

# Automatisering av skruvmontering



---

**Markus Persson**

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

# Automatisering av skruvmontering



**LUNDS  
UNIVERSITET**

Lunds Tekniska Högskola

**LTH Ingenjörshögskolan vid Campus Helsingborg  
Industriell elektroteknik och automation**

Examensarbete:  
Markus Persson

© Copyright Markus Persson

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  
Media-Tryck  
Biblioteksdirektionen  
Lunds universitet  
Lund 2013

## Sammanfattning

Målet med examensarbetet är att hitta och analysera olika lösningsmetoder som sen ska appliceras på ett industriellt problem. Problemet som kommer att granskas är montering av skruvar i en skiva. I lösningen används ett Pacdrive system som kommer att styra och synkronisera servomotorer under processen. Driften av motorerna kommer att ske med en elektrisk kam som kommer att programmeras i epas-4. I projektet kommer vi att jämföra olika metoder som att ha en elektrisk kam eller positionering som styrning och analyser av för- och nackdelar med de olika metoderna. Bland annat behandlas konstruktionsmöjligheter, synkronisering av motorer, styrning och positionering med servomotorer.

Nyckelord: Skruvmontering, Servomotorer, Pacdrive, Automatisering, Epas-4, Rörelsestyrning

## **Abstract**

The goal of the project is to identify and analyze different solution methods and to see how they should be applied to an industrial problem. The problem that will be examined is assembly of screws in a disc. The solution uses a Pacdrive system which is used to control and synchronize the servo motor. The operation of the motors will be done with an electronic cam that is completely virtual. In this project we compare different methods to have an electronic cam or positioning control and analyze the advantages and disadvantages of each method. Some issues that are discussed are design possibilities, synchronization of servo engines, steering and positioning with servo motors.

**Keywords:** Screw assembly, Servo engine, Pacdrive, automatization, Epas-4, Motion Control

## Förord

Detta är en kandidatuppsats på 22,5 högskolepoäng och är den avslutande delen för att ta ut examen på högskoleingenjörsutbildningen på programmet Elektro- och Automationsteknik vid Lunds Tekniska Högskola. Arbetet har utförts på Ecophon i Hyllinge.

Jag vill tacka Ingeman Lundstedt som var min handledare på Ecophon för all hjälp och vägledning han har gett.

Sen vill jag tacka alla på Ecophon för att gjort tiden under examensarbetet rolig och trivsamt.

Till sist vill jag tacka min examinator och lärare Mats Lilja för allt under min studietid.

# Innehållsförteckning

<b>1 Inledning</b> .....	<b>1</b>
<b>1.1 Bakgrund</b> .....	<b>1</b>
<b>1.2 Ecophon</b> .....	<b>1</b>
<b>1.3 Syfte och målsättningar</b> .....	<b>1</b>
<b>1.4 Problemformuleringar</b> .....	<b>2</b>
<b>1.5 Avgränsningar</b> .....	<b>2</b>
<b>2 Metod</b> .....	<b>3</b>
<b>2.1 Arbetsmetoder</b> .....	<b>3</b>
<b>2.2 Problemlösningar</b> .....	<b>3</b>
2.2.1 Drift och förflyttning .....	3
2.2.2 Synkronisering och motorval .....	4
<b>2.3 Källkritik</b> .....	<b>5</b>
<b>3 Teknik och Program</b> .....	<b>6</b>
<b>3.1 Materiallista</b> .....	<b>6</b>
<b>3.2 Teknisk bakgrund</b> .....	<b>6</b>
3.2.1 Pacdrive .....	6
3.2.1.1 <i>Kontrollenhet C400</i> .....	6
3.2.1.2 <i>Sercos</i> .....	6
3.2.1.3 <i>Servoförstärkare MC-4</i> .....	6
3.2.1.4 <i>Servomotor av typ SH</i> .....	6
<b>3.3 Epas-4</b> .....	<b>7</b>
3.3.1 Introduktion .....	7
3.3.2 POUs .....	7
3.3.2.1 <i>Funktioner</i> .....	7
3.3.2.2 <i>Funktionsblock</i> .....	7
3.3.2.3 <i>Program</i> .....	7
3.3.3 Visualisering.....	7
3.3.4 Resurser.....	7
3.3.5 Funktioner .....	8
3.3.5.1 <i>AxisModule</i> .....	8
3.3.5.2 <i>Axismodulecontroller och Modulcontroller</i> .....	8
3.3.5.3 <i>MultiCam</i> .....	9
3.3.6 Profiltyper .....	9
<b>4 Programmering</b> .....	<b>10</b>
<b>4.1 Struktur på programmet</b> .....	<b>10</b>
<b>4.2 Master-slav-struktur</b> .....	<b>10</b>
<b>4.3 Konfiguration och initiering</b> .....	<b>11</b>
4.3.1 PLC Konfiguration .....	11
4.3.2 AxisModuler .....	11
4.3.3 Övrig konfiguration .....	12

4.4	CMDT och CMDC .....	13
4.5	Manuell drift av Axlar .....	13
4.6	Manuell drift av Master med varmstart .....	14
4.7	Driftval .....	14
4.8	Återställning.....	14
4.9	Konfiguration av motorparametrar.....	14
4.10	Kampunkter.....	18
4.11	Kommunikation med operatörspanelen.....	18
4.12	Operatörspanel .....	18
5	Säkerhet.....	20
5.1	Riskbedömning.....	20
5.2	Säkerhetsfunktioner .....	20
5.2.1	Med eller utan inverter enable .....	20
5.2.2	Personsäkerhet.....	21
5.2.3	Maskinsäkerhet.....	21
6	Resultat.....	22
6.1	Konstruktion .....	22
6.2	Programmet .....	23
7	Slutsats.....	24
7.1	Framtid och utvecklingsmöjligheter .....	24
8	Terminologi .....	25
9	Referens .....	26
10	Bilagor .....	27
10.1	Elschema.....	27





# 1 Inledning

## 1.1 Bakgrund

Med en så snabb ökning av efterfrågan och stigande konkurrens sätts höga krav på industrin att automatisera processer. I dagens industri utförs ett konstant arbete med att utveckla och förbättra maskinerna, antingen för att öka produktionen eller att effektivisera processerna. När man utvecklar eller förbättrar en maskin är det en utmaning för att nästan varje maskin är unik, då det oftast inte finns färdiga lösningar utan man måste skraddarsy lösningar för varje specifik maskin. Där kommer ingenjörsyrket in för att analysera problemet och därefter skapa en lösningssidé som fungerar. I detta examensarbete undersöks metoder för att montera skruvar i en skiva.

## 1.2 Ecophon

Ecophon är en global leverantör av ljudabsorberande undertak och väggabsorbenter. Huvudkontor ligger Hyllinge och där utförs en stor del av all produktion. Det finns även mindre produktion i Danmark, Finland, Polen och Frankrike.

Ecophon AB skapades 1986 och ingår i Saint-Gobain koncernen som inriktar sig på produktion och distribution av funktionella material.

## 1.3 Syfte och målsättningar

Syftet med detta projekt är att undersöka och jämföra olika lösningar för att automatisera en process som ska utföra montering av skruvar i en skiva. Det finns många moment i processen som kan lösas på olika sätt, jag kommer att studera olika lösningsmetoder för de olika momenten i processen. Sen tillämpas resultaten för att automatisera processen på bästa sätt för att få en säker och effektiv process för att uppfylla de kriterier som finns.

Målsättningen för projektet kommer att vara att ta fram ett komplett förslag som innefattar:

- Hur maskinen ska vara konstruerad
- Material som krävs
- Fungerande program som kan köra processen
- Skapa ett HMI för maskinen

## 1.4 Problemformuleringar

Problemet i projektet är att montera skruvar i en skiva. Man ska kunna välja olika driftlägen där man monterar 4 eller 6 skruvar samtidigt i en skiva.

Det stora problemet är att skapa ett förslag på processen och sedan kan man lösa de andra problemen som är:

- Lämplig konstruktion på maskinen som kan utföra processen effektivt och säkert.
- Hitta lämpliga komponenter.
- Skapa ett effektivt och säkert program där man kan välja olika driftlägen och styra maskinen på ett kontrollerat sätt.
- Skapa ett HMI som är enkelt att använda och har allt som behövs för att köra maskinen.

På punkterna ovan ska alla lösningar vara utförda på ett sätt som liknar andra maskiner på Ecophon.

## 1.5 Avgränsningar

Avgränsningarna som finns i projektet är att en stor del av processen kommer att utföras för hand, som placeringen av skruvarna på verktygen och placering av skivan i maskinen. Inställning av verktygen för skruvmontering utförs manuellt vid formatomställning. Fixering av skivan sker med en luftcylinder. Det kommer inte att finnas någon maskin där man kan testa och provköra lösningen utan all provkörning kommer att simuleras. Projektet är väldigt stort och det finns många områden som tas upp. Jag kommer att lägga fokus på programmering och att skapa en bra lösning. Områden som säkerhet, mekanisk konstruktion och elkonstruktion kommer bara att kort sammanfattas.

## 2 Metod

### 2.1 Arbetsmetoder

Det första steget kommer att vara insamling av information som kan användas till lösningsidéer på problemen. Nästa steg kommer att vara att analysera de olika lösningsmetoderna och se vilken som kommer att lösa problemet bäst. Sista steget är att börja arbeta på lösningen. Under processen så kan det upptäckas fler problem som måste lösas och då börjar man om med första steget.

Fig.2.1 Gantt-schema för projektet, svarta x är vad som är planerat att göra och röda vad jag gjort

Aktivitet	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Intro/Analys	xx														
Undersökning av lösningside	x	xx	x	x											
Material planering		x		x	x										
Programmering till processen		x	x	x	xx	xx	xx	x							
Programmering till interface			x		x	x	x	x	x	x					
Rapport		x	x	x	x	x	x	x	x	x	x	x	x	x	x

### 2.2 Problemlösningar

Problemställningen är drivning av skruvdragare och förflyttning i höjddled samt synkronisering av dessa.

#### 2.2.1 Drift och förflyttning

För att utföra förflyttningen i höjd så fanns det två metoder.

Den första är att utnyttja rotationen från inskrivningen.

Den andra är att använda en separat motor för förflyttningen i höjddled.

För att utnyttja rotationen så skulle motorn vara kopplad till en gängad axel som har samma gängstigning som skruven, så att förflyttning i höjd är synkroniserad med rotationen. Motorn skulle fästas så den inte kan rotera fritt.

På den gängade axeln så finns ett fäste där man placerar skruven.

Man behöver en solenoid som kan flytta verktyget upp och ner för när skruven är i slutläget i skivan måste verktyget frigöras från skruven innan man kör till startpunkten.

En fråga var att välja vilken typ av motor man skulle använda sig av, pneumatisk eller elmotor.

Fördelar med en pneumatisk motor jämfört med en elmotor:

- Liten och låg vikt
- Lågt underhåll
- Säker om den fastnar eller den blir överbelastad
- Klarar av hård miljö
- Pris
- Genererar ingen värme

Nackdelen med en pneumatisk motor är att den drar mer effekt än en vanlig motor. Det är kompressorn som står för största delen av effektförlusterna. Vid starten av motorn så kan man inte få en mjuk start så skruven riskerar lossna från verktyget.

Den andra lösningen är att låta en annan motor göra förflyttningen, man behöver inte ha en solenoid som frigör verktygen. Men då kommer ett nytt problem att synkronisera motorerna så de jobbar ihop. Om man inte synkroniserar motorerna så kommer man skada produkten eller verktyget på maskinen.

Det är för stor risk med rotationsmetoden så förflyttningen kommer att ske med en separat motor.

### 2.2.2 Synkronisering och motorval

Vid motorval så fanns det två möjligheter, stegmotor eller servomotor. Val av motor är en viktig bit men man måste först se hur man ska synkronisera motorerna.

En lösningsidé för att synkronisera motorerna är att ha en mätgivare som känner av avståndet mellan plattan med skruvarna och skivan.

Sen kan man räkna ut hastigheten på förflyttningen i höjd och med det kan man räkna ut hur snabbt den ska rotera. Där hade man kunnat använda sig av stegmotorer för inskrivningen. Man hade troligen behövt en stegmotor med minst 300 helsteg per varv för att få en låg stegvinkel. Det stora problemet är att det krävs väldigt mycket programmering för att det ska fungera. Det blir ett stort program och rätt komplext, det brukar leda till att man minskar pålitligheten. Mätgivaren måste vara exakt och ha hög tillförlighet.

Kommunikationen mellan mätgivaren, programmet och motorerna måste vara snabb och pålitlig så all information blir skickad och mottagen korrekt.

En annan metod för att synkronisera motorerna är att använda ett servosystem som styrsystem för rörelsekontroll. Fördelarna med det är enklare programmering, stor noggrannhet och hög dynamik.

Vid val av styrning och synkronisering så finns det två val.

Antingen så kör man mastern i positioneringsmode med skruvdragare och höjdjustering som slavar till mastern.

Alternativt så kör man mastern i ändlös körning där en submaster är kamstyrd och där skruvdragarna och höjjustering är slavar mot submastern. En fördel med kamstyrning är att om programmet får stoppsignal eller nödstopp så kan man starta där stoppet kom i kamkurvan med hjälp av varmstart och även återställa maskinen till startposition med hjälp av kallstart. I positioneringsmode så måste man börja om då det inte finns någon varmstart.

### **2.3 Källkritik**

Under examensarbete så har det krävts information för att kunna göra lösningsförslagen och lösa de olika problemen.

De två informationstyper som jag har använt i detta projekt är fakta/kurs litteratur för att läsa upp högre förståelse inom vissa områden.

Där behöver man inte vara så kritisk till vad man läser. Det man ska tänka på är när litteraturen släpptes så det man läser fortfarande är aktuellt.

Den andra informationstypen är produktbeskrivningar och lösningsförslag från olika företag. Det skadar inte att vara källkritisk men företagen tjänar inte något på att skriva missledande information, men det kan förekomma slarvfel och att de har formulerat sig konstigt.

## 3 Teknik och Program

### 3.1 Materiallista

Elau Pacdrive C400

Operatörspanel Magelis XBTGT 5330

Pacdrive servo förstärkare typ MC-4

Pacdrive servomotorer typ SH-055 för skruvdragarna

Pacdrive servomotorer typ SH för höjdjustering

### 3.2 Teknisk bakgrund

#### 3.2.1 Pacdrive

Pacdrive är ett system som integrerar PLC och rörelsekontroll för servomotorer samt robotiserad kinematik i mjukvaran. Man kan synkronisera motorer, koordinera och generera positionsfunktioner till servomotorena. Man kan övervaka och kontrollera prestanda och man kan leverera MES data.

##### 3.2.1.1 Kontrollenhet C400

C400 är kontrollenheten som är mikroprocessorbaserad med VxWORK realtidsoperativsystem. Kontrollenheten kan synkronisera, styra och skapa rörelsefunktioner för maximalt 16 axlar.

Kontrollenheten är kompatibel till många HMI och klarar att köra de flesta HMI uppgifter.

Fältsbussar som C400 enheten kan kommunicera med är PROFIBUS DP, Can, Canopen, DeviceNet och Ethernet/IP.

##### 3.2.1.2 Sercos

Sercos är en realtidsbuss som används till att förbinda rörelsestyrning, sensorer, ställdon för numeriskt styrda maskiner och I/Os. Sercos är designat för seriell höghastighetskommunikation och finns i tre versioner.

Sercos 1 och 2 har en ringtopologi med fiberoptiska kablar och i Sercos 3 så används Ethernet. Skillnaden mellan Sercos 1 och 2 är att ASIC hastigheten på Sercos 1 är 2/4Mbit/s och Sercos 2 är 2/4/8/16Mbit/s Sercos 3 kan man få hastigheter upp till 100Mbit/s. Nyare Sercos versioner är bakåtkompatibla.

##### 3.2.1.3 Servoförstärkare MC-4

Servoförstärkaren MC-4 är till för kraftförsörjningen av servomotorn och den sköter sista steget med mjukvara till servoregulatören. Identifiering och konfiguration av motorer (motorer av typ SH) sker automatiskt för det finns en elektronisk typskylt i motorn. Det krävs inte något användarprogram, process signal eller multi-turn avkodare.

##### 3.2.1.4 Servomotor av typ SH

En servomotor i SH serien jämfört med en vanlig servomotor så har SH motorn lågt internt tröghetsmoment och har hög överbelastningskapacitet. Det leder till att man kan få väldigt bra precision, dynamik och effekt på motorn.

## 3.3 Epas-4

### 3.3.1 Introduktion

Epas-4 är ett kraftfullt program för att bygga upp projekt för rörelsestyrning. I Epas-4 finns en mall för styrning av servomotorer som kan användas för att enkelt skapa ett projekt. Epas-4 är uppdelat i olika områden, det finns POU's, Datatyper, visualisering och resurser. En stor fördel är att man kan inkludera bibliotek.

I bibliotek så får man allt som behövs för en specifik funktion.

### 3.3.2 POU's

I POU's skapar man funktioner, funktions block och program. Man kan skriva i något av IEC standard programspråken enligt 61131-3 som är IL, ST, SFC, FBD och LD sen utöver det kan man skriva i CFC.

#### 3.3.2.1 Funktioner

En funktion gäller bara ett dataelement och returnerar bara ett värde. Man kan få fler element om man använder sig av en datastruct men man får bara ett värde av funktionen.

#### 3.3.2.2 Funktionsblock

I funktionsblock så kan man ha inmatning av många dataelement men inga värden kan returneras. Påminner om void funktioner som finns i språk som java.

#### 3.3.2.3 Program

I program kan man ha inmatning av många dataelement och kan returnera många värden under drift och alla värden från förra programvarvet sparas. Ett program kan anropa ett annat program eller ett funktionsblock men inte en funktion.

### 3.3.3 Visualisering

Visualisering är ett kraftfullt verktyg som finns i Epas-4 där kan man kan simulera servomotorerna och provköra motorerna. Man kan rita geometriska element offline och sen kan man ändra dess form online beroende på variabelvärden. Där finns en larmlista som man kan använda för att hitta fel i sin programmering. Man kan också simulera motorfel för att se hur programmet hanterar de.

### 3.3.4 Resurser

I resurser så konfigurerar och strukturerar man projektet. Det finns en larmkonfiguration där man kan klassa larm. I PLC-konfiguration så ställer man in allt om systemets hårdvara och mjukvara som in och utgångarnas koppling. Där finns även konfigurationen av Sercosinterfacen och logiska avkodare. I task konfiguration så kan man ställa in hur ofta man ska anropa de olika programmen man har skrivit i POU's, man kan optimera programkörningen så man inte slösar resurser på program som inte behöver så snabb exekvering. Det finns ett oscilloskop med 8 kanaler där kan man till



exempel se status på motordrifter. Bibliotek och variabellistor hanteras också i resurser.

### 3.3.5 Funktioner

Här följer en kort beskrivning av de vanligaste funktionerna som används i projektet.

Det finns färdiga moduler som man kan hämta från generella eller importerade bibliotek som ingår i projektet.

#### 3.3.5.1 AxisModule

Detta är ett funktionsblock vars uppgift är att rotera axeln på en motor.

Funktionsblocket kontrollerar start, stopp, alarm angående axeln, jogg, log, hastighet, position, acceleration, deceleration, styrningsläge på allt angående axeldrift.

Fig.3.1 Visar Axismodul.



#### 3.3.5.2 Axesmodulecontroller och Modulcontroller

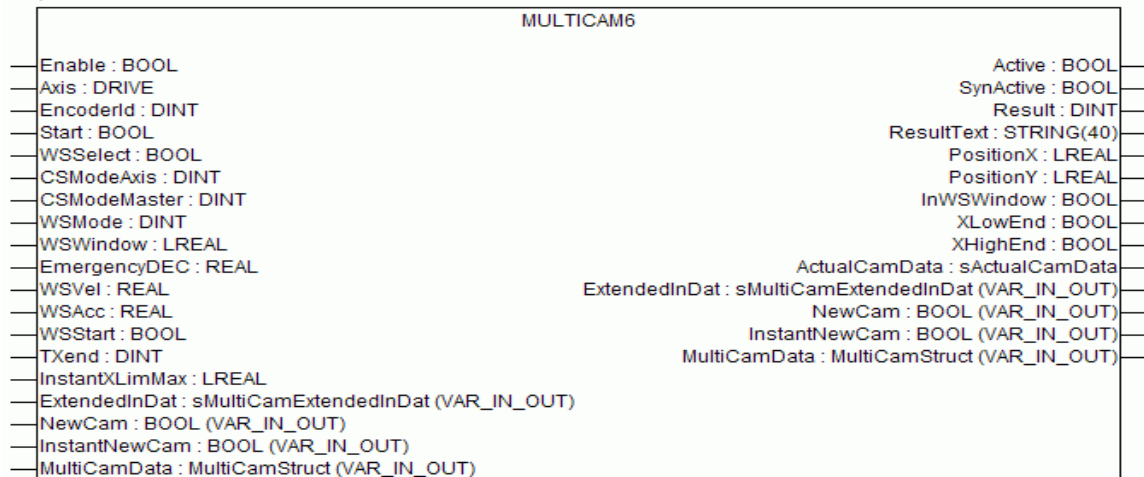
Modulcontroller är ett funktionsblock vars uppgift är att skicka kommandon till andra moduler. Axesmodulen har samma uppgift men skickar bara kommandon till axismoduler.

### 3.3.5.3 MultiCam

MultiCam funktionen är en elektrisk kam.

Där anges kampositioner, var den är i sin körning och hur den ska utföra varm och kallstart.

Fig.3.2 Visar en MultiCam6 modul



### 3.3.6 Profiltyper

Profiltyper är hur axeln kan köras.

Straight	Linjärt
Simplsin	Vanlig sinus
Inclisin	Lutande sinus
Modisin	Modifierad sinus
Modacctr	Trapetsformad acceleration
Poly5	Polynom av 5 graden
Parable2	Kvadratisk parabel
modisincom	Allmän modifierad sinus linje
modacctrcom	Allmän modifierad accelerations trapets
Harmocomb	Harmonisk kombination
Sinstraightcomb	Sinus-linjär kombination

De intressanta profilerna är straight och poly5.

Med straight så kör den med konstant hastighet, i en kamkurva så hade den rört sig linjärt mellan två punkter.

Med poly5 så räknar den automatiskt ut kurvan mellan två punkter med ett polynom av 5:e graden.

## 4 Programmering

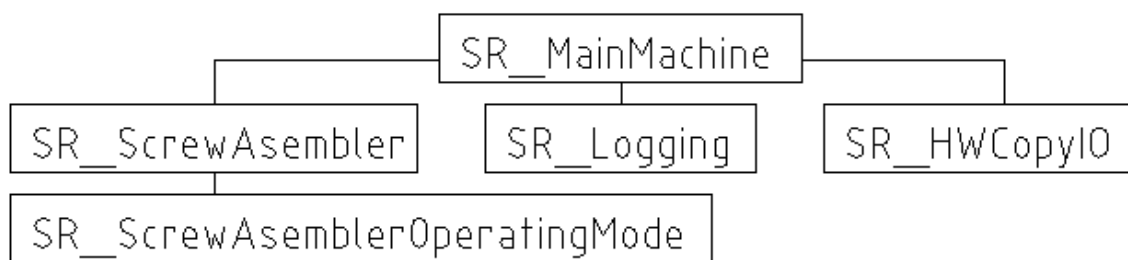
### 4.1 Struktur på programmet

Som visas i figur 3 så finns det 5 program som styr och sköter olika delar av maskinen. Programmet som heter SR\_MainMachine är ett huvudprogram som initierar andra program. Man hade kunnat lägga sitt program i MainMachine men nackdelen med det är att om man har flera maskiner som ska styras så är det bra att ha dem separerade. Fördelen med att ha programmen separerade är att programmeringen blir enklare att följa och felsöka i.

I SR\_ScrewAsembler finns nästan all programmering till maskinen.

SR\_ScrewAsemblerOperatingMode sköter vilket driftläge som körs. Det finns tre driftlägen som är setup, hand och auto. I SR\_Logging så hittar vi ett funktionsblock som sparar, tar bort och sorterar data. Det ligger i ett eget program så att anrop inte sker så ofta. Det sista programmet är SR\_HWCOPYIO och där har vi huvudkontaktorn, nödstopp och enable och programmet knyter ihop I/O och hårdvara med programmet.

Fig.4.1 Visar strukturen över programmen



### 4.2 Master-slav-struktur

Det finns 7 motorer som ska köras. Allt styrs från en virtuell master som styr en submaster.

Submastern styr sen skruvdragarna och höjdjusteringen

Mastern och de båda submasterna är virtuella.

Mastrarna och slavarnas moduler ingår i SR\_ScrewAsembler programmet.

Mastergrupp: Mastern

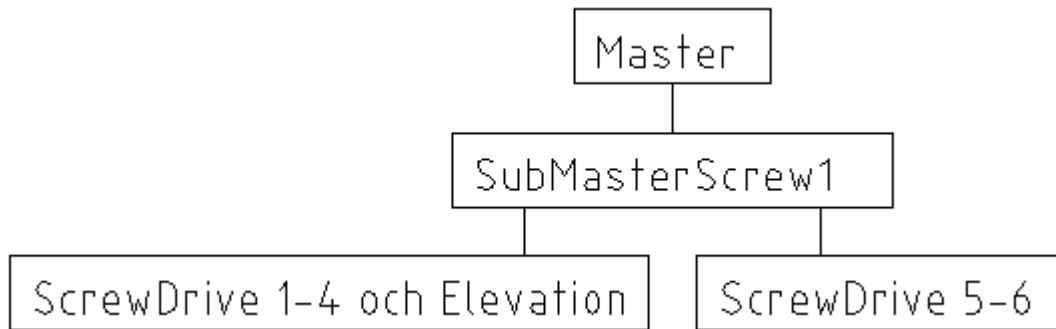
Submastergrupp: Submasterscrew1-2

Submasterslavegrupp1: Screwdrive 1-4 och höjdjustering

Submasterslavegrupp2: Screwdrive 5-6

Man grupperar modulerna för att enkelt kunna anropa en grupp istället för alla enskilt. Genom att göra programstrukturen på detta sätt så blir det mycket enkelt att göra valet om man ska köra med 4 eller 6 skruvdragare.

Fig.4.2 Visar grupperingen av axlarna



## 4.3 Konfiguration och initiering

### 4.3.1 PLC Konfiguration

Det första man gör i PLC-konfigurationen är att välja typ av enhet och tilldela den en IP-adress.

Det finns information om version, system och IEC program.

Sercosbussen konfigureras genom att lägga till servoförstärkarna som används, både virtuella och verkliga servomotorer.

Servoförstärkarna ska adresseras, om man sätter en adress högre än 99 så blir det en virtuell enhet.

Mekanisk information ska anges såsom utväxling, matningskonstant och tröghetsmoment.

Till varje servoförstärkare ska en logisk avkodare konfigureras som har hastighetsgenerator och fasgenerator.

### 4.3.2 AxisModuler

Axismodulen har 7 olika områden i initiering av modulen. De är Homing, Manual, Cam, Endless, Pos, MultiPos och generellt om modulen. Det är bara tre områden på modulen som måste ändras, det är Manual, Cam eller Endless och generellt. Under generellt sätter man namnet på axeln, adressen till motorn och avkodare och om den har en master och i så fall vilken.

Fig. 4.3 Visar den generella konfigurationen till Screwdrive 1

```

0001 (* System generated code, could be modified for your application *)
0002
0003 (* Standard interface *)
0004
0005 astSubModuleInterface[c_diScrewDrive1].stI.sModuleName      := CONCAT(STR1:=I_sModuleName, STR2:="ScrewDrive1:");
0006
0007 (* AxisModule specific interface *)
0008 astSubModuleInterface[c_diScrewDrive1].stI.xLogEnable       := TRUE;
0009 astSubModuleInterface[c_diScrewDrive1].stI.wLogFilter       := DINT_TO_WORD(TPD_GE_LOGFILTER_MODULE_DEFAULT);
0010
0011 (* astAxisModuleInterfaceis main drive / EndlessFeed*)
0012 stScrewDrive1Interface.stI.diMasterId                       := DRV_SubMasterScrew1.LogAdr;    (* Master *)
0013 stScrewDrive1Interface.stI.driveAxis                       := DRV_ScrewDrive1;
0014 stScrewDrive1Interface.stI.lencEncoder                    := LE_ScrewDrive1;
0015 stScrewDrive1Interface.stI.ginSensorTP                     := IM_1;
0016
0017 (* DeadTimeMode for Firmware V20 *)
0018
0019 diDeadTimeMode:=TPL_FC_GetDeadTimeMode();
0020 IF diDeadTimeMode = 1 AND                                  (* 1 = V20 ; 0 = V16 *)
0021   (DINT_TO_DWORD(stScrewDrive1Interface.stI.diMasterId) AND TPC_Gc_dwLogAdrTypeMask) = DINT_TO_DWORD(P_ENC_TYPE) THEN(* Physical encoder *)
0022   stScrewDrive1Interface.stI.lencEncoder.Delay := -stScrewDrive1Interface.stI.driveAxis.ShaftRefDelay;
0023 END_IF
0024
0025
0026 stScrewDrive1Interface.stIQ.stExtended.stI.rAxisMoveVel    := 15.0;    (* Limit for max. velocity of astAxisModuleInterface to switch output AxisMove in Units/s *)
  
```

I konfigurationen för manuell drift så ställer man in hastighet, acceleration och retardation. Hastigheten kan man sen också ändra på operatörspanelen. I manuellt läge kan man köra på två sätt, jogg eller stegkörning. I konfigurationen anges hur många grader den får rotera per steg vid stegkörning.

Alla moduler som användes konfigureras lika på dessa områden.

I kamkonfigurationen anges punkter för förhållandet mellan master och slav. Screwdrive1-6 har samma kamkurva, två punkter, en start och en stopposition och den kör linjärt mot submastern. Elevation modulen liknar Screwdrive modulerna, det som skiljer sig är att Screwdrive modulerna kör cykliskt 360 grader och sen börjar de om medan elevation kör bara ett varv och stannar på sista punkten. Submaster modulen kör inte cykliskt och har tre kampunkter och det är för att man ska kunna få ett mjukare stopp till alla motorer när de närmar sig slutpunkten i processen man får en kurvform som kommer förklara mer i kapitel 4.9.

Vid kallstart på screwdrive1-6 och submastern sätts positionen för dessa till första punkten i kamkurvan. Men för elevation modulen så kör den till första punkten i kamkurvan.

Mastern kör ändlöst (endless) och allt man ställer in där är hastighet, acceleration och retardation.

Fig. 4.4 Visar Kamkonfigurationen till elevation

```

0146 (** Cam **)
0147
0148 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.diCSModeAxis := 2; (* ColdStartMode Axis *)
0149 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.diCSModeMaster := 1; (* ColdStartMode Master *)
0150 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.diWSMode := 0; (* WarmStartMode *)
0151 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.IrWSWindow := 5.0; (* WarmStartWindow in Axis units *)
0152 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.rWSVel := 100.0; (* Velocity for WarmStart and CSModeAxis = 2 in Units/s *)
0153 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.rWSAcc := 18.0; (* Acceleration and deceleration for Warmstart and CSModeAxis = 2 in Units/s2 *)
0154 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.rEmergencyDec := 10000.0; (* Deceleration in Units/s2 with 'Enable' off / FastStop *)
0155 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.IrMasterEncOffset := 0.0; (* Master encoder offset *)
0156 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.diTXend := 2*LzsTaskGetInterval(); (* Time of XLowEnd and XHighEnd before Xend or Xstart in msec *)
0157 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.IrInstantXLimMax := 0.0; (* Switch off position at 'InstantNewCam' *)
0158 stElevationInterface.stIQ.stExtended.stI.diTableSelectNr := TPD_GE_USER_ID;
0159 stElevationInterface.stIQ.stExtended.stI.xEnableTabHandler := FALSE;
0160 stElevationInterface.stIQ.stExtended.stI.xWithoutEncSet := TRUE; (* extended In of the Multicam6 FB, can be used if needed! *)
0161 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.stMultiCamExtn.NoCyclicCams := TRUE;
0162 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.stMultiCamExtn.NoModuloMasterAtStart := TRUE; (* extended In of the Multicam6 FB, can be used if needed! *)
0163 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.stMultiCamExtn.NoModuloSlaveAtStart := TRUE; (* extended In of the Multicam6 FB, can be used if needed! *)
0164
0165
0166 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.stMultiCamExtn := (* Extended In of the Multicam6 FB, can be used if needed! *)
0167
0168 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.astMotionPar[TPD_GE_USER_ID].NumberOfCamPoints := 2; (* DataStruct for the next cycle *)
0169 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.astMotionPar[TPD_GE_USER_ID].XPeriode := 0.0;
0170 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.astMotionPar[TPD_GE_USER_ID].YPeriode := 0.0;
0171 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.astMotionPar[TPD_GE_USER_ID].CamPoint[0].X := 0.0;
0172 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.astMotionPar[TPD_GE_USER_ID].CamPoint[0].Y := 0.0;
0173 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.astMotionPar[TPD_GE_USER_ID].CamPoint[0].m := 1.0;
0174 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.astMotionPar[TPD_GE_USER_ID].CamPoint[0].CamType := Straight;
0175 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.astMotionPar[TPD_GE_USER_ID].CamPoint[1].X := 820.0;
0176 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.astMotionPar[TPD_GE_USER_ID].CamPoint[1].Y := 820.0;
0177 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.astMotionPar[TPD_GE_USER_ID].CamPoint[1].m := 1.0;
0178 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.astMotionPar[TPD_GE_USER_ID].CamPoint[1].CamType := Straight;
0179
0180
0181 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.astMotionPar[TPD_GE_IDLE_ID].UserTableId := TPD_GE_IDLE_ID;
0182 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.astMotionPar[TPD_GE_START_ID].UserTableId := TPD_GE_START_ID;
0183 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.astMotionPar[TPD_GE_CONT_ID].UserTableId := TPD_GE_CONT_ID;
0184 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.astMotionPar[TPD_GE_STOP_ID].UserTableId := TPD_GE_STOP_ID;
0185 stElevationInterface.stIQ.stExtended.stAxisDal.stCam.astMotionPar[TPD_GE_USER_ID].UserTableId := TPD_GE_USER_ID;

```

### 4.3.3 Övrig konfiguration

I update\_struct skapar man interface till axismodulerna och sen skapar man en deklaration för kommandotabeller som sen används i CMDT.

I Submoduls konfigureras vilka moduler som är i de olika grupperna, mer information om grupperna finns på 4.2.

## 4.4 CMDT och CMDC

I CMDT så skapar man tabeller med instruktioner till axismodulerna. Varje tabell är för ett kommando som varmstart av modulerna eller kallstart.

Det finns klara instruktioner för axismodulerna som man får från biblioteket. Det är här som man får användning av att man grupperar modulerna då man kan anropa en grupp istället för att anropa 5 moduler. Tabellerna blir mindre, vilket leder till att det blir enklare att följa och felsöka.

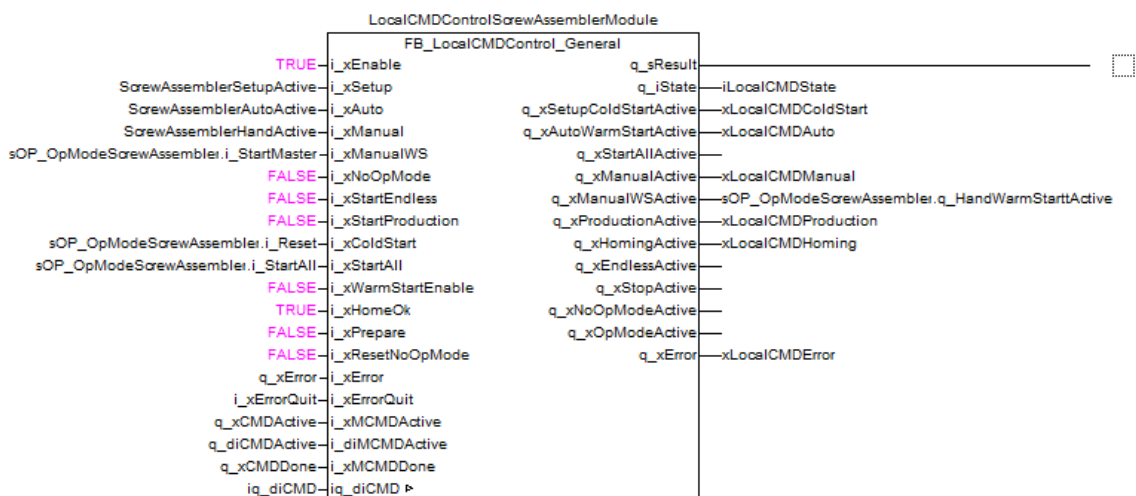
När man skriver en ny tabell så måste man deklarera kommandotabellernas interface i update\_struct, sen ska man uppdatera antal kommandotabeller i SR\_ScrewAssembler.

CMDC är ett funktionsblock där man skriver tillstånd. Tillstånden i funktionsblocket anropar en tabell i CMDT.

Varje tillstånd har en ingång som startar den men ett tillstånd behöver inte ha någon utgång. Ingångarna och utgångarna är kopplade till operatörspanelen eller till andra program i projektet.

Det är viktigt när man skriver villkoren för att gå in i ett tillstånd att de är skrivna på ett säkert sätt så man inte hoppar till olika tillstånd samtidigt eller att man fastnar i ett tillstånd.

Fig. 4.5 Visar CMDC modulen



## 4.5 Manuell drift av Axlar

För att sätta motorerna i manuell drift skapar man en tabell med instruktioner så alla motorer ställer sig i manuell driftläge. Det skriver man i CMDT, sen ska man skiva en funktion i CMDC som går till ett tillstånd som anropar tabellen i CMDT. Funktionen för att gå i manuell drift startas från operatörspanelen. På varje axismodul så finns det två adresser som joggar motorn fram eller back men bara om motorn är i manuell driftläge. På operatörspanelen så finns det en knapp för att jogga fram, en för jogga back, en knapp för val av jogg eller stegkörning och en variabel för att justera hastigheten.

## 4.6 Manuell drift av Master med varmstart

Vid manuell drift av mastern så sätter man master i manuell drift och submaster och alla slavar i kamstyrning med varmstart. Genom att jogga mastern fram eller back så följer skruvdragarna och höjdjusteringen submasterns kamkurva. Fördelen med detta är att man kan följa processen sakta. Funktionen styrs med hjälp av en tabell i CMDT.

## 4.7 Driftval

Det går att välja två olika driftval, drift med 4 skruvdragare eller med 6 skruvdragare. Lösningen till detta problem är att skriva två tabeller i CMDT. I den första tabellen sätter man mastern i ändlöst läge, skruvdragare 1-4, höjdjustering och submastern i kamstyrning med varmstart. Man startar detta tillstånd med autoknappen på operatörspanelen. I den andra tabellen startar man även skruvdragare 5-6 i kamstyrning med varmstart.

## 4.8 Återställning

För att återställa maskinen till sitt utgångsläge sätter man den i driftmode setup och funktionen aktiveras med knappen återställning. Funktionen styrs av en tabell i CMDT. Tabellen sätter submastern och alla slavarna i kamstyrning med kallstart.

## 4.9 Konfiguration av motorparametrar

Konfiguration av motorparametrar är viktig så att man kan göra mjuka start och stopp på maskinen, vilket leder till att man minskar slitaget på maskinen och ökar kvalitén på produkten.

För att skapa en mjuk start så ställer man in masterns acceleration och retardation, alla andra moduler kör i exakt samma hastighet som mastern. För att få ett mjukt stopp när skruven är helt inskruvad skapar man detta med punkter i kamkurvan för submastern.

Exempel på punkter i en kamkurva.

Drift 820 grader.

Punkt 1: master 0 grader, slav 0 grader, profiltyp straight

Punkt 2: master 720 grader, slav 720 grader, profiltyp poly5com

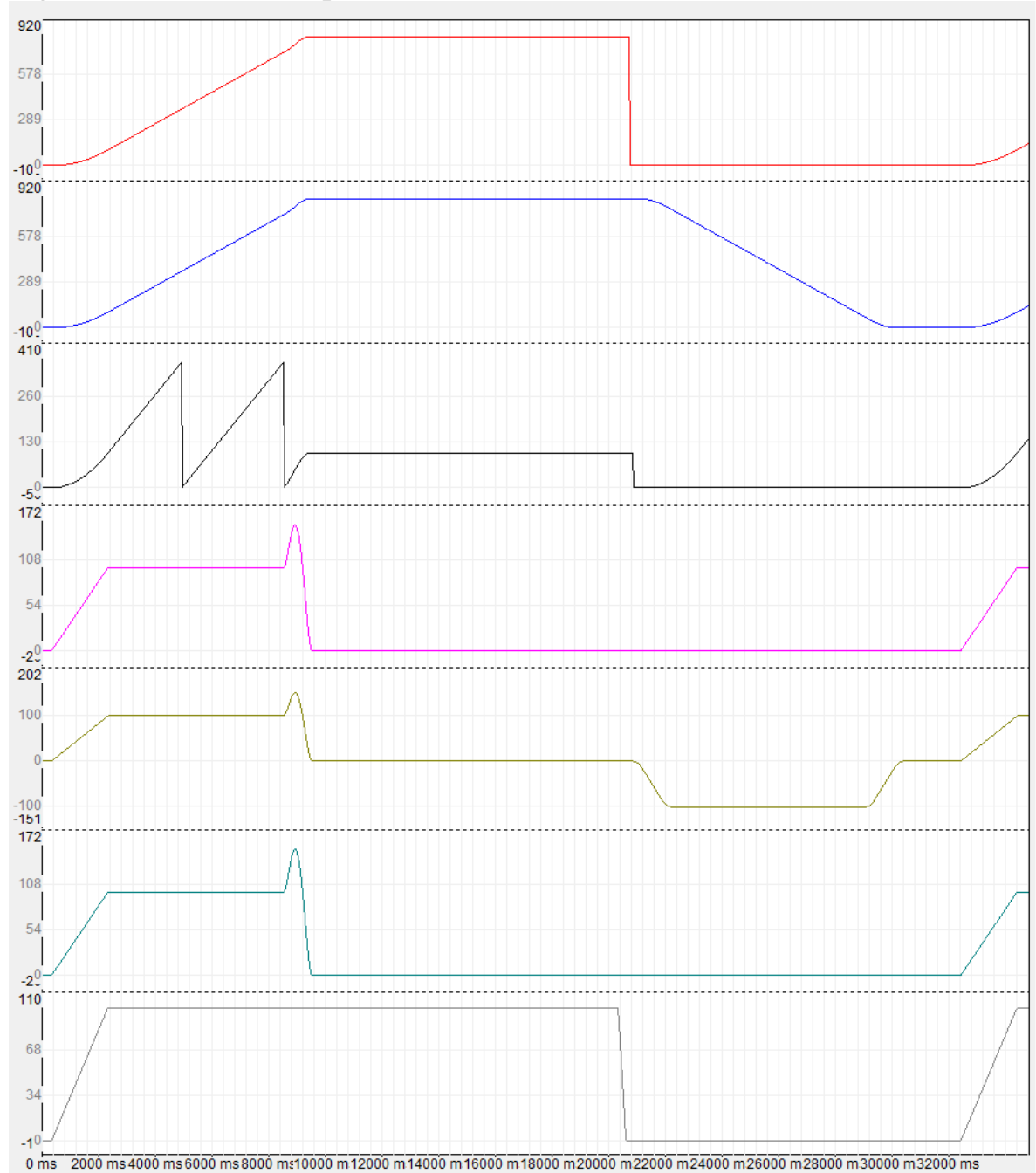
Punkt 2: master 960 grader, slav 820 grader, profiltyp poly5com

Punkt ett säger att det finns en startpunkt vid 0 grader och att den ska köra linjärt till nästa punkt. Så slaven kommer då att följa masterns hastighet fram till 720 grader. Sen ska mastern rotera till 960 grader medan slaven ska rotera till 820 grader, så slaven hastighet kommer att minska gradvis mellan punkterna. Hastigheten på slaven bestäms av ett polynom av 5:e graden.

Bilderna Fig. 4.6–4.8 visar driften med olika konfiguration på motorparametrarna.

Röda kurvan är Submaster position, blå kurvan är höjdjusteringens position, svarta kurva är skruvdragare 1 position, rosa kurvan submaster hastighet, lime kurvan är höjdjusteringens hastighet, turkosa kurvan är hastigheten för skruvdragare 1 och grå kurvan är masterns hastighet.

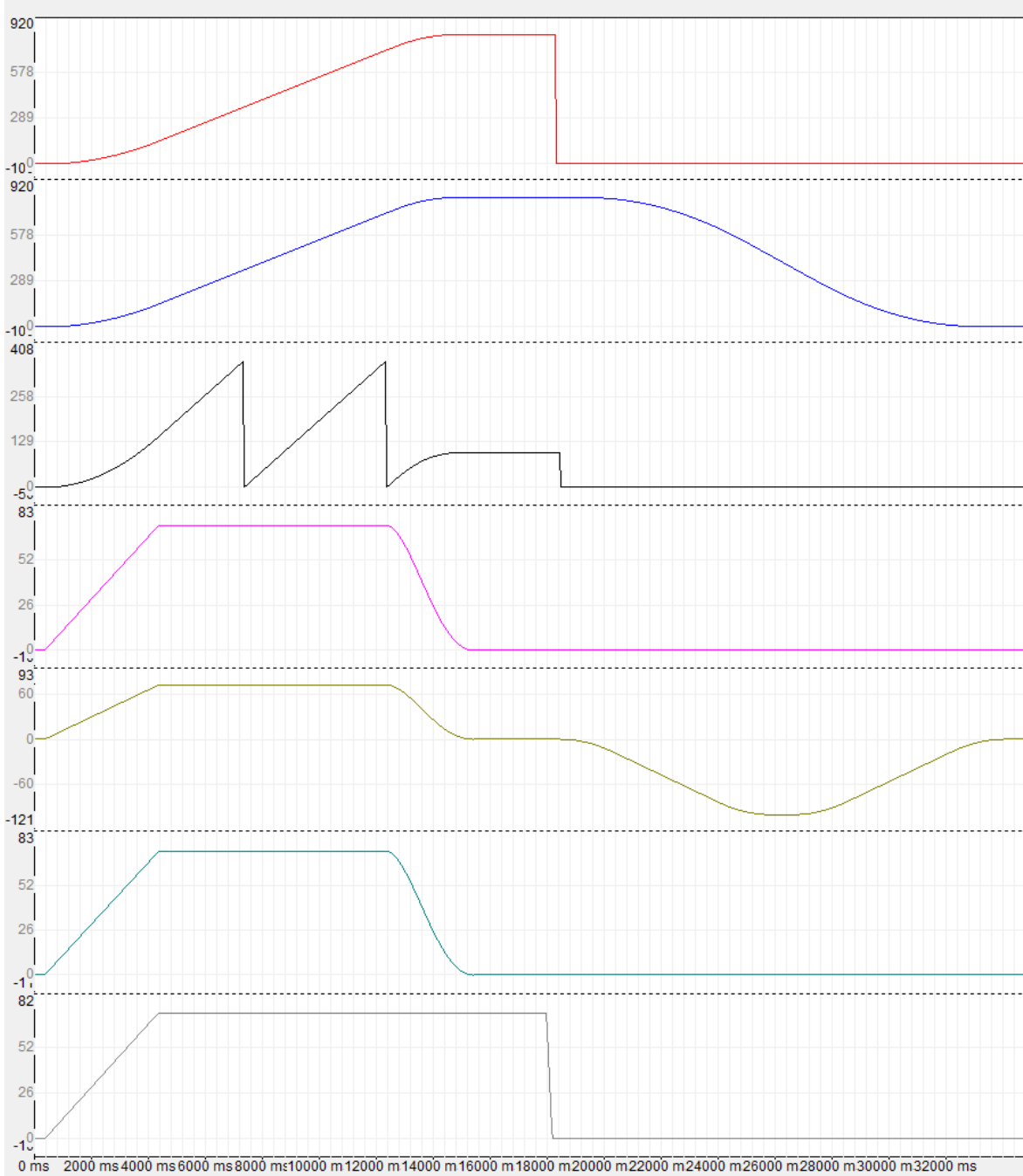
Fig. 4.6 Visar driftexempel 1



På Figur 4.6 visar drift med väldigt hög acceleration på mastern och submastern har inget mjukstopp. Det är inte bra för motorer med snabba start och stopp.

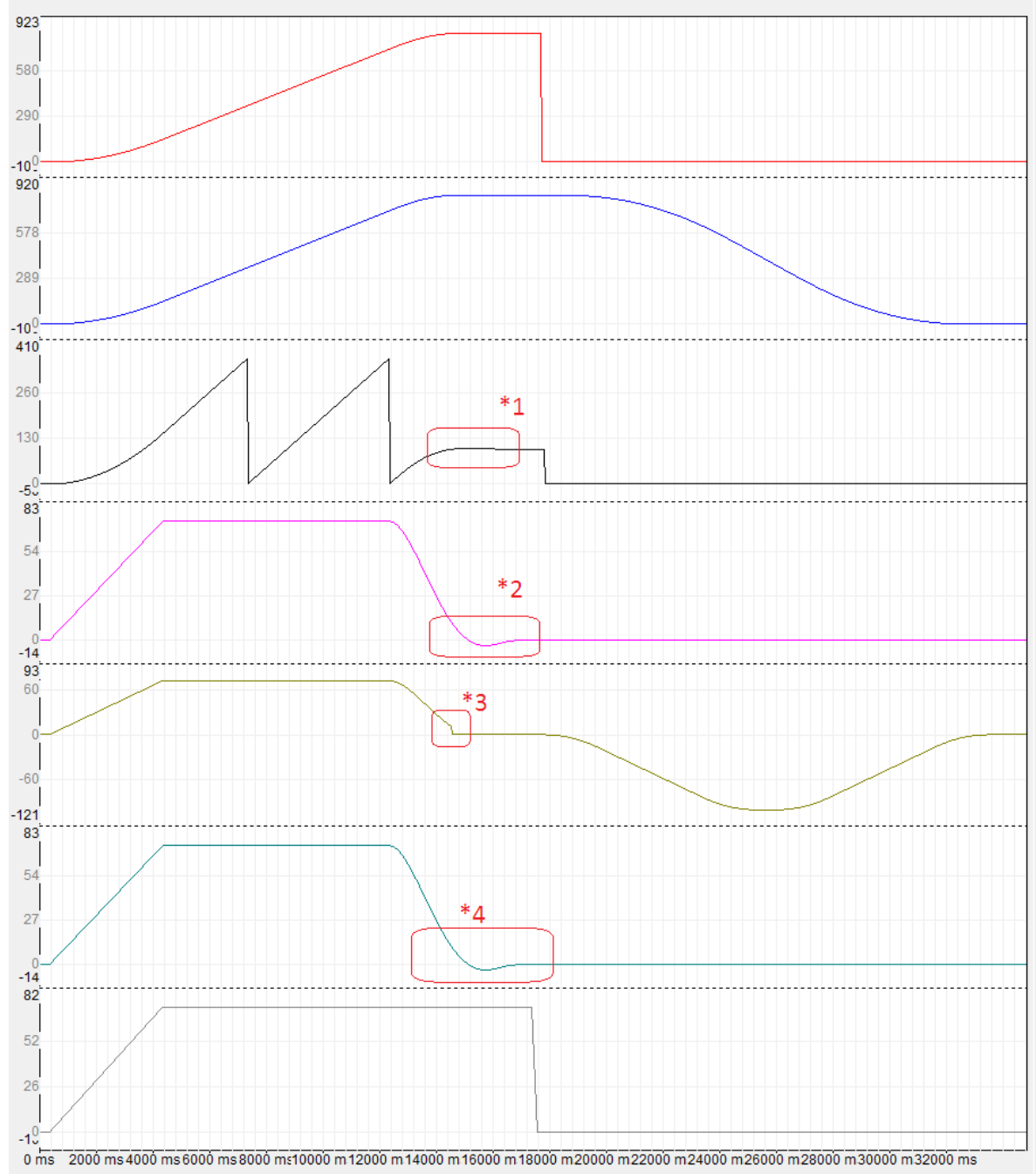


Fig. 4.7 Visar driftexempel 2



I figur 4.7 så är all konfiguration angående motorparametrarna inställd till en mjuk drift. Submasterna retarderar de sista 100 graderna.

Fig. 4.8 Visar driftexempel 3



I figur 4.8 så har submasterna 31 % hastighet de sista 100 graderna. Notera punkt 1 som är skruvdragare 1 position, den har gått 3-4 grader längre än vad den skulle. Höjdjusteringen gick också fel på 3-4 grader men det syns inte på kurvan. Punkt 2-4 visar att alla motorer börjar backa för att korrigera felet.

## 4.10 Kampunkter

Punkterna i kamkurvorna måste ha rätt utväxling mellan ökning i höjd och rotation. Skruven är 33.5mm. Tar man bort huvudet på skruven så är den 32mm och det är 6mm i stigning på skruven. Skruvdomkraften kommer ha en stigning på 6mm och en utväxling på 10. Man behöver plats för att montera skruven på verktyget det krävs minst 14mm utöver skruvens längd men väljer att ha 20.5mm för att det inte ska vara så trångt när man monterar skruven på verktyget.

Det blir totalt 54mm och det 9 varv, så submasterns slutpunkt bör vara på 3240 grader.

## 4.11 Kommunikation med operatörspanelen

Största delen av kommunikationen med operatörspanelen sker i programmet Status\_OP. I programmet så hanterar alla signaler som start av mastern, indikering av start och stoppläge för höjjustering. Information om position och statutstext hämtas från modulerna. Manuell hastighet och typ av jogg för modulerna och stopposition för höjjusteringen anges på operatörspanelen. Status för driftmod av alla moduler visas på operatörspanelen. Det finns även ett program så att man kan spara en ny stoppunkt i kamkurvan, och larm så att den nya stoppunkten är inom ett visst gränsområde.

Resten av kommunikationen sker till CMDC funktionsblock och axismodulerna.

## 4.12 Operatörspanel

Operatörspanelen är programmerad i Vijeo designer och består av 11 sidor som hanterar alarm, status, drift, inställningar och manuell drift. Det som är viktigt när man gör panelsidorna är att de är enkla att förstå och använda. Det måste också finnas säkerhet, så att vem som helst inte kan ändra de känsligare variablerna som stoppositionen för höjjusteringen. Säkerheten som finns är ett inloggningssystem så att man bara kan ändra variablerna om man har behörighet. Det finns en larmlista som visar aktuella larm och en som visar historik men det är bara larm som angår maskinens drift. Det finns även en larmlista på huvudsidan som ger larm angående Pacdrive systemet. Alla larm angående maskinens drift måste skapas i en larm grupp i Vijeo designer och en variabeldeklaration i Epas-4 som larmet är knutet till. Variabler som behöver justeras vid drift ska finnas på operatörspanelen så att man inte behöver ha någon dator när man ska justera något.

Fig. 4.9 Visar drift sidan för processen

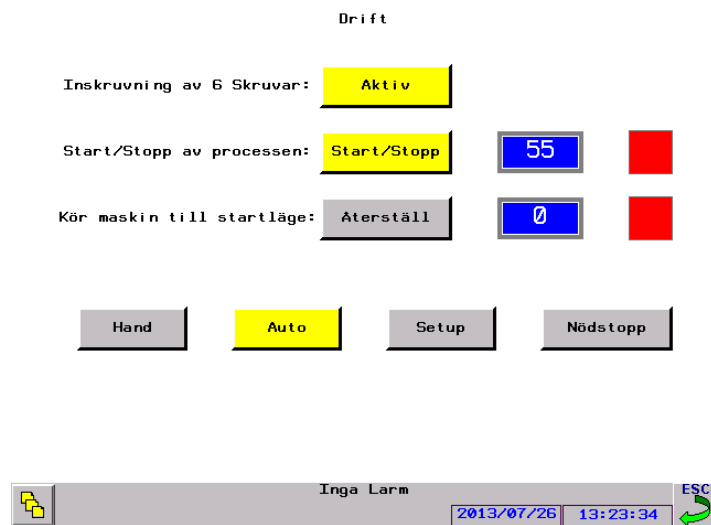


Fig. 4.10 Visar sidan för manuell drift för mastern

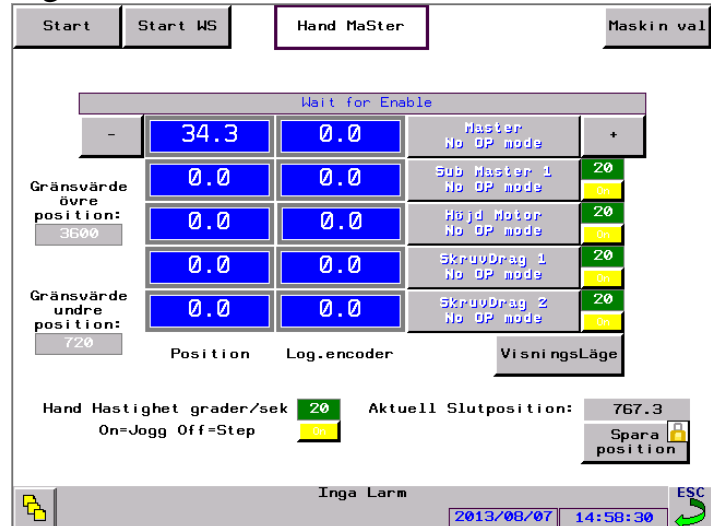
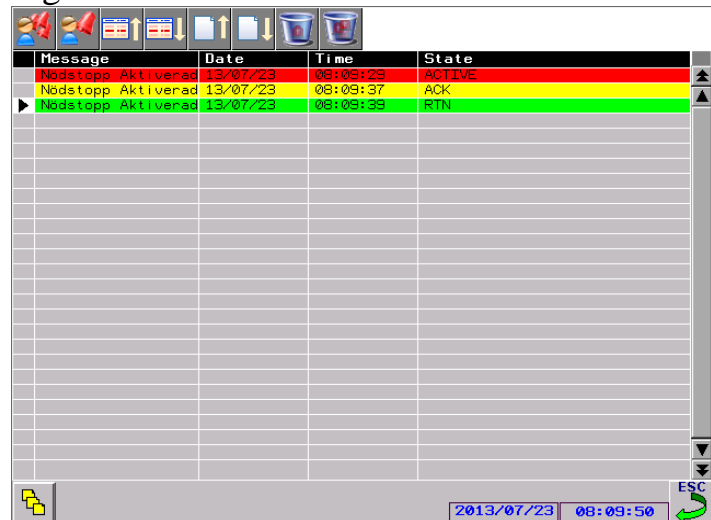


Fig. 4.11 Visar larmhistorik till maskin/drift



## 5 Säkerhet

### 5.1 Riskbedömning

Riskbedömningen kan man se ur två synvinklar, den första är personsäkerhet och den andra är maskinsäkerhet.

Personsäkerhet:

- Det finns klämrisk mellan den rörliga plattformen och mot konstruktionen.
- Det finns klämrisk mellan skruvdragarna och mot konstruktionen.
- Roterande axlar, skruvdragarna och höjjustering.

De roterande axlarna medför inte någon stor fara på grund av att de kommer att rotera med en låg hastighet och har ett begränsat antal varv som de kan rotera. Det medför att det blir svårt att få någon allvarlig skada.

Klämrisk mellan den rörliga plattformen och mot konstruktionen finns en större risk att få en skada av.

Klämrisk mellan skruvdragarna och mot konstruktionen är en stor risk, om man är två personer som jobbar på maskinen den första monterar ditt en skruv och den andra starta maskinen så kan person ett bli klämd eller få skruven genom handen.

Maskinsäkerhet:

- Servomotorerna startar eller stoppar i fel position

Det finns risk att motorerna startar/stoppar i fel positioner, vilket kan leda till att man förstör produkten eller skadar maskinen.

### 5.2 Säkerhetsfunktioner

#### 5.2.1 Med eller utan inverter enable

”Utan inverter enable” menas att när man får stopp eller nödstopp så bryter man huvudkontaktorn för att stanna enheterna. Nackdelen med denna metod är att det sliter på kontaktorn så om det ofta blir stopp under processen så har kontaktorn kortare livslängd. Inverter enable är alltid aktiv i denna metod.

Med inverter enable funktionens stopp så bryter man inte huvudkontaktorn utan bara kraften till motorn.

I en process där mycket sköts för hand så kommer det att ske många stopp och då är inverter enable den bästa metoden.

Om man ska ha med inverter enable så ställs följande krav

- MC-4 enheten måste vara i ett kopplingskåp eller ett skåp med minst IP54 klassat
- MC-4 enheten måste märkas på typskylten och måste ha en skyddande beläggning. Den skyddande beläggningen kommer som standard på MC-4
- Ett säkerhetsrelä behövs i konstruktionen

I bilagor under elscheman så finns ett kretsschema som visar hur man kan koppla inverter enable funktionen.

### **5.2.2 Personsäkerhet**

För att öka personsäkerheten på maskinen så bör man ha ljusridåer och nödstoppsknappar monterat på maskinen.

### **5.2.3 Maskinsäkerhet**

För att öka maskinsäkerheten så monterar man gränslägesbrytare som bryter höjjusteringen i min och maxläge.

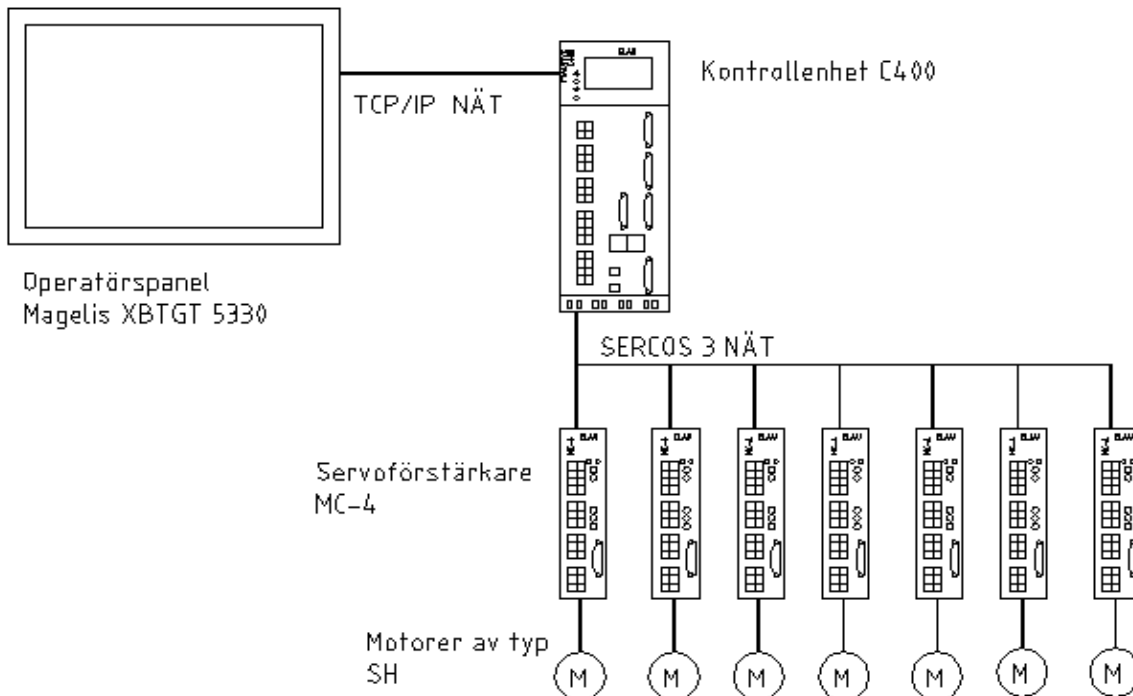
Gränslägesbrytaren för minläget används för att referensköra höjjusteringen.

## 6 Resultat

### 6.1 Konstruktion

Resultatet blev att man ska ha en plattform där man kan placera servomotorerna i 22 olika positioner beroende på formatet på skivan som ska köras. Sen flyttar man plattformen med en servomotor. Iskrivningen görs underifrån. Servosystemet är ett ELAU Pacdrive system. Kontrollenheten för styrningen är en C400 till den är en operatörspanel av typ Magelis XBTGT 5330 ansluten med Ethernet och protokollet är Modbus TCP. Från C400 enheten är det anslutet 7 st. servoförstärkare av typen MC-4 där kommunikationen sker med Sercos 3 buss. Till MC-4 enheten är en servomotor av typen SH ansluten.

Fig. 6.1 Visar arkitektur på pacdrive systemet.



## 6.2 Programmet

9 axismoduler styr servomotorerna, 6st skruvdragare, 1st höjjustering och 2st virtuella axlar.

En virtuell submaster är slav till en virtuell master som kör ändlöst, skruvdragare och höjjustering är slavar till den virtuella submastern. Sekvensen som styr monteringen av skruvarna startas med en knapp på operatörspanelen. Sekvensen startar mastern och därmed startar skruvdragarna och höjjusteringen som slavar mot submastrarna. Profilen på rörelsen bestäms av en kamkurva för submastern där sista punkten i kamkurvan är stoppositionen. Återställning av maskinen till sitt startläge sker med en knapp på operatörspanelen. Detta gör en kallstart av modulerna. Programmet har inlagt de uträknade kampunkterna för att det ska bli rätt utväxling och höjdregering. Eventuell justering av kamkurvorna kan göras vid drifttagning.

Som ett resultat på hur programmet är skrivet kommer skruvdragarna att rotera när man kör upp plattformen. Detta kan vara ett problem om man ska köra en längre sträcka innan man är vid skivan så måste man begränsa sin fart så skruven inte lossnar från verktyget. Man kan lösa det genom att man lägger elevation modulen i en ny submaster. Man måste ändra kamkurvorna för de två submastrarna så man får mjukstart på skruvdragarna, bromsning på elevation och att man synkroniserar skruvdragarna med elevation. Men när sträckan är kort skapar det bara mer problem och man sparar inte så mycket tid i processen.



## 7 Slutsats

Lösningen på problemet med montering av skruvar i skivor har lösts på ett effektivt sätt med hög grad av kontroll på processen. Konstruktionen med separat motor till höjjusteringen bedömdes vara den bästa lösningen. Fördelarna är att alla skruvdragare flyttas tillsammans. Lösningen är väldigt öppen för förändringar som är en fördel för framtida utvecklingar och ändringar.

Med rotationsmetoden så finns det mycket mekanik i processen som begränsar eventuella ändringar och det kräver mer styrning för att alla motorer går separat upp och ner.

Om man ser på programmet som sköter styrning och driftval så finns det kanske tillägg och justeringar att göra när maskinen driftsätts. Men programmet kan köra processen och är skrivet på ett bra sätt som gör att tillägg eller ändringar i framtiden blir enkelt.

Det finns en funktion i programmet så man kan spara nya positioner som kommer att vara användbara under driftsättning och vid kalibrering av maskinen.

Operatörspanelen har ett enkelt gränssnitt och är enkel att använda.

Överlag så tycker jag att projektet har gått bra och har gett bra resultat.

### 7.1 Framtid och utvecklingsmöjligheter

I framtiden så kommer maskinen byggas och driftsättas.

Det som ska göras i programmet (Epas-4) när maskinen är byggd, blir att lägga in ny larm som kommer från extern utrustning som nödstoppknapp. De nya larmen ska även läggas till på operatörspanelen.

Det finns många utvecklingsmöjligheter på maskinen.

- Man kan automatisera en skruvmatning så man inte behöver montera skruvarna på verktygen för hand.
- Man kan automatisera positionering av skruvdragarna för de olika skivformaten.
- Man kan använda en robot för iläggning och urplockning av skivorna i maskinen.

Utvidgningarna som beskrivs ovan blir enkla att göra styrning till, för strukturen på programmen i Epas-4 är uppdelade så man skapar bara ny program som utför de nya uppgifterna. Kommunikation mellan de nya och gamla programmen blir inga problem för att båda är knutna till SR\_MainMachine.

Det som kan behövas är en ny kontrollenhet för C400 klarar bara av 16 axlar.

## 8 Terminologi

ASIC = Application Specific Integrated Circuits

CAN = Controller Area Network är en fältbuss

CANOPEN = Är en fältbuss som är CAN baserad och är internationellt standardiserade enligt (EN 50.325-4)

CFC = Continuous function chart, programmeringsspråk

CMDC = Command control

CMDT = Command table

DeviceNet = Är en fältbuss som är CAN baserad

Epas-4 = Programmerings program till pacdrive

FBD = Function plan, programmeringsspråk

HMI = Human machine interface

IEC = International Electrotechnical Commission

IL = Instruction list, programmeringsspråk

I/Os = IN/OUT det är ingångar och utgångar

Jogg = Man kan kör fram och baka motorn

LD = Ladder diagram, programmeringsspråk

MES = Manufacturing Execution System är system för styrning och övervakning av produktionsprocess

PLC = Programmable logical controller är ett programbart styrsystem

POU = Program organisation units

PROFIBUS DP = PROFIBUS Decentralized Peripherals är en fältbuss

PWM = Pulse width modulation

SERCOS = **S**erial **r**eal-time **c**ommunication **s**ystem

SFC = Sequential function chart, programmeringsspråk

SH = Är en motor serie till Pacdrive systemet

ST = Structured text, programmeringsspråk

TCP/IP = Transmission Control Protocol/Internet Protocol  
datakommunikation över nätet

Vijeo designer = Programmerings program till operatörspanelen

VxWORK = Är ett realtidsoperativsystem från Wind river

## 9 Referens

### Literatur:

1. Beater, Peter, 2007. Pneumatic Drives. Springer e-books.
2. Burlington, 1997. Pneumatic Handbook. Butterworth-Heinemann.
3. PDF manual till pacdrive c400: [http://static.schneider-electric.us/docs/Motor%20Control/Operating%20Manual\\_%20PacDrive%20Controller%20C400\\_C400%20A8.pdf](http://static.schneider-electric.us/docs/Motor%20Control/Operating%20Manual_%20PacDrive%20Controller%20C400_C400%20A8.pdf)(2013-06-27)
4. EPAS-4 manual, åtkomst från programmet help→contents
5. Vijeo designer manual, åtkomst från programmet help→contents
6. Information om SERCOS <http://www.sercos.com/technology/index.htm> (2013-07-03)
7. Manual om epas-4: [http://static.schneider-electric.us/docs/Motor%20Control/Operating%20Manual\\_%20PacDrive%20-%20EPAS-4.pdf](http://static.schneider-electric.us/docs/Motor%20Control/Operating%20Manual_%20PacDrive%20-%20EPAS-4.pdf)(2013-07-04)
8. Information om elektrisk kam: <http://ehis.ebscohost.com.ludwig.lub.lu.se/eds/pdfviewer/pdfviewer?sid=1ca16637-12d1-442d-8ef4-a732651037bb%40sessionmgr113&vid=5&hid=102v> (2013-07-08)
9. PacDrive M Projecting manual,2002,Elau artikelnummer:17130058-001
- 10.Information om säkerhetsrelä: [http://www.eltron.pl/automatyka/pilz/pdf/przekazniki-bezpieczenstwa/wyl\\_bezp\\_drzwi\\_kurtyny/kat4/PNOZ\\_XV2\\_GB.pdf](http://www.eltron.pl/automatyka/pilz/pdf/przekazniki-bezpieczenstwa/wyl_bezp_drzwi_kurtyny/kat4/PNOZ_XV2_GB.pdf) (2013-08-02)

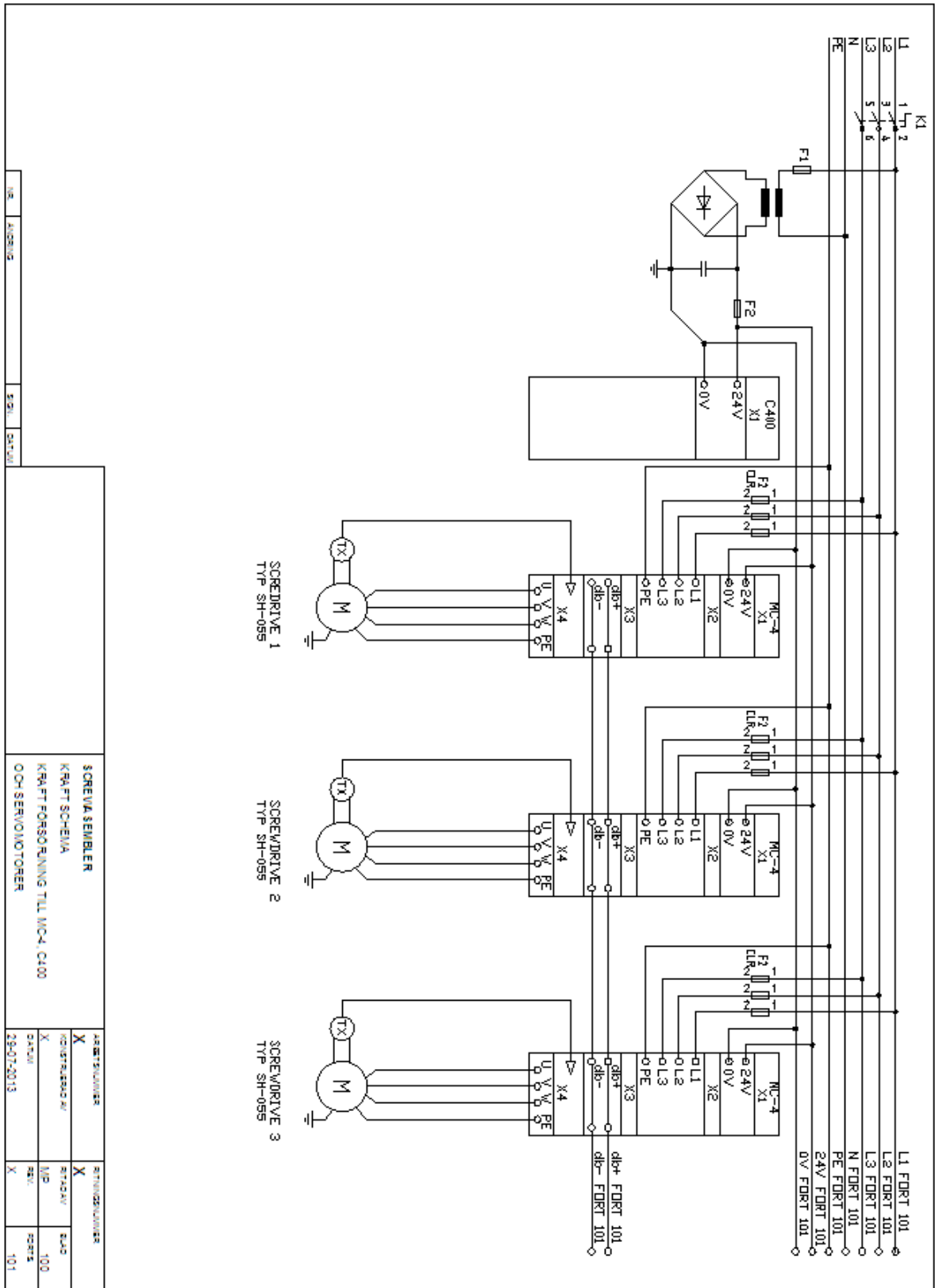
### Bilder:

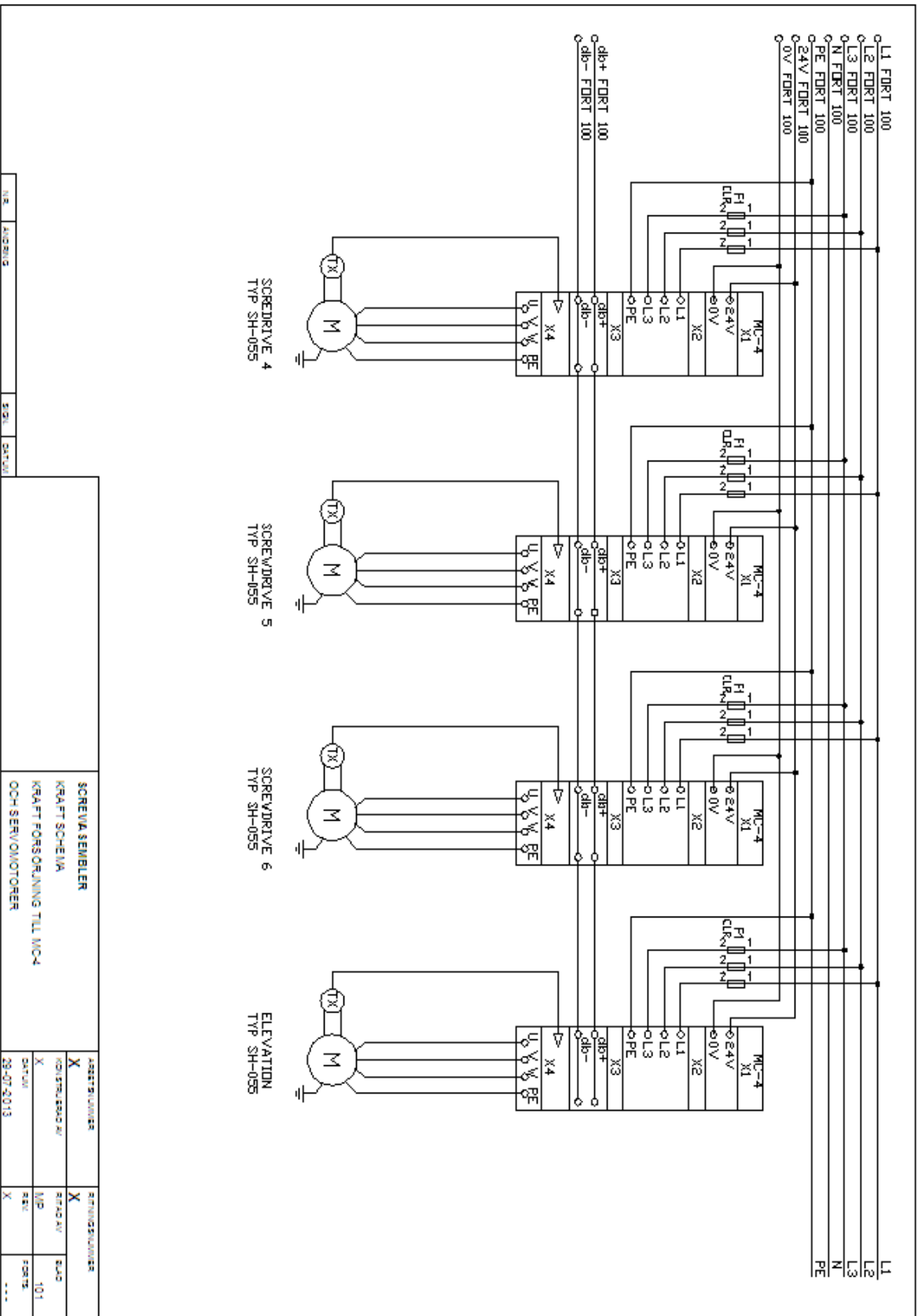
1. Figur 3.1 Axismodul från Epas-4 manual
2. Figur 3.2 Multicam6 modul från Epas-4 manual
3. Figur 4.1 Struktur på programmet i Epas-4 ritad i auto cad
4. Figur 4.2 Struktur på modulerna i Epas-4 ritad i auto cad
5. Figur 4.3 Kod från epas-4, Screwdrive 1
6. Figur 4.4 Kod från epas-4, Elevation
7. Figur 4.5 CMDC modulen från SR\_ScrewAsembler i Epas-4
8. Figur 4.6 Driftexempel 1 från sampling trace i Epas-4
9. Figur 4.7 Driftexempel 2 från sampling trace i Epas-4
- 10.Figur 4.8 Driftexempel 3 från sampling trace i Epas-4
- 11.Figur 4.9 Driftsidan på operatörspanelen från simulation i Vijeo designer
- 12.Figur 4.10 Manuell drift till submaster 1 på operatörspanelen från simulation i Vijeo designer
- 13.Figur 4.11 Larmhistorik på operatörspanelen från simulation i Vijeo designer
- 14.Figur 6.1 Struktur på pacdrive systemet ritad i auto cad

# 10 Bilagor

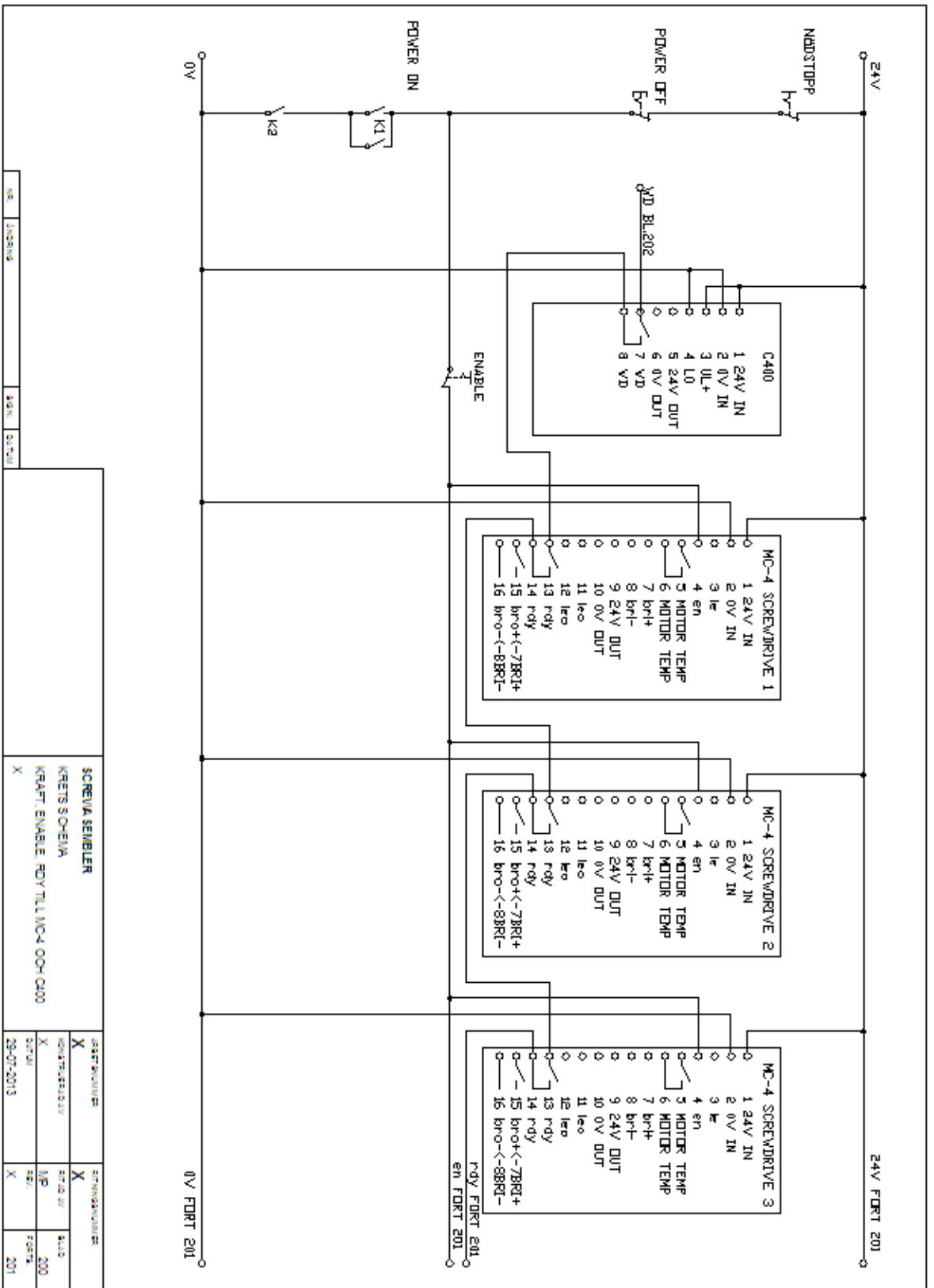
## 10.1 Elschema

Här visas kopplingscheman för att uppnå de olika funktionerna som beskrivs.

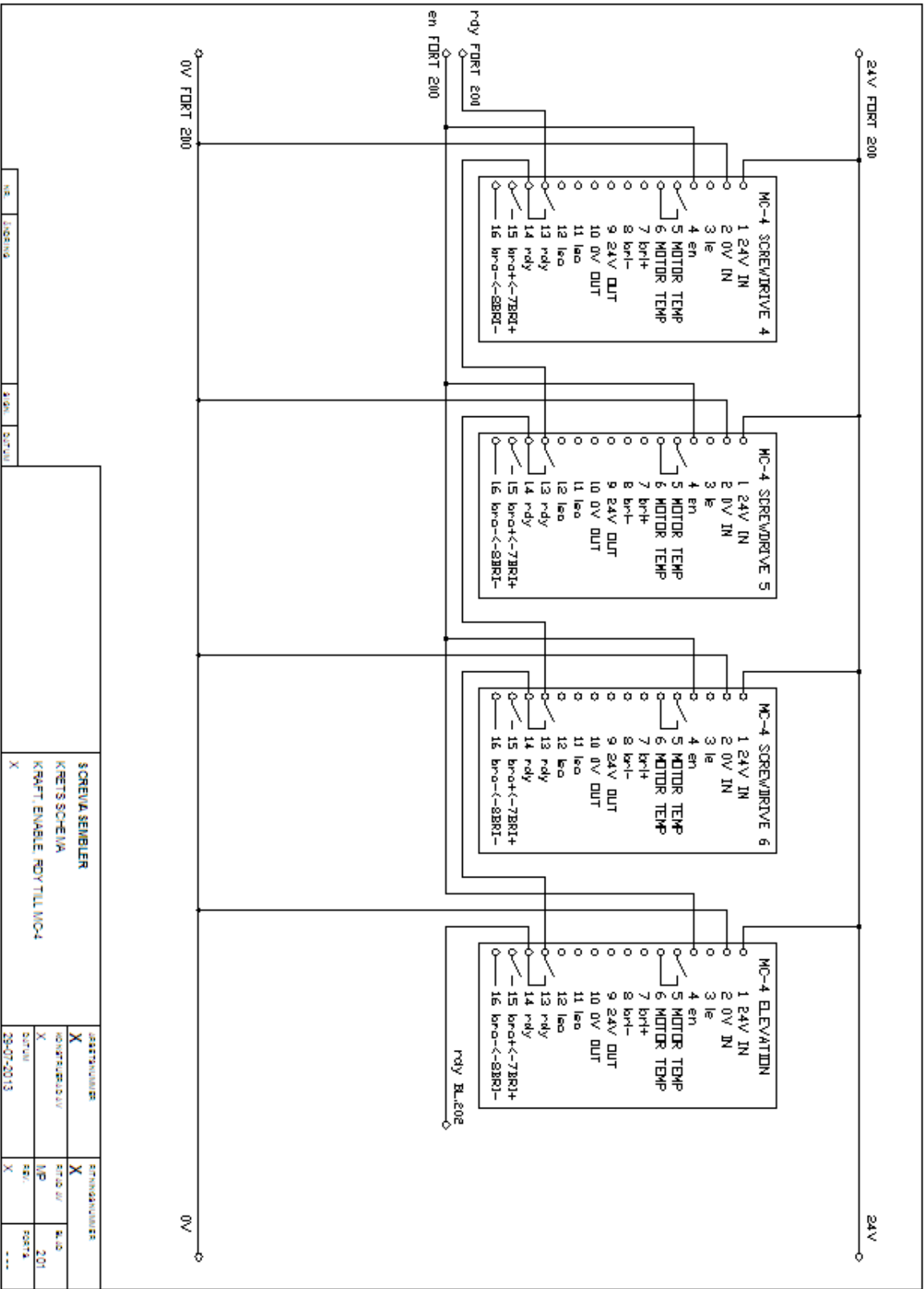




NR	ANORDNING	SRIT	STATUS	SCREWDRIVER	SCREWDRIVER	SCREWDRIVER	SCREWDRIVER
				SCREWDRIVER	SCREWDRIVER	SCREWDRIVER	SCREWDRIVER
				KRAFT SCHEMA	KRAFT SCHEMA	KRAFT SCHEMA	KRAFT SCHEMA
				KRAFT FÖRORJNING TILL MOC4	KRAFT FÖRORJNING TILL MOC4	KRAFT FÖRORJNING TILL MOC4	KRAFT FÖRORJNING TILL MOC4
				OCH SERVO MOTORER	OCH SERVO MOTORER	OCH SERVO MOTORER	OCH SERVO MOTORER
				28-07-2013	28-07-2013	28-07-2013	28-07-2013
				X	X	X	X
				101	101	101	101
				...	...	...	...



SÖREVA SENSLER		SÖREVA SENSLER	
KRETT SÖREVA	X	KRETT SÖREVA	X
KRAFT ENABLE	X	KRAFT ENABLE	X
FOY TILL MC4 OCH Q400	X	FOY TILL MC4 OCH Q400	X
29-07-2013		29-07-2013	



SE	INSERS	24V	24V					
<b>\$CREVA SEMBLER</b> KRETS SCHEMA KRAFT ENABLE RDY TIL MC-4				JAGTSNUMMER X KRAFT TIL MC-4	SKEMA X 29-07-2013	SKEMA X 29-07-2013	SKEMA X 29-07-2013	SKEMA X 29-07-2013

