

Elektronisk kamfunktion för en winder-applikation



Ola Nilsson

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

Elektronisk kamfunktion för en winder-applikation



LUNDS
UNIVERSITET

Lunds Tekniska Högskola

LTH Ingenjörshögskolan vid Campus Helsingborg
Industriell Elektroteknik och Automation

Examensarbete:
Ola Nilsson

© Copyright Ola Nilsson

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

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

Tryckt i Sverige
Lunds universitet
Lund 2018

Sammanfattning

Syftet med det här examensarbetet var att utveckla en generell lösning för en winder-applikation genom att undersöka möjligheten att styra spridarens rörelse med en elektronisk kam. Den elektroniska kammen synkroniserar spridaren med upprullaren genom att bestämma var spridaren ska befinna sig i relation till upprullarens position.

En prototyp byggdes för att återskapa det verkliga driftscenariot i en testmiljö. Prototypen bestod av två stegmotorer och drivsteg, PLC och HMI. En interaktiv visualisering skapades för att möjliggöra för en operatör att starta och stoppa applikationen samt göra en referenskörning av spridaren.

Resultatet av examensarbetet blev en modulär och skalbar automationslösning för en generell winder-applikation. Den elektroniska kammen kan modifieras för att passa det enskilda driftfallet. Mekanik och elmotorer kan bytas ut utan att ändra i programmets struktur. Lösningen är därför applicerbar på en godtycklig tvåaxlig winder-applikation.

Nyckelord: Elektronisk kam, PLC, HMI, stegmotorer, winder-applikation

Abstract

The aim of this thesis was to develop a general solution for a winder application by examining the possibility to control the spreader axis movement with an electronic cam. The electronic cam synchronizes the spreader axis with the winder axis by determining where the spreader axis should be positioned in relation to the winder axis position.

A prototype was built to recreate the application scenario in a testing environment. The prototype consisted of two stepper motors and drives, a PLC and HMI. An interactive visualisation was created that allowed the operator to start and stop the application and also do a homing operation of the spreader axis.

The thesis resulted in the development of a modular and scalable automation solution for a general winding application. The electronic cam can be modified to suit the application. The mechanics and electrical motors can be replaced without having to change the structure of the program. The solution is therefore applicable to any arbitrary winder application with two axes.

Keywords: electronic cam, PLC, HMI, stepper motors, winder application

Förord

Det här examensarbetet utfördes under våren 2018 hos SIGBI AB i Helsingborg, Sverige. Examensarbetet utgjorde den sista delen av högskoleingenjörsutbildningen *Elektroteknik med Automation* vid Lunds Tekniska Högskola.

Upplevelsen hos SIGBI har varit väldigt lärorik. Min problemlösningsförmåga har vid flera tillfällen satts på prov vilket har gjort att min personliga utveckling främjats. Jag har fått en god inblick i hur det är att arbeta som automationsingenjör. Jag är oerhört tacksam över hur SIGBI har bistått med resurser i form av material, tid och tålamod. Därför vill jag rikta ett stort tack till Stefan Svelenius, Mats Johannesson och min handledare Patrik Andersson på SIGBI AB.

Ett tack riktas även till Henrik Arnekull på Styrteknik Europe för mekaniken som användes i prototypen för att fixera tråden. Slutligen vill jag tacka Bengt Simonsson, min handledare vid LTH och Mats Lilja, min examinator.

Innehållsförteckning

1	Inledning	1
1.1	Bakgrund	1
1.2	Syfte	1
1.3	Målformulering	1
1.4	Problemformulering	1
1.5	Motivering av examensarbetet	2
1.6	Avgränsningar	2
2	Teknisk bakgrund	3
2.1	Olika typer av spridning	3
2.2	Motion Control	4
2.2.1	Mekanisk kam	4
2.2.2	Elektronisk kam	5
2.3	Stegmotorer	6
2.3.1	Drivsteg	7
2.4	PLC	8
2.4.1	VARAN	8
2.4.2	S-DIAS Bus	9
2.5	LASAL	10
2.5.1	Class 2	10
3	Konstruktion av prototyp	12
3.1	Funktionsbeskrivning	12
3.2	Mekaniskt utförande	12
3.2.1	Fastech	13
3.2.2	Upprullare	15
3.2.3	Spridare	15
3.3	CP102, ST011, KL090 och SE051	16
3.4	HMI – ETT 732	19
4	Implementering av applikationen	20
4.1	Konfigurering av anslutningar	20
4.2	Program	20
4.3	Styrning	21
4.4	Synkronisering	23

4.5	Den elektroniska kammen	24
5	Resultat	27
5.1	Kammen	27
5.2	Visualiseringen	27
6	Slutsats	29
6.1	Vidareutveckling	29
7	Terminologi	30
8	Källförteckning	31

1 Inledning

1.1 Bakgrund

SIGBI AB är ett företag grundat 1982 i Helsingborg som levererar automationslösningar där dimensionering av drivsystem, driftsättning samt support och service ingår. SIGBI är även återförsäljare av bland annat styrsystem, frekvensomriktare och elmotorer.

SIGBI har tidigare stött på fall där en automationslösning åt en winder-applikation varit önskad. Applikationen har bestått av ett tvåaxligt system, en upprullare som rullar upp en produkt på en rulle eller vinda och en spridare som formar upprullningen av produkten.

Idén har varit att synkronisera en upprullare och en spridare där spridaren följer upprullarens position. När upprullaren roterar ska spridaren löpa över upprullarens ändlägen. Hur spridaren löper över ändlägena påverkar mönstret på upprullningen.

1.2 Syfte

Syftet med examensarbetet var att ta fram en generell lösning för en winder-applikation med en spridare. Lösningen skulle innehålla möjligheten att synkronisera upprullare och spridare med en elektronisk kam.

1.3 Målformulering

Målet med examensarbetet var att utveckla en applicerbar lösning till winder-applikationen. För att göra lösningen modulär och skalbar skulle den elektroniska kammen kunna ändras för det enskilda ändamålet. Implementering skulle alltså med lätthet kunna skräddarsys för alla typer av tvåaxliga winder-applikationer genom att enbart behöva ändra den elektroniska kammen. Detta oavsett storlek på elmotorer, typ av elmotorer, typ av material som rullas upp eller hur materialet rullas upp.

1.4 Problemformulering

Huvuduppgiften i examensarbetet var att rulla upp en produkt enligt ett specifikt önskemål på en vinda eller spole. Detta medför att spridarens rörelse måste specificeras med avseende på upprullaren. Nedan följer några punkter som examensarbetet hade som intention att lösa:

- Implementera en elektronisk kam
- Realisera drift med olika typer av spridning
- Undersök för- och nackdelar med drifterna.
- Optimera drifterna
- Implementera en interaktiv visualisering för att kunna starta och stoppa applikationen.

1.5 Motivering av examensarbetet

Detta examensarbete valdes eftersom det återskapar en typisk situation där en automationsingenjör behöver tillverka en automationslösning. Examensarbetet gav därför en inblick i hur det är att arbeta som automationsingenjör. Problemlösningsförmågan sattes på prov vilket främjat den personliga utvecklingen. Även kunskaper kring styrsystem, programmering och elmotorer fördjupades.

1.6 Avgränsningar

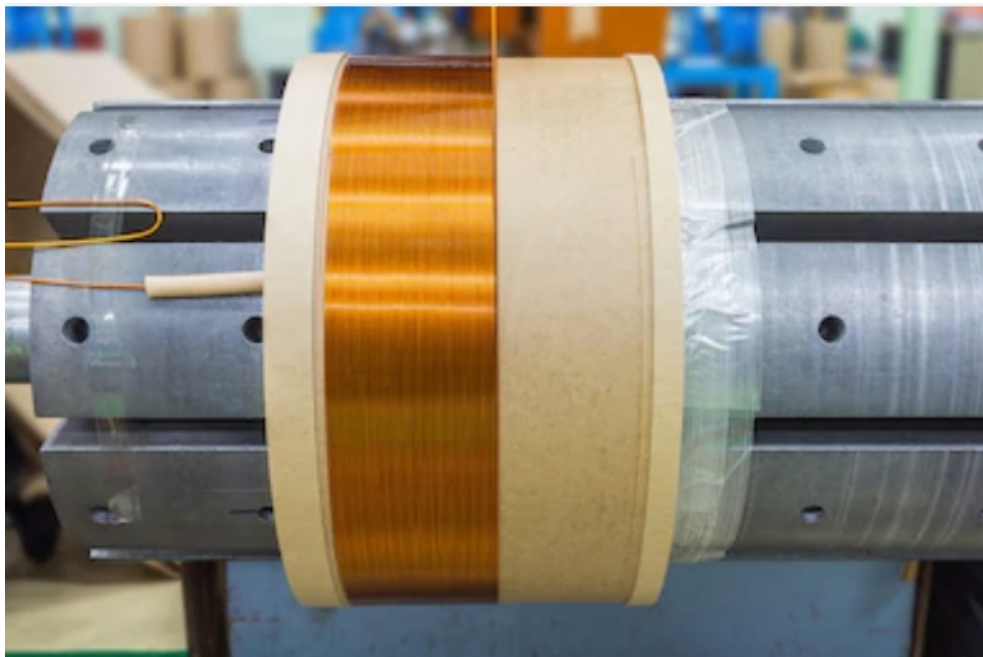
Examensarbetet implementerar en lösning till winder-applikationen med Sigmateks styrsystem och utvecklingsmiljö (LASAL). Framtagandet av den elektroniska kammen sker med verktyget *CAM-Designer* som finns integrerat i utvecklingsmiljön. Examensarbetet avser enbart den generella lösningen. Vid applicering av lösningen måste mekaniken anpassas till det enskilda driftfallet.

2 Teknisk bakgrund

Detta kapitel inleder med att beskriva olika typer av upprullning och exempel på tillämpningar. Detta kapitel beskriver även teorin bakom hårdvaran, mekaniken och mjukvaran som använts i examensarbetet. Detta för att få en bättre förståelse för implementeringen och resultatet.

2.1 Olika typer av spridning

Beroende på vad avsikten med upprullning av produkten är tillämpas olika former av spridning. En form av spridning sker vid lindning av transformatorer. Kopparråderna appliceras jämnt över lindningen utan att överlappa sig själv. Med begreppet *jämnt* menas även att kopparråderna ligger tätt. För varje varv som spolen roterar gör spridaren en förflyttning med samma längd som produktens bredd. Figur 1 illustrerar denna typ av spridning.



Figur 1. Transformatorlindning. Copyright: Matee Nursem [1]

En annan typ av spridning tillämpas när sytråd rullas upp. Vid analys av en sytrådsrulle bildar spridningen av sytråden ett mönster. När tråden rullas av löper tråden mellan rullens ändlägen. Slutsatsen som kan dras av detta är att tråden spridits ut över rullen med någon typ av intervall. Detta innebär att under ett givet antal rotationer av rullen löper tråden mellan rullens ändlägen.



Figur 2. Sytrådsrulle

För att kunna synkronisera förflyttningen av spridaren med avseende på upprullarens rotation behövs någon form av *Motion Control* (se kap.2.2).

2.2 Motion Control

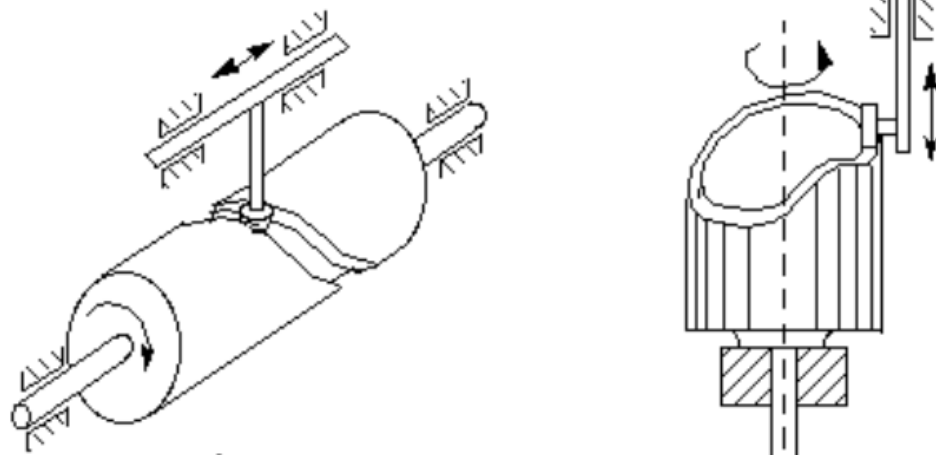
Motion Control [2] är ett delområde inom automation där mekanik styrs med hjälp av elmotorer men även pneumatik eller hydraulik. Med hjälp av ett styrsystem kan en elmotor utföra en förflyttning av en last med avseende på styrsignalen. Styrsignalen beräknas utifrån parametrar så som hur lång förflyttningen ska vara, vilken hastighet förflyttningen ska ha, hur motorn ska accelerera samt vilken profil accelerationskurvor får ha för att begränsa mekaniskt ryck.

Med hjälp av *Motion Control* kan mekaniska processer automatiseras. Exempel på automatiserade processer i industrin där *Motion Control* är ett inslag finns bland annat hos plockrobotar, packrobotar, sortering och transportband.

I detta examensarbete är *Motion Control* centralt. Med hjälp av ett styrsystem ihop med en elektronisk kam synkroniseras och styrs två elmotorers rörelse och position, i det här fallet stegmotorer.

2.2.1 Mekanisk kam

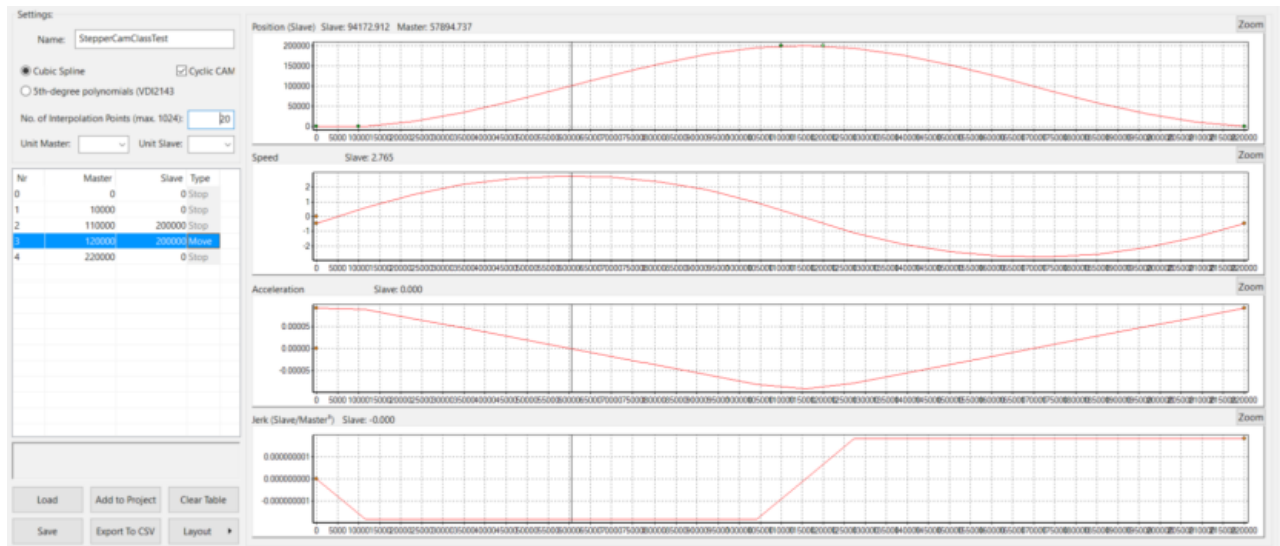
I mekaniska sammanhang definieras en kam som omformning av rörelse, exempelvis från en roterande rörelse till en linjär rörelse. Mekaniken bakom omformningen består oftast av två delar, "kammen" och "följaren". Kammen bestämmer följarens rörelse genom hur kontaktytan mot följaren är utformad. När kammen roterar rullar följaren över kammens yta och ändrar därmed position i x-led.[3]



Figur 3. Mekanisk kam[1]

2.2.2 Elektronisk kam

Principen bakom en elektronisk kam är den samma som en mekanisk. Men som benämningen *elektronisk kam* antyder implementeras den elektroniskt, i det här fallet som programkod. Den elektroniska kammen definierar hur slavaxeln (följaren) ska förhålla sig till huvudaxeln enligt en positionstabell. Positionstabellen utgör en kurva som interpolerats med hjälp av start och slutvärde för kammen. Vid exekvering av programkoden läses huvudaxelns position in. Slavaxelns position korrigeras enligt motsvarande värde i positionstabellen. Vid nästa exekvering återupprepas samma sekvens.



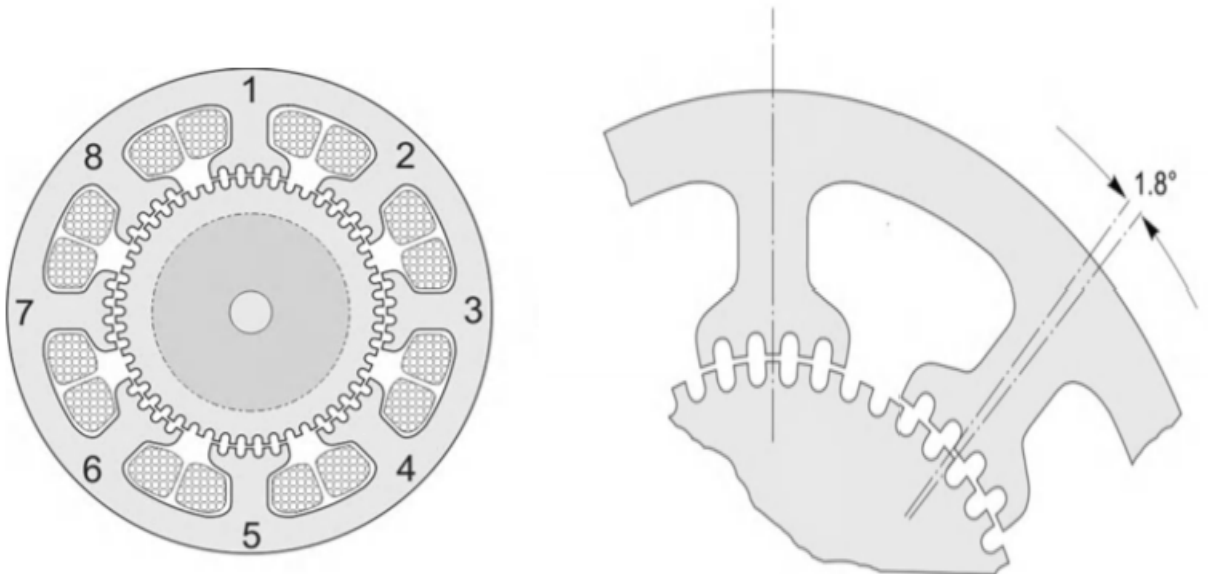
Figur 4. Elektronisk kam genererad med Sigmateks *CAM-Designer*

Fördelarna med en elektronisk kam är att den är relativt lätt att modifiera i förhållande till en mekanisk kam. Med nytt start- och slutvärde kan en ny tabell genereras. Den mekaniska kammen hade fått helt ersättas eller byggas om. Dessutom uteblir det mekaniska slitaget som skulle uppstått med en mekanisk kam.

2.3 Stegmotorer

Stegmotorer är en form av elmotorer som bygger på reluktansprincipen, till skillnad från exempelvis likströmsmotorer och asynkronmotorer som följer Lorentz Lag ($F = BIL$). Vridmoment från en reluktansmotor åstadkommer man genom ferromagnetiska materials tendens att vilja följa med magnetfält. Rotorn designas med poler som vill följa med magnetfältet statorlindningarna ger upphov till. Detta producerar ett vridmoment.[4]

Som namnet antyder roterar stegmotorn stegvis, där varje steg är en vinkelförflyttning. Förflyttningens längd beror på motorns konstruktion. I figur 5 illustreras ett exempel på en stegmotor med åtta lindningar som tillsammans bildar fyra statorfaser. Rotorn har 50 stycken tänder jämnt fördelade över varvet. Rotortänderna separeras med permanentmagneter. Genom att leda ström genom statorfaserna i en sekvens ger lindningarna upphov till ett magnetfält som får rotorn att rotera.[4]



Figur 5. Stegmotorns konstruktion [4]

Förflyttningen beräknas enligt ekvation 1:

$$\text{Step angle} = \frac{360^\circ}{(\text{rotor teeth}) \times (\text{stator phases})}$$

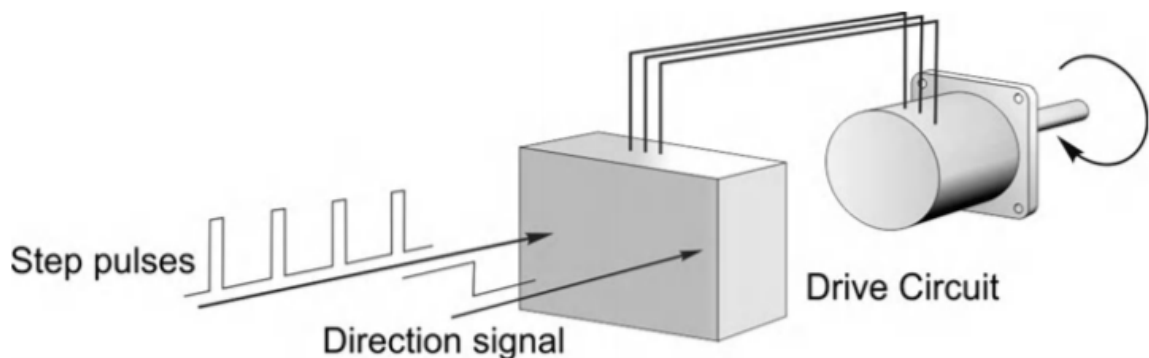
Ekvation 1. Beräkning av vinkelförflyttning per steg [4]

Vinkelförflyttningen är direkt proportionell mot antalet steg för ett varvs förflyttning. En vinkelförflyttning på 1.8° per steg motsvarar då 200 steg per varv.

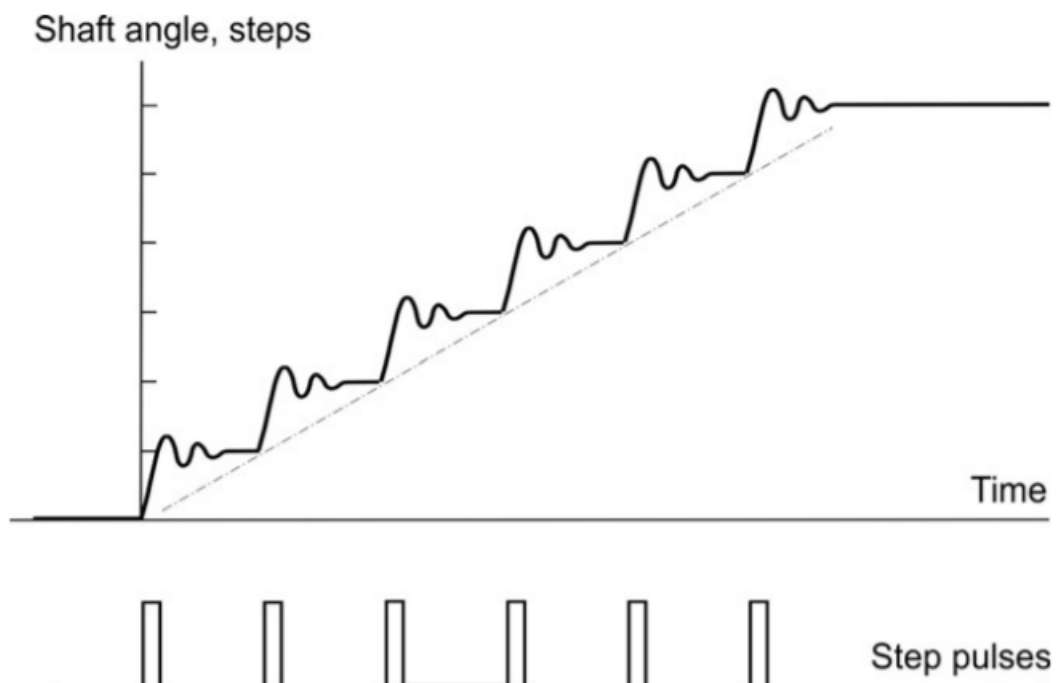
Det är svårt att tillverka stegmotorer med fler än 50-100 tänder. Det är även opraktiskt att ha mer än 4 eller 5 statorfaser. Vid applikationer där högre upplösning än 200 steg per varv krävs tillämpas därför en teknik som kallas *stegdelning*. Kort sagt innebär det att stegen delas upp i delsteg. Det är inte ovanligt med upp till 100 delsteg per steg. För en stegmotor med 200 steg per varv ger detta 20000 delsteg. [4]

2.3.1 Drivsteg

För att styra stegmotorn krävs ett drivsteg. Drivsteget tar emot ett inkommande pulståg från ett styrsystem eller mikrokontroller. Pulstågets längd och frekvens avgör hur långt respektive hur fort motorn ska rotera. En separat signal med en separat anslutning för att bestämma rotationsriktning skickas även från styrsystemet. Detta kallas styrning med puls/riktning. Drivsteget tar emot respektive signaler och omvandlar dessa till sekvenser. Sekvensen avser i vilken ordning statorlindningarna magnetiseras. [4]



Figur 6. Inkommande pulståg till ett drivsteg [4]



2.4 PLC

En PLC (Programmable Logic Controller) är en dator som är anpassad för industriellt bruk. PLC:n används bland annat för att styra maskiner eller industriella processer. PLC:n utvecklades som ett alternativ till reläer och har sedan dess intåg i industrin blivit ett oundgängligt verktyg.[5][6]

Generellt består en PLC följande delar[6]:

1. **CPU (Processor):** CPU:n exekverar programkod, utför beräkningar samt läser ingångar och sätter utgångar.
2. **IO-moduler (Input/Output):** Används som gränssnitt för att läsa in analoga eller digitala värden samt sätta ut signaler. In- och ut signaler kan vara analoga eller digitala.
3. **ROM:** Lagringsutrymme för instruktioner.
4. **Fältbussar:** Över fältbussarna sker kommunikationen mellan **CPU** och **IO-moduler** men även mellan PLC:er. Olika protokoll kan användas för kommunikationen så som **SERCOS**, **EtherCAT**, **PROFIBUS** och **VARAN**.

PLC:n arbetar cykliskt så till vida att vad den är programmerad till att utföra upprepas med förutbestämt intervall, en så kallad ”cykeltid”. Sekvensen brukar vara den följande[6]:

1. Ingångar läses.
2. Beräkningar utförs.
3. Utgångar sätts.

2.4.1 VARAN

VARAN står för Versatile Automation Random Access Network och är ett Ethernet-baserat protokoll för kommunikation i realtid. Protokollet följer en struktur med en överordnad nod, en så kallad *Manager* och en eller flera undernoder, så kallade *klienter*. Managern ser nätverket som ett 4 GB stort minnesutrymme där klienterna är allokerade minnesceller. Nätverket stödjer upp till 65280 klienter.[7][8][9]

Managern hanterar informationsflödet i nätverket med hjälp av läs- och skrivinstruktioner. Varje överföring av data i nätverket initieras av managern. Kollision av paket förekommer därför aldrig.[8]

Instruktionsuppsättningen i VARAN-kommunikationen är enkel och består endast av fyra kommandon[8]:

1. **Memory Read:** Managern läser data från klienten.

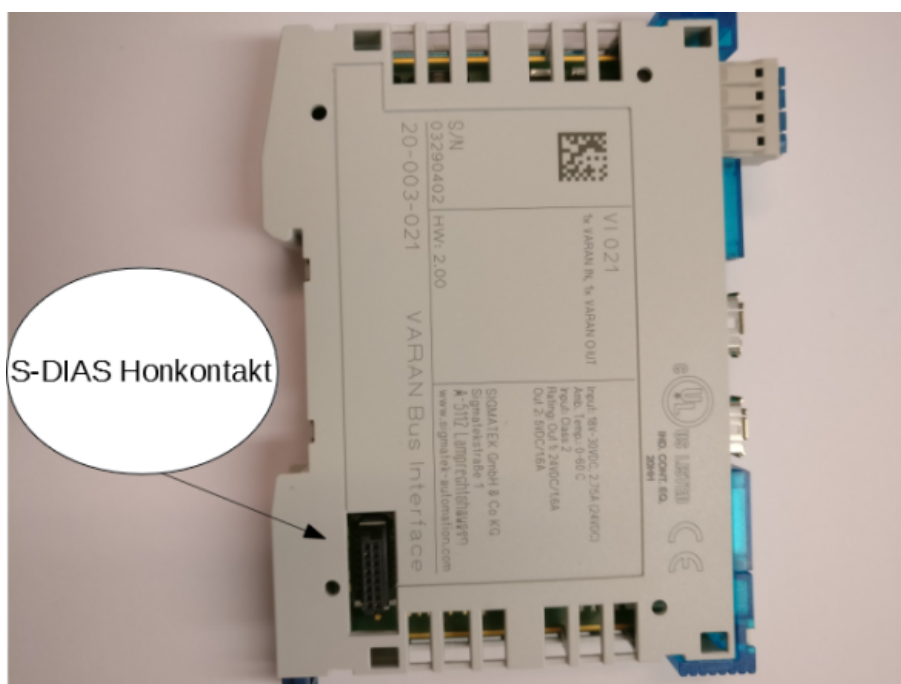
2. **Memory Write:** Managern skriver data till klienten.
3. **Global Write:** Alla klienter skrivs till samtidigt. Instruktionen används för att synkronisera alla klienter och för att göra en global återställning.
4. **Foreign Package Request/Response:** Initierar dataöverföring via VARAN-nätverket där datapaketet som skickas har ett annat protokoll, så som TCP/IP.

VARAN använder sig av en busscykel vid överföring av data. Cykeltiden är under 100 μ s. Vid varje ny busscykel skickas ett "Global Sync"-kommando för att synkronisera alla noder med managern. Busscykeln delas in i delmoment, så kallade "tasks" med olika prioritet. Högst prioritet har den isokrona task:en där uppdatering av exempelvis börvärden och inläsning av ärvärden i momentreglering sker. Data kan även skickas mellan manager och nod asynkront under busscykeln. Asynkron direktåtkomst till en nod sker i form av ett avbrott.[9]

Felkorrigering då en nod inte svarar vid ett anrop (timeout) upprepas upp till två gånger under samma busscykel. VARAN lämpar sig därför i tidskritiska applikationer utan att inverka på pålitligheten.[9]

2.4.2 S-DIAS Bus

S-DIAS Bus är Sigmateks implementation av VARAN-protokollet. Skillnaden ligger i hårdvaruimplementationen av bussystemet. Kommunikationen mellan modulerna sker genom sammankoppling av han- och honkontakter på sidorna av modulerna.[10]



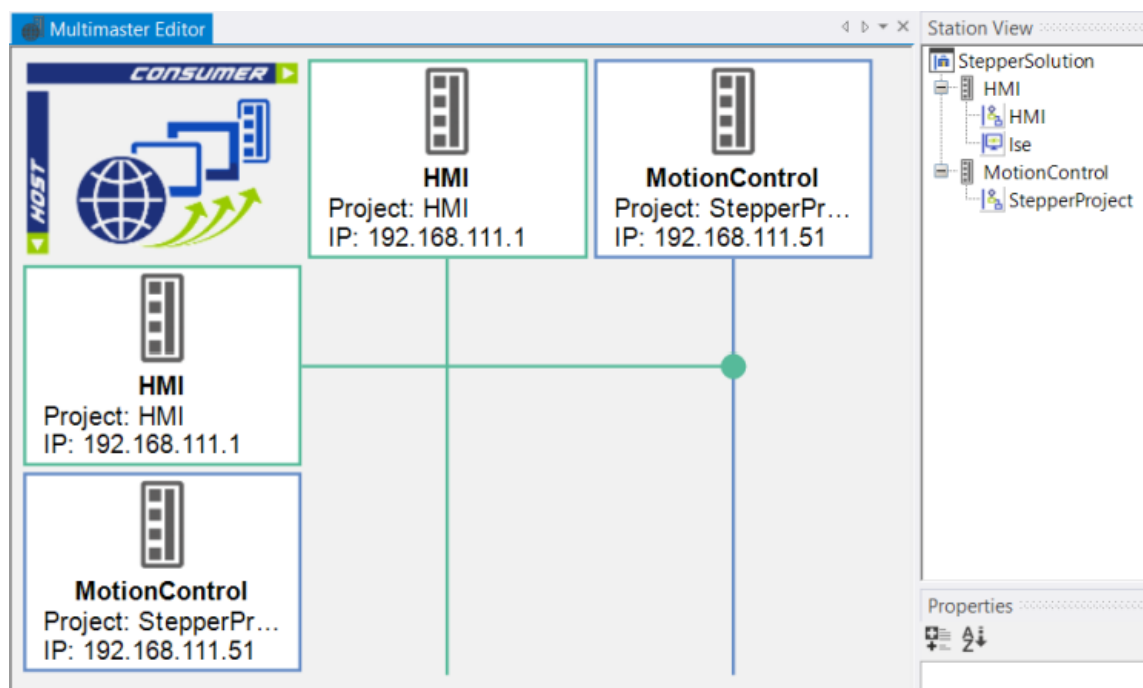
Figur 8. Modul från Sigmatek med en S-DIAS buss.

2.5 LASAL

LASAL är Sigmateks utvecklingsmiljö för programmering av PLC och skapande av HMI-visualisering. LASAL delas in i tre delar:

1. Machine Manager
2. Class 2
3. Screen

Machine Manager används för att knyta ihop projekt och konfigurera anslutningar mellan HMI och PLC eller rentav processer. En lång processkedja kan delas upp i subsystem, så kallade *stationer*. Machine Manager bestämmer sedan över informationsflödet i processen, enkelt sagt, vilken station som får kommunicera med en annan station. Att dela upp en lång processkedja i stationer gör det lättare att få en överblick över projektet. Varje station har ett LASAL Class-projekt. Stationen som berör visualiseringen har även ett LASAL Screen-projekt.



Figur 9. LASAL Machine Manager

2.5.1 Class 2

I LASAL Class 2 sker programmeringen. Class 2 stödjer de standardiserade språken IEC 61131-3 så som Ladder, Sequential Flow Chart och Strukturerad Text men i det här examensarbetet har endast strukturerad text använts.

Lasal Class 2 stödjer objektorienterad programmering med klasser, arv och polymorfism. Klasser representeras som objekt och ansluts till andra objekt grafiskt i så kallade nätverk. Klasserna kan innehålla globala metoder och privata metoder, tillstånd, globala variabler och privata variabler, servrar och klienter. I figur 10 representeras ett objekt av klassen *InitOutput*. InitOutput-klassen är ansluten till en hårdvaruutgång. Vid exekvering skriver klienten en etta till sin utgång, varav hårdvaruutgångens server läser detta värde och

ställer den fysiska hårdvaruutgången till *ett* och en LED-lampa på modulen tänds.

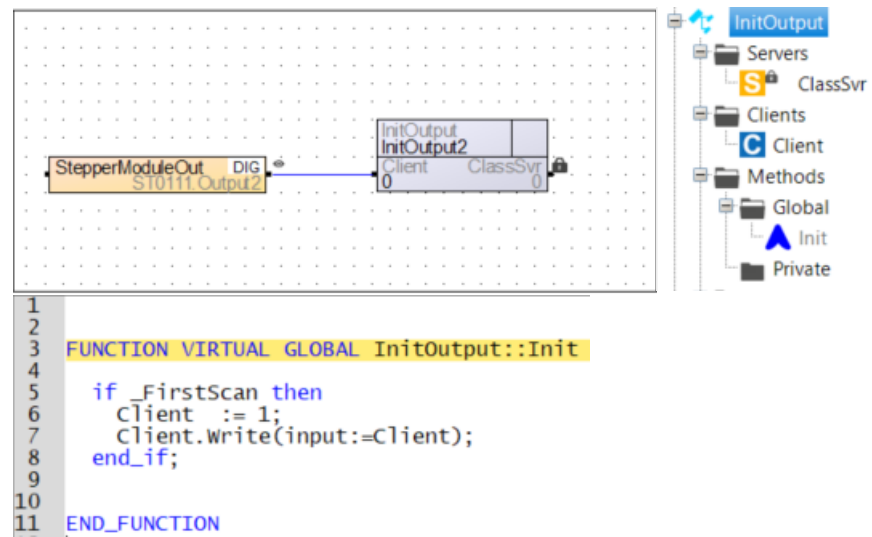
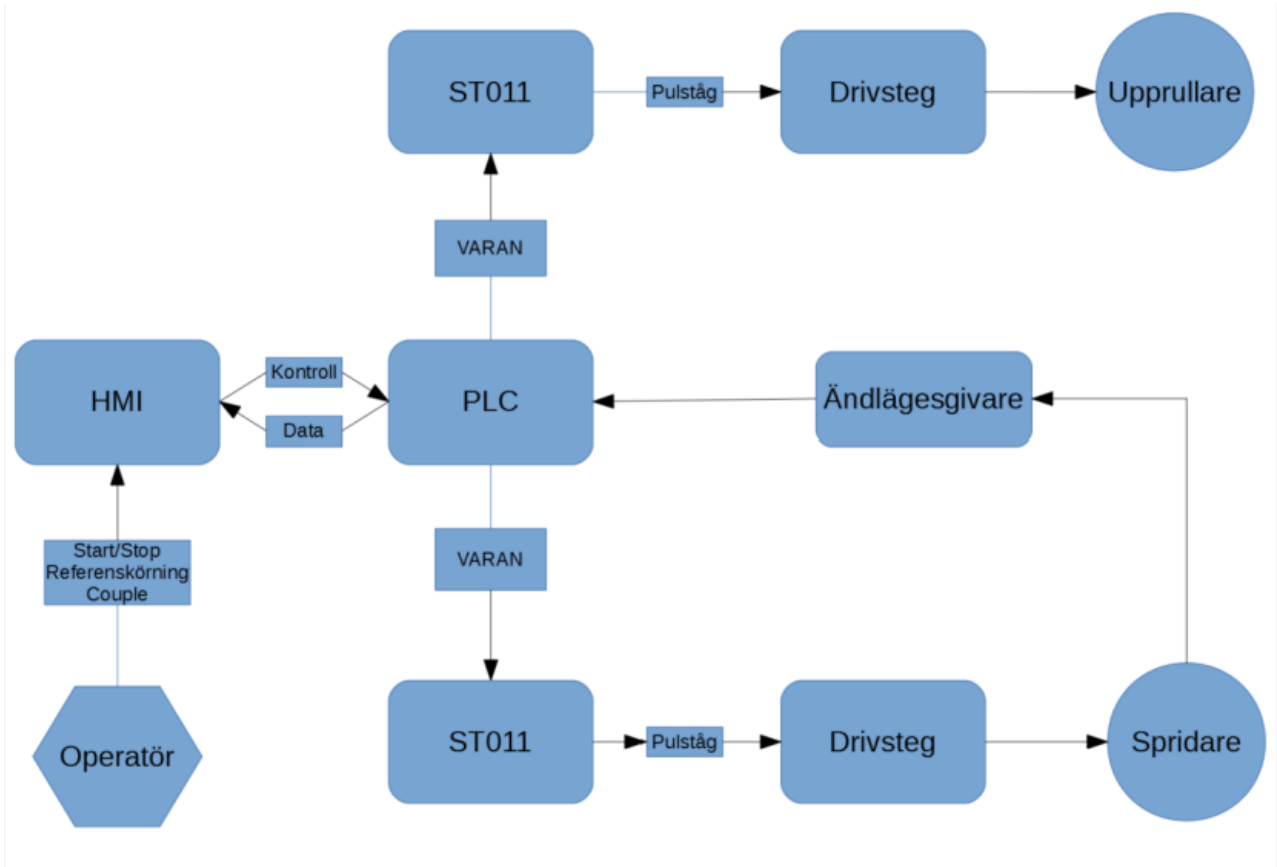


Fig 10. LASAL Class 2 Kodexempel

3 Konstruktion av prototyp

3.1 Funktionsbeskrivning

Funktionsbeskrivningen i figur 11 sammanfattar systemet. Operatören startar och stoppar applikationen via ett HMI (operatörsgränssnitt). Operatören kan även göra en referenskörning av spridaren samt synkronisera upprullare med spridare. Operatörsgränssnittet kommunicerar med en PLC över Ethernet. PLC:n styr i sin tur två stycken ST011 stegmotorkort över VARAN som genererar pulståg och riktning. Pulstågen och riktningssignalerna tas emot av drivstegen som omvandlar dessa till respektive stators magnetiseringssekvenser.



Figur 11 Funktionsbeskrivning av systemet

3.2 Mekaniskt utförande

För att kunna återskapa det verkliga driftscenariot i en testmiljö byggdes en prototyp. Prototypen är byggd av enklare mekanik och är inte tänkt att användas i ett annat sammanhang än detta examensarbete som demonstrationssyfte. Allt material vid byggandet av prototypen har tillhandahållits av SIGBI. Prototypen består av upprullare, spridare, två stegmotorer med tillhörande drivsteg, ändlägesgivare, styrsystem, HMI och strömförsörjning. Givetvis ingår även kablage, kopplingsplintar och DIN-skenor. Prototypens layout planerades i Googles 3D-verktyg *Google SketchUp*.

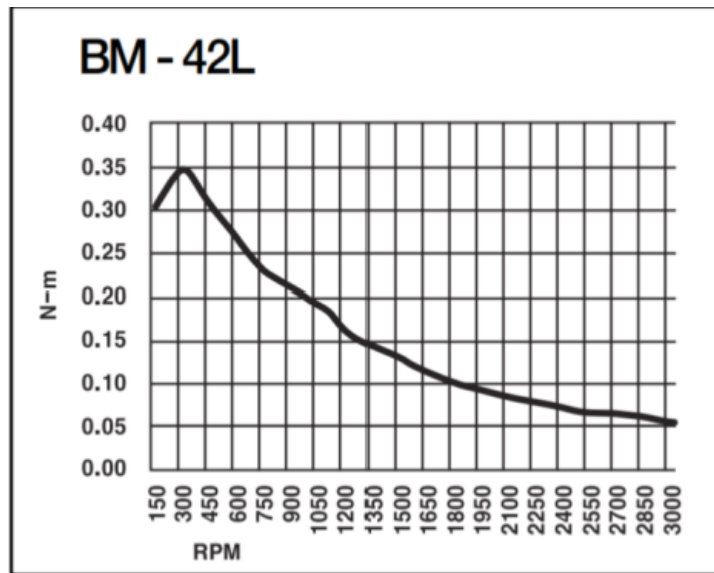


Figur 12. Prototyp

3.2.1 Fastech

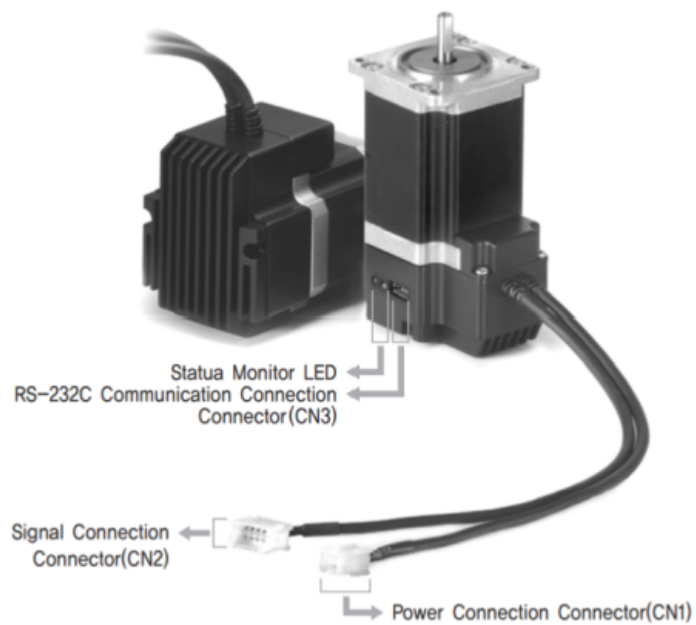
Stegmotorerna som använts i examensarbetet för att styra upprullare och spridare är tillverkade av Fastech. Valet av elmotorer föll på vad som fanns tillgängligt hos SIGBI. Stegmotorer är kostnadseffektiva eftersom priset är lågt och de är enkla att styra. Stegmotorer är även robusta och mekaniskt enkla då rotorn inte har några lindningar, kommutatorer eller borstar. [4]

Ezi-Step BM-42L är den näst största motorn i BM-42 serien[11]. Den här motorn driver linjärenheten. Motorn är ansluten till en Ezi-Step Bipolar Drive. Drivsteget är konfigurerat för puls/riktning och upplösningen är satt till 4000 pulser per varv. Upplösningen kan ändras manuellt med DIP-switcharna på drivsteget. I figur 13 illustreras momentkurvan för stegmotorn. Maximalt moment ges vid ≈ 320 varv per minut. Vid högre varvtal försvagas motorn kraftigt. Denna momentkaraktistik gäller generellt för stegmotorer[12].



Figur 13. Momentkurva BM-42L [11]

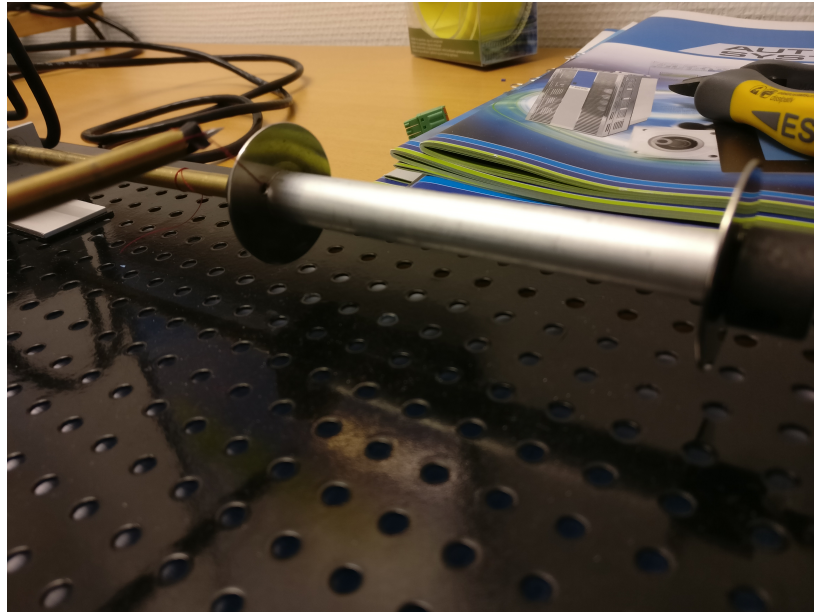
Ezi-Step BT-42XL är den största motorn i BT-42 serien.[13] Den här motorn driver upprullaren. Drivsteget är integrerat med motorn. Motorn behöver därför endast kabel för matningsspänning och signalkabel från styrsystemet. Det är fördelaktigt eftersom det sparar plats vid exempelvis skåpbygge.



Figur 14. Ezi-Step BT-42XL [13]

3.2.2 Upprullare

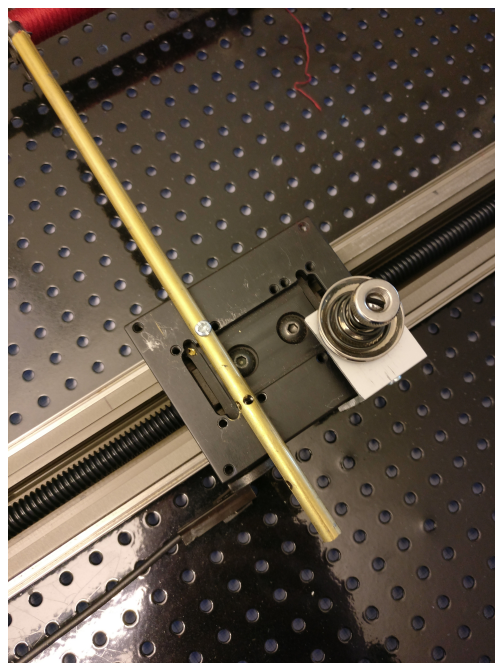
Upprullaren är ett mässingsrör monterat mellan två vinkeljärn. Spolen träs på mässingsröret och kläms fast. Spolen är 10 cm lång och har en diameter på 12 mm. Detta ger en omkrets på ≈ 37.68 mm vilket innebär att för varje varv upprullaren roterat har spolen färdats 37.68 mm.



Figur 15. Spole monterad på upprullare

3.2.3 Spridare

Spridaren är en linjärenhet som drivs av en Ezi-Step BM-42L stegmotor. Linjärenheten har en skruv med stigning 2 mm vilket innebär att när stegmotorn roterat ett varv motsvarar det 2 mm förflyttning. På linjärenheten är en spridararm fixerad med en synål i spetsen som tråden träs igenom. För att undvika att tråden trasslar in sig löper tråden genom en spännmekanik. Spännmekaniken klämmer åt tråden med hjälp av en fjäder. Motståndet justeras för att ge rätt spänning i tråden.



Figur 16. Spridararm och spännmekanik monterat på linjärenheten

Stegmotorernas positioner är inte återkopplade vilket innebär att informationen om hur långt motorerna har roterat försvinner om programmet startas om eller PLC:n blir strömlös. Spridararens position återställs vid omstart genom att operatören gör en referenskörning mot en ändlägesgivare för att hitta startpositionen.

Ändlägesgivaren är en brytande givare av modell KL3048 från Hoerbiger. Givaren matas med 24 volt från en av styrsystemets digitalutgångar. Signalen från givaren är återkopplad till en digitalingång på styrsystemet. När ett magnetfält appliceras på givaren öppnas brytaren. Styrsystemet känner då av att digitalingången går från hög till låg (från 24 volt till 0 volt). Detta indikerar att linjärenheten har nått sin startposition och styrsystemet stoppar motorn.

3.3 CP102, ST011, KL090 och SE051

I det här examensarbetet användes Sigmateks PLC CP102 ihop med två stycken ST011 stegmotormoduler för generering av pulståg. En KL090-modul användes som strömbrygga och en SE051 Ethernet-switch för kommunikation mellan PLC och HMI. Sigmateks moduler är väldigt kompakta. Anslutning mellan modulerna sker över S-DIAS med hjälp av kontakten på sidan av modulen. Det är därför väldigt lätt att expandera in- och utgångar genom att koppla på en extra modul. Modulerna monteras sedan på en DIN-skena.

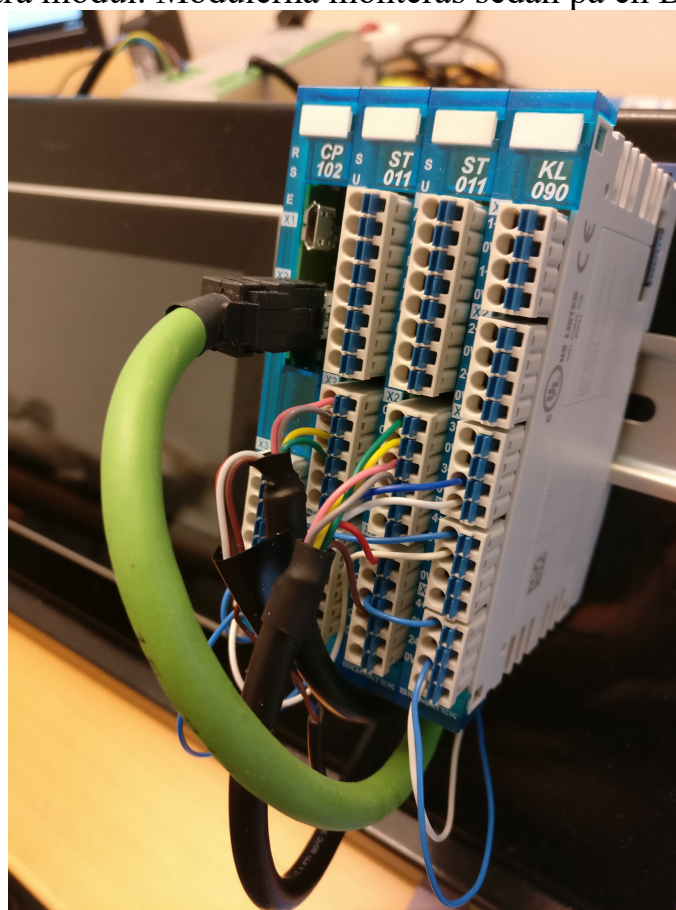


Fig 17. Montering av moduler på din-skena

CP102 är en processor-modul med anslutningar över Ethernet, CAN-bus och USB. Enheten har en ARM-processor och 256 MB DDR3 RAM. Ett 256 MB lagringsutrymme för program finns i form av ett flashminne. En del av RAM-minnet är avbrottsskyddad, det vill säga att information som hålls i den delen av minnet inte försvinner vid ett strömavbrott. Detta möjliggörs genom att informationen i RAM-minnets avbrottsskyddade del kopieras till flashminnet. [10]

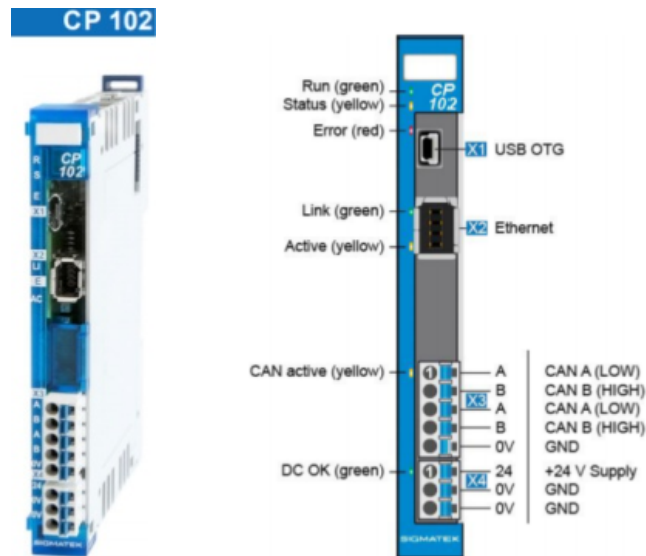


Fig 18. CP102 med beskrivning av in- och utgångar [10]

ST011 är Sigmateks stegmotor-modul för styrning av stegmotorer. Modulen har en utgång med styrsignaler. Ingångar finns för återkoppling av motorn med inkrementell encoder över RS422-protokollet.[14] Det finns även två digitala in- och utgångar. Dessa kan användas för enkel logik så som inläsning av ändlägesgivare.

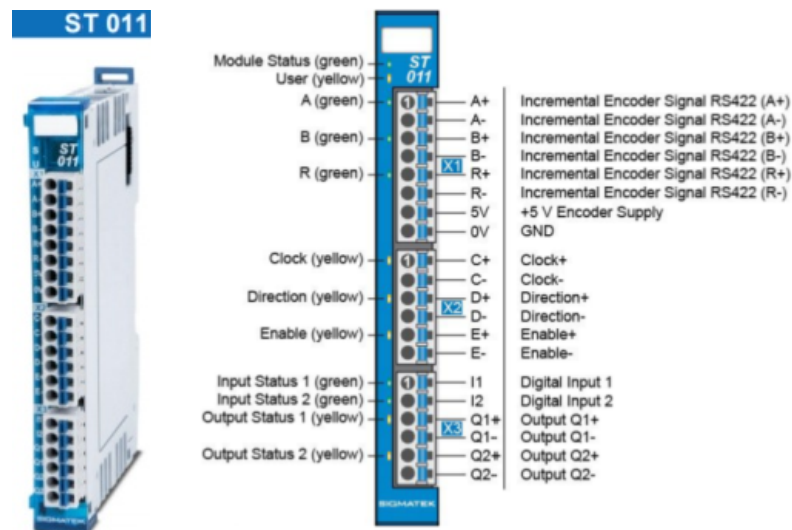


Fig 19. ST011 med beskrivning av in- och utgångar [14]

3.4 HMI – ETT 732

ETT732 är en av Sigmateks operatörspaneler. Panelen har en 7 tums ”multi touch”- display, integrerad ARM-processor, RAM-minne och ett flash-minne för lagring. Anslutningar i form av CAN-bus, USB, Ethernet och RS232 finns tillgängliga. Ethernet används i detta examensarbete för kommunikation med CP102 via en ethernet-switch (SE051). Panelen används som ett gränssnitt mellan operatör och maskin eller process. Med en interaktiv visualisering kan operatören starta och stoppa en process eller maskin. Visualiseringen skapas i LASAL Screen. [15]



Figur 20. ETT 732 HMI [15]

4 Implementering av applikationen

4.1 Konfigurering av anslutningar

För att PC, PLC och HMI ska kunna kommunicera med varandra måste anslutningarna konfigureras. Kommunikationen sker över Ethernet. Alla tre delar måste befinna sig på samma nätverk och ha samma subnätmask. Eftersom PLC:n varit använd tidigare var den inte konfigurerad med den förinställda adressen som den har då den levereras från tillverkaren. Den förinställda adressen står angiven i databladet. För att ta reda på adressen anslöts PLC:n till datorn via USB. När den seriella förbindelsen mellan dator och PLC var upprättad kunde filen för konfigurering, **autoexec.lsl**, öppnas på PC:n. Den ställda IP-adressen för PLC:n kunde där efter läsas av. Datorns nätverkskort gavs en ny statisk adress enligt tabell 1.

Konfigurering	PC	PLC	HMI
IP-Adress	192.168.111.10	192.168.111.51	192.168.111.1
Mask	255.0.0.0	255.0.0.0	255.0.0.0

Tabell 1

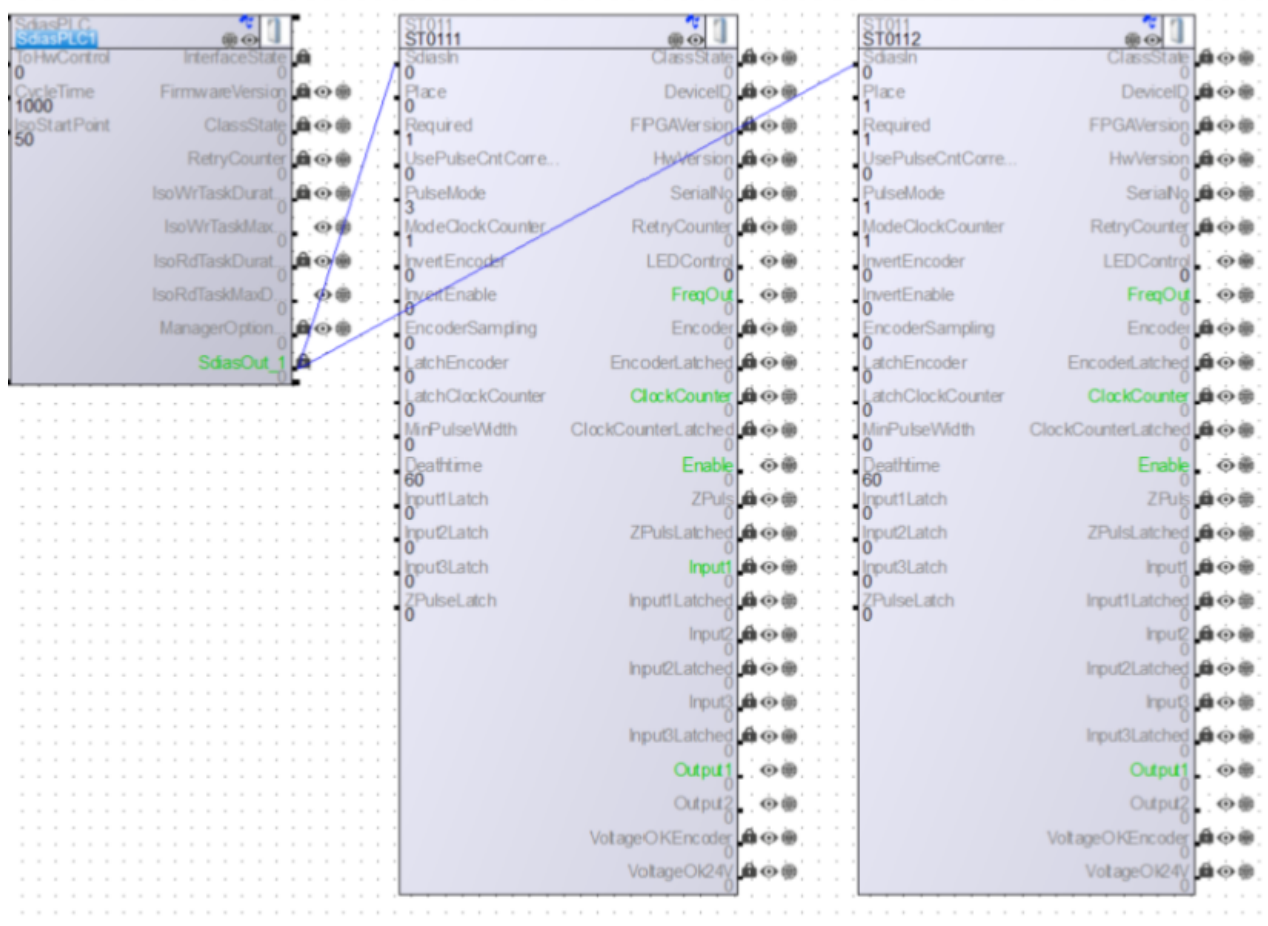
4.2 Program

Implementationen av programmet gjordes i LASAL Class 2. Programmet avser delen av automationslösningen som berör styrningen och synkroniseringen av upprullare och spridare. Delar av programmet bygger på Sigmateks *Motion*-bibliotek med tillhörande klasser. Vissa av klasserna har ärvts enligt den objektorienterade principen. De ärvda klasserna har sedan anpassats för den här applikationen. All programkod är skriven i strukturerad text.

Programmet i sin helhet består av tre nätverk:

- HW_Network
- CoupleNet
- StepperNet

HW_Network innehåller grafiska representationer (objekt) av den fysiska hårdvaran. Nätverket ihop med objekten utgör ett mjukvarugränssnitt mot hårdvaran. HW_Network genereras automatiskt vid varje nytt projekt. Detta genom att LASAL kan läsa av vad för hårdvara datorn är ansluten till. I detta fall CP102 och efterföljande moduler som är anslutna via s-dias-bussen. Båda ST011-objekten i figur 21 knyts till varsitt `_LMCAxisStepper`-objekt i StepperNet.



Figur 21. HW_Network med objekt som representerar hårdvaran

4.2.1 Styrning

I StepperNet finns objekten som styr spridare och upprullare. Eftersom både spridare och upprullare drivs av stegmotorer används samma klasser. `_LMCAxisStepper` är klassen som används för styrning av stegmotorerna. Klassen är placerad som ett objekt i nätverket. Objektet anropar ST011-objektet i HW_Network. Anropen ger `_LMCAxisStepper` information om hur många pulser som skickats och vilken position motorn i så fall bör ha, samt vilken hastighet och acceleration motorn roterar med. Anropen kan även bestå

av kommandon. Ett kommando kan vara att motorn ska göra en förflyttning med en viss längd och hastighet.

Parametrarna i `_LMCAxisStepper` ställs med avseende på vilken stegmotor som är ansluten till objektet. Spridarens drivsteg är inställt på 4000 pulser per varv. Motsvarande värde måste därför matas in i `sExtUnitsStepper`. Ett varv motsvarar en förflyttning av linjärenheten på 2mm. Detta värde matas in i `sIntUnitsStepper`.

Parametern `sMaxRPM` bestämmer det maximala varvtalet motorn får rotera med. Detta bestämmer i sin tur den maximala frekvensen av utgående pulståg från ST011-modulen. Drivstegen tolererar pulståg upp till 500 kHz.

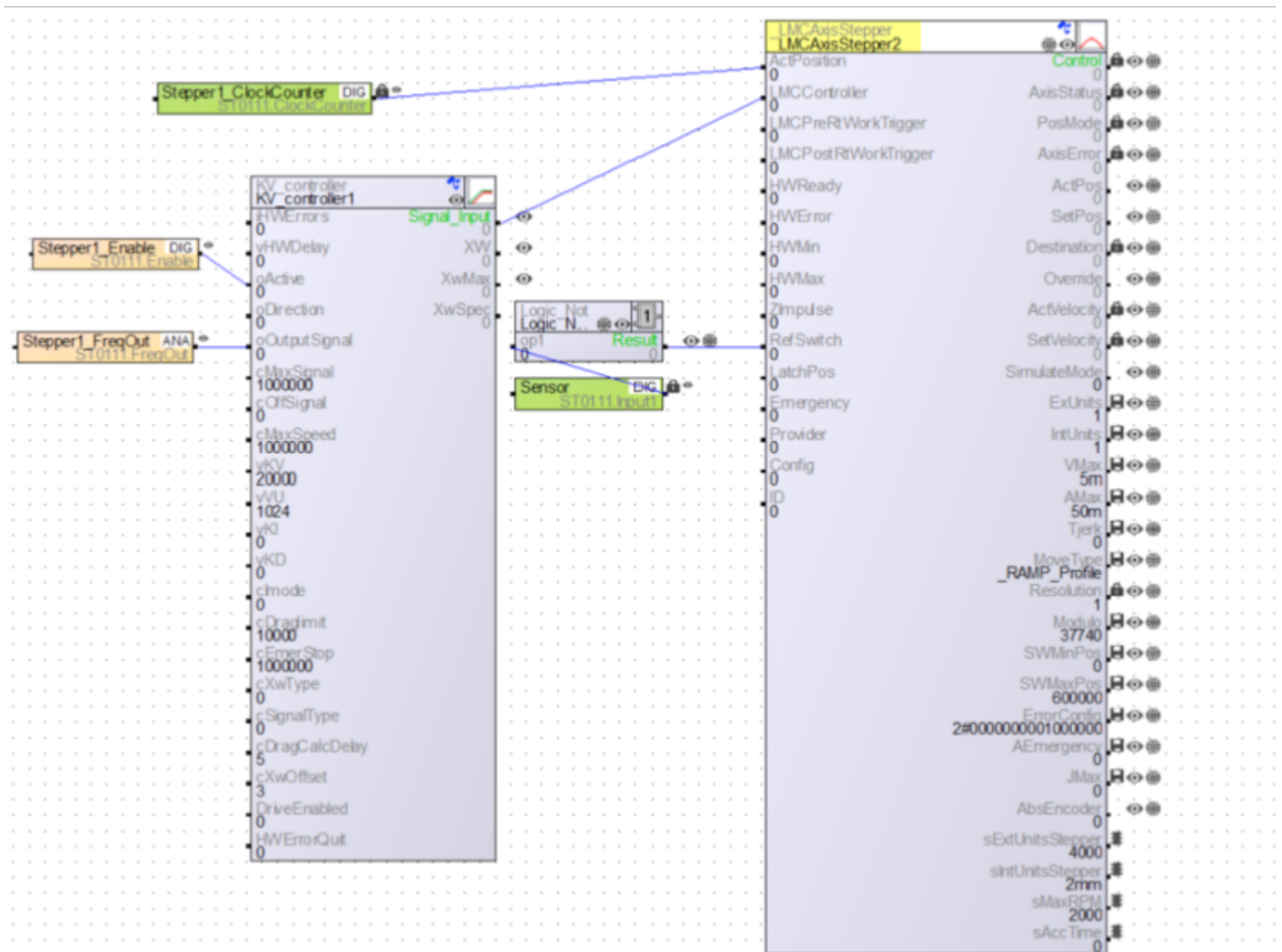


Fig 22. `_LMCAxisStepper` i StepperNet. Objekten i StepperNet som kontrollerar spridare.

`_LMCAxisStepper` är resultatet av ett arv från Sigmateks `_LMCAxis`-klass ur `Motion`-biblioteket. Arv innebär att alla egenskaper från den ärvda klassen följer med. Den nya klassen kan sedan fritt skriva över metoder som ärvt samt utöka funktionaliteten utan att påverka basklassen. Basklassens metoder kan fortfarande anropas. Detta gör arv-funktionen till ett kraftigt verktyg.

Alla metoder från arvet som hanterar parametrar så som position, hastighet, acceleration och ryck skrevs över i `_LMCAxisStepper`. Anledningen till detta

var för att storheterna behövdes räknas om till motsvarande pulser. När ett förflyttningskommando skickas med parametrar som specificerar hur förflyttningen ska ske beräknas parametrarnas motsvarighet i pulser. Hastigheten omvandlas exempelvis från mm/s till pulser/s. Omvandlingen sker med hjälp av funktionen *CalcPulses* som illustreras i figur 23.

```
FUNCTION _LMCAxisStepper::CalcPulses
VAR_INPUT
    Param    : DINT;
END_VAR
VAR_OUTPUT
    OutParam : DINT;
END_VAR

    OutParam := (Param*sExtUnitsStepper)/sIntUnitsStepper;
END_FUNCTION
```

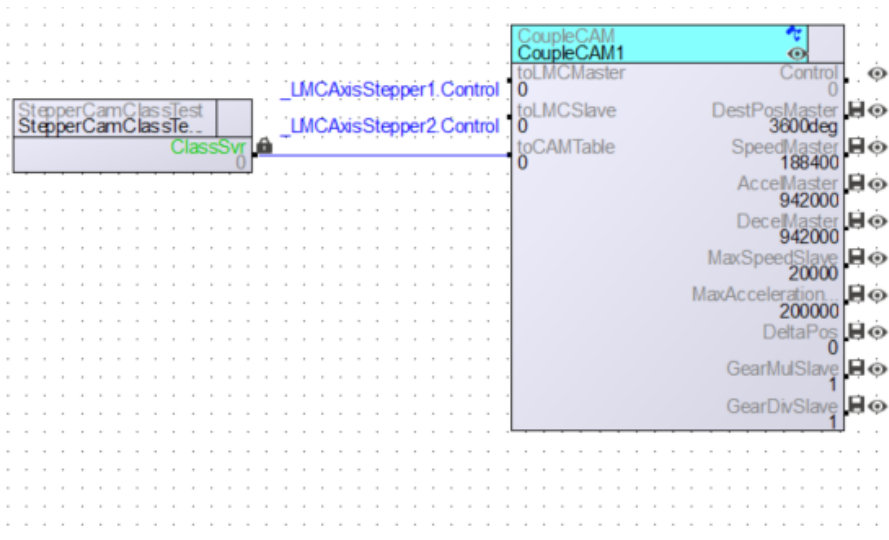
Figur 23. Funktionen *CalcPulses* som omvandlar inkommande parameter till motsvarande pulser

4.2.2 Synkronisering

_LMCAxisStepper-objekten i StepperNet ansluts till ett *CoupleCam*-objekt i CoupleNet. Objektet definierar vilken av *_LMCAxisStepper*-objekten som är huvudaxel respektive slavaxel. Spridaren är i detta fall slavaxel åt upprullaren. *CoupleCam*-klassens metoder är följande:

- PowerOn
- PowerOff
- MoveReference
- StepCoupleAxis
- MoveEndless
- MoveAbsolute

PowerOn samt *PowerOff* används för att ge motorerna spänning respektive göra motorerna spänningslösa. *MoveReference* är metoden som används för referenskörning av slavaxeln. *MoveEndless* och *MoveAbsolute* är kommandon för förflyttning av huvudaxeln. Antingen att huvudaxeln rör sig tills ett manuellt stopp från operatören sker med *Stop Move* eller tills den nått den angivna positionen. *StepCoupleAxis* är metoden som används för att koppla samman slav med huvudaxel enligt den elektroniska kam-kurvan.



Figur 24. Ett "CoupleCam"-objekt i "CoupleNet"

Vid anropet av *StepCoupleAxis*-metoden skickas ett kommando till slavaxelns *_LMCAxisStepper*-objekt. Kommandot innehåller bland annat slavens maximala hastighet och acceleration och en pekare till positionstabellen. Positionstabellen innehåller värdena som utgör den elektroniska kammen och finns lagrade i *StepperCamClassTest*-objektet. För att göra kammen cyklisk sätts ett modulo-värde för huvudaxelns position. När huvudaxeln når detta värde börjar kammen om från det första värdet i positionstabellen.

```

FUNCTION VIRTUAL GLOBAL CoupleCAM::StepCoupleAxis
VAR_OUTPUT
dRetcode : DINT;
END_VAR
VAR
retcode : _LMCAXIS_CMDERROR;
END_VAR

retcode := toLMCSlave.CoupleCAM(
MaxVel:=MaxSpeedSlave, MaxAcc:=MaxAccelerationSlave,
DeltaPosMaster:=DeltaPos, pMaster:=toLMCMaster.ReadMasterInfo(usDelay:=0),
GearMulMaster:=100, GearDivMaster:=100, Modulo:=202500,
GearMulSlave:=GearMulSlave, GearDivSlave:=GearDivSlave,
OffsetSlave:=0, Mode:=0, pCurveTable:=toCAMTable.Read(), MaxJerk:=0);

if retcode = 0 then
dRetcode := 1;
else
dRetcode := -1;
end_if;
END_FUNCTION

```

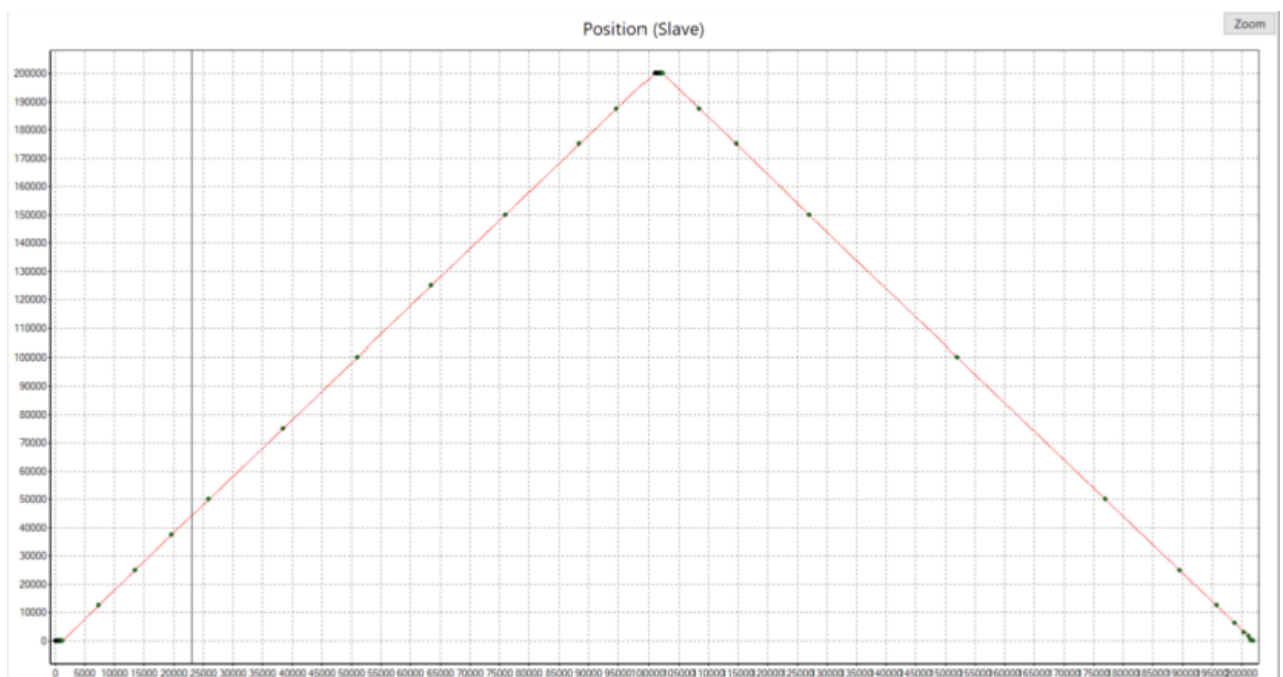
Figur 25. StepCoupleAxis-metoden

4.3 Den elektroniska kammen

Målet med upprullningen är oftast att produkten appliceras jämnt över vindan eller spolen. Det är alltså önskvärt att ytan som utgörs av den upprullade produkten inte blir bulig. För att åstadkomma en jämn upprullning måste därför spridaren förflyttas kontrollerat. Hastigheten bör vara konstant under förflyttningen av spridaren. Den mekaniska begränsningen blir särskilt

märkbar när spridaren nått vindans ändlägen och ska byta färdriktning. Systemet måste bromsa in spridaren och sedan accelerera motorn åt motsatt riktning. Systemet brukar överkompensera för att hinna med vändningen vilket resulterar i en översläng av hastigheten. Detta i sin tur resulterar i en ojämn förflyttning av spridaren.

Idealet hade varit att spridaren vid vändningen omedelbart får rätt hastighet. Detta är förstås omöjligt i en fysikalisk mening eftersom oändlig acceleration hade behövts. Men styrningen kan optimeras för att nå ett resultat som är tillräckligt bra. I figur 26 illustreras den färdiga kammen som tillverkats med CAM-Designer. Kammen importerar sedan i LASAL Class 2-projektet som ett objekt med och ansluts till CoupleCam-objektet.



Figur 26. Interpolerad kam-kurva

För att fördela produkten jämnt, i det här fallet sytråd delades spolen upp i åtta delar. Vid start av applikationen väntar spridaren vid startpositionen ett tiondels varv. Spridaren löper sedan linjärt över spolen tills den når det andra ändläget. Spridaren står stilla under en tiondels rotation av spolen och återvänder sedan till startpositionen. Under en förflyttning av spridaren till ett ändläge roterar spolen tio varv. Den här sekvensen återupprepas till applikationen stoppas, antingen då upprullaren har nått sin position eller vid ett manuellt stopp.

Spridarens sekvens förskjuter sytrådens placering med en åttondel av omkretsen för varje förflyttning mellan ändlägena. Detta gör att sytråden sprids ut jämnt över spolen utan bulor.

Inmatningen av kontrollpunkterna för axlarnas position görs i pulser. Från startvärdet står slavaxeln stilla under en åttondels rotation av huvudaxeln. Detta motsvarar 1250 pulser ($10000/8$). Under följande tio rotationer av

huvudaxeln roterar slavaxeln 50 varv. Detta motsvarar i sin tur 200 000 pulser och en total förflyttning på 100mm.

Cubic Spline Cyclic CAM

5th-degree polynomials (VDI2143)

No. of Interpolation Points (max. 1024):

Unit Master: Unit Slave:

Nr	Master	Slave	Type
0	0	0	Stop
1	100	0	Stop
2	200	0	Stop
3	202	0	Stop
4	300	0	Stop
5	400	0	Stop
6	500	0	Stop
7	600	0	Stop
8	700	0	Stop
9	800	0	Stop
10	900	0	Stop
11	1250	0	Speed
12	7375	12500	Speed
13	13500	25000	Speed
14	19625	37500	Speed
15	26000	50000	Speed
16	38500	75000	Speed
17	51000	100000	Speed
18	63500	125000	Speed

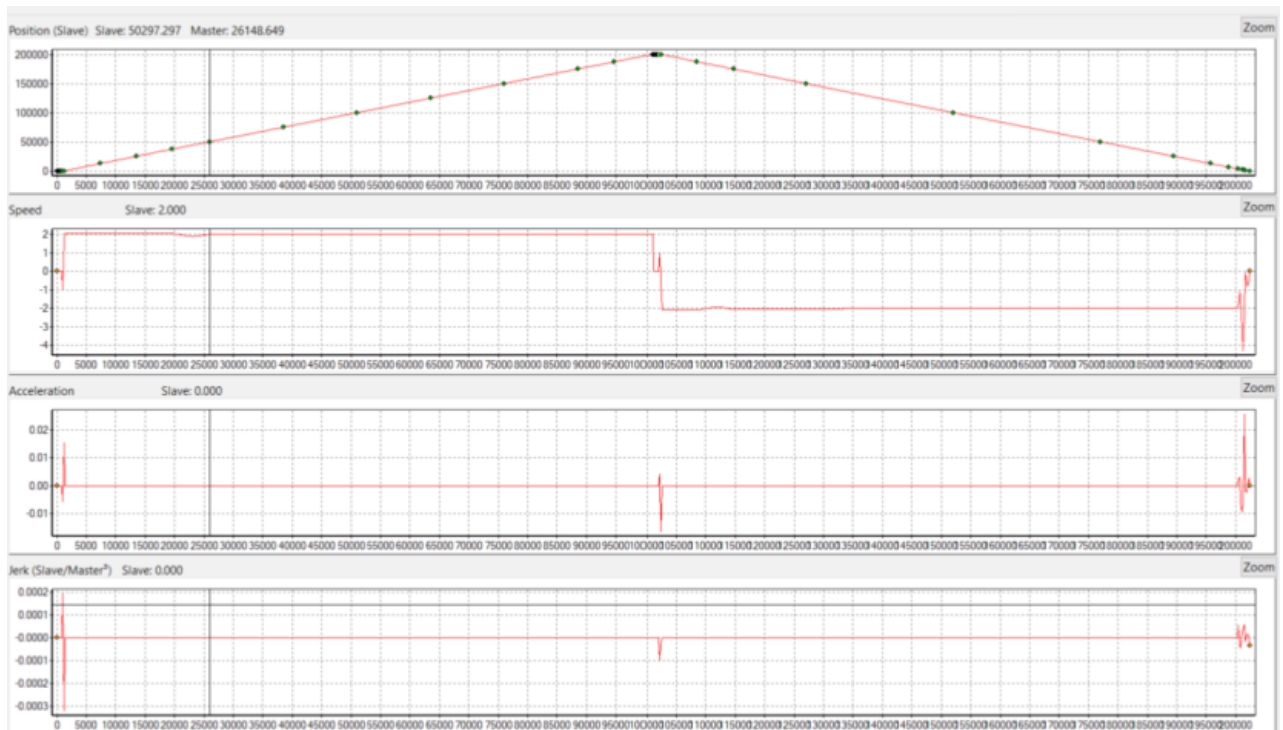
Figur 27. Inmatning av kontrollpunkter

5 Resultat

Resultatet av det här examensarbetet blev en färdig automationslösning för en winder-applikation med spridare och upprullare. En prototyp har byggts för att demonstrera lösningen i verkligheten. Specificering av hur produkten ska rullas upp kan med enkelhet tillgodoses genom att ändra den elektroniska kammern för att passa det enskilda ändamålet. Detta gör lösningen modulär. Lösningen är även skalbar i den mening att den kan appliceras på winder-applikationer av alla storlekar. Stegmotorer och mekanik kan bytas ut och ersättas av kraftigare enheter utan att programmet behöver göras om. Stegmotorerna kan även ersättas av servomotorer för att få en snabbare upprullning.

5.1 Kammen

Kamkurvan är resultatet av en interpolering av ett femtegradspolynom. Interpoleringen gjordes med avseende på de kontrollpunkter som matats in. CAM-Designer placerar sedan ut ett bestämt antal mellanliggande punkter för att göra kurvan jämnare. Antalet mellanliggande punkter har i det här fallet valts till 1000. Kurvorna i figur 28 visar även hur hastighets-, accelerations- och ryck-profilen för kammen ser ut.

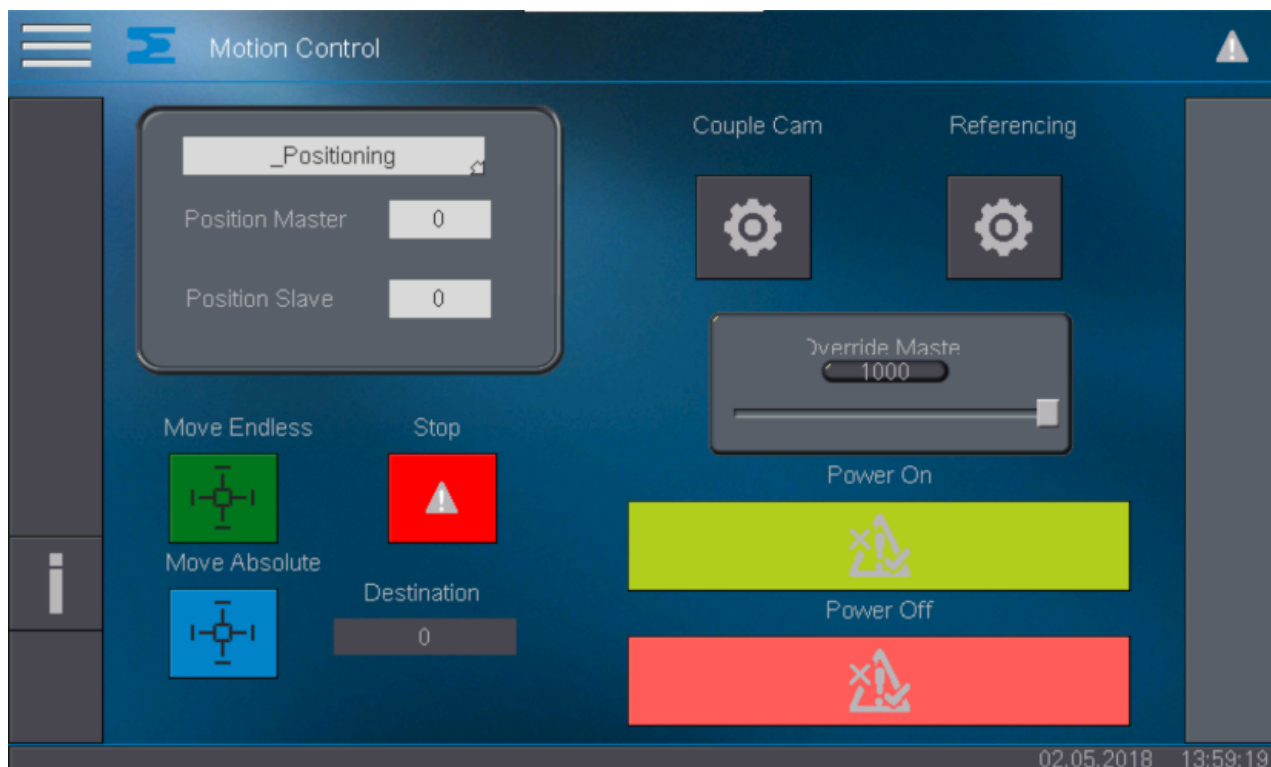


Figur 28. Den färdiga kamkurvan

5.2 Visualiseringen

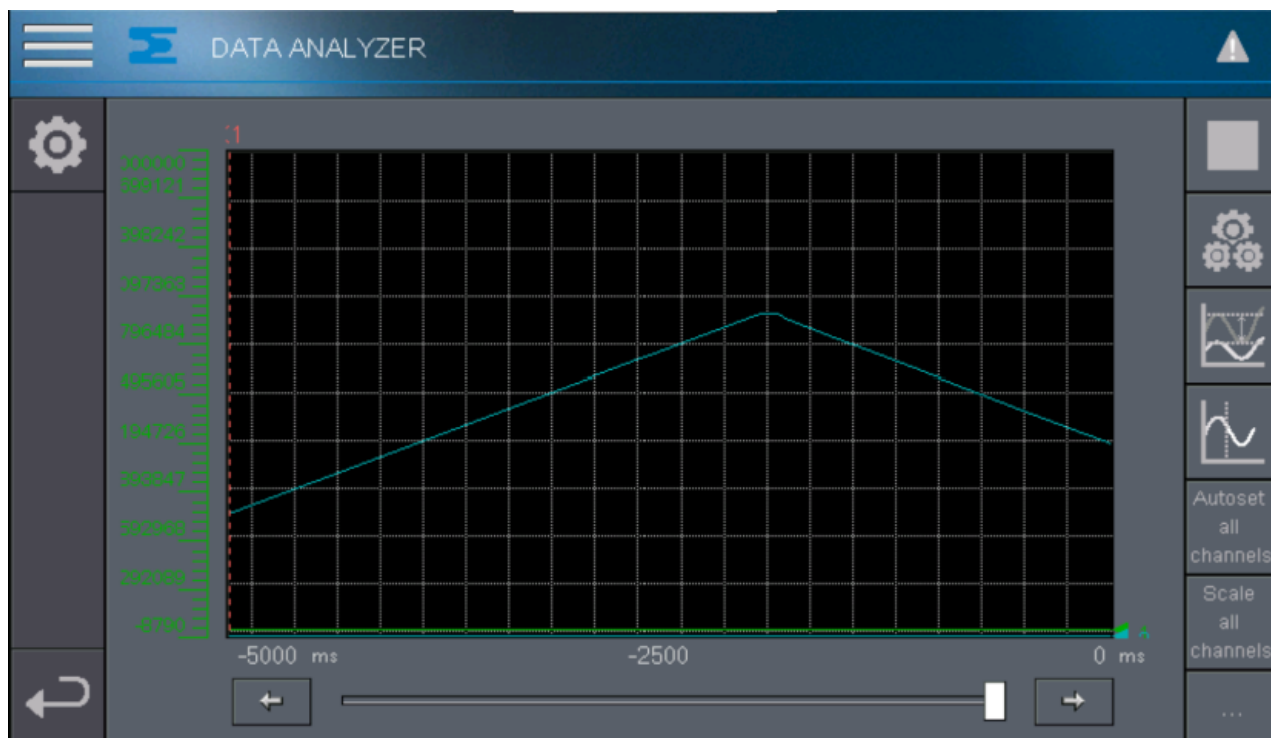
Det skapades även en interaktiv visualisering för start och stopp av applikationen via ett HMI. Visualiseringen låter även operatören göra en

referenskörning av spridaren för att den ska nå sin startposition. Återkoppling av applikationens status och axlarnas positioner finns även integrerat. Operatören kan även göra en *override* av huvudaxelns hastighet med reglaget.



Figur 29. Visualiseringen för start och stopp av applikationen.

I visualiseringen finns även en graf-ritare integrerad som visar axlarnas position i realtid.



Figur 30. Graf-ritare i visualiseringen som visar axlarnas position

6 Slutsats

En elektronisk kam implementerades som kunde styra spridarens rörelse med avseende på upprullarens position. Kammen kunde modifieras för att möjliggöra olika typer av spridning, men enbart en typ av spridning realiserades. Fördelen med den enskilda driften av spridaren är godtycklig eftersom den beror på vad målet med upprullningen är. Fördelen med lösningen för att realisera en typ drift där emot är att den är modulär och skalbar. Visualiseringen som skapades för att starta och stoppa applikationen kan återanvändas eller vidareutvecklas för att utöka funktionalitet.

6.1 Vidareutveckling

Det finns ett API från Sigmatek som tillåter generering av den elektroniska kammen under *runtime*. Implementering av detta API skulle möjliggöra för operatören att ändra formen på upprullningen utan att kompilera om programmet. Operatören hade kunnat bestämma spridarens rörelse genom att själv mata in värden i positionstabellen som används via visualiseringen.

I nuläget kan operatören indirekt bestämma hur mycket material som ska rullas upp genom att bestämma hur långt upprullaren ska rotera. Det kan bli en besvärlig huvudräkning eftersom hänsyn måste tas till att diametern på rullen växer allt eftersom fler och fler varv av materialet rullas upp. För att förenkla för operatören skulle en metod kunna implementeras för att beräkna hur mycket material som ska rullas upp. Detta genom att operatören skickar med den önskade mängden material, exempelvis längden på sytråden samt materialets tjocklek som inparametrar.

7 Terminologi

PLC Programmable Logic Controller

HMI Human Machine Interface

VARAN Versatile Automation Random Access Network

I/O Input/Output

CPU Central Processing Unit

RAM Random Access Memory

ROM Read Only Memory

8 Källförteckning

- [1] M. Nuserm [Photograph]. Tillgänglig:
https://www.123rf.com/photo_66561796_round-enamelled-wire-being-wound-to-form-the-transformer-coil-on-the-winding-machine.html
[5 maj 2018]
- [2] Architecture Technology Corpor ”Motion Control Report”, 1th edition, Elsevier, 1990, [E-bok] Tillgänglig: Science Direct
- [3] Y. Zhang, S. Finger, S. Behrens, Introduction to Mechanisms, Carnegie-Mellon University, 2003, [ONLINE] Tillgänglig:
<http://www.cs.cmu.edu/~rapidproto/mechanisms/chpt6.html> [15 April, 2018]
- [4] A. Hughes, B. Drury, Electric Motors and Drives – Fundamentals, types and application, 4th edition, Elsevier, 2013, [E-bok] Tillgänglig: Science Direct
- [5] E.R. Aphonsus, M.O. Abdullah “A review on the applications of programmable logic controllers”, Elsevier, 2016, Tillgänglig: Science Direct
- [6] W.Bolton, Programmable Logic Controllers, 6th edition, Elsevier, 2015, [E-bok] Tillgänglig: Science Direct
- [7] VARAN BUS USER ORGANIZATION “Technology” [Online] Tillgänglig: <http://www.varan-bus.net/en/home-en/> [24 April, 2018]
- [8] Sigmatek [Online] Tillgänglig:
http://www.sigmatek-automation.com/fileadmin/user_upload/downloads/real-time-ethernet-varan-e.pdf [27 April, 2018]
- [9] SigmaControl [Online] Tillgänglig:
<http://www.sigmacontrol.eu/images/stories/SIGMATEK/SKVARAN.pdf> [27 April, 2018]
- [10] Sigmatek [Online] Tillgänglig:
http://www.sigmatek-automation.com/fileadmin/user_upload/downloads/CP-102-eng.pdf [28 April, 2018]
- [11] SIGBI [Online] Tillgänglig:
http://www.sigbi.se/files/manual/Ezi-STEP_ST_OperatingManual.pdf [22 April, 2018]

[12] M.Kukla, P Tarkowski, I Malujda, K Talaska, J Górecki, “Determination of the torque characteristics of a stepper motor”, Elsevier, 2015, Tillgänglig: Science Direct

[13] SIGBI [Online] Tillgänglig:
http://www.sigbi.se/files/manual/Ezi-STEP_BT_OperatingManual.pdf [23 April, 2018]

[14] Sigmatek [Online] Tillgänglig:
http://www.sigmatek-automation.com/fileadmin/user_upload/downloads/ST-011-eng.pdf [29 April, 2018]

[15] Sigmatek [Online] Tillgänglig:
http://www.sigmatek-automation.com/fileadmin/user_upload/downloads/ETT-732-eng.pdf [1 Maj, 2018]