | Rittal | Stack Light | SG3374.010 | - | - | 1 |
| Rör 250mm ALU | Rittal | Stack Light | SG3374.020 | - | - | 1 |
| Vinkel rörmontage | Rittal | Stack Light | SG3374.050 | - | - | 1 |
| Texttyvla signalpelare | Rittal | Stack Light | SG3374.150 | - | - | 1 |
| Relämodul | Phoenix Contact | Kompaktrelä | PLC-RSC-24DC/21 | 2966171 | 35 271 49 | 6 |
| Relä | Phoenix Contact | Plug-in miniatur-relä | REL-MR-24DC/21 | 2961105 | 97 190 36 | 6 |
| Raddlämma | Phoenix Contact | Skruvanslutning | USLKG 2,5N - 0441119 (Gul/Grön) PE | 0,2mm2 - 4mm2 | 19 631 28 | 2 |
| Raddlämma | Phoenix Contact | Fjäderkraftanslutning | ST 2,5 PE - 3031238 (Gul/Grön) | 0,08mm2 - 4mm2 | 19 635 26 | 6 |
| Raddlämma | Phoenix Contact | Fjäderkraftanslutning | ST 2,5 - 3031212 (Grå) | 0,08mm2 - 4mm2 | 19 635 29 | 38 |
| Ändklämma | Phoenix Contact | Clipfix | CLIPFIX 35-5 - 3022276 (Grå) | - | 19 631 33 | 2 |
| DIN-skena | Weidmuller | TS (W35XH7,5mm) | - | Hål 6,3mm X 18mm | - | 1m |
| Kabelkanal | Rittal | Segma (LxK60x40) | PW21156 | - | 14 711 56 | 1,5m |
| Spiralrör | Rehan | 6mm | - | - | 13 716 02 | 1m |
| Ethernet Cat 6 kabel | Schneider | VDLP185X46010 | - | - | 72 003 61 | 1m |
| Dvärgbrytare | ABB | S201 C 10 A | C 10 A 10kA IP20 | 2CDS251001R0104 | 32 107 82 | 1 |
| Strömbrytare | ABB | 3-polig (Front operated) | OT16FT3 | 1SCA104838R1001 | 36 014 26 | 1 |
| Strömbrytare handtag | ABB | Genomgående, Snap On (Skåp) | OHB52PJ | 1SCA105231R1001 | 36 619 11 | 1 |
| CPU | Siemens | 1215C DC/DC/DC | 14D/10DO, 2AI/2AO | 6ES7215-1AG40-0XB0 | 27 020 77 | 1 |
| HMI | Siemens | KTP900 Basic | Key & Touch, Profinet | 6AV2123-2JB03-0AX0 | - | 1 |
| Power Module 24VDC | Siemens | PM 1207 | Power Module 24VDC/2,5A | 6EP1332-1SH71 | 27 020 81 | 1 |

SIEMENS

Data sheet

6ES7215-1AG40-0XB0

SIMATIC S7-1200, CPU 1215C, COMPACT CPU, DC/DC/DC, 2 PROFINET PORT, ONBOARD I/O: 14 DI 24V DC; 10 DO 24V DC 0.5A 2 AI 0-10V DC, 2 AO 0-20mA DC, POWER SUPPLY: DC 20.4 - 28.8 V DC, PROGRAM/DATA MEMORY: 125 KB



| General information | |
|---------------------------------------|--------------------------|
| Product type designation | CPU 1215C DC/DC/DC |
| Firmware version | V4.1 |
| Engineering with | |
| • Programming package | STEP 7 V13 SP1 or higher |
| Display | |
| with display | No |
| Supply voltage | |
| Rated value (DC) | Yes |
| • 24 V DC | Yes |
| permissible range, lower limit (DC) | 20.4 V |
| permissible range, upper limit (DC) | 28.8 V |
| Reverse polarity protection | Yes |
| Load voltage L+ | |
| • Rated value (DC) | 24 V |
| • permissible range, lower limit (DC) | 5 V |
| • permissible range, upper limit (DC) | 250 V |

SIEMENS

Data sheet

6AV2123-2JB03-0AX0

SIMATIC HMI, KTP900 BASIC, BASIC PANEL, KEY AND TOUCH OPERATION, 9" TFT DISPLAY, 65536 COLORS, PROFINET INTERFACE, CONFIGURATION FROM WINCC BASIC V13/ STEP7 BASIC V13, CONTAINS OPEN SOURCE SW WHICH IS PROVIDED FREE OF CHARGE FOR DETAILS SEE CD



| Product type designation | |
|--------------------------------|--|
| Display | |
| Design of display | TFT widescreen display, LED backlighting |
| Screen diagonal | 9 in |
| Display width | 198 mm |
| Display height | 111.7 mm |
| Number of colors | 65 536 |
| Resolution (pixels) | |
| • Horizontal image resolution | 800 |
| • Vertical image resolution | 480 |
| Backlighting | |
| • MTBF backlighting (at 25 °C) | 20 000 h |
| • Dimmable backlight | Yes |
| Control elements | |
| Keyboard | |
| • Function keys | |
| — Number of function keys | 8 |
| • Keys with LED | No |
| • System keys | No |
| • Numeric/alphabetical input | |
| — Numeric keyboard | Yes; Onscreen keyboard |
| — Alphanumeric keyboard | Yes; Onscreen keyboard |
| Touch operation | |
| • Design as touch screen | Yes |

SIEMENS

Data sheet

6EP1332-1SH71

POWER SUPPLY S7-1200 PM1207
 SIMATIC S7-1200 POWER MODULE PM1207 STABILIZED
 POWER SUPPLY INPUT: 120/230 V AC OUTPUT: 24 V DC/2.5 A

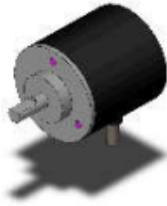


| Input | |
|--|---|
| Input | 1-phase AC |
| Supply voltage | |
| <ul style="list-style-type: none"> • 1 at AC Rated value • 2 at AC Rated value • Note | 120 V 230 V Automatic range selection |
| Input voltage | |
| <ul style="list-style-type: none"> • 1 at AC • 2 at AC | 85 ... 132 V 176 ... 264 V |
| Wide-range input | No |
| Overvoltage resistance | 2.3 × Vin rated, 1.3 ms |
| Mains buffering at Iout rated, min. | 20 ms; at Vin = 83/187 V |
| Rated line frequency 1 | 50 Hz |
| Rated line frequency 2 | 60 Hz |
| Rated line range | 47 ... 63 Hz |
| Input current | |
| <ul style="list-style-type: none"> • at rated input voltage 120 V • at rated input voltage 230 V | 1.2 A 0.67 A |
| Switch-on current limiting (+25 °C), max. | 13 A |

Incremental 40-mm-dia. Rotary Encoder

E6B2-CWZ5B 500P/R 0.5M

Incremental, Resolution 500P/R, 40-dia., 12 to 24 VDC, PNP open collector output, Pre-wired models (0.5 m)



Image

| | |
|-------------------------------|---------------------------------------|
| Encoding method | Incremental |
| Resolution (Single turn type) | 500 P/R |
| Output phases | A, B and Z |
| Control output (Output type) | PNP open collector output |
| Connection method | Pre-wired models(Cable_length: 0.5 m) |

Ratings/Performance

As of November 11, 2015

| | |
|-----------------------------------|--|
| Encoding method | Incremental |
| Power supply voltage | DC12 to 24 V (ripple (p-p): 5% max.) |
| Current consumption | 100 mA DC Max. |
| Inrush current | approx. 9A(Time : approx.0.3ms) |
| Resolution (Single turn type) | 500 P/R |
| Output phases | A, B and Z |
| Control output (Output type) | PNP open collector output |
| Control output (Load current) | 35mA max.(source current) |
| Control output (Residual voltage) | 0.4 VDC max.(load current:at 35mA Max.) |
| Starting positional point | Equipped |
| Load power supply voltage | 30 VDC max. |
| Max. response frequency | 50 kHz |
| Phase difference on output | 90 DEG -45 to +45 DEG |
| Rise and_fall times of output | Rise time of output: 1 μs max. Fall time of output: 1 μs max. (Cable length: 2m, source current: 10mA) |
| Starting torque | Room temperature: 0.98 mN.m max. |
| Moment of inertia | 3×10^{-7} kg.m ² max. |
| Shaft loading | Radial: 30 N Thrust: 20 N |
| Max. permissible rotation | 6000 r/min |
| Mechanical life expectancy | 100 million rotations min. |
| Protective circuit | Output short cut protection Power supply reverse polarity protection |
| Ambient temperature | Operating: -10 to 70 °C Storage: -25 to 85 °C |

Bilaga C
Installationsbilder
sid 1



Driftövervakningssystemet hopmonterat till vänster och driftsatt till höger.

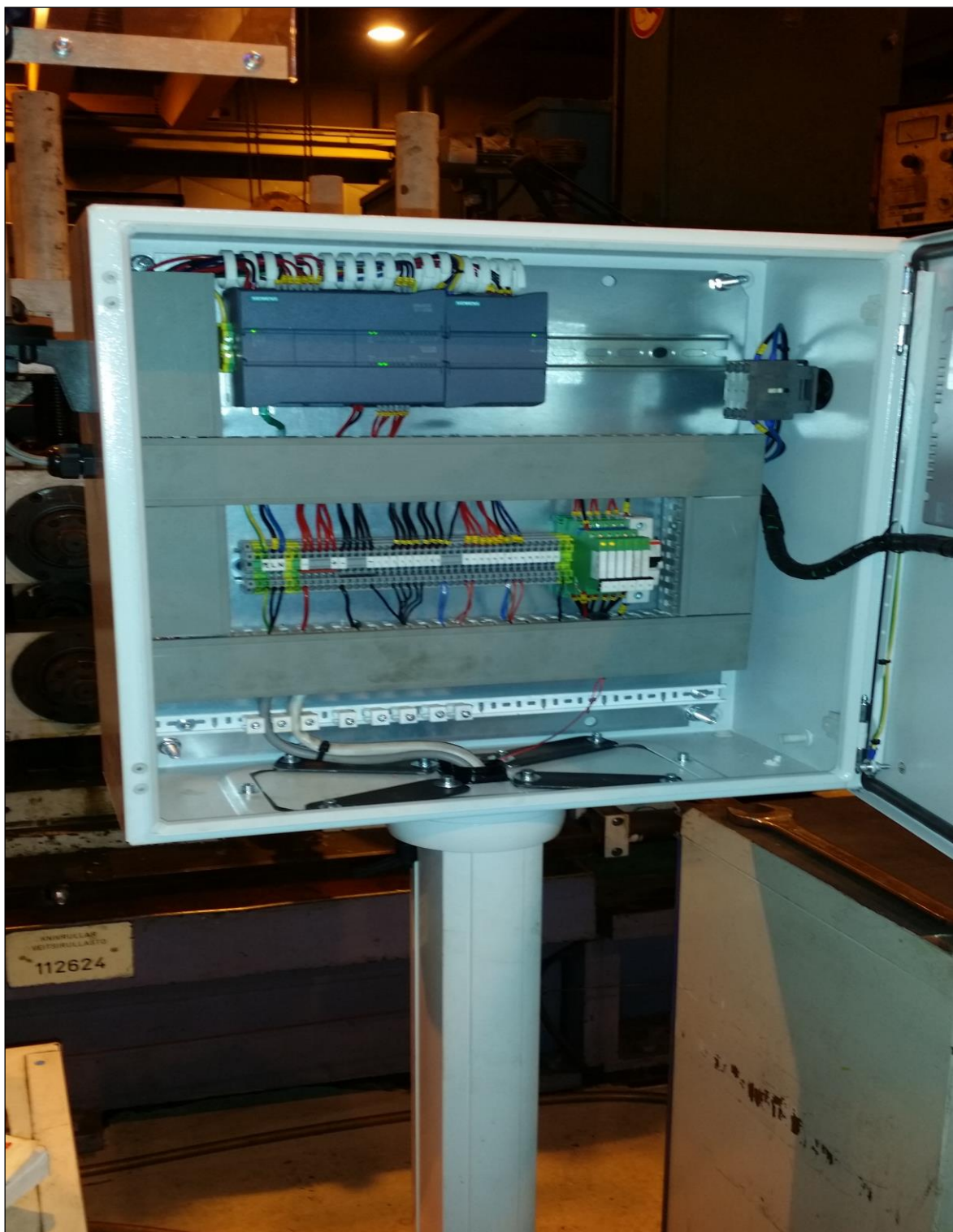


Bild på elskåpets inandöme (1/2).



Bild på elskåpets inandöme (2/2).

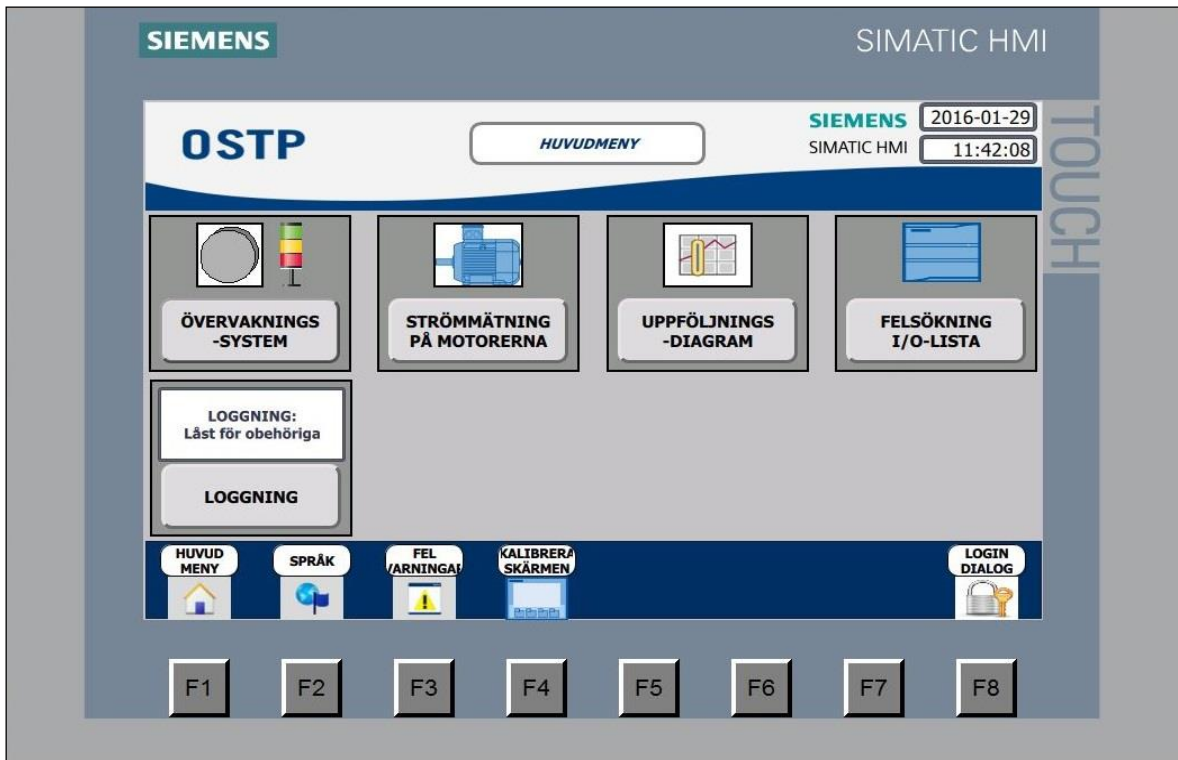


Bild på driftövervakningssystemets sida: huvudmeny.

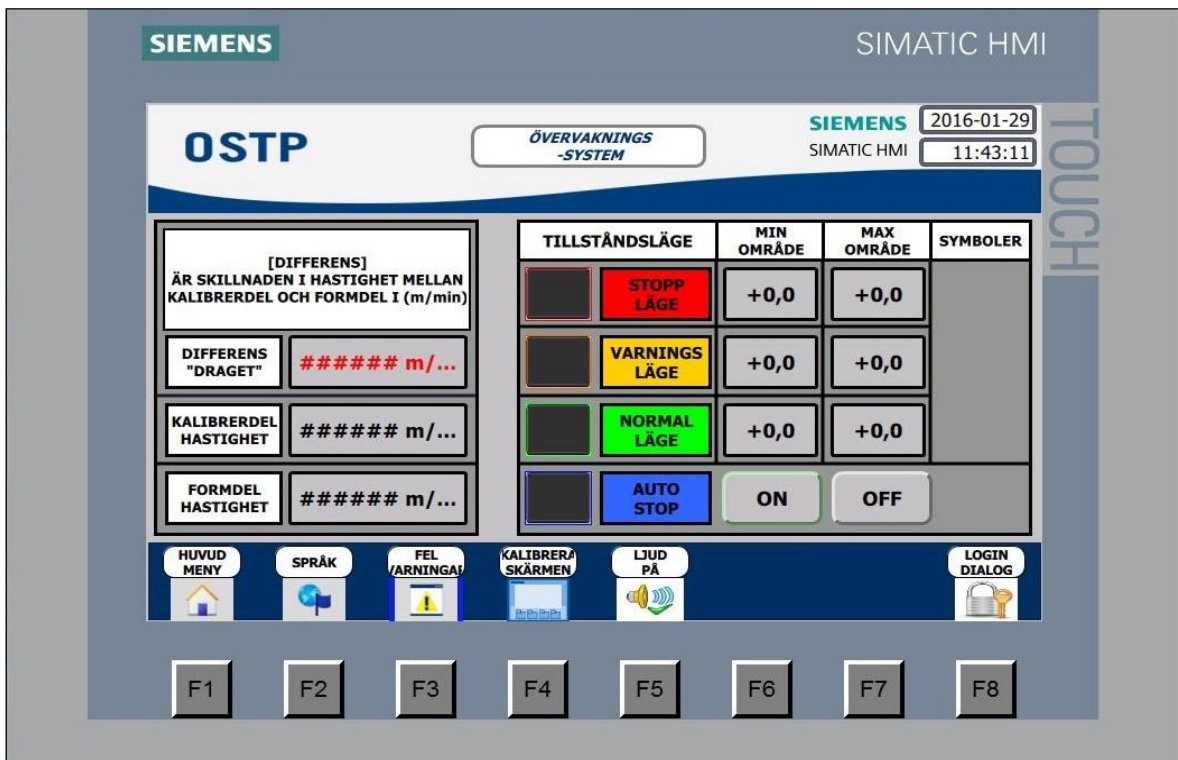


Bild på driftövervakningssystemets sida: övervakningssystem.

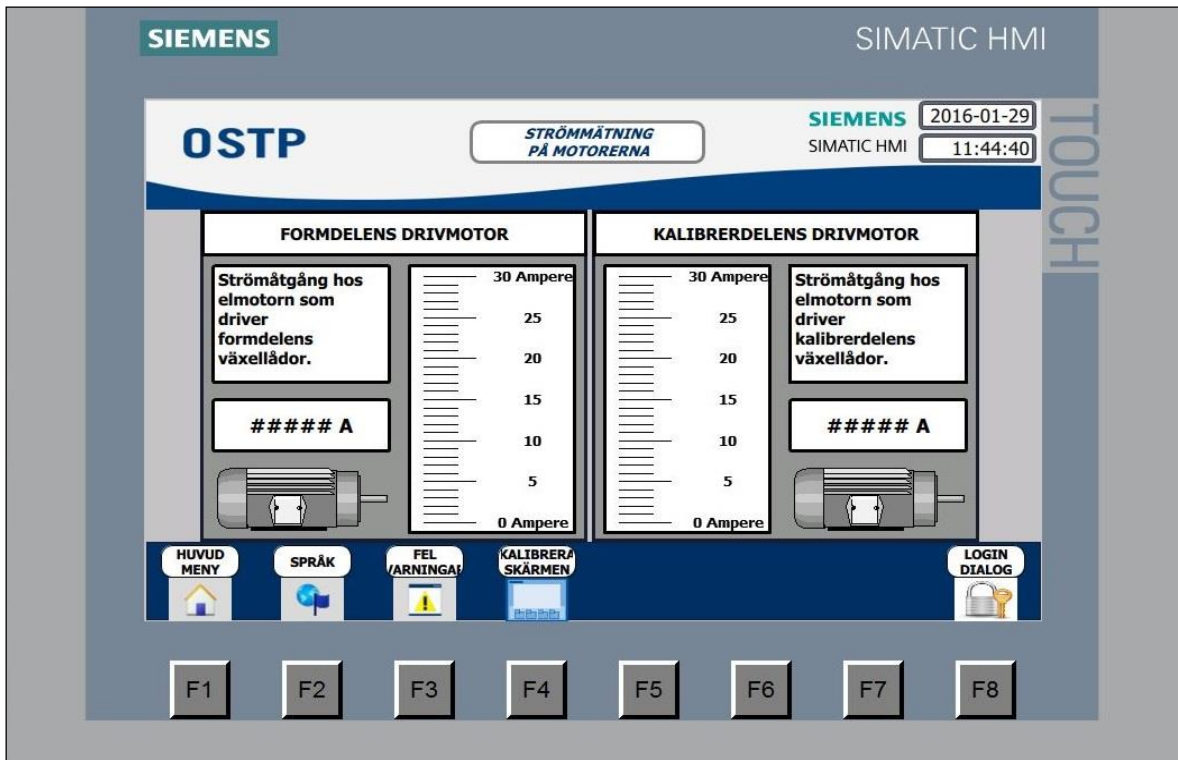


Bild på driftövervakningssystemets sida: strömmätning på motorerna.

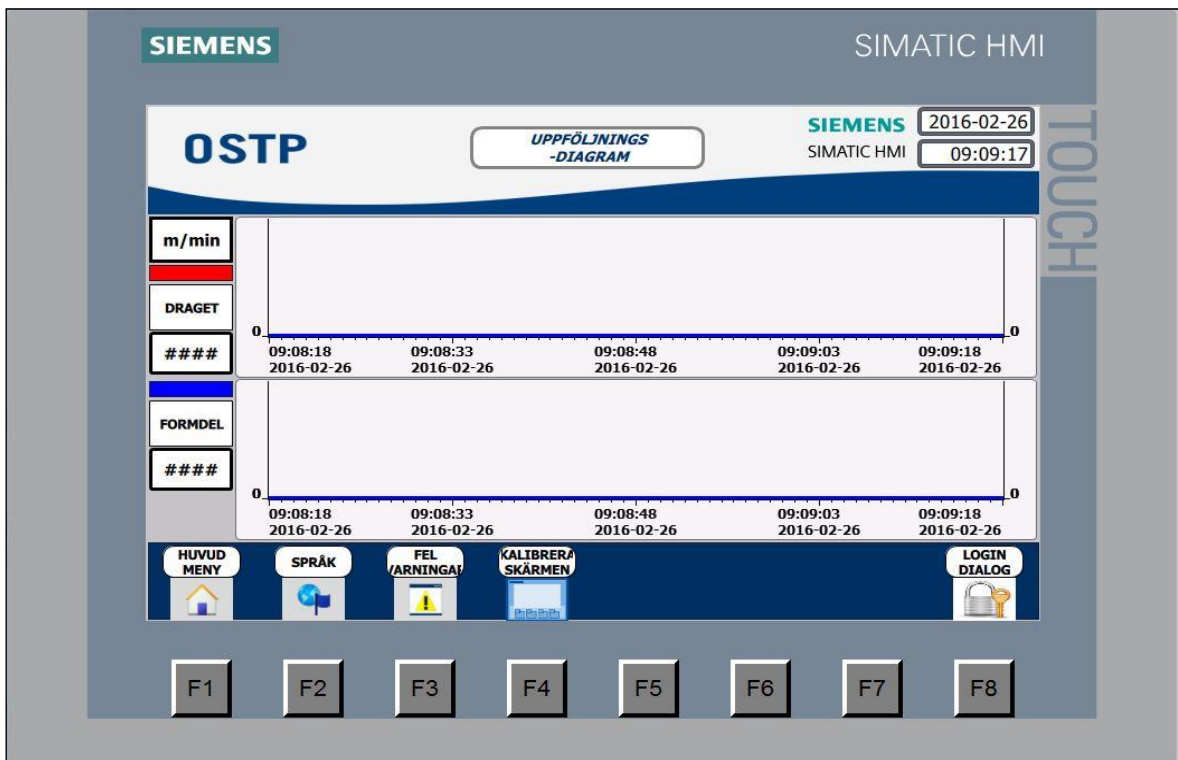


Bild på driftövervakningssystemets sida: uppföljningsdiagram.

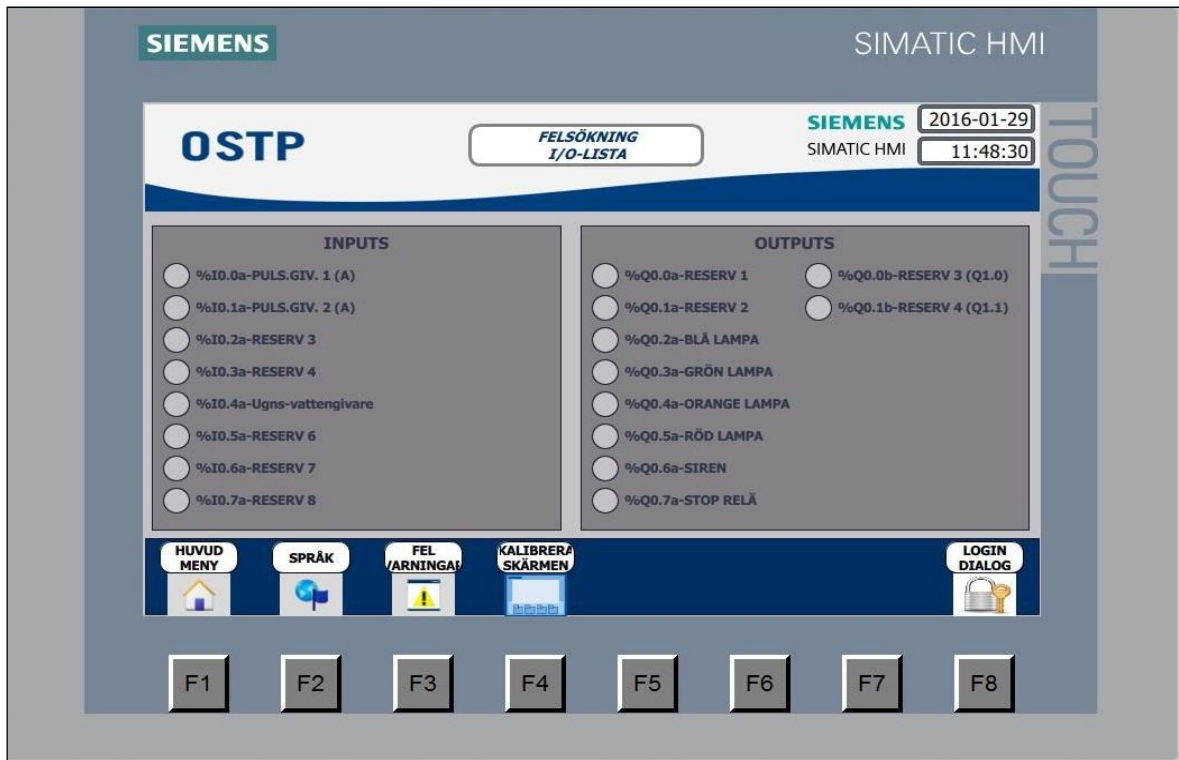


Bild på driftövervakningssystemets sida: felsökning.