

Realisering av testloop

Dokumentering och programmering av testanläggning



Niclas Andersson
Sam Ebadeh

Division of Industrial Electrical Engineering and Automation
Faculty of Engineering, Lund University

Realisering av testloop

Dokumentering och programmering av testanläggning



LUNDS UNIVERSITET
Campus Helsingborg

LTH Ingenjörshögskolan vid Campus Helsingborg
Avdelningen för Industriell Elektroteknik och Automation

Examensarbete:
Niclas Andersson
Sam Ebadeh

© Copyright Niclas Andersson, Sam Ebadeh

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

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

Tryckt i Sverige
Avd. för Industriell Elektroteknik och Automation
Lund 2014

Sammanfattning

KNAPP-Moving AB i Åstorp har specialiserat sig i främst automatisk pallhantering. Allt jobb exklusive rapporten utfördes i huvudkontoret i Åstorp.

KNAPP-Moving har en enklare variant av en testloop i Åstorp där de testar nya komponenter. Företaget testar exempelvis nya motorer eller nya givare i sin egen testloop för att förenkla utvecklingen av kundernas anläggning. I uppdraget KNAPP-Moving gav oss ingick det att dokumentera deras testloop. I uppdraget ingår det även att programmera loopen med ett nytt PLC, Siemens S7-1200, programmera och designa en ny HMI och att jämföra effekten på deras nyinförskaffade motorer med deras äldre motorer. Resultatet av arbetet blev en PLC som arbetar som deras tidigare PLC. En HMI som har en renare design och ett effektresultat som uppfyllde tillverkarnas beskrivning av motorerna.

Nyckelord: PLC, TIA-portalen, KNAPP-Moving, Siemens, automation, smarta motorer

Abstract

KNAPP-Moving AB in Åstorp specialize in automatic storage and materials handling. We did all of the major work of the thesis in Åstorp.

KNAPP-Moving owns a testloop where they test their new components such as new motors and new switches to simplify the development for the customer. The assignment KNAPP-Moving gave us were to document their testloop. This includes programing the loop with their new PLC, Siemens S7-1200, program and design a new HMI and to measure the efficiency of their new smart motors and compare it with their old motors. The result of the project was a PLC that works like KNAPP-Movings earlier PLC. A HMI with a cleaner design than the before and an efficiency result that matched the manufacturers description of the smart motors.

Keywords: PLC, TIA-portal, KNAPP-Moving, Siemens, automation, smart motors

Förord

Detta examensarbete har utförts under femton veckor som den avslutande kurs i högskoleingenjörutbildningen inom elektroteknik med inriktning mot automation vid Lunds tekniska högskola. Projektet utfördes på KNAPP-Moving AB i Åstorp som har varit uppdragsgivaren i detta projekt.

Vi vill tacka våra handledare på KNAPP-Moving Pär Eliasson, Magnus Lundh för en utmärkt handledning under projektets gång som guidat oss till ett lyckat projekt. Vi vill även säga tack till all personal på KNAPP-Moving som hjälpt och stöttat oss med att lyckas med projektet.

Vi vill även tacka vår handledare och examinator på LTH, Mats Lilja för att ha hjälpt oss att komma igång och färdigställa vårt arbete.

Niclas Andersson
Sam Ebadeh
Åstorp, 2014-08-15

Innehållsförteckning

1.1 Bakgrund.....	2
1.2 Problemformulering	2
1.3 Syfte	3
1.4 Målsättning	3
1.5 Metod.....	3
1.6 Avgränsningar	3
2 Teknisk bakgrund	4
2.1 Layouten för testloopen.....	4
2.2 Ellayout	4
2.3 Apparatlistan	6
2.4 Kretsschema.....	7
2.5 I/O-lista	8
2.6 Kommunikation mellan PLC och I/O	9
2.6.1 Fysisk koppling	9
2.6.2 Profinet, switchar, realtidskommunikation & fältbussystem	10
3 Utförande.....	11
3.1 Konfigurering av PLC.....	11
3.2 Programmering.....	12
3.2.1 Mainprogrammet.....	12
3.2.2 Stegfunktion	12
3.2.3 Waitfunktion	13
3.2.4 Bandfunktion	13
3.2.5 HMI-funktion	14
3.2.6 Filterfunktion	14
3.2.7 Tidmättningsfunktion	14
3.3 Tillvägagångssätt för programmering	15
3.4 Implementering av HMI	16
3.4.1 Manuell körning.....	16
3.5 Felsökning	19
3.6 Effektprovet	20
4 Resultat.....	21
5 Slutsats.....	22
6 Framtida utvecklingsmöjligheter	25
7 Källor	26
7.1 Digitala källor	26
7.2 Litterär källa	26
8 Bilaga: HMI-manual.....	27

Terminologi

Apparatlista - En lista med information om alla apparater som används i anläggningen.

AS-i - Actuator Sensor Interface, en busslinga som ger komponenter unika adresser.

Busscykeltid - En tid för busscykeln att börja om igen.

D-koppling - En variant på koppling mellan faserna. Används när en högre spänning ska användas. Ingen nollpunkt är inkopplad vid en D-kopplad motor.

Ellayout - En 2D ritning på anläggningen som visar hur den är ritad.

Ethertype - Del av ett ethernettelegram som anger protokolltypen.

Frekvensomvandlare - Omvandlare som styr vilken frekvens motorn ska köra i.

Fotoelektriska givare - Sensor, en elektrisk apparat som omvandlar elektricitet till ljus. I fall ljusstrålen bryts aktiveras givaren.

Fältbuss - Se AS-i

Fyrvägsväxel - Ett transportsätt där föremålet som transporteras får tre stycken vägar, fram, höger & vänster.

Gateway - Övergång mellan två nätverk med olika hårdvara och mjukvara

HMI - Human to Machine Interface - användargränssnitt

I/O - Inputs/Outputs, berättar vilken ingång och utgång som används.

Induktiva givare - Sensor, en speciell givare som aktiveras när ett föremål som kan leda när elektricitet närmar sig ytan.

IRT - Isokron Realtidskommunikation, en kommunikationstyp som sker enbart på nätverk.

Kabelstege - En "stege" där kablarna följer för att hålla kablarna ihop.

Kretsschema - Ritats i CAD för att förklara hur de olika komponenterna är kopplade.

Ladderdiagram - Grafiskt programspråk som använder sig utav logiska grindar som villkor.

Ljusbom - En infraröd laser skjuts från fyra stycken givare och en spegel reflekterar tillbaks strålen. Om strålen bryts i fel ordning så avbryts driften i anläggningen.

M16 anslutning - En kontakt, i detta examensarbete används det främst för att plocka upp bussadress eller ge en signalanslutning.

Moment 22 - Ett moment där hela anläggningen står stilla, ingenting körs för alla funktioner väntar på varandra.

Off delay - Frånslagsfördröjning, fördröjer frånslaget på funktionen med x sekunder.

OP-panel - Operatörsskåpet, i detta skåp finns det PLC, Gateway och majoriteten av kommunikationen mellan hårdvaran och anläggningen.

PLC - Programmable Logic Controller. Ett programmerbart styrsystem, används för att styra anläggningens in och utgångar.

Prosbeteckning - En unik beteckning för att enklare förstå vilken komponent eller delsträcka det handlar om.

Proxy - En "mellanhand" för PLCn och busslingan.

RT - Realtidskommunikation, se stycke 2.5.2 för beskrivning av systemet

Tagg - En variabel som representerar ingång eller utgångsminne i programmeringsdelen.

Y-koppling - En variant på koppling mellan faserna, nollpunkt existerar. Används när det behövs en mer variation på spänningen.

Inledning

1.1 Bakgrund

Testloopen som KNAPP-Moving äger används för att optimera både tjänster och produkter som de erbjuder till sina kunder. Testloopen används idag för att testa olika PLC och komponenter. På företaget testas även nya motorer som styr transportbanden och diverse givare.

KNAPP-Moving har nyligen införskaffat sig en ny PLC, Siemens S7-1200 och nya smarta motorer. Smarta motorer kallas smarta av flera olika anledningar. En anledning är att den inte behöver kopplas till en extern frekvensomvandlare utan motorn har en inbyggd. Fördelen med detta är att företaget slipper att koppla ihop komponenterna med kablar och det minskar även antalet kablar ute på anläggningen. En annan fördel med smarta motorer enligt tillverkarna är att de har en exceptionellt låg energiförbrukning. Denna fördel var en av anledningarna till att KNAPP-Moving införskaffade sig dessa smarta motorer och därför är de intresserade av att verifiera att tillverkarnas beskrivning av motorn stämmer. Om tillverkarnas beskrivning stämmer är det högst sannolikt att KNAPP-Moving kommer ersätta sina äldre motorer med dessa nya smarta motorer så att deras anläggning blir effektivare.

1.2 Problemformulering

Anläggningen har idag flera funktioner som den nya PLCn också ska ha exempelvis att flera pallar kan köra samtidigt, att när flera pallar körs så ska anläggningen kunna undvika kollisioner. Däremot kan anläggningen inte köra två pallar på ett och samma band. Detta är ett problem som ska lösas genom lämplig programmering. Anläggningen ska även programmeras så att den har samma funktioner som den äldre PLCn. De nya smarta motorerna som nyligen köpts in ska undersökas för att klargöra ifall de är effektivare än de äldre motorerna. Anläggningen ska även testas för att visa hur många pallar som kan köras på hela anläggningen för att optimera pallhanteringen.

1.3 Syfte

Syftet med examensarbetet är att skapa gynnsammare förutsättningar för KNAPP-Moving AB genom att mäta och jämföra de nya smarta motorerna med de äldre motorerna. Även att uppdatera anläggningen med den senaste PLCn och att förse KNAPP-Moving AB en fulländad dokumentationslista om deras anläggning. Detta innefattar Layout, apparatlista, I/O-lista och kretsschema. En HMI ska även implementeras för att underlätta drifttestning för framtida bruk.

1.4 Målsättning

Målsättningen för detta examensarbete är en testanläggning med en ny programmerad PLC som skall ha liknande funktioner som den tidigare PLCn. En fullständig dokumentation om testanläggningen i form av ellayout, apparatlista, I/O-lista och kretsschema skall också ingå. Det skall även skapas en ny design och omprogrammera anläggningens HMI samt att få ett konkret resultat av effektmätningarna för de smarta motorerna och att fastställa resultatet av testerna till KNAPP-Moving.

1.5 Metod

Eftersom examensarbetet har varit praktiskt inriktat så har det inte följts någon känd metod utan utförandet har utförts genom empiriska studier. Alla delmoment har krävts att undersöka hur testanläggningen och testanläggningens komponenter fungerar. För att underlätta för användaren försökte vi skapa en HMI med så användarvänlig interface som möjligt. Efter att ha sett KNAPPs standardlayout på interface bestämde vi oss för att inkorporera den i vår design. När det kommer till PLC-programmeringen valde vi att utgå från den strukturen som vi har lärt oss under studietiden.

1.6 Avgränsningar

I projektet kommer vi inte att gå in på djupet på hur vissa funktioner fungerar eller gå in på djupet på hur komponenter och program fungerar. I mån av tid kommer vi att undersöka ifall en handhållen HMI kan införas samt övervakning med hjälp av kameror på anläggningen för att underlätta felsökning och driftsättning.

I examensarbetet kommer även vissa ord att vara kursiverade och dessa kursiverade ord förklaras i terminologin.

2 Teknisk bakgrund

2.1 Layouten för testloopen

Loopen är en kombination av olika transportsätt för att transportera en pall från punkt A tillbaka till punkt A igen. Loopen är av en rektangulär form. Varje del av maskinen har sin egen motor med sin egen *prosbeteckning*. Alla givare får beteckningen B och sedan en siffra efter beroende på vilken ordning givaren kommer på en speciell sträcka.

Motorn betecknas med bokstaven M och en siffra efter bokstaven. Siffrans betydelse beror på ifall det finns fler än en motor på en av transportsträckorna. Om det finns fler än en motor på sträckan så tyder siffran på vad den har för funktion på sträckan.

Exempel på detta är om en motor driver band, kedja, lift eller ett vrid. För att på smidigaste sätt kunna läsa av layouten är det bestämt att motorena ska betecknas i en viss ordning.

Prioriteringsordningen för siffran skrivs i följande ordning:

Rullbandet > Horisontella färdriktningar > Vertikala färdriktningar

I detta arbete har det första bandet prosbeteckningen +11000, komponenterna som finns på detta band har också sin unika prosbeteckning. Exempel bandet +11000 har en givare som heter B1, denna givare kommer få prosbeteckningen +11000-B1. Att återanvända B1 och M1 på andra sträckor har ingen betydelse då prosbeteckningen för transportsträckan ändras. Bandets prosbeteckning ökar med tjugo åt gången, det vill säga nästa prosbeteckning skulle bli +11020. Bandet efter det skulle bli +11040. Detta utförs på grund av att om kunden någon gång i framtiden skulle vilja utöka antalet transportband i sin anläggning så monteras bandet in mellan två transportband. Vill kunden ha ett transportband mellan +11000 och +11020 så får bandet emellan namnet +11010. I anläggningen finns det även apparatskåp som bland annat innehåller *frekvensomvandlare*, relä och kontaktorer. Dessa apparatskåp har prosbeteckningen +A1101. Beteckningen ökar med ett för varje skåp precis som med givarna. I *ellayouten* finns även I/O-modulerna och dessa följer samma mönster som apparatskåpen och börjar på +A1111.

Komponenter som används i denna anläggning är *fotoelektriska givare*, *induktiva givare*, *ljusbommar*, elektriska motorer och *smarta motorer*.

2.2 Ellayout

Automationsingenjören ska kunna rita en layout på en anläggning i CAD. Ellayouten är i 2D och visar hur anläggningen ser ut ovanifrån. Layouten ska kunna läsas av så enkelt som möjligt och för att detta ska ske så måste ritningen vara enkel. Den ska heller inte gå in för mycket på detaljer utan layouten ska vara överskådlig så att läsaren kan hitta alla komponenter och skåp. Alla layouts i KNAPP-Moving ritas i programmet MicroStation.

I detta examensarbete var redan testanläggningen färdigmonterad. Detta betyder att projektet har fått utföras i fel ordning, det vill säga att projektet utgår från verkligheten till ritbord istället för hur det vanligtvis utförs vilket är från ritbordet till verkligheten.

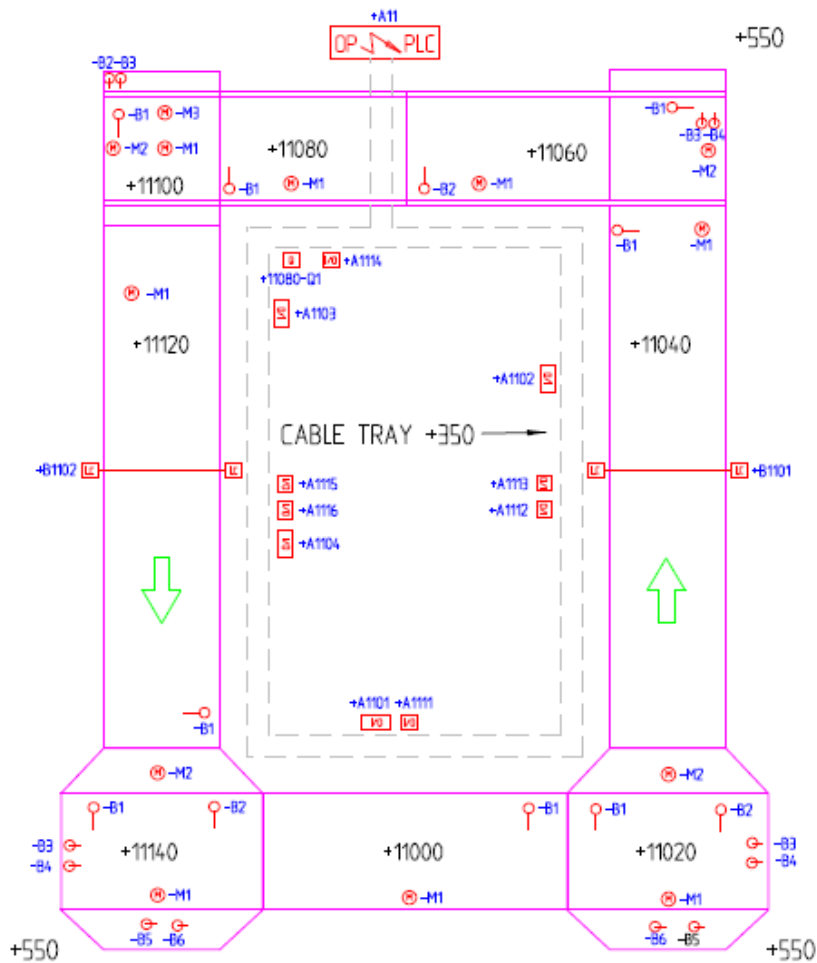
Innan layouten påbörjades behöver följande saker undersökas:

- Hur och var är komponenterna placerade på anläggningen?
- Vad har komponenterna för ändamål?
- Vilka mått har transportsträckorna (längd, bredd och höjd)?

Layouten som ritats innehåller följande information:

- Samtliga transportband
- Samtliga komponenter på loopen
- Samtliga apparatskåp
- *Kabelstege*
- Kabeltyper
- *OP-panel*
- Färdriktning
- Unika prosbeteckningar
- Elektrisk information om anläggningen

I större anläggningar kan det finnas strekkodsavläsare och andra sorters mätverktyg som beräknar massa, höjd eller någon annan form av avläsning. I KNAPP-Movings anläggning finns inget avläsningsverktyg, då detta är en enklare anläggning som enbart transporterar pallar.



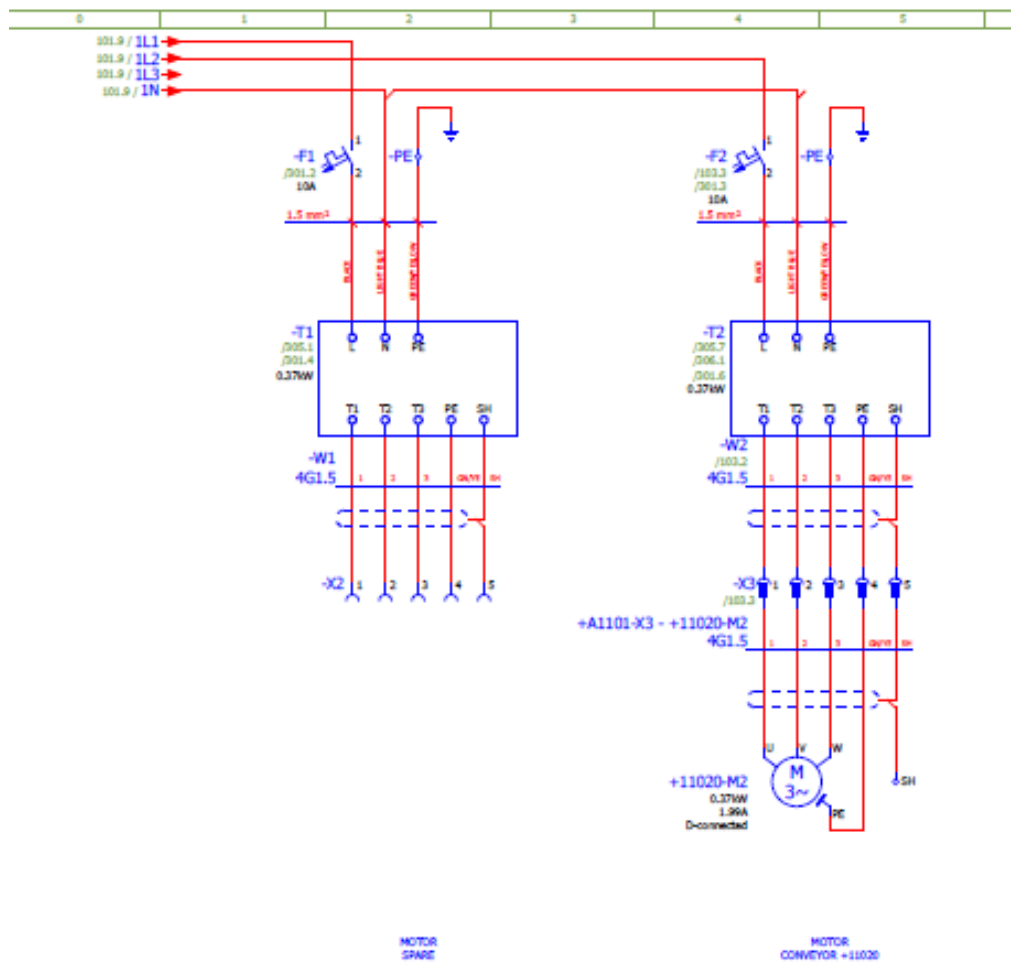
Figur 2.2: Bild på layouten som ritades i MicroStation. Kraftkabel, signalkabel och busskabel är bortfiltrerade.

2.3 Apparatlistan

KNAPP-Movings anläggning är, som tidigare nämnts redan färdigmonterad, så har alla motorer och andra komponenter är installerade. En del av uppgiften blev att ta reda på artikelnummer på samtliga komponenter som finns i anläggningen. För motorerna behövs det information som exempelvis om motorn är *D-* eller *Y-kopplad* eller annan information återfinns på motorskylten. Detta behövs för att kunna skriva en fullständig apparatlista. Listan skrivs i Excel.

2.4 Krettschema


Programmet som användes för att rita alla *kretsscheman* var E-Plan. Skillnaden mellan E-plan och andra CAD-program är huvudsakligen att E-plan är kopplad till en datorbas. Detta gör att det blir snabbare och smidigare att arbeta med detta program ifall användaren har en informationsrik databas. En annan skillnad är istället för att rita ut streck så sätter ritaren enbart ut förbindningspunkter mellan komponenterna. Även i detta arbetsmoment blev en del av uppgiften att undersöka hur komponenterna var kopplade för att kunna rita ett fullständigt kretsschema i E-Plan.



Figur 2.4: Bild från kretsschemat. I bilden så visas motorn M2 som är kopplad till frekvensomvandlaren T2

2.5 I/O-lista

Anledningen till att man har en I/O-lista är för att snabbt se var in- och utgångarna är placerade. I listan ska det finnas information om var komponenten är placerad på bandet och vad komponenten är kopplad till. Den beskriver även vilken bussadress komponenten har. Även listan för I/O är skriven i Excel.

 I/O-LIST PROJECT ??????? TESTLOOP KNAPP CONVEYOR SYSTEM =KES1						
MODULE	I/O	DEVICE 1	DESIGNATION	VARIABLE	DISTR. BUS ADDRESS	FIELD BUS ADDRESS
+A1101-K12	O3	+11020-M2	FREQUENCY CONVERTER	IO DrvUStartBreak	A11-K2	1-8A
+A1102						
+A1102-K11	I0	-Q1 + -F1 + -F2 + F3	MAIN CIRCUIT BREAKER + CONTACTOR + MINIATURE CIRCUIT BREAKER + MOTOR PROTECTOR	IO MainSw	A11-K2	1-6A-7A
+A1102-K11	I1			IO FreqCvtrOK	A11-K2	1-6A-7A
+A1102-K11	I2	+11040-M1	FREQUENCY CONVERTER	IO FreqCvtrOK	A11-K2	1-6A-7A
+A1102-K11	I3			IO PosSensFwd	A11-K2	1-6A-7A
+A1102-K11	I4	+11040-B1	PHOTOELECTRIC REFLEX SWITCH	IO PosSensStop	A11-K2	1-6A-7A
+A1102-K11	I5	+11060-B1	PHOTOELECTRIC REFLEX SWITCH	IO PosSensLRTbUp	A11-K2	1-6A-7A
+A1102-K11	I6	+11060-B3	INDUCTIVE PROXIMITY SWITCH	IO PosSensLRTbDwn	A11-K2	1-6A-7A
+A1102-K11	I7	+11060-B4	INDUCTIVE PROXIMITY SWITCH		A11-K2	1-6A-7A
+A1102-K11	O0				A11-K2	1-6A-7A
+A1102-K11	O1				A11-K2	1-6A-7A
+A1102-K11	O2				A11-K2	1-6A-7A
+A1102-K11	O3				A11-K2	1-6A-7A
+A1102-K11	O4	+11040-M1	FREQUENCY CONVERTER	IO DrvUStartFwd (bwd)	A11-K2	1-6A-7A
+A1102-K11	O5	+11040-M1	FREQUENCY CONVERTER	IO DrvUStartHighSpeed	A11-K2	1-6A-7A
+A1102-K11	O6	+11060-M2	CONTACTOR	IO DrvUStartLift	A11-K2	1-6A-7A
+A1102-K11	O7				A11-K2	1-6A-7A
+A1103						
+A1103-K11	I0	-Q1 + -F1 + -F2 + F3	MAIN CIRCUIT BREAKER + CONTACTOR + MINIATURE CIRCUIT BREAKER + MOTOR PROTECTOR	IO EDM	A11-K2	1-3A-4A
+A1103-K11	I1	+11100-M1	FREQUENCY CONVERTER	IO FreqCvtrOK	A11-K2	1-3A-4A
+A1103-K11	I2	+11100-M2	FREQUENCY CONVERTER	IO FreqCvtrOK	A11-K2	1-3A-4A
+A1103-K11	I3			IO PosSensFwd	A11-K2	1-3A-4A
+A1103-K11	I4	+11100-B1	PHOTOELECTRIC REFLEX SWITCH	IO PosSensLRTbUp	A11-K2	1-3A-4A
+A1103-K11	I5	+11100-B2	INDUCTIVE PROXIMITY SWITCH	IO PosSensLRTbDwn	A11-K2	1-3A-4A
+A1103-K11	I6	+11100-B3	INDUCTIVE PROXIMITY SWITCH		A11-K2	1-3A-4A
+A1103-K11	O0				A11-K2	1-3A-4A
+A1103-K11	O1	+11100-M1	FREQUENCY CONVERTER	IO DrvUStartFwd (bwd)	A11-K2	1-3A-4A
+A1103-K11	O2	+11100-M1	FREQUENCY CONVERTER	IO DrvUStartHighSpeed	A11-K2	1-3A-4A
+A1103-K11	O3				A11-K2	1-3A-4A
+A1103-K11	O4	+11100-M2	FREQUENCY CONVERTER	IO DrvUStartFwd ???	A11-K2	1-3A-4A
+A1103-K11	O5	+11100-M2	FREQUENCY CONVERTER	IO DrvUStartHighSpeed	A11-K2	1-3A-4A
+A1103-K11	O6	+11100-M3	CONTACTOR	IO DrvUStartLift	A11-K2	1-3A-4A
+A1103-K11	O7				A11-K2	1-3A-4A
+A1104						
+A1104-K11	I0	-Q1 + -F1 + -F2 + -F3	MAIN CIRCUIT BREAKER + MINIATURE CIRCUIT BREAKER + MOTOR PROTECTOR + CONTACTOR	IO MainSw	A11-K2	1-12A-13A
+A1104-K11	I1	+11140-M1	FREQUENCY CONVERTER	IO FreqCvtrOK	A11-K2	1-12A-13A
+A1104-K11	I2	+11120-M1	FREQUENCY CONVERTER	IO FreqCvtrOK	A11-K2	1-12A-13A
+A1104-K11	I3	+11140-M2	FREQUENCY CONVERTER	IO FreqCvtrOK	A11-K2	1-12A-13A
+A1104-K11	I4	+11120-B1	PHOTOELECTRIC REFLEX SWITCH	IO PosSensFwd	A11-K2	1-12A-13A
+A1104-K11	I5			IO PosSensTwo	A11-K2	1-12A-13A
+A1104-K11	I6	+11140-B2	PHOTOELECTRIC REFLEX SWITCH	IO PosSensInOne	A11-K2	1-12A-13A
+A1104-K11	I7	+11140-B1	PHOTOELECTRIC REFLEX SWITCH		A11-K2	1-12A-13A
+A1104-K11	O0	+11140-M1	FREQUENCY CONVERTER	IO DrvUStartFwd	A11-K2	1-12A-13A
+A1104-K11	O1				A11-K2	1-12A-13A
+A1104-K11	O2	+11140-M1	FREQUENCY CONVERTER	IO DrvUStartHighSpeed	A11-K2	1-12A-13A
+A1104-K11	O3				A11-K2	1-12A-13A

Figur 2.5: Skärmdump från I/O-listan.

2.6 Kommunikation mellan PLC och I/O

2.6.1 Fysisk koppling

Alla in- och utgångar som hela anläggningen kommunicerar med *PLC* på följande sätt: Givarna och de smarta motorerna är kopplade till en I/O-moduls ingång eller utgång med en *M16-anslutning*. I I/O-modulen finns det flera ingångar som kan hämta en signal från en *AS-i* slinga eller en signalslinga med en 24 Volts insignal eller hämta signal från både *AS-i*'n och signalslingan. Utgångarna på I/O-modulen är kopplade till antingen givare eller motorer. *AS-i* slingan och busslingan är en enda stor slinga runt anläggningen.

Uppgiften för anläggningens apparatskåp är att styra de motorer som inte är smarta. Skåpens innehåll varierar efter motorernas specifika uppgift. Uppgiften kan vara ifall någon av motorerna ska kunna bromsa eller att någon av motorerna ska kunna ändra hastighet under körningen. Detta utförs med hjälp av en frekvensomvandlare. Samtliga apparatskåp får in I/O modulens signal och de förser motorer och givare med en signalmatning men även en kraftmatning från en separat kraftmatningsslinga.

Varje modul har en specifik bussadress som bestäms av programmeraren. Bussadressen kan ändras och tillsättas på en angiven komponent genom att koppla in ett instrument i den I/O-modulen som styr komponenten och därefter programmera modulen och tilldela eller ändra en adress. Ett exempel för att illustrera detta är ifall en givare sitter i en I/O-modul på modulens ingång 0 och den ingången har adressen 5A. Då tilldelas givaren adressen 5A-I0.

I *OP-panelen* som också är huvudskåpet i anläggningen är en *AS-Interface Master/Gateway* inkopplad. Denna hämtar upp eller tilldelar alla komponenters bussadresser genom *AS-i* slingan. *Gatewayen* kan ha upp till 64 stycken olika adresser. *Gatewayen* kommunicerar med *PLCn* och kommunikationen sker med hjälp av en ethernetkabel som kommunicerar genom *Profinet*.

2.6.2 Profinet, switchar, realtidskommunikation & fältbussystem

Profinet är en automationslösning som är baserad på industriellt *Ethernet*, vars uppgift är att på ett smidigt sätt sprida I/Os och tidskritiska applikationer. Profinet har även möjligheten att fritt mixa olika system av *fältbussar* och ethernet.

Vid kommunikation kan Profinet använda sig av olika sätt att kommunicera beroende på behovet. Vid överföring av icke tidskritisk data använder sig Profinet standarden av överföringskanaler med hjälp av TCP, UDP och IP. Detta är grunden till att kunna skapa en förbindelse till andra nätverk. Dock brukar detta inte vara tillräckligt när det kommer till industriapplikationer. När Profinet överför tidskritisk processdata används funktionen *RT (realtidskanal)* eller vid mer precision och hastighetskrävande fall, *IRT (Isokron realtidskanal)*. Kommunikationen är optimerad med för hastighet och kan ske på samma AS-i-kabel.

Kanalerna som TCP- och UDP-/IP använder har en mängd kontroller som gör att datatrafiken blir långsammare. För att kommunikationen ska ske så snabbt som möjligt väljer Profinet att inte använda dessa delar via RT-kommunikation. Sättet som ethernet kommunicerar på är mer lämpat för detta. I KNAPP-Movings anläggning används denna kommunikationstyp.

Vid denna typ av kommunikation inom ett nätverk behövs ingen information om mottagarnätverkets adress. Datan identifieras istället vid ett mottagande med hjälp av *Ethertypen*. RT-kommunikation har en *busscykeltid* på ca en millisekund.

Switchar är nätverksenheter som mellan aktiva komponenter fungerar som en överföringsväg i ett nätverk. Deras huvudsakliga uppgift är att skicka vidare alla signaler som de tar emot. Switchar kan både skicka och ta emot information via samma port utan några kollisioner.

I KNAPP-Movings anläggning används ett fältbussystem som är anslutet till Profinet. Detta med hjälp av en *proxymfunktion* som finns i gatewayen. Proxyn fungerar som en talesman för fältbussens enheter på ethernet. Detta ger flera fördelar. En av fördelarna med detta är att snabbheten som används från fältbussen kommer att kunna användas i Profinet.

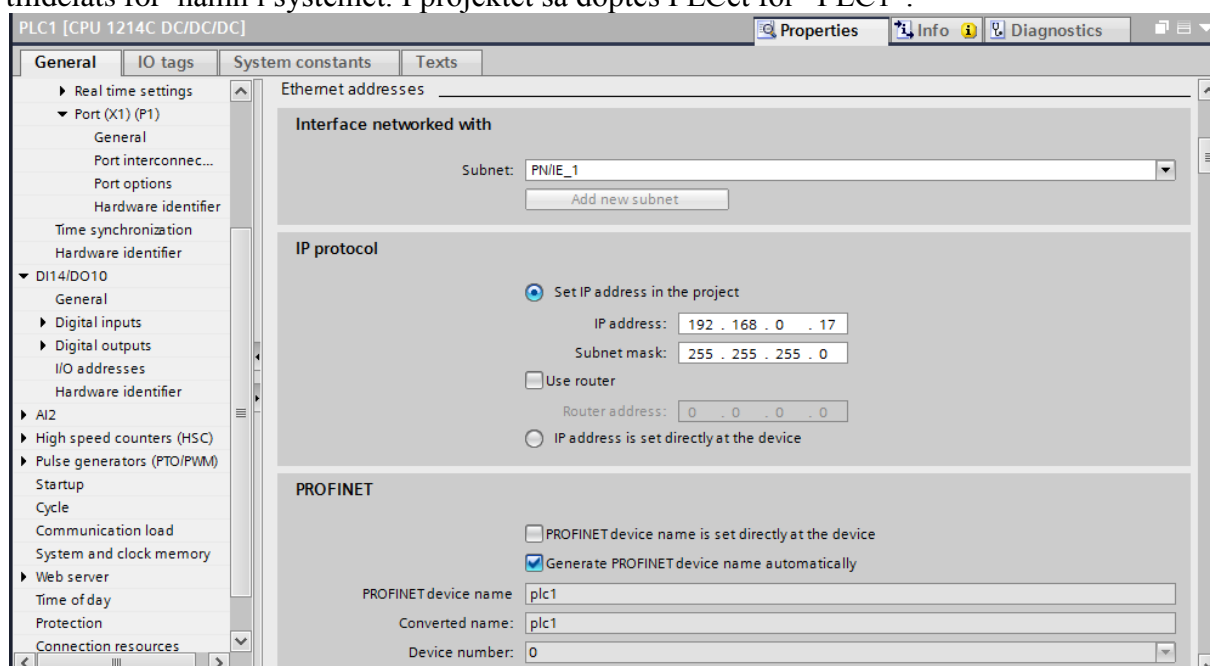
3 Utförande

3.1 Konfigurering av PLC

PLCet i anläggningen får sina in och utgångar från Profinet. Det är väldigt vanligt att företag använder fältbussar eftersom större anläggningar har ett större behov av in och utgångar än vad som finns fysiskt på PLC:et

Med TIA-portalen och Siemens PLC går det inte att koppla in ethernetsladden mellan dator och PLC och förvänta sig att de ska börja kommunicera med varandra. I Siemens måste ändringar göras i konfigurationen i själva TIA-portalen. I TIA-portalen måste användaren välja vilket PLC som används, vilken typ av PLC som används och ge PLCn en unik IP-adress. I vårt fall måste vi även berätta att PLCet får sina I/O ifrån en AS-i-gateway med Profinet.

I konfigurationen måste användaren ta fram en AS-i-gateway i programmet precis som användaren var tvungen att göra med PLCet. Profinet har även en speciell egenskap i TIA-portalen. När Profinet gör ett anrop av enheter i systemet så förväntar sig inte Profinet att ta emot en ip-adress som användaren har tilldelat utan efter vad den har tilldelats för namn i systemet. I projektet så döptes PLCet för "PLC1".



Figur 3.1: Skärmdump från TIA-portalen som visar PLC-konfigurationen bl.a av IP-adressen.

3.2 Programmering

Programmeringen utfördes i Siemens egna program Siemens TIA-portalen. Koden skrevs mestadels med funktionsblock men även datablock förekom. Programmeringsspråket som används i projektet är *Ladderdiagram*.

3.2.1 Mainprogrammet

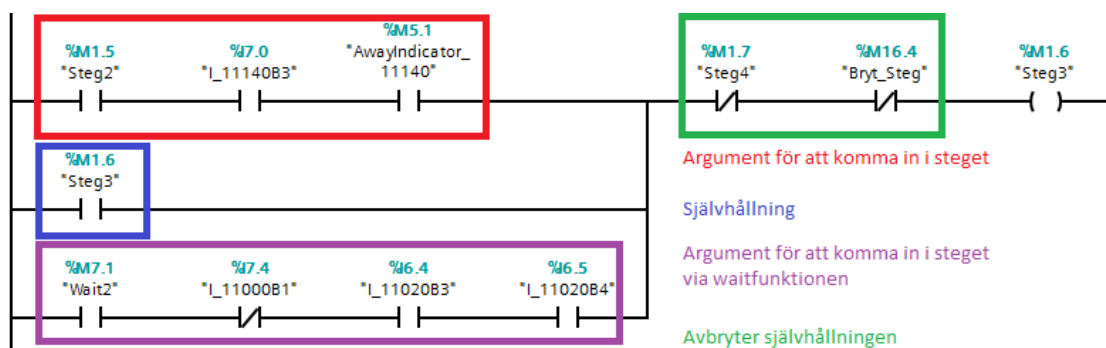
I varje program finns det en main metod, vilket är den första funktion som PLCet exekverar kod. PLC:et börjar alltid att läsa av koden uppifrån och arbetar sig neråt till sista raden i programmet. Mainmetoden är alltså högst upp i hierarkin och bestämmer när kod ska exekveras från andra funktioner.

3.2.2 Stegfunktion

Metoden som valdes för att programmera PLCet med är med hjälp av ladderdiagram. En "ladder" består av flera steg där varje steg fungerar som ett minne. När steget är aktiverat så kommer den utföra ett arbete som beskrivs av koden. Stegen är en cykel som går runt hela tiden det vill säga att när det första steget har arbetat klart så kommer programmet att gå vidare till det andra steget, när det andra är klart så går den vidare till det tredje steget. Till slut når den sitt sista steg och börjar därefter om igen. För att komma in i ett steg är ett av villkoren att det föregående steget är aktivt. Om detta uppfylls så aktiveras steget och börjar sin självhållning tills den bryts. Självhållning innebär att ett steg håller sig själv aktivt. Steget kommer att vara aktivt tills självhållningen bryts. Detta görs av nästa steg i cykeln.

Ett exempel på hur detta kan se ut:

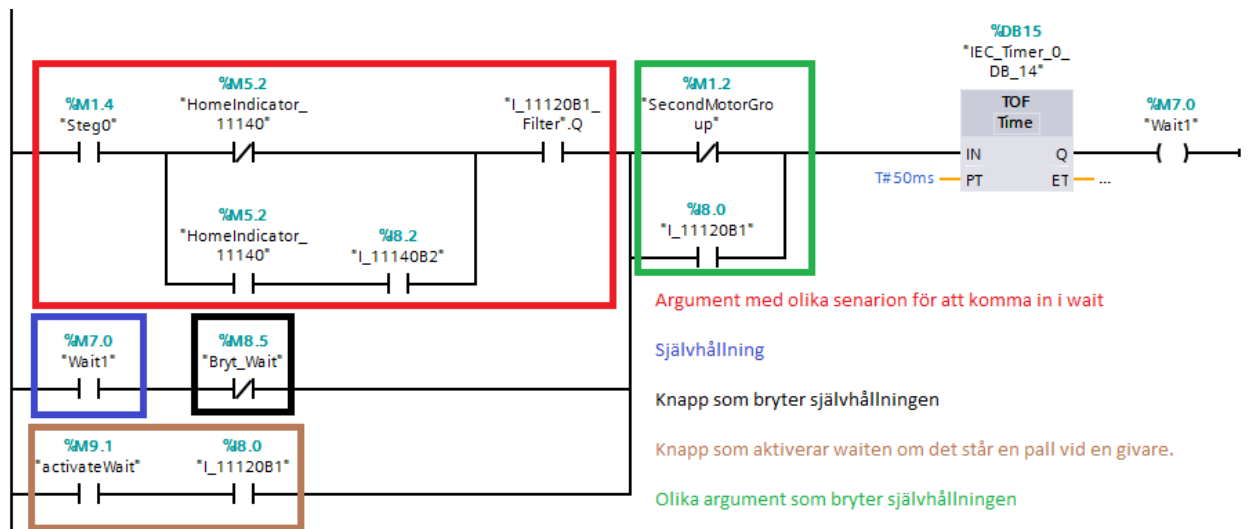
Steg 2 är aktiverat och PLCet vill utföra Steg 3. För att PLCet ska kunna utföra Steg 3 behöver Steg 2 att vara aktiverat och specifika givare ettställs, detta sker när pallen når givaren och vridbordet ställer in sig så den står över rätt givare vid denna position. När detta sker så går PLCet vidare till nästa steg och stegets självhållning startar. Steg 3 avbryts när Steg 4 eller när operatören väljer att stoppa driften. PLCet kan inte komma in i Steg 3 ifall Steg 4 är aktiverat.



Figur 3.2.2: Exempel på hur stegfunktionen har implementerats. Funktionen för koden beskrivs i exemplet ovan.

3.2.3 Waitfunktion

Waitfunktionen används när en pall kommer till slutet av ett steg och ska precis komma in i nästa steg. Då spanar systemet framåt och känner av ifall det är fritt från pallar. Är det upptaget, det vill säga att bandet redan är i bruk eller att det står en väntande pall, aktiveras waitfunktionen för att förhindra kollision. Waitfunktionen inväntar att det ska bli ledigt i nästa steg och när det blir ledigt tar stegfunktionen över igen och sätter in den väntande pallen i ett steg. Pallen skickas alltså vidare till nästa moment i transporten. Designen på waitfunktionen följer samma mönster som stegfunktionerna. Skillnaden är att waitfunktionen har en *off delay* som gör att den aktiverade waiten stängs av lite senare. Detta görs för att garantera att den skickas in i nästa steg.



Figur 3.2.3: Exempel på hur waitfunktionen har implementerats.

3.2.4 Bandfunktion

Bandfunktionernas uppgift är att köra motorerna så att bandet, kedjorna, vridbordet eller liften körs. Funktionen är konstant aktiv i mainfunktionen och den väntar på att rätt steg ska aktiveras. Då aktiverar bandfunktionen någon av utgångarna som styr en motor. När ett steg blir aktivt så aktiveras det samtidigt ett minne i bandfunktionen som även självhåller sig själv. Detta minne aktiverar de motorer som ska köras i detta steg. När steget är klart bryts minnet och motorerna slutar köra.

3.2.5 HMI-funktion

Den här funktionen är precis som bandfunktionen alltid aktiv och innehåller speciella minnen. Dessa minnen representerar en viss händelse som visas i *HMI*n. Ett exempel är när det ena vridbordets båda givare är aktiverade. Detta betyder det att det finns en pall på vridbordet och programmet kan använda dessa minnen i *HMI*n för att aktivera en visuell pall i panelen.

3.2.6 Filterfunktion

Filterfunktionen är en säkerhetsfunktion som är tillämpad för givarna så att det inte ska kunna ske ett speciellt givarfel. Alla givare har en filtertid det vill säga att det finns en timer i programmet som räknar så fort givaren blir aktiv. Tiden som startar i timern är väldigt liten den kan exempelvis vara femtio millisekunder. Ifall timern inte hinner räkna klart då utförs det inget på anläggningen som vanligtvis skulle ske om givaren gick igång. Detta används för att förhindra givarfel om exempelvis en fluga eller en hand skulle träffa givaren så att anläggningen inte körs igång av olyckshändelse.

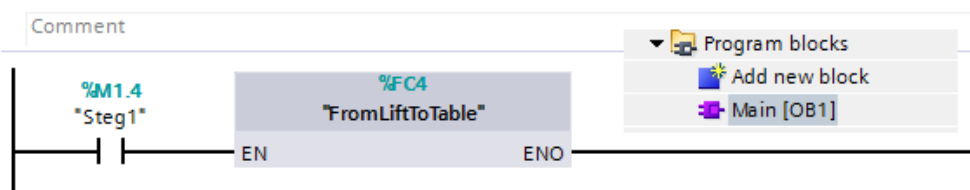
3.2.7 Tidmättningsfunktion

För att transportbandet ska vara säkert implementerades det en säkerhetsfunktion till utöver filtertiden. Säkerhetsfunktionen som implementerats är en tidmättningsfunktion. Funktionen finns där för att förhindra transportband att köra ifall en pall har fastnat någonstans på bandet. Funktionen fungerar så att ett band måste börja köra och då startar en timer med en angiven tid för att pallen ska kunna ta sig fram till en speciell plats på ett band. Om pallen inte har hunnit ta sig fram så kommer timern att utlösas och alla band att stoppas. Tiden det handlar om här varierar från fem sekunder upp till tjugo sekunder. Detta ändras beroende på hur långa sträckorna är.

3.3 Tillvägagångssätt för programmering

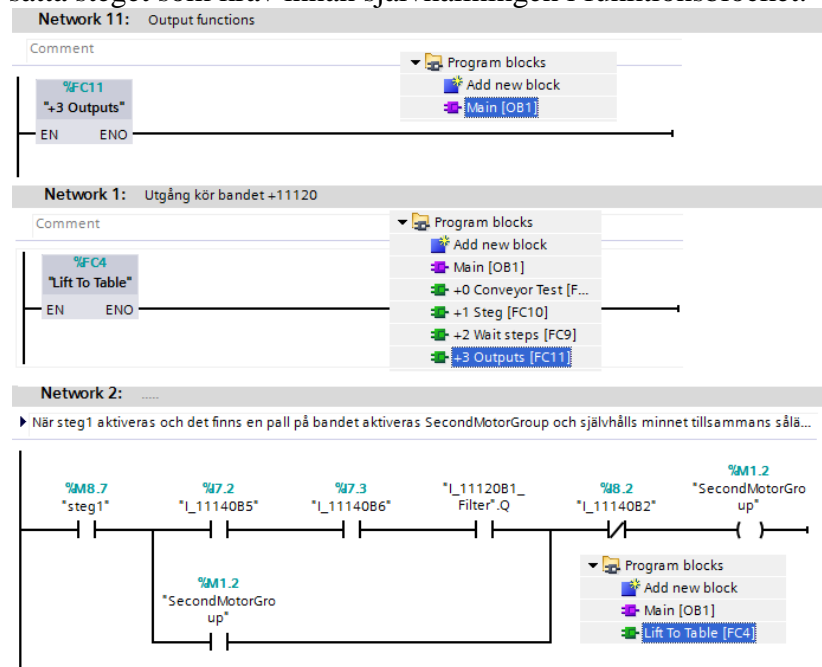
Programmeringsprocessen startades med att undersöka anläggningen och ta reda på hur transporten ska gå till, exempelvis vilken riktning banden rullar i eller vad olika komponenter på givare gör. Efter att planeringsstadiet avklarats skulle tillvägagångssättet för programmeringen fastställas. Strukturen som programmet programmerades med blev med hjälp av “stegfunktioner”.

Från början programmerades alla utgångar i funktionsblocken och mainblocket bestod av stegfunktionen. Utgångarna till motorerna blev aktiva när de stegen som styrde just den motorn var aktiv.



Figur 3.3: Ett äldre utdrag från koden där funktionsblocket “FromLiftToTable” endast får bli aktivt när “Steg1” aktiveras. På detta sätt så skapades det problem för motorerna.

Den här strukturen skapade en hel del problem. När ett funktionsblock bryts sparas alla minnen i det läget som det just var i innan bryningen aktiverades, det vill säga att ifall ett minne som styr en motor är aktivt när funktionsblocket avbryts så kommer motorn att fortsätta köra. Detta problem löstes genom att alltid ha funktionsblocket aktivt och sätta steget som krav innan självhållningen i funktionsblocket.



Figur 3.3: Ett uppdaterat utdrag från TIA-portalen där funktionsblocket “FromLiftToTable” alltid är aktivt och har istället steg1 som krav innan självhållningen.

För att kunna transportera fler än två pallar samtidigt på anläggningen behövdes det en typ av kommunikation mellan transportbanden för att undvika kollisioner. Då ett av målen var att kunna köra fler än två pallar på anläggningen uppstod ett problem med detta programmeringsätt. Problemet var att uppbyggnaden av programmet bestod av att två närliggande steg, exempelvis "Steg1" och "Steg2" aldrig kunde aktiveras samtidigt. Lösningen till detta problem blev att införa så kallade väntesteg som placerades ut vid alla givare. Väntestegets funktion är att blicka framåt i anläggningen alltså vid nästa sträcka och se ifall det är fritt att köra eller om nästa sträcka är upptagen av en annan pall. Väntesteget uppfattar om nästa sträcka är upptaget med hjälp av givaren som finns på den sträckan. Om givaren är aktiverad eller om nästa motor är igång då aktiveras väntesteget och pallen som egentligen skulle köra vidare får vänta. Pallen kommer att vara i vänteläget tills nästa sträcka är ledig igen.

3.4 Implementering av HMI

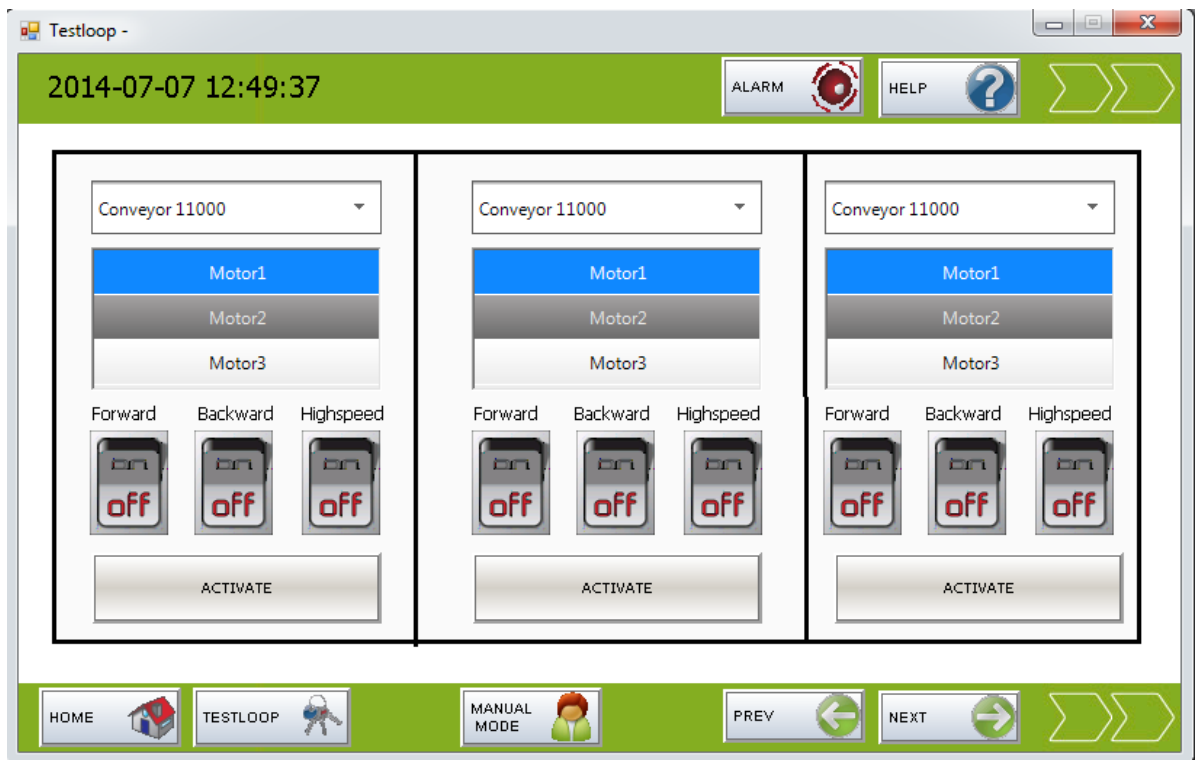
För att fullfölja implementeringen av HMIn så behövdes det programmeras i TIA-portalens. (Se HMI funktion). Implementeringen av HMIn utfördes i programmet iX Developer. Detta program användes för att själva displayen är från Beijer och iX är ett program från Beijer. Programmet är ett drag & drop program, ett program där det redan finns objekt i programmet så att användaren kan dra objektet och släppa inom displayens ramar.

Själva objektet kan ha flera olika egenskaper. Dessa egenskaper aktiveras med hjälp av att de ansluts ihop med en *tagg*, vilken importerades från TIA-portalens *tagglista*.

3.4.1 Manuell körning

HMIn behöver en funktion som gör att operatören själv kan styra och köra maskineriet manuellt. Under manuellkörning tar det flesta säkerhetsfunktioner bort och allt ligger i operatörens händer. Operatören bestämmer vilket band och vilken motor som ska köra och håller in activate-knappen så länge som operatören vill köra.

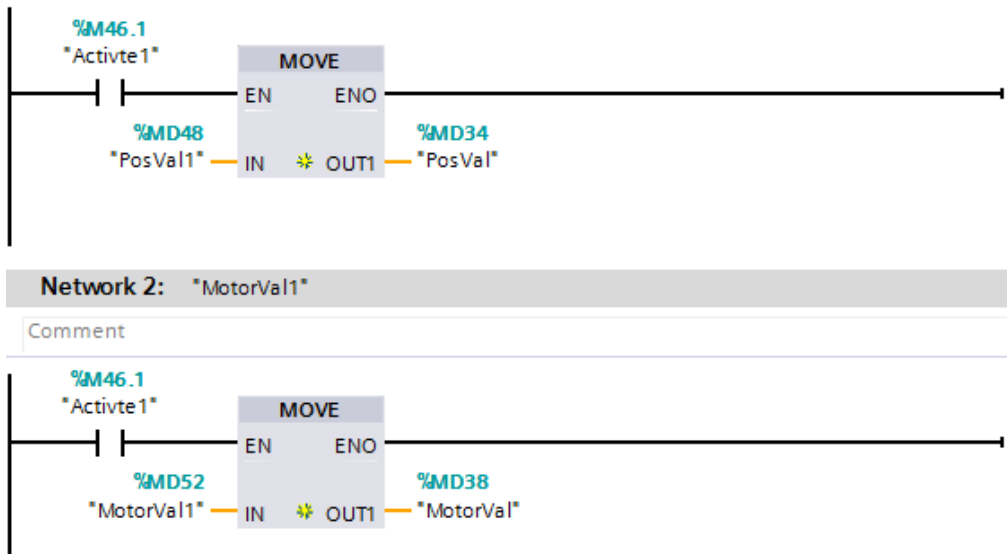
Skulle operatören välja en motor som inte existerar exempelvis motor tre på ett transportband som är kopplad till max två motorer då händer ingenting och operatören får välja en annan motor.



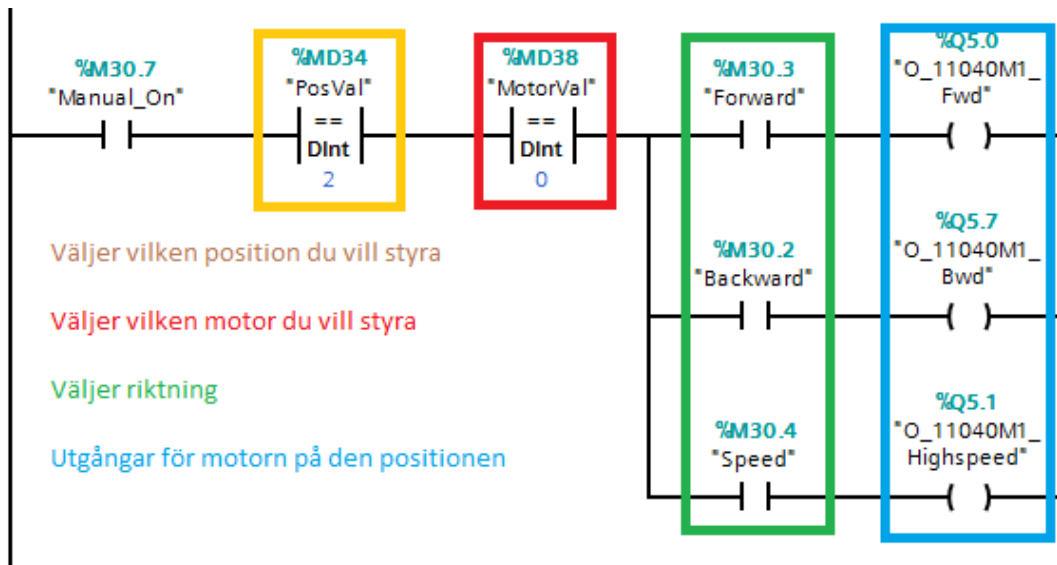
Figur 3.4: Sidan tillhör den manuella körningen i HMI:n och har tre kolumner med samma innehåll. Klickar operatören på dropboxen högst upp kan operatören välja transportband. Därefter väljs en specifik motor, en riktning och om den ska köra i höghastighet i den riktningen. Därefter trycker operatören in och håller inne "ACTIVATE" knappen för att drifisätta motorn. Operatören kan välja olika kombinationer i de olika kolumnerna.

Den manuella körningen har gjorts med hjälp av ett funktionsblock i TIA-portalen. Detta utförs av ett verktyg som heter "Move" i TIA-portalen. När "Move" funktionen blir aktiv tar den det värdet den har i inbenet och flyttar värdet till outbenet. "PosVal" och "MotorVal" är huvudvalen som är kopplade till alla utgångar. Varje kolumn i manuellt läge har egna taggar som skickar vidare sitt värde till "PosVal" eller "MotorVal" när knappen för den kolumnen är aktiverad. Detta gör att manuellkörningssidan kan ha flera val uppe som är samtidigt redo att köras med "Activate"-knappen.

"PosVal" taggen kan få åtta stycken olika värden, värdena varierar mellan heltalen noll och sju. Dessa värden är anslutna till vars sin specifika transportsträcka. "MotorVal" fungerar på samma sätt, fast värdena varierar endast mellan noll och två.



Figur 3.4: Move verktyget från TIA-portalen. "PosVal" tar emot "PosVal1" värde.



Figur 3.4: Varje band har sitt unika nummer i TIA-portalen. I detta kodavsnittet har band 11040 positionsnummer 2. Motorerna för på varje band fungerar på samma sätt.

3.5 Felsökning

Under tiden som projektet genomfördes uppstod det olika typer av problem som behövdes lösas innan projektet kunde gå vidare. Problem som uppstod var oftast hårdvarufel som exempelvis att komponenten som nödstoppet sitter på gick sönder och var tvungen ersättas med en ny komponent. När komponenter byts ut och ersätts i huvudskåpet behöver dessa oftast byta AS-i adresser eller tilldela den en ny adress då den nya komponenten oftast inte har en adress. Komponenterna behöver även integreras i anläggningen. Detta görs då med hjälp av en snabbinställning på gatewayen som gör detta själv. Denna snabbinställning gör följande sak, gatewayen gör ett anrop i anläggningen som hämtar och lagrar alla adresser den kan få tag i från AS-in och sparar dessa adresser.

Andra hårdvarufel som uppstod är att PLCn gav ut ett fel som "Error in lower level component", detta fel uppstod i TIA-portalen. Detta fel löstes genom att ändra portingång på gatewayen genom att stoppa in ethernetsladden i den fria porten. Detta fel uppstod vid flertalet tillfällen

Andra fel som uppstod under projektets gång var när motorerna skulle effektberäknas. Motorernas strömmar skulle mätas med hjälp av en strömtång. Till en början skulle detta göras med hjälp av ett mer avancerat mätverktyg som mätte storheter och därefter visas mätresultatet på den grafiska displayen. Displayen visar hur den uppmätta storhetens våg ser ut. Storheten kan exempelvis vara strömmens eller spännings RMS värde. Detta mätverktyg var som ett litet oscilloskop. Problemet som uppstod var att detta mätverktyg visade inte ett korrekt värde av strömmen utan ett mycket litet resultat på 0.3 mA, vilket är högst osannolikt på grund av att en motor som denna och av denna storlek ska driva ett stort rullband med en tung last behöver ha en mycket större ström. Detta löstes genom att skaffa en enklare strömtång som direkt visar ett resultat på strömmen. Med detta verktyg uppmättes ett mer korrekt värde som kommer att redovisas i resultatet.

3.6 Effektprovet

För att genomföra effektberäkningen på motorerna behövs en del uppgifter, som till exempel, $\cos \rho$ (där ρ är fasvinkeln), spänningen över motorn, strömmen i en av motorns inkommande faser under drift och om motorn är D-kopplad eller Y-kopplad. Majoriteten av dessa storheter går att ta reda på från motorskylten som finns på motorn. Den enda storheten för oss som var okänd var strömmen och den behövdes mätas med hjälp av en strömtång. Detta mättes i en av motorns inkommande faser när motorn var i drift. Strömtången mätte *RMS-värdet* för strömmen och under tiden motorn var i drift växlade RMS-värdet kraftigt. För att förenkla det uppmätta resultatet, skrevs alla uppmätta värden ner och ett medelvärde togs fram av dessa värden för att få fram ett bättre resultat.

Effektprovet bestod av två delar, ett tomgångsprov och ett maxlastprov. I tomgångsprovet så körde rullbandet och kedjorna så mättes strömmen genom att inga pallar körde på anläggningen eller att en tom pall kördes. Resultatet blev likvärdigt. I maxlastprovet så lyfts det upp en tungt belastad pall på 1350 kg och körde runt på anläggningen medan mätningen av ström pågick.

Kända storheter var huvudspänningen $U_h = 400 \text{ V}$ och $\cos \varphi = 0,99$.

Den aktiva effekten för ett symmetriskt belastat trefassystem ges av:

$$\begin{aligned} &\text{Effektformeln} \\ P &= \sqrt{3} * U_h * I_f * \cos \varphi \end{aligned}$$

4 Resultat

Målen för detta examensarbete har uppnåtts genom att anläggningen har ett program med säkerhetsfunktioner och att programmeringen utfördes på ett sätt som gör att anläggningen kan köra flera pallar. Max antalet pallar är sex stycken, skulle sju pallar köras så skulle transportbandet hamna i *moment 22*. HMI:n blev även färdigprogrammerad och går att använda.

Vi har skrivit ett välfungerande PLC-program som kör loopen på ett felfritt sätt som dessutom har möjligheter att utökas ifall fler segment i loopen skall införas. Detta tack vare den strukturen som vi har valt att använda i vår kod (se Programmering).

Resultatet av HMI:n blev även väldigt lyckat då vi lyckades med att skapa en likvärdig HMI. Det finns tyvärr ingen jämförelse med den tidigare HMI:n som KNAPP-Moving ägde, deras kod sparades aldrig och vi dokumenterade aldrig hur deras HMI såg ut.

Dokumentationen som utfördes, var enligt KNAPP-Movings standard. Ritningarna som ritades och listorna som skrevs gjordes enligt deras mallar.

Effektresultatet syns i nedanstående tabell där en standardmotor sattes på prov under maxlastprovet för att jämföra resultaten jämfört med tre smarta motorer. Två av de smarta motorerna som heter Leuze är av samma tillverkare men en motor driver ett band och den andra driver en kedja. Den tredje smarta motorn är av SEW Eurodrive vilken kör ett band. Den gamla motorn är av SEW Eurodrive som också körde ett band. Resultaten av effektberäkningarna visas i Tabell 4.1.

Motor	Leuze Smart (band)	SEW Eurodrive Smart (band)	Leuze Smart (kedja)	SEW Eurodrive Standard motor (band)
Maxlastprov (W)	142,0	489,0	713,4,0	1 016,5
Tomgångsprov (W)	48,0	61,7	102,8	-

Tabell 4.1: Värdena är tagna från bilagan "Effektmedelvärde". Beräkningarna som utförts för att få fram resultaten finns att se på nästa figur.

Beräkningarna som utförts för att få fram den aktiva effekten vid tomgång och maxlastprovet kommer på figuren nedan.

Vid maxlastprovet så varierade det uppmätta strömvärdet i stor omfattning. Därav blev metoden att ta fram ett medelvärde på strömmen. Vid tomgångsprovet var den uppmätta strömvärdet konstant och därav behövdes inget strömmedelvärde beräknas. Ett mätresultat kunde inte plockas fram för den vanliga motorn på grund av svårigheter att koppla runt en av motorernas faser.

Motor	11000-M1 Smart Motor Leuze	11060-M1 Smart Motor SEW Eurodrive	11080-M1 Smart Motor Leuze Chain	11120-M1 SEW Eurodrive Standard Motor Efeq
Mätvärde (A)	0,15	0,31	0,42	0,43
(alla värden som strömlången växlade mellan när motorn körde med tung last)	0,18	0,6	0,77	0,71
	0,16	0,78	1,06	1,33
	0,1	0,76	1,09	1,27
Maxlast 1350 kg	0,29	0,83	1,38	1,38
	0,33	0,78	1,25	2,34
	0,32	0,72	1,29	2,49
	0,26	0,78	1,16	1,82
	0,22	0,73	1,2	1,65
	0,31	0,79	1,19	1,92
	0,4	0,77	1,28	1,25
	0,14	0,82	1,09	1,43
	0,08	0,61	1,07	1,4
	0,17		1,03	1,67
	0,26		1,08	1,5
	0,31		0,31	1,31
	0,18			1,32
	0,13			1,29
	0,34			1,25
	0,31			1,37
	0,17			1,5
	0,1			1,35
	0,23			1,27
	0,13			1,28
	0,14			1,26
	0,19			1,29
	0,16			1,26
	0,15			1,42
	0,19			2,56
	0,12			2,6
				2,3
				1,33
				1,26
				1,28
				1,33
				1
Strömmedelvärde (A)	0,207333333	0,713846154	1,041875	1,483888889
Effekt (W)	142,03992	489,0417231	713,767725	1016,5826
Tongång				
Strömmedelvärde (A)	0,07	0,09	0,15	
Effekt (W)	47,9556	61,6572	102,762	

Figur 4.1: Skärmdump taget ifrån "Medelvärde watt". Den röda kolumnen visar strömvärdet och den aktiva effekten för den vanliga motorn. De övriga kolumnerna visar värdet för de smarta motorerna.

Resultatet av programmeringen blev en fullt fungerande testanläggning med både säkerhetsfunktioner och en HMI med egenskaper att styra anläggningen med även manuellt.

5 Slutsats

Examensarbetet gick ut på att lära sig om hur en ingenjör ser ut och utföra dessa arbetsuppgifter. Syftet med arbetet var även att undersöka ifall de smarta motorerna var effektivare än de gamla motorerna som KNAPP-Moving använder sig av.

En av målsättningarna var att göra fullständigt kretsschema och ellayout i KNAPP-Movings standard. Detta krav gällde även för apparatlistan och I/O-listan. Tidsmässigt gick dokumentationen av testloopen snabbare än beräknat, vilket gav mer utrymme till programmeringen

Skillnaden mellan vårt programmeringssätt och KNAPP-Movings programmering är att vi skapade programmet med hjälp av så kallade stegfunktioner och KNAPP-Movings programmeringsätt är att det är uppbyggt på nivåer. Ett exempel på deras nivåer är att andra funktionen byggs upp av den första precis som en stapel. För att den andra nivån ska fungera så måste första nivån vara igång. Om första nivån nu avaktiveras så rasar hela stapeln och stängs av stegvis. Tyvärr så har vi ingen tillgång till deras kod då deras äldre PLC är kodad i ett annat program.

Vi var så ivriga med att komma igång med vårt projekt av vi missade att kolla deras kodning och deras HMI. Hade vi gjort de hade vi kunnat jämföra vårt resultat med deras.

Däremot så stötte vi på flera problem under själva driftsättningen, exempelvis att kunna driftsätta två pallar på ett och samma band. Detta var mer komplicerat än vad vi tänkt oss, eftersom anläggningen hade behövt använda sig av fler minnen samt att anläggningen skulle behöva utföra sina moment med en väldigt hög precision.

Med detta menas att pallarna måste köra med ett speciellt avstånd mellan varandra. Ett exempel är att två pallar placerade efter varandra kör på en lång raksträcka. När den främre pallan rullat in på nästa band och för att bakre bandet ska utan några stopp så måste den bakre pallan befinna sig vid en givare. Då kan den bakre pallan fortsätta köra på transportbandet utan att påverkas av den främre pallan.

För att detta skulle kunna fungera behövs det fler givare på anläggningen. Då hade det kunnats skapa flera väntepositioner men även då hade det blivit svårt. Skulle den bakre pallan köra för tätt eller för långt bort så finns det en väldigt stor risk att den bakre pallan skulle missa en givare när den främre ställer sig i position som egentligen ska stänga av transportbandet. Detta skulle även vara mycket svårt ifall det skulle finnas flera givare på anläggningen. Inte äns KNAPP-Movings egna programmerare har löst detta problem, så vi valde att inte prioritera detta. Ett av programmeringsmålen var att kunna köra minst två pallar samtidigt på anläggningen. Detta var vi framgångsrika med då vårt program lyckades undvika kollisioner och hålla pallarna i ett vänteläge.

Under arbetets gång upptäckte vi saker som vi hade kunnat göra bättre. Vi hade kunnat filma deras HMI och jämföra med vår. Arbetet bestod också av väldigt många överraskande fel som exempelvis när nödstoppet gick sönder och vi var tvungna att byta ut det.

Effektprovet som utfördes på de smarta motorerna, Lenze smart motor och två stycken SEW Eurodrive Movigear motorer var också väldigt lyckade. Lenze motorn kör kedjan som har liften, och SEW eurodrive drar den andra kedjan och den andra SEW motorn kör ett rullband. Strömmarna som går igenom de smarta motorerna varierar beroende på vad för tyngd och vilken sorts sträcka dem kör. Vad som upptäcktes under mätningarna var att det var mer krävande för en motor att dra en kedja än ett rullband.

De smarta motorerna är betydligt mer effektsnåla än de vanliga motorerna. Om vi jämför de båda motorerna på bandtransportsträckorna så kan vi se att den smarta motorn spenderar 86 % mindre effekt, vilket syns på figur 4.1. Detta resultat svarar på frågan om det är värt att satsa på dessa motorer. Fast de smarta motorerna är dyrare så tjänar man in de pengarna i längden när det kommer till strömkostnaden.

Vi tror att vår mätning hade blivit ännu bättre om vi hade fått Fluken, (det mer avancerade mätverktyget) att fungera. Eftersom den har en display likt ett oscilloskop hade det varit lättare att få ut bättre värden. Metoden som vi använder var att använda

oss av en strömtång. När vi märkte hur mycket värdena varierade, började vi med att filma strömtången. Då blev det mycket lättare att se alla värdena som strömmen varierade mellan och vi kunde få fram ett medelvärde.

Sättet vi fick fram medelvärdet på var att skriva ner alla värden som visades under mätningen och sen dividera med antalet värden. Den lilla felmarginalen som uppstod berodde på att vissa värden höll i sig längre än andra. Detta medelvärde hade kunnat bli mer exakt om vi hade skrivit upp alla värdena för varenda bild och sen dividerat med hur många bilder hela mätningen var på.

Vi kom även fram till att anläggningen hade varit mer effektiv ifall det hade funnits fler givare. Hade det funnits fler givare hade det även funnits fler vänte positioner och man hade även kunnat konstatera när hela pallen hade lämnat ett visst band tidigare. Detta hade gjort att motorerna kunnats stänga av tidigare och att pallarna inte hade behövt ligga i wait lika länge. Eftersom motorerna även drar effekt när de kör på tomgång hade även detta sparat på pengar.

Vi undersökte även hur många pallar loopen kunde ha på en gång och vara som mest effektiv. Med tanke på de väntesteg och de givarna som finns, så fick vi detta värde till fem pallar. Även detta hade ökat ifall vi haft fler givare.

Under de sista veckorna hos Knapp ville de ha hjälp med lite extra undersökningar i sin anläggning. En av deras kunder använder sig av svarta plastpallar istället för vanliga träpallar. Både den svarta färgen och att det är väldigt mycket håll i pallen gjorde att givare ibland missade att registrera pallen. Just vid den positionen som givaren missade eller reagerade olika, så fanns det en matningsstation. För att pallen skulle kunna komma in i matningen var den tvungen att stanna på en position med mindre än två cm marginal. Vi gick uppgiften att prova diverse nya givare för att se ifall vi kunde hitta en givare som var bättre. Just på vår testanläggning fanns det en liknande matningsposition som vi kunde göra våra tester. Vi använde vårt nuvarande program för att köra runt pallen 50 gången och markerade var pallen stannade, hur många gånger givaren reagerade, d.v.s. om den höll signalen eller blinkade. Detta gjorde vi för de tre olika plastpallarna och för de olika givarna vi fått.

Vi fick fram ett bra resultat på en av givarna. Den hade en variation i stopp positionen på 0,7cm och givaren blev aktiverad 50 gånger, vilket räcker utmärkt i detta fall. Med detta kan vi säga att vår testloopsprogrammering uppfyller syftet.

Tillvägagångssättet vi hade för att dokumentera anläggningen antagligen hade det varit lättare att göra ifall testloopen var gjord efter dokumentationen och inte tvärt om. Eftersom alla sladdar redan var inkopplade och fastripsade i kabelstegen var det ibland svårt att följa dem. Även vissa givare och motorer att på stället som man var tvungen att ha verktyg för att öppna upp och se hur de var kopplade. Det hela gjorde det jobbigare att se vad som gick till vad och vi fick dubbelkolla för att vara på den säkra sidan.

6 Framtida utvecklingsmöjligheter

Det finns hur många olika utvecklingsmöjligheter som helst för denna testloop. Som tidigare nämnt så hade fler givare kunnat installeras för att få en ännu mer effektivare testloop. KNAPP-Moving hade även kunnat byta ut alla sina vanliga motorer till de smarta motorerna eftersom dessa har en lägre driftkostnad.

Eftersom den nya PLCns kod har en viss typ av struktur, se tidigare kapitel, så går det att återanvända samma kod ifall man vill lägga till liknande transportsträckor i loopen. Uppdatering i dokumentationen är också väldigt smidigt att göra eftersom numreringen av segmenten är gjorda med ett mellanrum. Då kan man lätt se vad som har tillkommit senare.

För att underlätta för driftsättningen hade man kunnat göra en mobil applikation med samma funktion som HMI:n.

7 Källor

7.1 Digitala källor

Bihl Wiedermann

<http://www.bihl-wiedemann.de/en/products.html>

Hämtat: 2014-06-18

Profibuskälla

http://www.profibus.se/PDF/PROFINET_200609.pdf

Hämtat: 2014-06-23

Givarbeskrivning

<http://www.pepperl-fuchs.com/global/en/2747.htm>

Hämtat: 2014-06-30

AS-i

<http://w3.siemens.com/mcms/industrial-controls/en/industrial-communication/as-interface/asi-power24/Pages/tabcard.aspx>

Hämtat: 2014-06-23

Smarta motorer

<http://www.lenze.com/en-se/products/motors/three-phase-ac-motors/lenze-smart-motor/>

Hämtat: 2014-07-07

<http://www.sew-eurodrive.se/produkt/movigear-dac.htm>

Hämtat: 2014-07-07

7.2 Litterär källa

A.Alfredsson & R.K. Rajput, (2009). *Elkretsteori*. Liber

8 Bilaga: HMI-manual

Startsida



Detta är HMIns startsida. När kan man navigera sig till del olika alternativen genom att trycka på diverse knappar. Alla sidor har samma valfönster.

Home: Kommer till startsidan

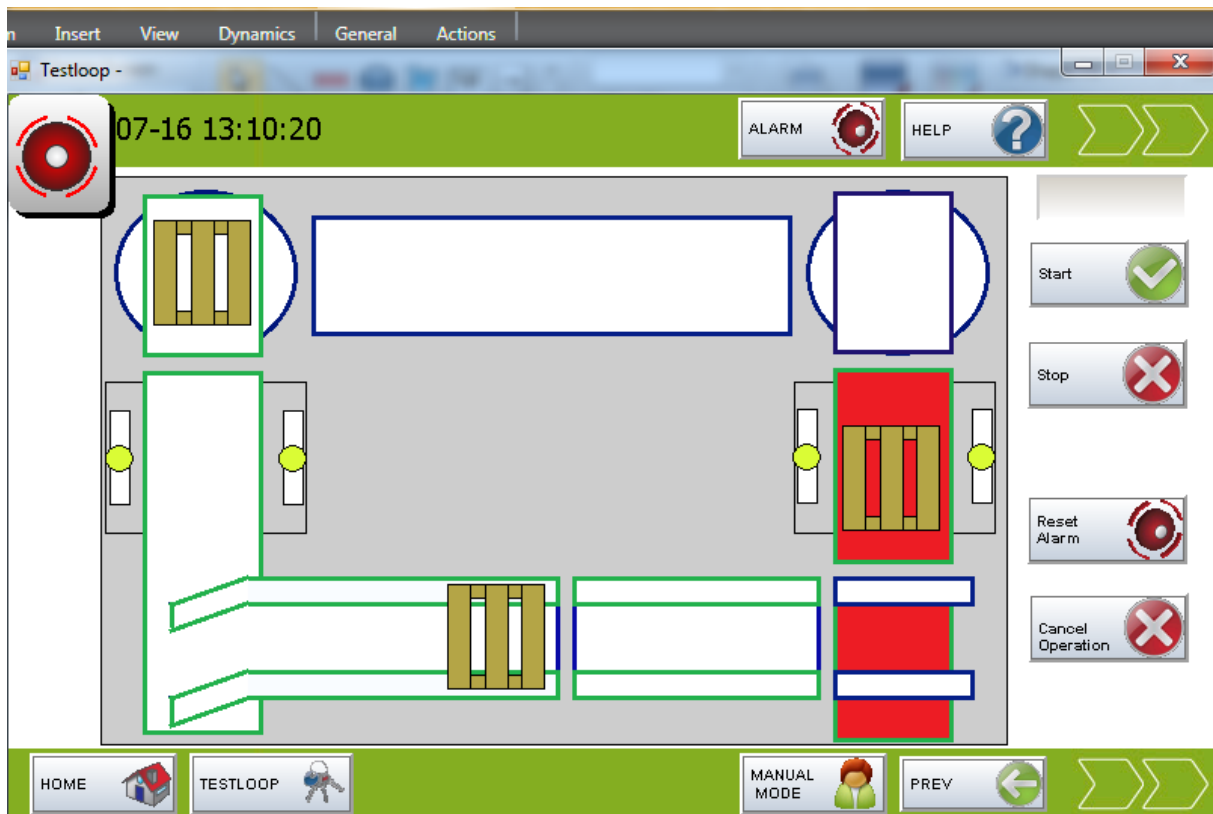
Testloop: Växlar till sidan för att köra anläggningen automatiskt.

Manual mode: Sidan där man kan manuellt köra anläggningen.

Prev: Går till föregående sida

Alarm: Går in på larmsidan där man kan kolla larm.

Testloops-/automatsida



Detta är huvudskärmen för testloopen. Här kan man välja att köra anläggningen.

Start: Startar den automatiska körningen

Stop: Stoppar körningen

Reset alarm: Återställer aktiva alarm

Cancel operation: Bryter alla steg och väntesteg. Detta gör att all information som anläggningen har försvinner. För att starta anläggningen igen måste pallarna ligga på givare.

När det finns en pall på bandet visualiseras det på HMI:n

När en ram har en blå färg betyder det att bandet inte är i bruk.

En grön färg betyder att bandet är i bruk och motorn är aktiv.

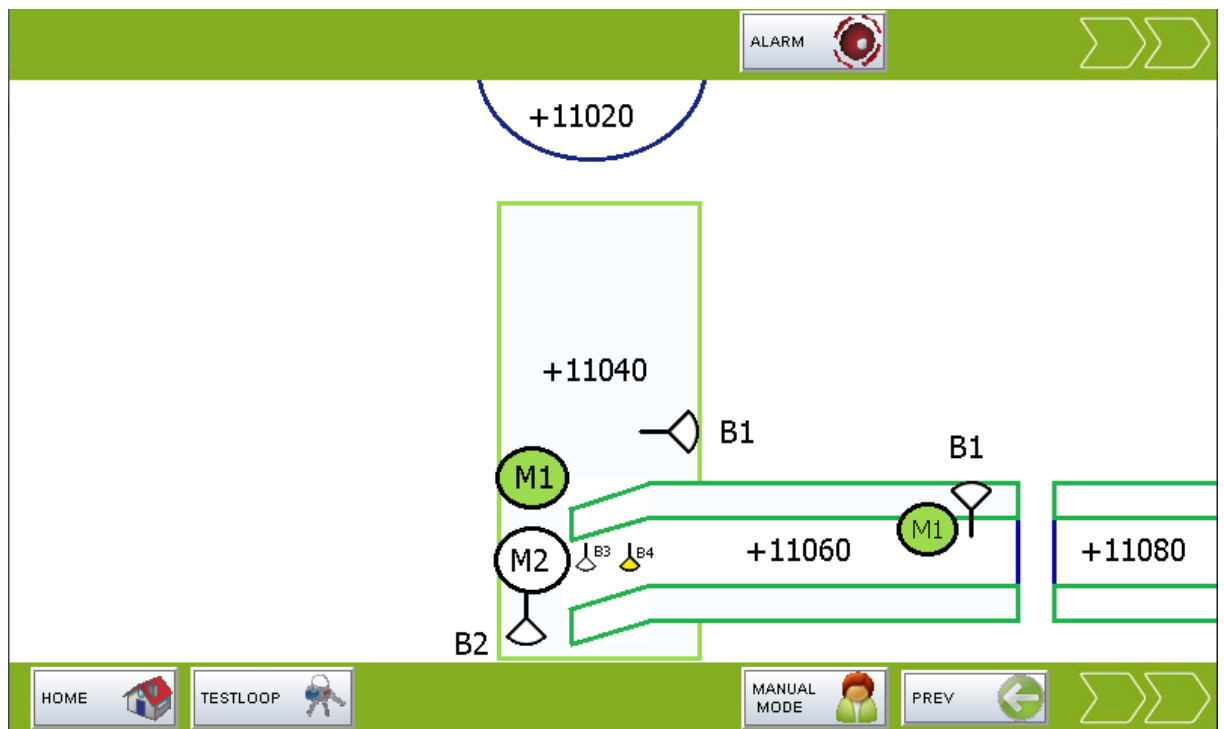
Ifall innerfärgen blinkar rött betyder det att ett larm har utlöst. Detta stoppar alla motorer i anläggningen.

Tryck på ett band för att få en mer detaljerad inzoomning av bandet.

Tryck på "HELP" för att få upp ett popup-fönster för att få hjälp i HMI-panelen.

Tryck på fyrkanten med ljusbommarna för att ta få komponentinformation om ljusbommarna.

Inzoomnings-/detaljsida

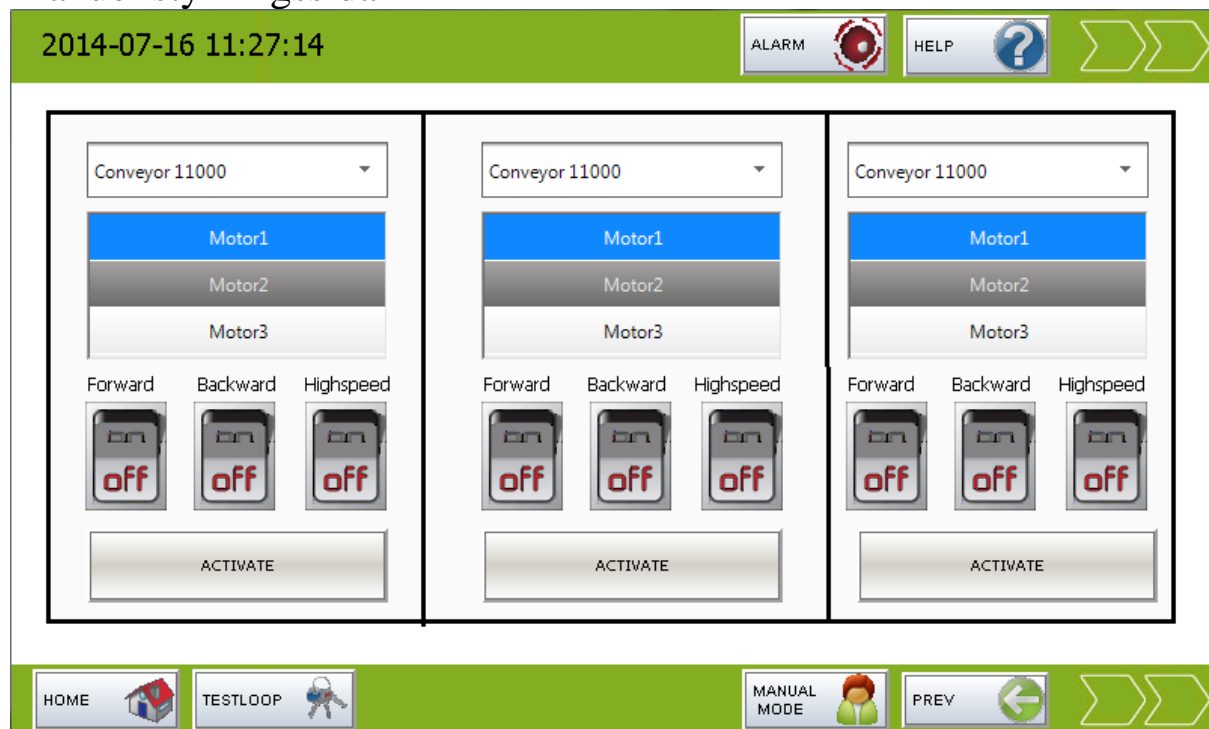


I den detaljerade skärmen får man en bild på vilka komponenter som finns på varje band.

Symbolerna med M symboliserar motorerna. Om fyllningsfärgen är:
 GRÖN: Så betyder det att motorn är aktiv och driver ett band, kedj, lift eller vridbord.
 VIT: Så är motorn inaktiv och inväntar körning.
 Givarna markeras med bokstaven B, om givaren är
 GUL: Så är givaren aktiv, vilket innebär antingen står där en pall på givaren eller så representerar den om liften är nere/uppe.

För att ta sig till ett närliggande band går det bra att trycka på figuren med texten i. Då skickas du vidare till den transportören med dess egna komponenter.

Manuellstyrningssida



Ifall man trycker på "MANUAL MODE" knappen så skickas man vidare till skärmen där man kan styra anläggningen manuellt. Så fort "Manual mode" trycks så släpper alla befintliga funktioner och informationen om pallarna som står där försvinner.

Automatiken försvinner.

I manuell drift kan man välja upp till tre olika motorer med olika riktningar för att köra. Däremot går det inte att köra alla tre motorer samtidigt.

För att välja transportör så tryck på dropboxen där "Conveyor 11000" står som alternativ, detta är automatiskt första alternativet. Tryck och välj en annan transportör.

Vissa av transportörerna har fler än en motor att välja mellan. För att välja rätt motor så ändrar du genom att trycka på en av de andra motorerna. "Motor2" eller "Motor3".

Observera att "Motor1" alltid är förstahandsvalet. Om du skulle välja ett transportband som bara har en motor och trycker på "motor2" eller "motor3" så händer det ingenting om du väljer att driftsätta motorn.

Välj antingen:

"Forward" för att köra motorn i rätt riktning.

"Backward" för att köra motorn i backriktning.

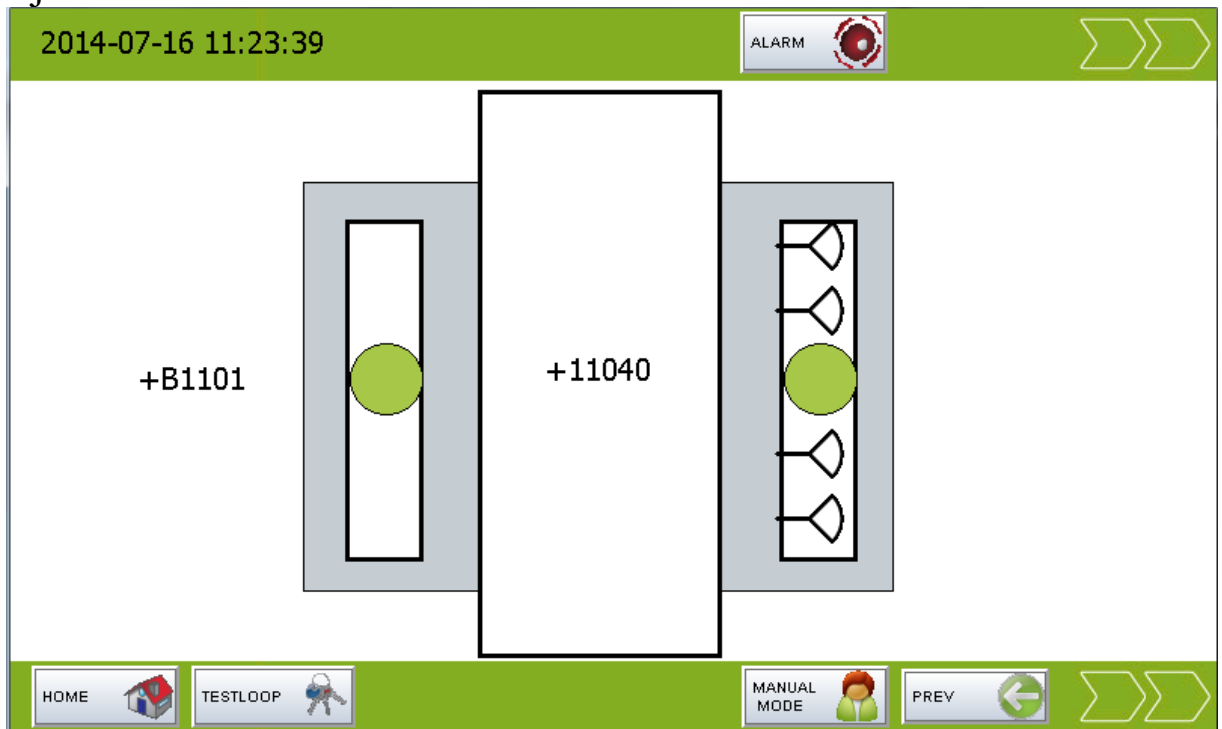
"Highspeed" för att aktivera körning i hög hastighet.

När en av riktningarna/hastigheten är vald då kommer knappen ändras till "ON" med en grön text.

Tryck på "ACTIVATE" för att driftsätta motorerna.

Ifall "HELP" knappen trycks så kommer en popup med en hjälptext om hur man manövrerar i manuell styrning.

Ljusbomssida



Om knappen med ljusbommarna trycks i visas denna skärmen. I skärmen så ser vi vilket band den står på och hur sensorerna är uppställda.

Om givarna är:

VITA: Då är givarna avaktiverade och ingenting kör förbi givarna.

GULA: Då är givarna aktiverade och då kör pallar förbi givarna.

Tryck på transportbandet för att ta dig in i den inzoomade bilden av "11040" transportband.

Larmsida

State	Active Time	Text
Active	2014-07-16 13:53:32	Pallen har inte hunnit fram till givare i tid. +1110
Acknowledge	2014-07-16 13:53:32	Pallen har inte hunnit fram till givare i tid. +1108
Inactive	2014-07-16 13:52:04	Pallen har inte hunnit fram till givare i tid. +1110
Normal	2014-07-16 13:52:04	Pallen har inte hunnit fram till givare i tid. +1112
Normal	2014-07-16 13:10:18	Pallen har inte hunnit lyfta i tid. +11100
Normal	2014-07-16 13:10:18	Pallen har inte hunnit fram till givare i tid. +1110
Normal	2014-07-16 13:10:18	Pallen har inte hunnit fram till givare i tid. +1112

Active: 1 Inactive: 1 Ack: 1 Normal: 4 [7 / 7]

När ett larm löst ut kan man trycka på Alarmknappen för att komma in på denna sida.
Här kan man se vilket larm som löst ut och vart de har löst ut.

Grön acknowledge: Ifall du har tyckt på Ack, dvs tryckt i att du har sett larmet.

Röd active. Detta är ett nytt larm som du inte har sett tidigare.

Gul inactive. Du har återställt larmet innan du har tryckt på Ack.

Vit normal: Tidigare larm som är ackade.

Tryck på clear för att rensa historiken.