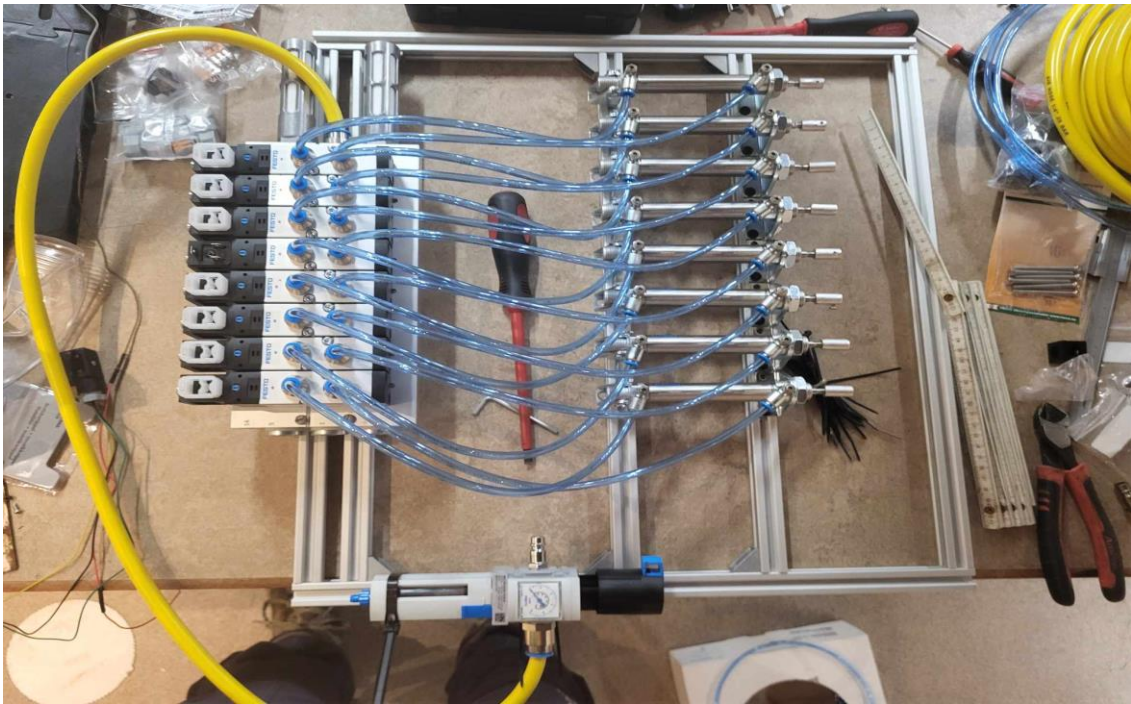




CHALMERS



# Digitalisering av hålkort för en vävmaskin

Examensarbete inom högskoleingenjörsprogrammet Mekanik

LANA BERG  
ERIK LINDGREN

INSTITUTIONEN FÖR RYMD-, GEO- OCH MILJÖVETENSKAP

---

CHALMERS TEKNISKA HÖGSKOLA  
Göteborg, Sverige 2024  
[www.chalmers.se](http://www.chalmers.se)



EXAMENSARBETE 2024

## **Digitalisering av hålkort för en vävmaskin**

LANA BERG  
ERIK LINDGREN



**CHALMERS**

Institutionen för rymd-, geo- och miljövetenskap  
CHALMERS TEKNISKA HÖGSKOLA  
Göteborg, Sverige 2024

Digitalisering av hålkort för en vävmaskin  
Lana Berg, Erik Lindgren

© LANA BERG & ERIK LINDGREN, 2024.

Handledare: Hans Bulthuis, Mallas Stenstugu AB  
Examinator: Arto Heikkilä, Institutionen för rymd-, geo- och miljövetenskap

Examensarbete 2024  
Institutionen för rymd-, geo- och miljövetenskap  
Chalmers tekniska högskola  
SE-412 96 Göteborg  
Telefon +46 31 772 1000

Omslagsbild: Fullständigt system för ersättning av hålkort med tryckluft.  
Författad i Microsoft Word  
Göteborg, Sverige 2024

Erkännande, dedikation och liknande personliga uttalanden återspeglar författarnas egna åsikter.

Digitalisering av hålkort för en vävmaskin  
Digitalisation of Punch-Cards for a Weaving Machine  
LANA BERG, ERIK LINDGREN  
Department of Space, Earth and Environment  
Chalmers University of Technology

## **Abstract**

This report describes one way to modernise and automate an older weaving machine. Mallas Stenstugu AB is a company located in Visby, Gotland which amongst other businesses runs a factory that produces yarn made from wool.

The following report contains the documentation of the process from the early stages where various ideas and proposed solutions are presented and listed, to the final stages when the entire system with its components has been added to the machine. The punch-cards and with their related components are the parts of the machine that have been replaced.

With the project being completed, the employees can now control the thread selectors by themselves. This is done through a computer, an Ethernet cable and a simple code in the programming language Python, which replaces the older parts of the machine that served the same purpose. The result which is a working system.

This report is written in Swedish.

Keywords: Weaving machine, PLC, Punch-Cards, Pneumatics, Wool, Textile industry

## **Sammanfattning**

I denna rapport beskrivs ett tillvägagångssätt för att modernisera och automatisera en äldre vävmaskin. Företaget Mallas Stenstugu AB beläget i Visby, Gotland driver bland andra verksamheter en fabrik där garn gjort på ull tillverkas och hanteras.

Denna rapport består av en dokumentation av arbetets gång från det inledande skedet då olika idéer och förslag till lösningar tas upp och listas till den avslutande delen då hela systemet med tillhörande komponenter monterats på maskinen och funktionen testats. De delar på maskinen som har ersatts är hålkorten och dem tillhörande komponenter.

Efter det avslutade arbetet kan de anställda nu själva styra trådväljarna. Detta genom en dator, en Ethernetkabel och en simpel kod i programspråket Python som ersätter de äldre delarna av maskinen som haft samma syfte. Resultatet är ett fungerande system.

Nyckelord: Vävmaskin, PLC, Hålkort, Pneumatik, Ull, Textilindustrin

## **Förord**

Detta examensarbete på 15 hp för programmet Mekatronik högskoleingenjör 180 hp på Chalmers tekniska högskola är utfört under en period av fem månader i 50 % studietakt. Rapporten är en del av examensarbetet och dess dokumentation.

Vi vill tacka de som gjort det möjligt att utföra arbetet och denna rapport. Däribland Arto Heikkilä som stöttat oss under arbetets gång och som tagit på sig rollen som både examinator och akademisk handledare.

Ett stort tack till Hans Bulthuis, Jenny Andersson och August Gydemo från Mallas Stenstugu AB som låtit oss få ta del i deras verksamhet och gett oss en inblick i branschen och vad som krävs för att lyckas få igenom ett projekt som detta.

Tack till ABB och Anna Ridell Björk som hjälpt till med information och gjort det möjligt att få in de komponenter som projektet skulle kräva inom dess tidsram.

Vi vill även tacka Hydsupply och Jonas Paulsson som varit väldigt tillmötesgående och underlättat när de sista delarna till projektet saknades.

Lana Berg & Erik Lindgren, Göteborg, Juni 2024





# Innehåll

## Innehållsförteckning

<b>Figurer</b> .....	<b>xi</b>
<b>Tabeller</b> .....	<b>xii</b>
<b>1. Introduktion</b> .....	<b>1</b>
1.1 Bakgrund .....	1
1.2 Syfte .....	1
1.3 Mål.....	2
1.4 Avgränsningar .....	2
1.5 Precisering av frågeställning.....	2
<b>2. Bakomliggande tekniker</b> .....	<b>3</b>
2.1 Komponenter .....	3
2.1.1 Luftfilterregulator .....	3
2.1.2 Pneumatiskt 5/2 ventil.....	3
2.1.3 Output Manifold.....	3
2.1.4 Dubbelverkande pneumatisk kolv .....	3
2.1.5 Aluminiumfästen.....	5
2.1.6 Optiska ögat .....	5
2.1.7 Roterande pulsgivare (Rotationsgivaren).....	6
2.1.8 Programmable logic controller .....	6
2.1.9 Mikrokontroller.....	6
2.1.10 Singleboard computer .....	7
2.1.11 Stegmotor.....	7
2.1.12 Solenoid-don.....	7
2.2 Programvaror.....	7
2.2.1 Autodesk Fusion 360 .....	7
2.2.2 Automation Builder 2.7.....	7
2.3 Intervju med Borås textilhögskola .....	8
2.4 Livslängdsanalys för cylindrarna .....	8
<b>3. Metod</b> .....	<b>9</b>
3.1 Vilka möjliga komponenter.....	9
3.2 Krav för PLC .....	16
3.3 Proof Of Concept.....	18
<b>4. Systembeskrivning, hårdvara</b> .....	<b>19</b>
4.1 Test av PLC-lampa .....	19
4.2 POC .....	19
4.3 Pneumatik .....	24
4.3.1 Cylinder .....	24
4.3.2 Strybackventil .....	24

4.3.3	5/2 Ventil.....	24
4.3.4	Output Manifold.....	24
4.3.5	Ljuddämpare .....	24
4.3.6	Luftfilterregulator .....	25
4.4	Elkopplingarna .....	26
<b>5.</b>	<b>Programlogik.....</b>	<b>27</b>
5.1	PLC programmering.....	27
5.1.1	PLC proof of concept.....	27
5.1.2	PLC Vävmaskinen .....	28
5.2	Programmering i Python .....	33
<b>6.</b>	<b>Resultat .....</b>	<b>39</b>
6.1	Proof of Concept.....	39
6.2	Fullskalig montering på maskinen.....	40
<b>7.</b>	<b>Diskussion.....</b>	<b>43</b>
<b>8.</b>	<b>Slutsats.....</b>	<b>45</b>
	<b>Litteraturförteckning.....</b>	<b>46</b>
	<b>Bilagor .....</b>	<b>I</b>
	Bilaga A PLC flöde.....	II
	Bilaga B PLC kod.....	III
	Bilaga C Python flöde.....	V
	Bilaga D python kod.....	VII

## Figurer

Figur 1 Illustration av kolven som gjort i programmet pages. ....	4
Figur 2 Ritning för fästen mellan maskin och kolv. ....	5
Figur 3 Resultat på simulering av pneumatisk kolv med SMC simulerings program....	15
Figur 4 Mätning av tid för kolvens utslag med hjälp av en microswitch .....	20
Figur 5 Testtrigg med delarna för poc:en .....	21
Figur 6 3D- modellen för testtriggen där ett perfekt avstånd på microswitchen kan ställas in med hjälp av en skena.....	22
Figur 7 3D-modell, påbyggnadsdel till testtriggen för att hålla uppe slangen till inslaget. ....	22
Figur 8 Färdig testtrigg kopplad med övriga komponenter och maskinen för att garantera att maskinen går att styra som planerat. ....	23
Figur 9 Schema över all pneumatik som använts i examensarbetet .....	25
Figur 10 Elschemat för systemet .....	26
Figur 11 SFC program för tidtagning .....	28
Figur 12 Flödet av PLC koden .....	28
Figur 13 Rotationsgivarens program .....	29
Figur 14 Signal formerna från rotationsgivaren .....	30
Figur 15 Illustration av hur Excell arket ser ut i arrayformat .....	32
Figur 16 Flöde för hela koden som beskriver alla loopar och villkor. Koden skapar det mönster som ska klistras in i PLC koden. Se bilaga C för större bild .....	34
Figur 17 Mönstret i Excel.....	35
Figur 18 Dikten i Python .....	35
Figur 19 Del av figur Figur 16.....	36
Figur 20 Pythonkod för Figur 19.....	36
Figur 21 Del av Figur 16 .....	36
Figur 22 Pythonkod för Figur 21 .....	36
Figur 23 Del av Figur 16 .....	37
Figur 24 Pythonkod för Figur 23.....	37
Figur 25 Pythonkod av Figur 26.....	38
Figur 26 Del av Figur 16 .....	38
Figur 27 Optiskt öga, rotationsgivare och 3D-printat fäste. ....	40
Figur 28 Ramen med pneumatiken monterad .....	40
Figur 29 Ramen med pneumatiken monterad på maskinen bakifrån .....	41
Figur 30 Ramen med pneumatiken monterad på maskinen framifrån .....	41

## **Tabeller**

Tabell 1 Jämförelse av olika komponenter inför urval.....	9
Tabell 2 Uppmätta tider vid olika tryck och monterings positioner.....	39

## 1. Introduktion

### 1. Introduktion

#### 1.1 Bakgrund

Mallas Stenstugu AB är ett företag placerat på Gotland där ullproduktion och hantverk har en lång tradition. Deras verksamhet består av tvätteri, skrädderi och vävning. Väveriet utförs av stora, äldre vävmaskiner från 70-talet.

Den typ av vävmaskin som arbetas med i detta projekt är gjord 1974. Tillverkarna är tyska och går under namnet Dornier.

I en vävstol finns uppspända trådar, dessa kallas varp. Varpen separeras och en annan tråd som kallas inslag går igenom varpen. Inslaget är det som hanteras i detta projekt. Maskinen har åtta trådar som styr inslaget.

Till skillnad från en vanlig vävstol finns ingen skyttel som för tråden genom varpen utan detta sker med två grip-armar där den första griparmen tar tråden och för den till mitten av varpen och där tar den andra armen emot tråden och drar den igenom varpen, därefter skärs tråden av och då har ett inslag skett.

För att modernisera och anpassa dessa maskiner till dagens industristandarder har man valt att avlägsna de delar som är mer kostsamma och sämre när man ser på det ur ett miljöanpassat perspektiv. Bland dessa finns hålkorten, som används bland annat för att hantera val av tråd i vävprocessen. Dessa hålkort behöver bytas ut med jämna mellanrum på grund av skador och slitage. Hålkorten kräver även en hel del plats och förvaringsutrymme i förhållande till den lilla mängden data som de som mest kan lagra, dessutom krävs nya hålkort när företaget är i behov av en uppdatering eller förändring av mönster.

Ett hålkort gör ett och samma mönster och för att byta mönster måste ett nytt anpassat hålkort placeras i maskinen [1]. Detta är en process som kräver mycket tid och arbetskraft samt har begränsad flexibilitet.

För att arbeta mot en mer hållbar utveckling inom industrin, där material med kortare livscykler och komponenter av plast fasas ut har Mallas Stenstugu AB valt att endast ersätta de delar som eventuellt skulle kunna vara problematiska. Företaget eftersträvar att bevara så mycket som möjligt av de ursprungliga maskinernas konstruktion.

#### 1.2 Syfte

Syftet med detta arbete är att ersätta maskineriet som styr valet av tråd, i dagsläget så består det av en låda som kontrolleras med hålkort. Ersättningen ska digitalisera maskinen och med det kunna öka flexibiliteten hos maskinen då de fysiska hålkorten har begränsningar som tex att korten får slitageskador, det tar tid att skapa och byta korten samt att de är fysiskt begränsade i längd.

Mallas vill även rusta upp maskinen så att den mer liknar den digitala teknik som finns tillgänglig idag men hålla lösningen så enkel och okomplicerad som möjligt.

## 1. Introduktion

### 1.3 Mål

Målet med examensarbetet är att ta fram en teknisk lösning som ersätter hålkorten. Mer specifikt det hålkort som väljer vilken tråd som skall användas. Detta ska lösas på ett så effektivt och billigt sätt som möjligt.

### 1.4 Avgränsningar

För att definiera och avgränsa projektets storlek har följande punkter fastställts:

- Examensarbetets omfattning har begränsats med avseende på livslängdsanalyser av de utvalda komponenterna. En fullständig livslängdsanalys har uteslutits då detta skulle kräva mer tid än vad som fanns att tillgå. Det görs enbart en kortare analys på kolvarna då dessa är de mest relevanta delarna för examensarbetet.
- Examensarbetet hanterar enbart inslaget och inte varpen.
- Maskinen opererar med ett varvtal på ca 175 varv per minut (rpm). Delarna som ska monteras ihop med maskinen behöver kunna hantera detta.
- Avgränsning i användargränssnitt till koden i kontrollsystemet. Ingen skärm skapas eller kopplas in. Alla förändringar av instruktioner görs istället direkt i koden.
- I examensarbetet riktas uppmärksamheten mot vad vi behöver ta in från maskinen och vad vi ska ge ut. Enbart dessa input och output tas hand om och kollas. Därav begränsas övrig insikt i maskinens funktioner och maskinen betraktas som en svart låda.

### 1.5 Precisering av frågeställning

För att tydliggöra frågeställningen har följande frågor utformats:

- Kan hålkorten bytas ut till en mer teknisk lösning?
- Vilken teknisk lösning skulle vara mest användarvänlig? Mikrokontroller eller PLC, motor eller pneumatik?
- Vilken lösning skulle vara den bästa i det avseende på kraven som miljön i fabriken ställer? Mikrokontroller eller PLC, motor eller pneumatik?
- Hur kan användningen och kontrollen av vävmaskinen förbättras med en mer modern lösning jämfört med det traditionella hålkortssystemet?

## 2. Bakomliggande tekniker

### 2. Bakomliggande tekniker

Det här avsnittet beskriver den nödvändiga teori och de beräkningar som krävs för att färdigställa examensarbetet. Här finns även en mer detaljerad beskrivning på de olika delarna som använts.

#### 2.1 Komponenter

Nedan görs en kortare beskrivning av de delar som krävts för att konstruera systemet.

##### 2.1.1 Luftfilterregulator

Luftfilterregulator är en kombination av ett luftfilter och en tryckluftsregulator. Luftfilter filtrerar luften till en viss partikelstorlek så att de delarna efter filtret i det pneumatiska systemet får luft av en viss kvalitet och håller därför längre. Tryckluftsregulatorn reglerar lufttrycket så att trycket längre ner i systemkedjan uppnår ett stabilt värde som inte varierar allt för mycket [2, 3].

##### 2.1.2 Pneumatiskt 5/2 ventil

Ett don med fem inkopplingar och två lägen som flyttar luften till och från. De styrs med pneumatik antingen för att flytta mellan de två lägena eller med en fjäder för att ha ett "normal läge" [4].

##### 2.1.3 Output Manifold

Komponenten som kopplar samman flera ventiler för att kunna kontrollera dem tillsammans [5].

##### 2.1.4 Dubbelverkande pneumatisk kolv

I detta arbete används en dubbelverkande kolv med en slaglängd (hur långt kolvens stång rör på sig) på 60 mm och en innerdiameter på 16 mm. Dubbelverkande betyder att kolven styrs enbart med tryckluft från båda ändarna [6]. Kolven består av en cylinder och en skiva med en stång i mitten som pressas ut. Detta illustreras i Figur 1. När trycket är på sidan med endast skiva så åker kolven ut och påverkas av en kraft [7]:

$$F = P \cdot A .$$

## 2. Bakomliggande tekniker

Sidan där stången är fäst inne i cylindern blir långsammare då arean där är mindre. De olika storheterna är:

**Kraft (F),**

**Kolvarea (A),**

**Tryck (P).**

I fabriken ges trycket i enheten bar, omvandlingen till SI-enhet pascal ska ge  
 $1 \text{ bar} = 10^5 \text{ Pa}$

Typiskt värde på använt tryck är 6 bar. Kolvens innerdiameter är 16 mm vilket ger en tryckkraft på ca 121 N.



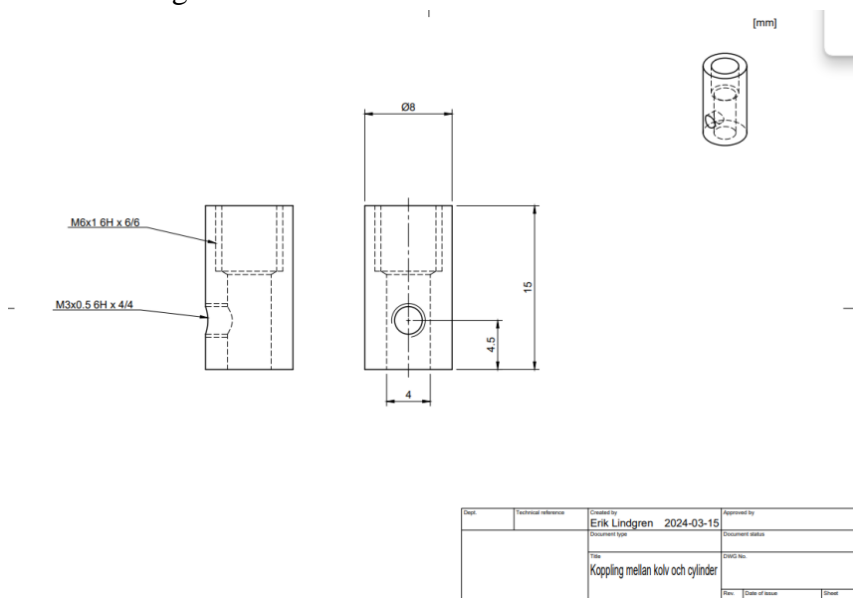
*Figur 1 Illustration av kolven som gjort i programmet pages.*



## 2. Bakomliggande tekniker

### 2.1.5 Aluminiumfästen

En koppling i 6032 aluminium 10 mm rundstång som producerades med pelarborr enligt ritningen i Figur 2. Anledningen till att en egen koppling skapades var att inga standardprodukter hittades och priset för de kopplingar som skulle kunna fungerat var för stora för budgeten.



Figur 2 Ritning för fästen mellan maskin och kolv.

### 2.1.6 Optiska ögat

Består av en ljuskälla, oftast i form av en lysdiod och en ljussensor som ger en signal när ljuskällan inte “syns” för sensorn och kan därmed avgöra om något hamnat emellan ljuskällan och sensorn. Denna elektroniska del märker skillnader i ljusnivåer. Den tar därefter informationen den fått i form av ljusstyrka och omvandlar detta till elektriska signaler. Detta för att kunna kommunicera med andra delar i ett elektriskt system. Ofta används denna i till exempel automatiska dörröppnare, förpackningsmaskiner, inbrottsystem för att detektera rörelser och skuggor [8]. Det är en väldigt träffsäker komponent med hög noggrannhet och kan upptäcka väldigt små förändringar. Den har även en förmåga att ge snabb feedback. Däremot är den negativa sidan med denna sensor att ljus från miljön runtomkring kan påverka och störa signalerna [8].

## 2. Bakomliggande tekniker

### 2.1.7 Roterande pulsgivare (Rotationsgivaren)

Roterande pulsgivare (som hädanefter kommer refereras som “rotationsgivaren”) består av två delar: en mekanisk del och en elektrisk del och utgörs vanligen av en magnetsensor eller en fotosensor som läser av “streck” på en mekanisk skiva som sitter på en roterande axel. På detta vis kan man med hög noggrannhet få rotationsriktning, rotationshastighet och antal varv som axeln har roterat [9, 10].

### 2.1.8 Programmable logic controller

Programmerbart logiskt styrsystem (PLC) används i industrin idag för att kunna styra maskiner och olika processer i olika miljöer. Det finns många olika tillverkare och sorter. De kan användas för att styra olika maskiner och motorer. Några användningsområden är rulltrappor, trafikljus, förpackningsmaskiner eller vindkraftverk [11].

Ett PLC gör det möjligt att kontrollera och programmera dessa precis som man önskar med hjälp av bland annat digitala I/O portar. De digitala in-portarna läser signaler från bland annat sensorer och brytare medan de digitala ut-portarna styr signaler till bland annat reläer och lysdioder [11].

PLC använder enligt IEC-61131-3 standarden fem programmeringsspråk men kan beroende på märke och programvara kodas med fler språk. De fem olika språken som IEC-61131-3 använder är Sequencial Function Charts (SFC), Ladder Diagrams (LD), Instruktion Lists (IL), Function Block Diagram (FBD) och Structured Text (ST). SFC, LD och FBD är grafiska kodspråk som är skapta för att efterlikna de gamla krets-och flödesscheman som användes innan PLC programmering fanns. IL och ST liknar assembler och vanliga högnivåspråk såsom C/C++ eller Java.

Dessa språk kan kombineras för att skapa ett program som utnyttjar alla olika språks styrkor och gör koden lättläst. [12, 11]

### 2.1.9 Mikrokontroller

En mikrokontroller är en integrerad krets som består av bland annat en Central Processing Unit (CPU), ett minne, inputs och outputs. CPU:erna är vanligtvis 32 bitars CPU:er eller mindre. Mellan CPU:n och input och Output brukar det finnas kretsar såsom ADC, DAC, PWM och kommunikationsmedium såsom I2C, SPI och UART. Mikrokontroller används idag som en mindre kraftfull dator och finns i allt från routrar till diskmaskiner och bilar [13].

## 2. Bakomliggande tekniker

### 2.1.10 Singleboard computer

En singleboard computer är en mindre dator såsom en laptop eller pc med tillgång till ett operativsystem och skärmslutningar men på ett enda kretskort. Likt mikrokontrollerna kan de ha I/O utgångar som är helt tillgängliga för användaren. Dessa klassas som ett mellanting mellan en PC och Mikrokontroller [13].

### 2.1.11 Stegmotor

Stegmotorn är en elektrisk motor som använder en tandformad disk som är kopplad till axeln. När magneterna kring disken aktiveras, stegar disken sig framåt.

Upplösningen på stegen kan öka genom att ta halvsteg, vilket uppnås genom att aktivera två magneter samtidigt. Detta gör att tänderna stannar mellan de magneterna som är aktiverade [14].

### 2.1.12 Solenoid-don

Ett solenoid-don består av fyra delar: en kopparspole, en ferromagnetisk metallstav och en fjäder. Kopparspolen skapar ett magnetfält som attraherar staven och drar den till sin mitt när det går ström genom spolen. När strömmen är av så trycker fjädern staven tillbaka till sitt ursprungliga läge. De är ett don som går att styras elektriskt mycket snabbt mellan två lägen med få rörliga delar [15].

## 2.2 Programvaror

Avsnitten nedan beskriver de programvaror som använts i projektet för att kunna programmera och rita upp systemet.

### 2.2.1 Autodesk Fusion 360

Autodesk Fusion 360 är ett molnbaserat CAD-program som inriktar sig mot industri samt skolarbete. Molnfunktionen gör att det är lämpligt för ett projekt med flera personer då många kan jobba tillsammans för att skapa en modell. I detta projekt har det använts för att skapa delar och ritningar till examensarbetet bland annat kopplingen mellan kolv och stång, samt de 3D-printade delarna för testriggen av Proof of Concept:en (POC:en).

### 2.2.2 Automation Builder 2.7

Automation Builder 2.7 är en programmeringsmiljö med inriktning på företaget ABB:s egna produkter. Miljön är baserad på CODESYS som är ett open-sourceprogram för PLC-programmering och används eftersom det är ABB:s produkter som används i detta projekt. Utöver det har Automation builder 2.7 också möjlighet till användarvänliga Human-Machine Interface (HMI) som förenklar ifall en instrumentpanel skulle krävas.

## 2. Bakomliggande tekniker

### 2.3 Intervju med Borås textilhögskola

Innan själva byggandet påbörjades och alternativen fortfarande vägdes mot varandra gjordes ett studiebesök på Borås textilhögskola för att få en bättre insikt i hur maskinen fungerade och om vår idé var en lämplig lösning. På högskolan finns en hel del liknande maskiner som den som används i examensarbetet och detta gav en chans för närmare analys.

På textilhögskolan gjordes en intervju med Fredrik Wennersten som är undervisningstekniker och hjälpte Mallas med att sätta ihop vävmaskinerna på Gotland. Detta gav en chans till att ställa relevanta frågor, såsom om det tänkta konceptet och initiala idéer är rimliga och genomförbara, om de tänkta donen kan användas i detta projekt, placering av sensorer och hur dessa skulle kunna utformas. Den viktigaste informationen som intervjun gav var angående öglorna som för ner tråden i färgväljaren och att dessa ska vara nere när huvudaxelns rotationsvinkel är mellan  $30^\circ$  och  $230^\circ$  [13].

### 2.4 Livslängdsanalys för cylindrarna

En kolv har enligt tillverkarens specifikationer en livscykel på 50 miljoner cykler [6]. För att beräkna kolvens uppskattade livslängd krävs en uppskattning av den dagliga drifttiden för vävmaskinen. En arbetsdag antas vara åtta timmar (480 minuter).

Antal varv per minut för maskinen är 175 rpm.

Cylindern gör en cykel per två varv.

Antal cykler per minut blir då  $\frac{175 \text{rpm}}{2} = 87,5$  cykler/minut.

Med dessa parametrar kan det dagliga antal cykler beräknas:

*Aktiveringar per dag*  $\Rightarrow 87,5 \text{rpm} \cdot 480 \text{min} = 42\,000$  cykler.

Därefter beräknas totalt antal dagar med livslängden på 50 miljoner cykler.

*Antal dagar*  $= \frac{\text{Totalt antal cykler}}{\text{Antal cykler per dag}} = \frac{50\,000\,000}{42\,000} \approx 1190,48$  dagar.

Vilket motsvarar 4,6 år (räknat på 260 arbetsdagar)

Detta motsvarar en livslängd på ca 4,6 år under förutsättningen att maskinen körs konstant under en åtta timmars arbetsdag, det vill säga 40 timmars arbetsvecka. Detta baseras dock på att maskinen opererar på maximalt varvtal under alla dagarna.

### 3. Metod

## 3. Metod

För att tydliggöra arbetet och lyckas uppnå de mål som satts upp i avsnitt 1.3 utfördes arbetet metodiskt uppdelat enligt följande: För att kunna fatta ett beslut om vad som skulle bli det mest effektiva sättet att bygga systemet på jämfördes olika förslag på komponenter. Därefter gjordes val av komponenter med hjälp av en tabell med dess fördelar och nackdelar. När denna tabell var färdigställd beslutades bland annat att arbetet skulle kräva en PLC och en fungerande PLC miljö. För att hitta det PLC:t som var mest lämpad för arbetet togs en kravlista fram.

I strävan att få en garanti på att byggandet skulle fungera som tänkt utfördes ett proof of concept på plats i fabriken där minimalt antal delar beställdes och testades.

### 3.1 Vilka möjliga komponenter

När ett urval för den mest lämpade utrustningen för examensarbetet skulle göras, beslutades det att ta fram en lista eller tabell. Denna togs fram för att brainstorma fördelar och nackdelar med varje komponent.

Genom att ha informationen framför sig var förhoppningen att snabbt kunna se och avgöra vilka komponenter som var optimalt anpassade för examensarbetets behov.

Tabell 1 Jämförelse av olika komponenter inför urval

<b>Fördelar:</b>	<b>Nackdelar:</b>
<p><b>PC</b></p> <ul style="list-style-type: none"><li>• Vanligt och lätt att få tag på.</li></ul>	<p><b>PC</b></p> <ul style="list-style-type: none"><li>• Komplex att programmera.</li><li>• Oklart om säkerheten för industrimiljö.</li><li>• Dyr i förhållande till budget.</li><li>• Känslig för damm och vibrationer.</li><li>• Lösningen blir en speciallösning med c-kod eller liknande som är svår att bygga vidare på eller förändra.</li><li>• Hög underhållstid om ett operativsystem (OS) såsom Windows eller Mac används.</li></ul>

### 3. Metod

<p><b>Mikrokontroller</b></p> <ul style="list-style-type: none"> <li>• Billigt i förhållande till budget.</li> <li>• Enkelt att köpa in.</li> </ul>	<p><b>Mikrokontroller</b></p> <ul style="list-style-type: none"> <li>• Kräver externa elkretsar för att kontrollera industridon.</li> <li>• Lösningen blir en speciallösning med c-kod eller liknande som är svår att bygga vidare på eller förändra.</li> <li>• Komplex att programmera.</li> </ul>
<p><b>Singelboard computer</b></p> <ul style="list-style-type: none"> <li>• Billigt i förhållande till budget.</li> <li>• Enkelt att köpa in.</li> </ul>	<p><b>Singelboard computer</b></p> <ul style="list-style-type: none"> <li>• Kräver externa elkretsar för att kontrollera industridon.</li> <li>• Komplex att programmera.</li> <li>• Lösningen blir en speciallösning med c-kod eller liknande som är svår att bygga vidare på eller förändra.</li> <li>• Svårt att felsöka hur programmet interagerar med OS.</li> </ul>
<p><b>PLC</b></p> <ul style="list-style-type: none"> <li>• Enkelt att programmera</li> <li>• Förutsägbart beteende.</li> <li>• Kräver ej externa kretsar för projektet.</li> <li>• Gjord för industrimiljö.</li> <li>• Möjlighet att expandera.</li> <li>• Standard för industrier.</li> </ul>	<p><b>PLC</b></p> <ul style="list-style-type: none"> <li>• Dyr i förhållande till budgeten.</li> </ul>
<p><b>Pneumatisk kolv</b></p> <ul style="list-style-type: none"> <li>• Finns tillgång till tryckluft i lokalen.</li> <li>• Slaglängd i varierande storlekar.</li> <li>• Enkel styrsignal.</li> <li>• Enkel Princip jämfört med andra alternativ.</li> <li>• Har använts i liknande projekt innan.</li> </ul>	<p><b>Pneumatisk kolv</b></p> <ul style="list-style-type: none"> <li>• Kolvstångens fram och tillbaka rörelse varierar i tid och är därmed inte fullt pålitlig.</li> <li>• Kräver ytterligare komponenter för att styra.</li> </ul>
<p><b>Solenoid-don</b></p> <ul style="list-style-type: none"> <li>• Snabb jämfört med de andra donen.</li> </ul>	<p><b>Solenoid-don</b></p> <ul style="list-style-type: none"> <li>• Kort slaglängd.</li> </ul>

### 3. Metod

<b>Stegmotor</b> <ul style="list-style-type: none"><li>Efter observationer från Borås textilhögskola så har detta tidigare använts på liknande maskiner.</li></ul>	<b>Stegmotor</b> <ul style="list-style-type: none"><li>Kräver mer ström ju högre motståndet är.</li><li>Kräver ytterligare komponenter för att styra.</li><li>Svår att programmera jämfört med de andra alternativen.</li></ul>
<b>Inkrementell Encoder</b> <ul style="list-style-type: none"><li>Billig jämfört med absolut encoder.</li><li>Enkla signaler.</li></ul>	<b>Inkrementell Encoder</b> <ul style="list-style-type: none"><li>Kräver mer elektronik för att veta sin position.</li><li>Tappar sin position om elen går.</li></ul>
<b>Absolut Encoder</b> <ul style="list-style-type: none"><li>Vet hela tiden sin position.</li><li>Har kvar sin position om elen går bort.</li></ul>	<b>Absolut Encoder</b> <ul style="list-style-type: none"><li>Dyr jämfört med inkrementell.</li><li>Kräver kommunikations moduler för att kommunicera med datorn.</li></ul>

### 3. Metod

#### **Styrenhet**

Baserat på Tabell 1 ovan gjordes bedömningen att en PC inte är anpassad till industrimiljö samt att utvecklingen av mjukvaran skulle bli för svår. Alternativet att skapa elektroniska kretsar till detta skulle ta för lång tid samt att lösningen skulle bli unik och kräva en person med liknande grad av utbildning för att fortsätta eller ändra på arbetet i framtiden och därmed uteslöts PC som alternativ.

På samma sätt är en singleboardcomputer utesluten då lösningen återigen hade blivit unik och svår att fortsätta utveckla. Ett av problemen med detta är att ett operativsystem kan ta resurser ifrån programmet som ska styra maskinen. Elkretsen till en singleboardcomputer är betydligt enklare att konstruera då t.ex. en Raspberry pi har general-purpose inputs/outputs i form av stift. En Raspberry pi behöver därmed bara en krets som styr en isolerad krets, med ena sidan kopplad till datorns stift och andra sidan till 24 V. Där kan 24 V sidan styra don. Detta val uteslöts återigen på grund av operativsystemet samt att om detta skulle genomföras med baremetal-programmering (Då programmeraren programmerar direkt på hårdvaran utan operativsystem) så skulle detta ta för mycket tid av examensarbetet.

Mikrokontrollern skulle kunnat vara ett bra alternativ. Den är billig, har ingen okänd kod utan bara det programmeraren gör, enkel att utveckla, enkel att köpa in. Dessutom skulle elkretsen för att kunna styra don likna den för singleboardcomputern, då även mikrokontrollern har GPIO-stift. Problemen är dock liknande ovanstående: svår för någon annan att ändra eller bygga på det som gjorts, begränsad möjlighet för att expandera i framtiden och att mikrokontrollers inte är robusta.

Sista alternativet är PLC som är en dator utvecklad för industrimiljö, kräver inga externa kretskort för att styra 24 V don, har expansionsmöjlighet, möjlighet att kommunicera i alla standardiserade kommunikationssätt såsom RS-232, RS-485, CAN-buss, TCP och Ethernet för att nämna några. Att välja en PLC möjliggör även implementering av Supervisory Control And Data Acquisition (SCADA, system för processövervakning) i framtiden. Med dessa faktorer i åtanke togs beslutet att välja PLC (programmerbar logikstyrenhet) framför en SCB (Single Board Computer) eller en mikrokontroller mest på grund av industristandarder och bortse från de högre kostnaderna och längre leveranstiderna som PLC:t skulle kunna ha.



### 3. Metod

#### **Don**

Solenoid kan uteslutas då slaglängden av det som hittades var för kort och kan därmed inte uppfylla ett av grundkraven för sin uppgift. Dessa undersöktes närmare då de är enkla att kontrollera och har en utmärkt responstid från på-signal till att de får fullt utslag (ändan går från 0 mm till 60 mm)

#### **Stegmotor**

Det som talar emot är att använda stegmotor från mjukvarusidan är att det kommer krävas flera outputs per motor, minst två outputs och två inputs per motor för att känna av ändlägena. Detta innebär att det skulle krävas 16 stycken inputs och outputs endast för donen, dessutom kan det tillkomma styrenheter för motorerna och ytterligare mekaniska delar för att omvandla rotationsrörelsen till en linjär rörelse. Det innebär att 3-4 problem kommer behöva lösas medan motorn bara löser ett problem och därmed så väljs det alternativet bort.

#### **Pneumatiska kolvar**

Det som talar emot att välja pneumatiska kolvar är att de inte är lika snabba som solenoider och att det finns en responstid mellan signal och start på rörelse då det tar tid för trycket att byggas upp, samt att tryckluft tenderar att läcka och ha små förseningar i hela systemkedjan. Då behövs det styrenheter för att kontrollera var luften ska komma. Till skillnad från de andra donen så finns det standardiserade lösningar för att kontrollera tryckluft och då krävs enbart att specificera styrenheterna istället för att konstruera en egen lösning. De pneumatiska kolvarna kommer då att behöva ventiler som styrs av solenoider. Enligt stycket ovan är solenoider snabba och enkla att styra med endast på- och-av signal för att kontrollera dem. Med tanke på att de är enkla att styra, det finns tryckluft i lokalen och att kolvarna ger en linjär rörelse utan extra komponenter görs en närmare undersökning för att se om de uppfyller krav för tid och kraft.

### 3. Metod

#### Uppmätning av kraftkrav

För att specificera kolven behöver en uppskattning av kraften göras då kolven får onödigt slitage om detta har ett för stort förhållande. Förhållandet som mäter detta blir:

$$\frac{\text{Max kolv kraft}}{\text{Belastning på kolven}}$$

För att uppskatta kraften som krävs gjordes grova mätningar med en egenkonstruerad newtonmeter då det inte fanns någon sådan i fabriken. Newtonmetern skapades genom att ta en fjäder och en känd vikt för att kunna tillämpa hooks lag [7]:

$$F = k \cdot \Delta l \text{ där } F = \text{känd vikt} \cdot g.$$

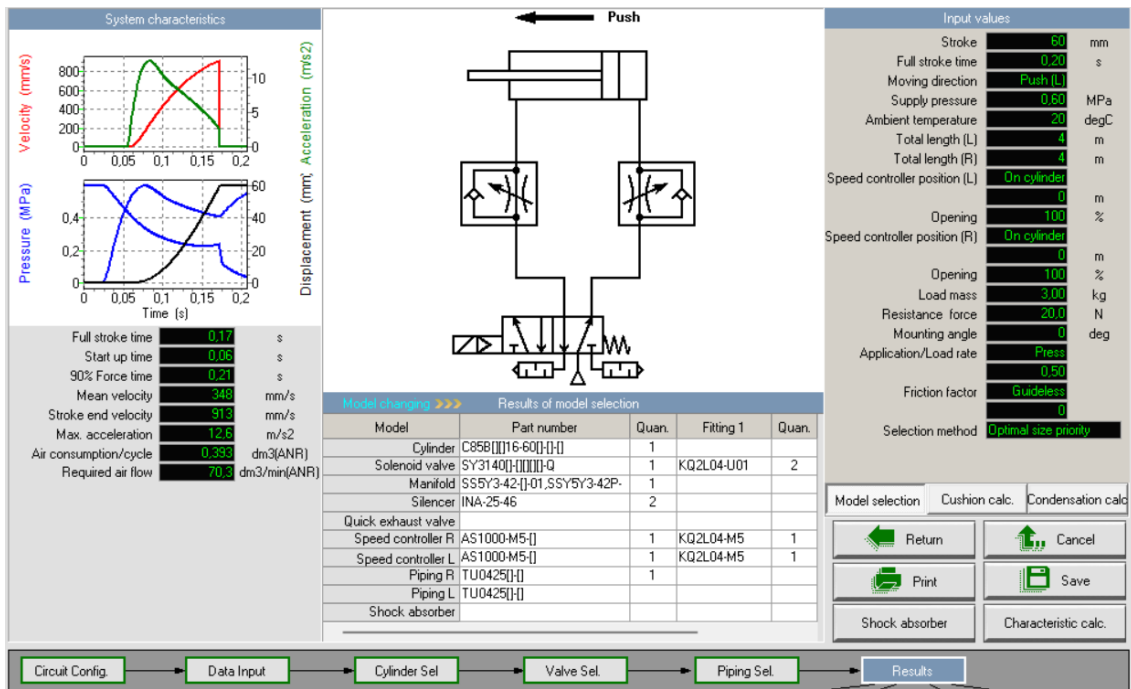
Detta för att få fram värdet för  $k$ . När detta var gjort så uppmättes  $\Delta l$  och en kraft på  $10,8 \text{ N}$ , vilket var den kraft som krävdes för att förflytta slangen och åstadkomma rörelsen. Med detta uppskattades onödigt slitage till en faktor 15 sådant att:

$$\text{max kolvkraft} = \text{Belastning på kolven} \cdot 15 = 10,8 \cdot 15 = 162 \text{ N},$$

Vad detta betyder är att inga kolvar som har en maxkraft som är högre än  $162 \text{ N}$  betraktades.

### 3. Metod

Med motståndet på 10,8 N och en uppskattning på 3 kg användes SMCs (grundat som Sintered Metal Corporation) Model selection för simulera en kolv med en hålstorlek på 16 mm och ett tryck på 6 bar. Enligt Figur 3 uppnåddes en tid på 0,17 s. Från detta drogs slutsatsen att kolven klarar kraven för tid och kraft med extra marginal på alla variabler.



Figur 3 Resultat på simulering av pneumatisk kolv med SMC simulering program

### Givare

För att få vinkeln från huvudaxeln behövs en sensor som kan registrera rotationsrörelser. Därför undersöktes encoders.

Encoders finns i två typer, inkrementell och absolut [16]. Inkrementell encoder ger bara pulser när en förändring i rotation sker och behöver ytterligare elektronik för att få sin hemposition. Absoluta encoders känner hela tiden sin position men skickar den informationen via ett protokoll. Detta skulle innebära ytterligare en enhet till PLC:t som kan tolka det protokollet. Därför gjordes istället valet att använda en inkrementell med en optisk sensor då dessa var ett billigare alternativ.

### 3. Metod

#### 3.2 Krav för PLC

Vid användning av PLC i en fabrik sådan som Mallas är det viktigt att ta hänsyn till miljön och användningsområdet för att arbetet ska gå så smidigt som möjligt men även leva upp till de förväntningar som fabriksägaren önskar. För att hitta det bäst lämpade PLC:t för detta projekt, både vad gäller hårdvara och mjukvara listades en del krav för att underlätta den processen och avgöra vilka egenskaper som värdesätts mest.

Nedan följer en lista över de mest värderade kraven som fastställts:

- IP-klass:  
Ip-klass talar om hur skyddad en elektrisk produkt är eller hur pass tätt höljet runt elektroniken är. Detta talar om hur tålig en produkt är när det kommer till både fasta och flytande föremål som potentiellt skulle kunna förstöra elektroniken i PLC:t. Denna klassificering består alltid av två siffror varav den första indikerar hur skyddat något är mot fasta föremål och damm och den andra siffran talar om hur resistent den är mot vatten. För att försäkra att PLC:t fortsätter hålla samma kvalitet är en Ip-klass på minst IP54/55 att föredra för att skydda mot damm, smuts och olja som kan finnas i en fabrik [17]. Följs detta kan livslängden på PLC:t öka och en hög prestanda hållas.

För att kunna få ner prisnivåerna och hålla budgeten för detta projekt valdes ett PLC från ABB i serien PM50XX med klass IP20. Tvåan i IP20 innebär att PLC:t har ett skydd mot fasta föremål som är större än 12 mm och nollan talar om att inget skydd mot väta finns [18]. Detta beslut fattades med hänsyn till att fabriken inte hanterar vatten i sina lokaler för produktionen. Det genomfördes en diskussion med Mallas och handledare för att hitta ett sätt som skulle kunna utöka skyddet. För att säkerställa både livslängd och säkerhet beslutades det att PLC:t förvaras i en plastlåda för el. Lådan kommer skydda enheten från en potentiell skada.

#### Temperaturtolerans:

PLC:t ska kunna operera utan problem under de temperaturer som kan uppstå i fabriken. En väldigt viktig punkt som PLC:t skulle leva upp till var en temperaturgräns på mellan +10°C till +30°C då detta är temperaturer som kan uppstå i fabriken.

PM5032 är en modell som tål ett temperaturintervall på 0°C till +60°C vid horisontell montering och 0°C till +40°C om den skulle monteras vertikalt.

- Resistans mot smuts och fett:  
Eftersom lokalen hanterar ull vilket i sig är väldigt rikt på fett [19] och att oljor och annat skadligt finns i lokalen var ett av kraven att PLC:t skulle klara detta. Detta förhindras genom att PLC:t står inne i plastlådan och inte exponeras för saker som kan ge oljeavtryck.

### 3. Metod

- I/O-Portar och möjlighet till utbyggnad:

För att kunna styra alla trådar på inslaget krävdes ett sätt att hantera alla åtta trådväljare på maskinen. Kravet på I/O portar blev då minst åtta stycken, en för varje trådväljare.

Efter att ha gått igenom flera datablad och kataloger på olika företag och olika modeller av PLC hittades ABB PM5032. Denna modell levde upp till detta krav. Modellen har 12 in-portar och åtta ut-portar och två som är programmerbara till in eller ut –port.

- Svenska alternativ:

För att undvika språkliga hinder ifall problem med PLC:t skulle uppstå var det ett krav att det skulle finnas svensk kundtjänst och att det skulle vara enkelt för fabriken att komma i kontakt med företaget. Något som var önskvärt var att stötta ett svenskt företag. Att beställa från ett EU-land innebär även att konfliktmetaller som kan finnas i datorer är reglerade enligt EU:s lagar och regler [20].

- Stand-alone:

Efter en diskussion med företagsledningen beslutades att ett så kallat Stand-alone PLC skulle vara ett krav. Detta då fabriken inte önskade någon vidare utbyggnad för tillfället utan hellre ett billigare alternativ än ett mer flexibelt.

- Användargränssnitt:

Ett av kraven var en simpel programmeringsmiljö, intuitivt och lättanvänt interface. Detta framför allt då ingen skärm implementeras och eftersom de ändringar som fabriken vill göra kommer att behöva ske direkt i programmet.

- Kablage:

För att underlätta arbetet och de tester som gjordes sattes kravet för en simpel sammankoppling. Det slutliga resultatet blev en Ethernet-kabel.

- Kostnad:

En balans mellan kostnad och funktioner var viktig i detta fall då företaget är ett start-up med en mindre budget. Examensarbetet krävde med andra ord en PLC som erbjuder de funktioner som var nödvändiga utan att överstiga ett för stort pris. Därav kravet att hitta ett så kostnadseffektivt alternativ som möjligt.

### 3. Metod

#### 3.3 Proof Of Concept

Proof Of Concept(POC) är en arbetsmetod där man gör det minsta möjliga för att bekräfta att ens koncept fungerar. För detta projekt har en del krav satts upp för att kunna bekräfta POCen.

Dessa krav är:

- Hela systemet ska gå från utsignal till att kolven är helt ute på 0,2 s. Mätningar på maskinens topphastighet gjordes. Enligt dessa mätningar kunde maskinen uppnå en topphastighet på 175 rpm. Detta ger en frekvens på  $\frac{1}{T} = \frac{1}{\frac{175}{60}} = 0,34$  s.
- Det här innebär att varje varv tar 0,34 s. När ett inslag sker ska kolven vara inne under tiden som axelns vinkel är mellan 30° och 230°. Överlämningen av tråden sker vid 180°, vilket innebär att  $\frac{230-30}{360} = 0,556$  det vill säga 55,6 % av tiden ska kolven vara ute. Det ger en tid på  $(1 - 0,556) \cdot 0,34 = 0,15$  s. Detta är den teoretiska tiden ner, men i praktiken ska tiden mellan kolv och griparm matchas så att programmet kan börja föra ut en kolv i förtid. Simulationer med mjukvaruprogrammet från företaget SMC för motsvarande delar gjordes. Enligt programmet kan det pneumatiska systemet få utslag på 0,17 s, där har kravet för konceptet satts till 0,2 s för att ge det mer marginal.
- Få kolven att reagera med sensorerna sådant att vi har ett START-läge när axeln är 0° med den optiska givaren samt att rotationsgivaren skall få rotationshastighet och visa gradantal från huvudaxeln.
- Bygga en testtrigg med alla sensorerna och en mikroswitch för att med PLC:t mäta tid från Start till Stopp. Tiden för tidsfunktionen i koden kommer att försummas då PLC:t är betydligt snabbare än pneumatiken och mätningen kommer göras i millisekunder och inte mikrosekunder.

## 4. Systembeskrivning, hårdvara

### 4. Systembeskrivning, hårdvara

Här beskrivs hela proceduren för konstruktionen och integrationen av systemet i en textilfabrik. Detta inkluderar de tester som gjordes, proof of concept samt den slutliga implementeringen av pneumatiken, elektroniken, och mekaniska komponenter. Nedan beskrivs både teoretiska överväganden och praktiska tillämpningar som krävdes för att till sist få ett system som fungerar. Till att börja med utfördes tester på PLC:t för att garantera dess funktionalitet och få en djupare förståelse för just den modellen. Detta gjordes också för att säkerställa att PLC:t inte skadats efter transport och att dess delar svarade enligt specifikationerna.

#### 4.1 Test av PLC-lampa

Då PLC:t hämtats ut och konfigurerats genomföres ett enkelt test. En normalt sluten knapp kopplades in i en krets tillsammans med PLC:t för att säkerställa att allt fungerade korrekt.

På PLC:t finns en programmerbar lampa. Med hjälp av knappen var målet att försöka få lampan att tändas då knappen aktiverades det vill säga att lampan skulle tändas då låg signal gavs.

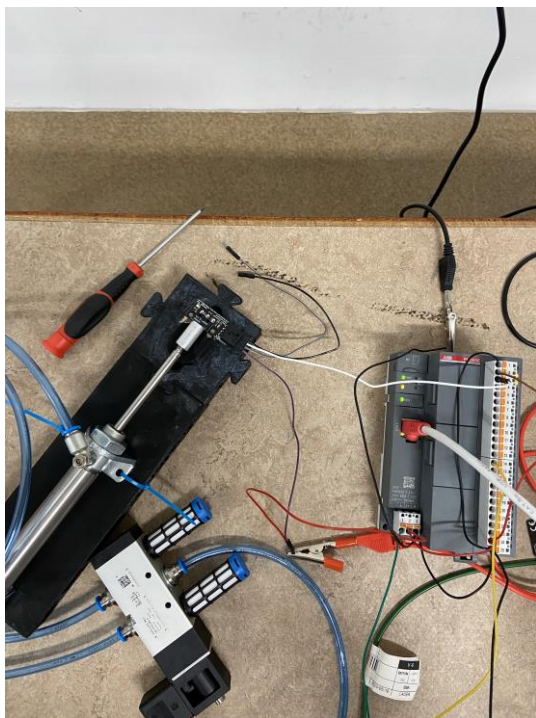
#### 4.2 POC

För att testa konceptet och de komponenter som slutligen användes skapades ett proof of concept. Detta genomfördes innan alla delar beställdes för att garantera att teorin fungerade som tänkt och för att mäta så att tiden som cylinderns cykel hade stämde överens med de beräkningar som tidigare gjorts.

För att allt skulle fungera som tänkt krävdes att cylindern skulle slå ut på 0,2 s.

För att avgöra om teorin skulle fungera som tänkt gjordes en tillfällig lösning där enbart en utav de åtta enheterna beställdes. Detta demonstreras i Figur 4 där ett testsystem en cylinder, ett 5/2 valv och två ljuddämpare satts samman. Till detta krävdes ett PLC system som kan programmeras och bestämma rytmen på när ventilen ska öppnas eller stängas, luftfilterregulatorn och luftslangar.

#### 4. Systembeskrivning, hårdvara



*Figur 4 Mätning av tid för kolvens utslag med hjälp av en microswitch*

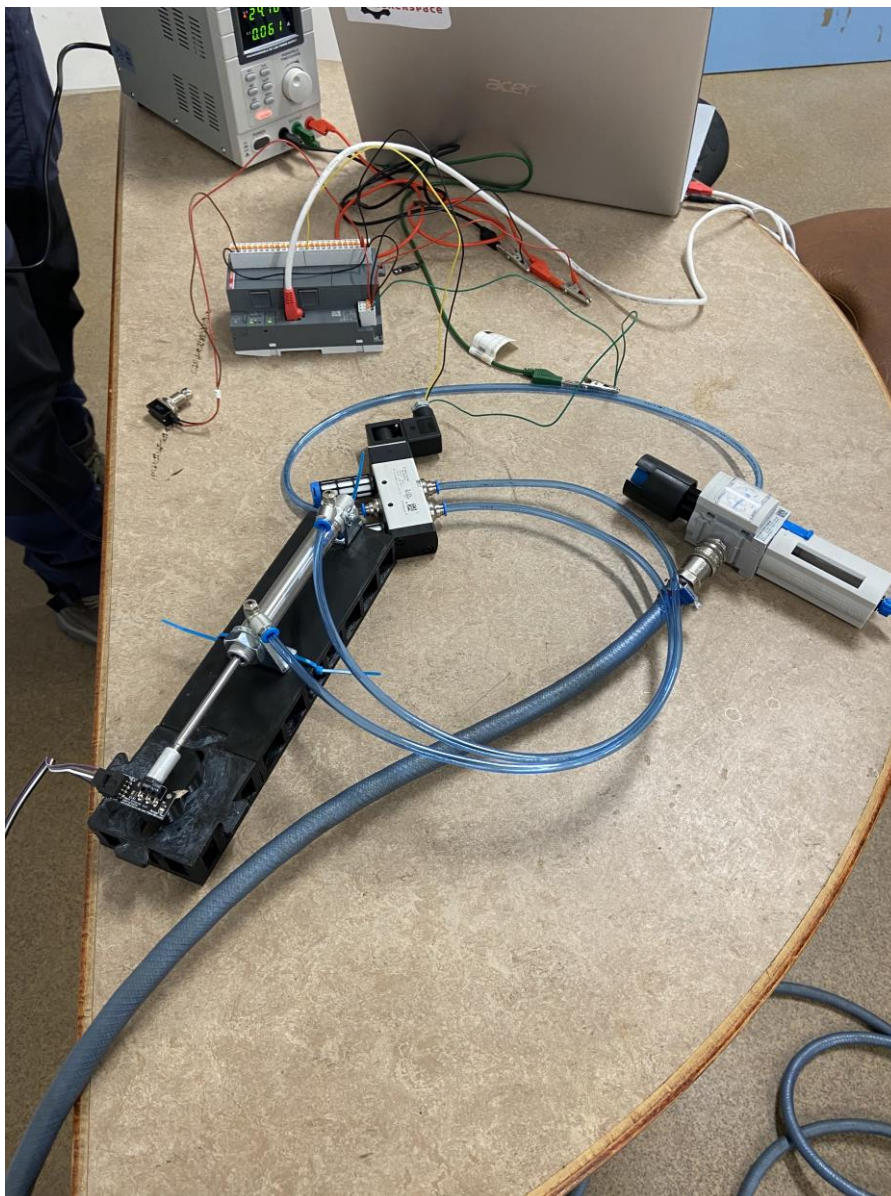
För att kunna mäta tiden från start tills att hela cylindern är ute behövdes något som registrerade detta, som exempelvis en knapp. Här är microswitchen en central del.

För att kunna mäta tiden skrevs ett program i ABB automation builder som registrerade och skrev ner tiden på varje utslag då cylindern sköts ut och aktiverade microswitchen. Konstruktionen för detta illustreras i Figur 4.

För att säkerställa att systemet levde upp till tidskravet som ställts gjordes försöket om 100 gånger med olika högt tryck utan att få ett enda resultat som dröjde längre än 0,2 s. Med detta resultat gick det att fastställa att cylindern och resten av testsystemet skulle klara av tidskravet.



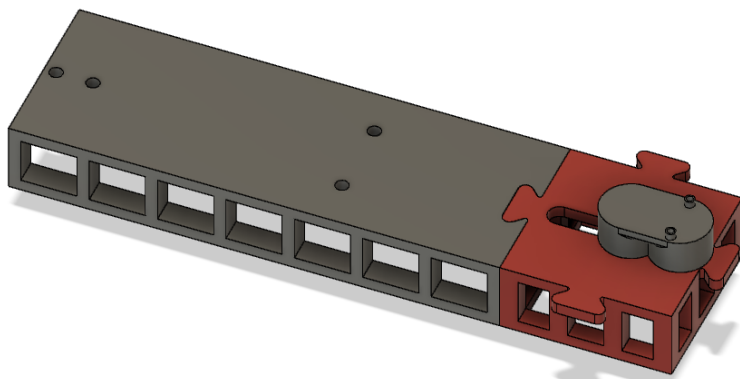
#### 4. Systembeskrivning, hårdvara



*Figur 5 Testrigg med delarna för poc:en*

För att stabilisera systemet byggdes en rigg med hjälp av en 3D-printad modell som skissats upp i fusion 360. I Figur 5 visas denna rigg ihop med andra pneumatiska delar.

#### 4. Systembeskrivning, hårdvara

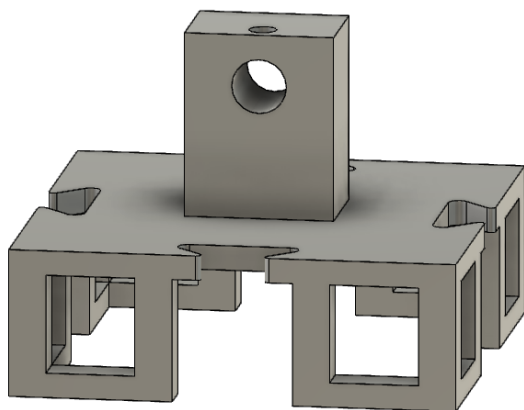


*Figur 6 3D- modellen för testriggen där ett perfekt avstånd på microswitchen kan ställas in med hjälp av en skena*

På riggen från Figur 6 placerades cylindern och microswitchen. De båda delarna fästes på riggen med hjälp av buntband. En skena och ett fäste konstruerades för att möjliggöra justering av avståndet mellan cylindern och microswitchen.

Efter att testerna genomförts utan motstånd påbörjades testerna med motstånd. Systemet fästes då på vävmaskinen.

Hela riggen, inklusive den sista delen som visas i Figur 7 placerades på maskinen enligt Figur 8 med hjälp av buntband. Då cylindern fortfarande behövde aktivera microswitchen och kravet på 0,2 s fortfarande behövde uppnås för att systemet skulle bli godkänt genomfördes samma test som tidigare. Testet genomfördes efter att cylindern och trådväljaren kopplats samman med hjälp av ett egentillverkat fäste gjort av aluminium. På detta aluminiumfäste placerades en skruv som kunde aktivera microswitchen. Då testsystemet klarat alla försök under 0,2 s var den mest kritiska delen av examensarbetet avklarat.



*Figur 7 3D-modell, påbyggnadsdel till testriggen för att hålla uppe slangen till inslaget.*

#### 4. Systembeskrivning, hårdvara



*Figur 8 Färdig testtrigg kopplad med övriga komponenter och maskinen för att garantera att maskinen går att styra som planerat.*

## 4. Systembeskrivning, hårdvara

### 4.3 Pneumatik

Systemet som presenteras i Figur 9 Schema över all pneumatik som använts i examensarbetet visar alla de slutliga, pneumatiska komponenter som använts för examensarbetet.

#### 4.3.1 Cylinder

Åtta dubbelverkande cylindrar (kolvar) som fungerar som en övergång mellan övriga systemet och maskinen. Till skillnad från enkelverkande cylindrar som använder sig av en fjäder eller tyngdkraften för att komma tillbaka till startposition kan dessa cylindrar utföra ett arbete på både in och utgång då de är dubbelverkande.

Detta beror på vart i den dubbelverkande cylindern som tryckluften tillförts [6].

#### 4.3.2 Strypbackventil

Kopplat till var och en utav cylindrarna sitter två strypbackventiler. Dessa reglerar hastigheten av luftflödet och hindrar luften från att färdas åt fel håll. Detta minskar risken för övertryck eller ett skadat system och sparar de pneumatiska komponenternas livstid. Dessa ventiler gör också att man får ett utfall på cylindrarna mycket snabbare [21].

#### 4.3.3 5/2 Ventil

Varje cylinder styrs av ett eget 5/2 valv(ventil). Med dessa valv kan luftflödet kontrolleras och reglering av in- och ut-luft ske. Valven har fem anslutningar och två positioner som de kan skifta mellan, därav namnet [4].

#### 4.3.4 Output Manifold

Längre ner i systemet sitter en så kallad Output manifold. Denna del skulle kunna betraktas som en central punkt, eftersom den fungerar som en anslutningspunkt för flera av komponenterna som används. Alla åtta valv som går ut till cylindrarna är sammankopplade i detta manifold. Även luften från luftfilterregulatorn är kopplat till manifoldet. Denna ger en mer effektiv lösning till systemet och hanteringen av luften då fler utgångar erhålls. Alternativet skulle vara att koppla på rör direkt till valven vilket skulle vara mer otympligt och ta mer plats men också kunnat ge en ojämnare distribution av luften [5].

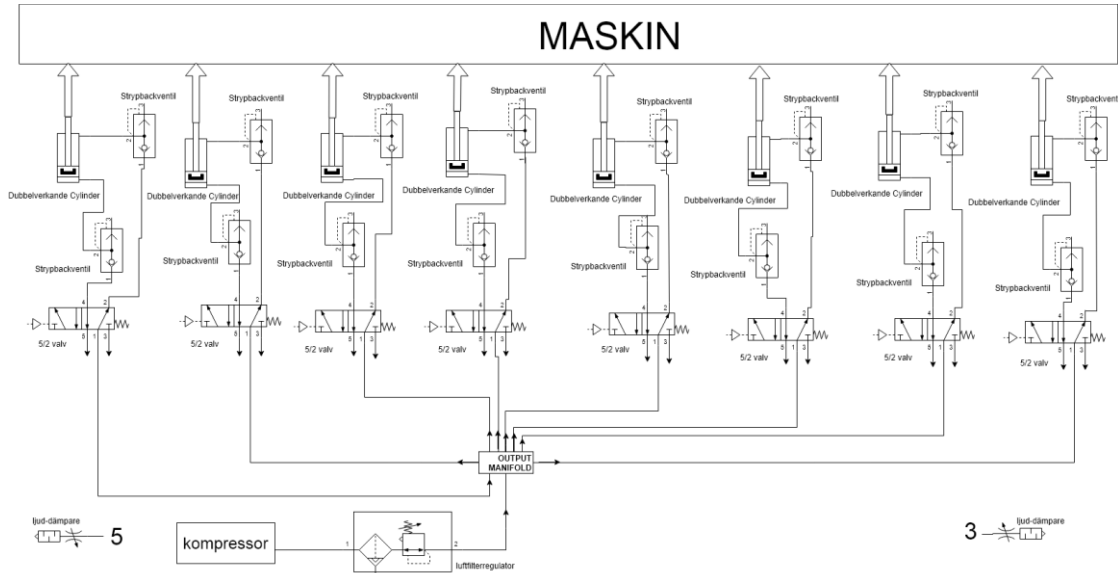
#### 4.3.5 Ljuddämpare

På manifoldet sitter två ljuddämpare då ljudet från cylindrar och tryckluft kan bli väldigt högt och minskade bullernivåer i fabriken är önskvärt för dem som ska vistas i lokalen längre perioder [22].

## 4. Systembeskrivning, hårdvara

### 4.3.6 Luftfilterregulator

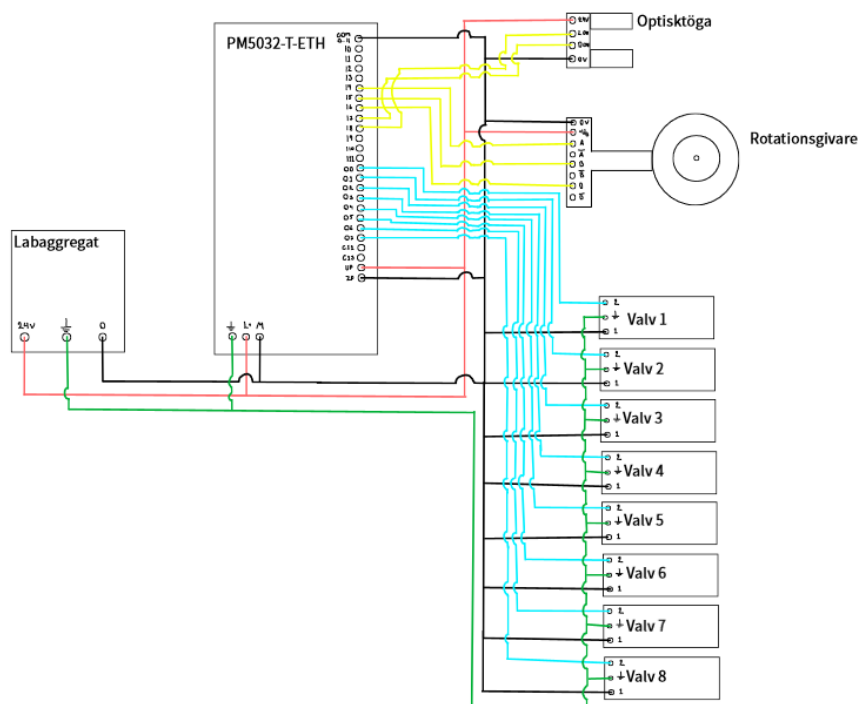
Kopplat till ett av manifoldets andra uttag sitter en luftfilterregulator. Där kan önskat tryck till systemet ställas in. Luftfilterregulatorn är också den komponent som ser till att luften som går in i ett system är ren och fri från smuts och damm. Detta är en väsentlig del då smuts och partiklar minskar livslängden av resterande komponenter [2, 3].



Figur 9 Schema över all pneumatik som använts i examensarbetet

## 4. Systembeskrivning, hårdvara

### 4.4 Elkopplingarna



Figur 10 Elschemat för systemet

Valven är 5/2 valven som är beskrivna tidigare i texten, de är kopplade till output o0-o7 enligt Figur 10 Elschemat för systemet. Valvens solenoider fungerar med av och på funktionen. När de får ström har de ett läge och när de är av trycker fjädern tillbaka dem till sitt ursprungliga läge.

Rotationsgivaren har sex signaler men bara tre behövs: A, B och 0. Dessa kopplades till input I4, I5 och I6. Det optiska ögat är kopplat till input I7 och I8 med dark-on respektive light-on.

## 5. Programlogik

### 5. Programlogik

I denna del av rapporten beskrivs mer utförligt hur kodningen är gjord. Examensarbetet innefattar dels PLC programmering och programmering i Python. Detaljerade förklaringar av de olika språk-koncepten, de använda biblioteken och koden för de olika testerna presenteras i detta kapitel.

#### 5.1 PLC programmering

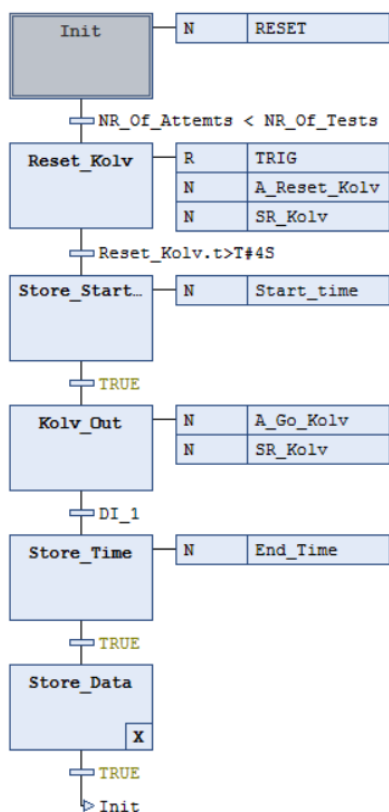
För att beskriva PLC-koden närmare har detta delats upp i två delar: PLC-koden för proof of concept och PLC-koden för vävmaskinen.

##### 5.1.1 PLC proof of concept

Då tiden enligt kraven var 0,2 s och tiden som gavs av realtidsklockan har en upplösning på 0,001 s var detta mer än tillräckligt för att uppfylla kraven. Tidfunktionen kan anses försumbar då PLC-processorn har en frekvens på 300 MHz vilket innebär att den inte bör påverka mätningar i millisekundintervall [12]. Koden som beskrivs nedan kommer bortse all tidsdata som inte är millisekunder.

I början av tidsexperimenten försäkrades att skillnaden från start till stopp alltid var mindre än en sekund. Flödet för koden ges enligt SFC-programmet nedan i Figur 11.

## 5. Programlogik



Figur 11 SFC program för tidtagning

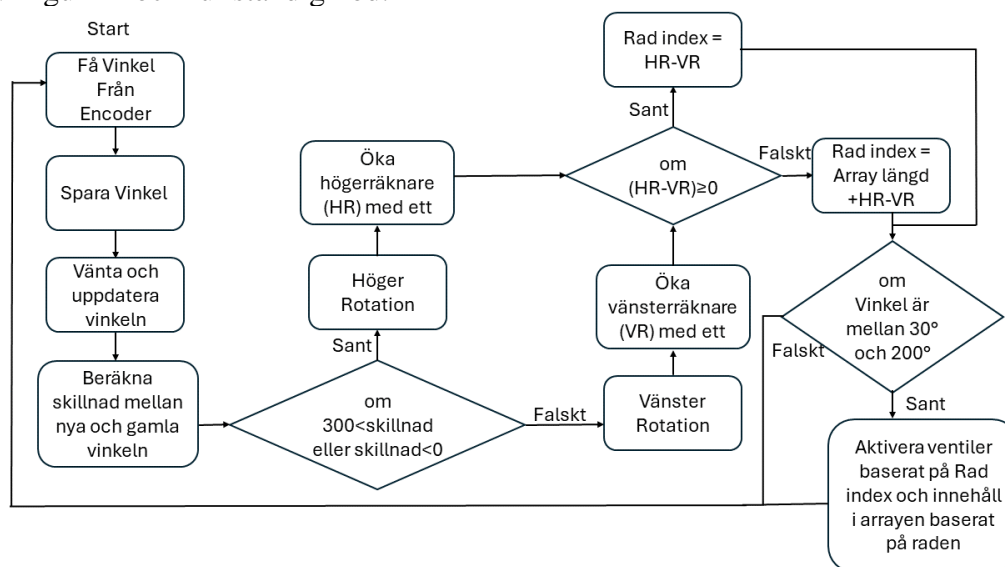
Reset sätter antal försök till noll men användes ej och Trig är kod som blev kvar från gamla iterationer.

A\_Reset\_Kolv sätter variablerna i SR\_Kolv så att kolven åker bakåt. Detta tog mindre än fyra sekunder men för att säkerhetsställa att kolven alltid skulle gå helt tillbaka så valdes fyra sekunder som ett mellanting mellan hastighet och väntetid.

Store\_Start\_Time har händelsen store time och lagrar tiden så fort det steget är klart sätts variablerna i A\_Go\_Kolv så att SR\_Kolv får kolven att åka ut och går vidare till nästa sekvens då DI\_1 blir TRUE, DI\_1 är i detta fall microswitchen som kolven trycker till när den är fullt uttryckt. Efter det så sparas sluttiden i End\_Time samt att skillnaden mellan End\_time och Start\_Time sparas. När programmet går ur Store\_Data så sparas tidsdatan in i en array och NR\_Of\_Attempts inkrementeras med ett. och sedan hoppar programmet tillbaka till Init.

### 5.1.2 PLC Vävmaskinen

Flödet för koden går att se i Figur 12. Det illustrerar alla moment och tillstånd som koden har. Koden har gjorts med en blandning av FBD och ST. Se bilaga A och B för större bild av Figur 12 och fullständig kod.

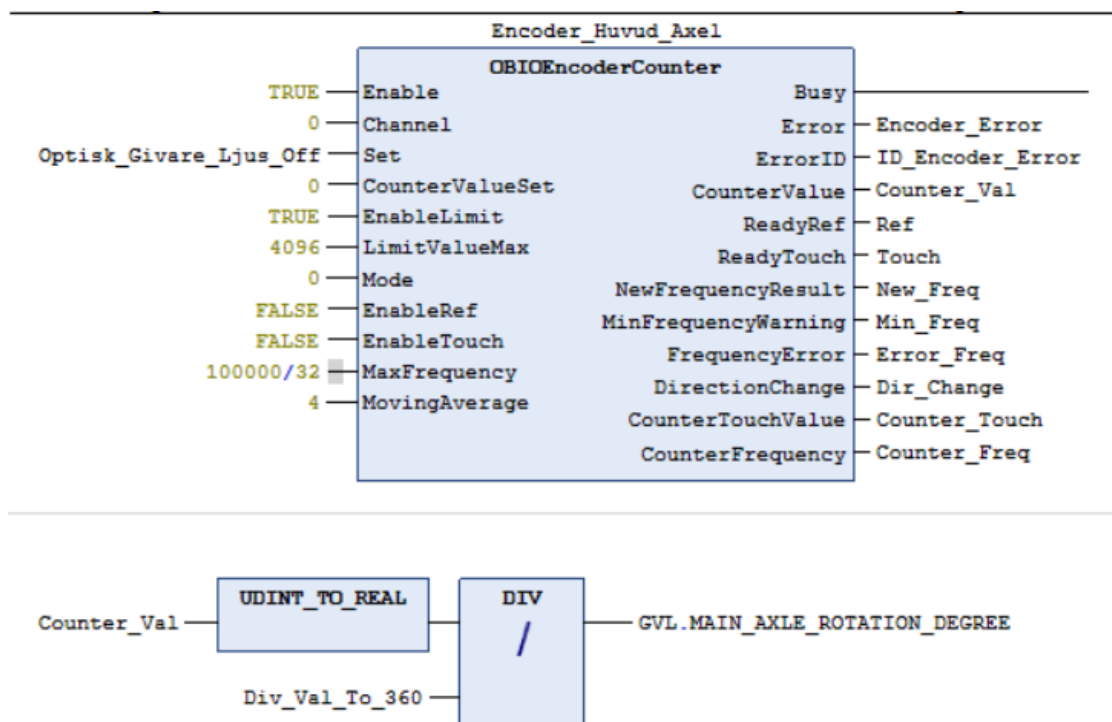


Figur 12 Flödet av PLC koden



## 5. Programlogik

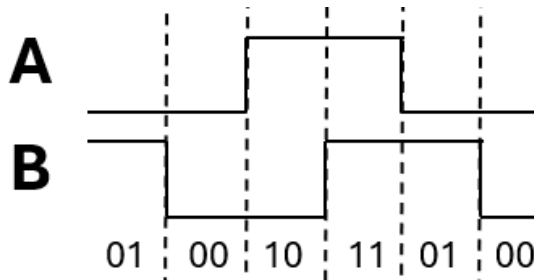
För att få vinkeln från encodern så specificerades input fyra som encoder track A, input fem som track B, input 6 som encoder reset och input sju som input för det optiska ögat när ljusstrålen bryts. Med hjälp av det färdiga funktionsblocket för encoders i ABBs AC500\_OnboardIO bibliotek så skapades blocket enligt Figur 13.



Figur 13 Rotationsgivarens program

Det som är värt att notera från Figur 13 Rotationsgivarens program är variabeln **LimitValueMax** då den sätter maxvärdet för **CounterValue**. **CounterValue** ökar eller minskar baserat på rotationsriktning och när tillstånd på encodern ändrats. Med encodern som används i detta projekt på 1024 pulser per track (A och B) ger detta fyra tillstånd. Tillstånden 00, 10, 11 och 01 som visas i Figur 14 ger 4096 tillståndförändringar per varv.

## 5. Programlogik



Figur 14 Signal formerna från rotationsgivaren

Set återställer CounterValue till det värde som CounterValue\_Set har. Set använder det optiska ögat för att återställa värdet på samma fysiska plats varje gång maskinen antingen sätts på eller när axeln roterat ett helt varv.

För att få ut ett värde som är enklare för programmerarna och användarna att förstå så konverteras CounterValue till antal grader på huvudaxeln med hjälp av en konstant. Detta fås enligt  $\frac{4096}{360} = 11,377 \dots$ . Resterande variabler används inte.

Det första som sker i ST programmet är att längden på arrayen beräknas. Denna array kontrollerar alla ventiler. Koden skrivs på följande vis:

```
Array_Len:=ABS(UPPER_BOUND(Ventil_Array,1)-LOWER_BOUND(Ventil_Array,1))+1;
```

Därefter så sparas vinkeln av encodern följt av en for-loop som kontinuerligt uppdaterar encoderns vinkel för att i nästa steg spara det nya värdet och beräkna skillnaden mellan det nya och det gamla, det resultat som fås kan då tolkas som rotationsriktning baserat på om skillnaden är positiv eller negativ.

Ett undantagsfall hittades som inträffa när encodern precis passerar  $0^\circ$ . Beroende på riktning går programmet från  $360^\circ \rightarrow 0^\circ$  eller  $0^\circ \rightarrow 360^\circ$  och gjorde att skillnaden  $T3 = T2 - T1$  blir stor och med fel tecken. Detta åtgärdas med if-satsen i följande kodrader:

```
Encoder();
T1 := GVL.MAIN_AXLE_ROTATION_DEGREE;
//A wait loop could possibly be decreased
FOR x:= 0 TO 1000 DO
    Encoder();
END_FOR

T2 := GVL.MAIN_AXLE_ROTATION_DEGREE;
T3 := T2-T1;
IF(T3<0 OR T3>300)THEN
    Right:=TRUE;
ELSE
    Right:=FALSE;
END_IF
```

## 5. Programlogik

Nästa steg är att öka räknare beroende rotationsriktning enligt följande kod:

```
IF(NOT RIGHT)THEN
    CounterUP_Right(CU:=Optisk_Givare_Ljus_Off
,RESET:=CounterUP_Right.Q,PV:= DINT_TO_WORD(Array_Len));
END_IF
IF(Right)THEN
    CounterUP_Left(CU:=Optisk_Givare_Ljus_Off
,RESET:=CounterUP_Left.Q,PV:= DINT_TO_WORD(Array_Len),);
END_IF
```

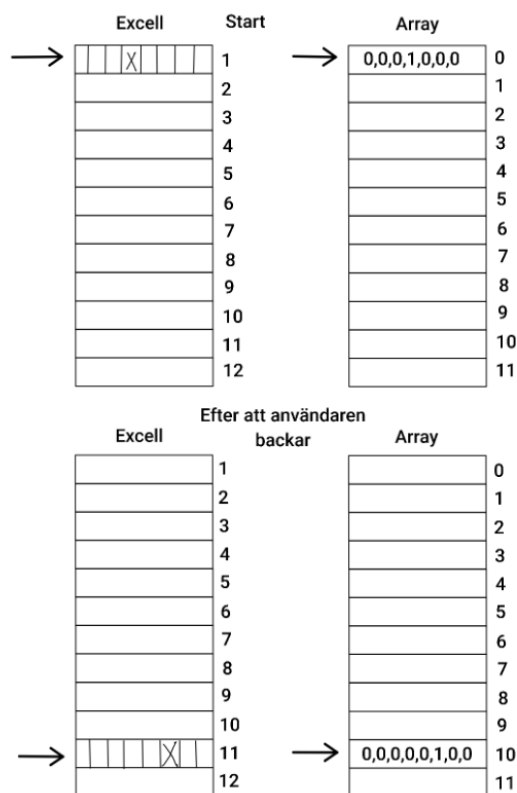
Värdena som fås av dessa räknare bildar tillsammans det index som används för att visa vilken rad i arrayen som ska exekveras. Det är viktigt att notera att båda räknarna har en övre gräns baserad på arrayens längd. När räknarna når denna gräns, återställs deras värde till noll. Indexet fås baserat på ett villkor, det villkoret möjliggör maskinens förmåga att kunna avancera i arrayen och backa. Villkoret är så länge CounterUP\_Right är större än eller lika stor som CounterUP\_Left.

Ett scenario på detta är då arrayen har 12 instruktioner. Användaren gör åtta rotationer, märker att något blev fel och backar sedan två rotationer, vilket resulterar i att Row\_Index blir sex och instruktion sex utförs. Om CounterUP\_Right är mindre än CounterUP\_Left kan samma beräkning inte utföras eftersom negativt index inte existerar. Lösningen till detta är att ta:

$$\text{arraylängden} + \text{CounterUP\_Right} - \text{CounterUP\_Left}.$$

Scenario två, om programmet är på instruktion noll och användaren backar två steg blir  $\text{CounterUP\_Right} - \text{CounterUP\_Left} = -2$ , Med en arraylängd på 12 kommer programmet ta in excell rad 11 vilket motsvarar arrayinstruktion tio då  $\text{arraylängden} + \text{CounterUP}_{Right} - \text{CounterUP}_{Left} = 12 + (-2) = 10$  se Figur 15 nedan.

## 5. Programlogik



Figur 15 Illustration av hur Excell arket ser ut i arrayformat

Sedan stegar programmet igenom varje instruktion beroende på rotationsriktning. Detta görs med koden nedan:

```
IF(UINT_TO_INT(CounterUP_Right.CV - CounterUP_Left.CV)>=0)THEN
    Row_Index :=UINT_TO_INT(CounterUP_Right.CV - CounterUP_Left.CV);
ELSE
    Row_Index := DINT_TO_INT(Array_Len) +
    UINT_TO_INT(CounterUP_Right.CV - CounterUP_Left.CV);
END_IF
```

När Row\_index är beräknat kontrollerar programmet vilka ventiler som ska åka ut och vilka som ska åka in. Det görs med följande kod:

```
//The pneumatics should be set and reset when the main axle of the weaving machine is
between 30 and 200 deg
IF(GVL.MAIN_AXLE_ROTATION_DEGREE > 30 AND
GVL.MAIN_AXLE_ROTATION_DEGREE <200) THEN
    FOR Index:= 0 TO 7 DO
        CASE Ventil_Array[Row_Index,Index] OF
            0:
                Reset_Ventil(Ventil:=Index);
```

## 5. Programlogik

```
1:
Set_Ventil(Ventil:=Index);

END_CASE
END_FOR
END_IF
```

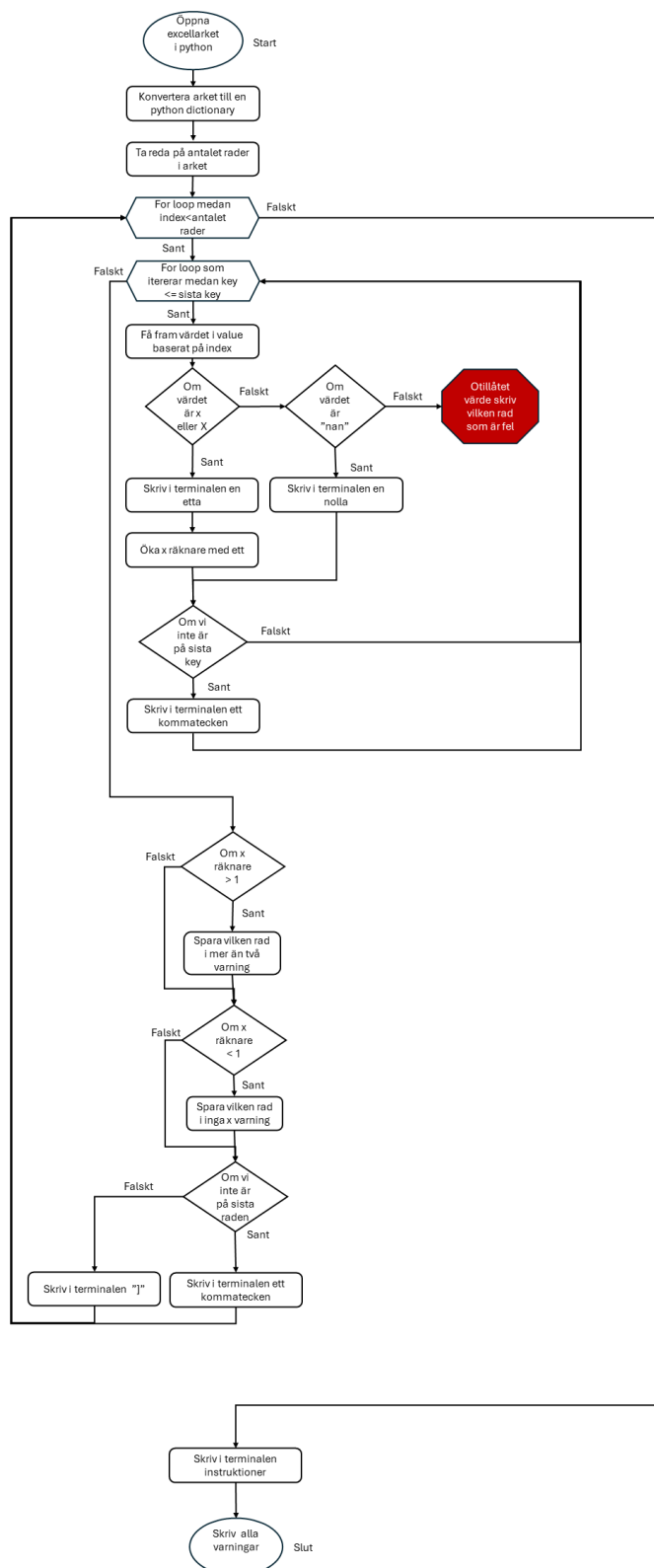
Om axeln är mellan 30° och 200° utför ventilerna sin nya instruktion baserat på arrayen där raden loopas igenom med en for-loop. Om en nolla dyker upp ska ventilen återställas (cylinder ska åka in) men om det är en etta ska ventilen sättas (cylinder åka ut).

### 5.2 Programmering i Python

Då användargränssnitt inte var ett fokus för detta projekt skapades det ett program för att förenkla mönsterbyte utan att behöva göra en array. Detta genomfördes på enklast möjliga sätt med det språk som används av flest programmerare och är ett av det enklaste att plocka upp därför valdes Python som programmeringsspråk [23].

Då ingen bakom detta arbete hade någon större erfarenhet av Python användes ChatGTP som hjälpmedel för syntax och för att påskynda programmeringen. Flera iterationer och tester gjordes av den kod som gavs av ChatGTP då AI-modellen ofta missförstod syftet med koden eller ändrade på viktig kod som redan hade skrivits. ChatGTP användes för att skriva transponeringen av datan samt villkoren för när ett kommatecken eller hakklammer ska skrivas. Det enda bibliotek som används i koden nedan är pandas och det används för att kunna öppna Excel-dokument i Python. Kodflödet för hela Pythonkoden visas i Figur 16. För större bild av Figur 16 se bilaga C och för hela koden se Bilaga D.

## 5. Programlogik



Figur 16 Flöde för hela koden som beskriver alla loopar och villkor. Koden skapar det mönster som ska klistras in i PLC koden. Se bilaga C för större bild

## 5. Programlogik

För att göra koden enklare att använda än att manuellt skriva en array av ettor och nollor, skapades en kod som tar x eller X i en Excel-cell och konverterar dem till ettor. När en cell är tom så skapas en nolla. I koden kallas en tom ruta för 'nan'.

Det första programmet gör är att hämta in ett Excel-dokument med namnet Pattern.xlsx och konverterar det till en python dictionary enligt koden under:

```
excel_file = 'Pattern.xlsx'  
df = pd.read_excel(excel_file)
```

När biblioteket hämtar in datan till dicten (dict är en datatyp som består av nyckel värde par) så transponeras Excel-arket enligt följande bilder från Figur 17 till Figur 18.

Kolv_1	Kolv_2	Kolv_3	Kolv_4	Kolv_5	Kolv_6	Kolv_7	Kolv_8
	x						
		x					
			x				
		x					
			x				
				x			

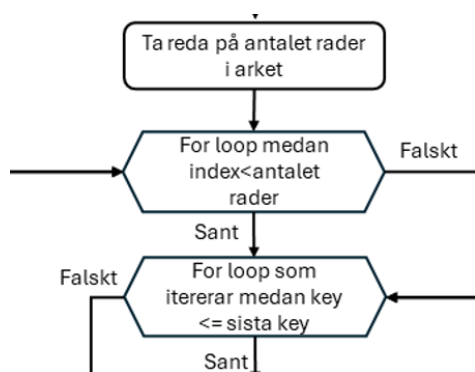
Figur 17 Mönstret i Excel

```
> 'Kolv_1': {0: nan, 1: nan, 2: nan, 3: nan, 4: nan, 5: nan}  
> 'Kolv_2': {0: 'x', 1: nan, 2: nan, 3: nan, 4: nan, 5: nan}  
> 'Kolv_3': {0: nan, 1: 'x', 2: nan, 3: 'x', 4: nan, 5: nan}  
> 'Kolv_4': {0: nan, 1: nan, 2: 'x', 3: nan, 4: 'x', 5: nan}  
> 'Kolv_5': {0: nan, 1: nan, 2: nan, 3: nan, 4: nan, 5: 'x'}  
> 'Kolv_6': {0: nan, 1: nan, 2: nan, 3: nan, 4: nan, 5: nan}  
> 'Kolv_7': {0: nan, 1: nan, 2: nan, 3: nan, 4: nan, 5: nan}  
> 'Kolv_8': {0: nan, 1: nan, 2: nan, 3: nan, 4: nan, 5: nan}
```

Figur 18 Dicten i Python

Detta innebär att samtidigt som x, X och 'nan' konverteras till ettor och nollor behöver datan transponeras till sitt ursprungliga format. Detta görs genom ett index som itererar i en for-loop och en for-loop inuti som går igenom dicten och dess data enligt Figur 19 och Figur 20.

## 5. Programlogik



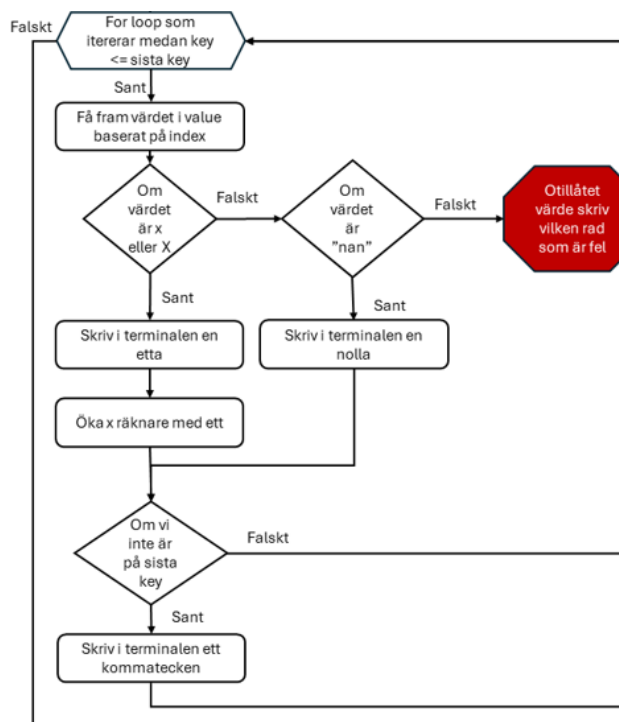
Figur 19 Del av Figur 16

```
for index in range(num_rows):
    x_counter=0

    # Iterate over each "Kolv" key
    for key, value in new_dict.items():
```

Figur 20 Pythonkod för Figur 19

Till exempel om värdena är: index = 0, key = kolv\_2 och value = array som key är kopplad till kommer val vara ett 'x' då kolv\_2s array på index noll granskas (rad ett i Figur 17). Nästa sekvens är det som sker inuti andra for-loopen enligt flödes-schemat och Figur 21 och Figur 22 nedan.



Figur 21 Del av Figur 16

```
val = value[index]
if val == 'x' or val == 'X':
    print("1", end='')
    x_counter+=1
elif pd.isna(val):
    print("0", end='')
else:
    return index

# Add a comma if it's not the last "Kolv"
if key != list(new_dict.keys())[-1]:
    print(",", end='')
```

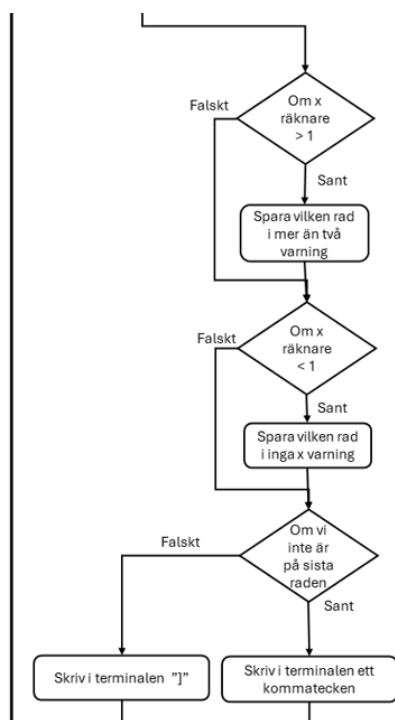
Figur 22 Pythonkod för Figur 21

Val är det värde som ska undersökas enligt Figur 22 ovan. om värdet är x eller X skrivs en etta i terminalen och x\_counter ökas med ett. x\_counter används för att se om någon varning behövs läggas till. Om värdet är 'nan' skrivs en nolla och om något annat dyker upp i en cell avslutas programmet och skriver vilken rad som var fel.



## 5. Programlogik

Sista if statmentet kollar ifall programmet är på kolv\_1-kolv\_7, om så är fallet skrivs ett kommatecken efter den etta eller nolla som precis har skrivits. Denna loop upprepas åtta gånger då det är åtta kolvar som ska styras vid varje instruktion. När dessa åtta loopar har gjorts kontrollerar programmet tre villkor: om programmet stött på fler än ett x på samma rad, om inga x har stötts på, och om sista raden har nåtts. Detta illustreras i Figur 23 och Figur 24 nedan.



Figur 23 Del av  
Figur 16

```
41     if x_counter > 1:
42         Warning_more_than_two_x.append(index+2)
43     if x_counter < 1:
44         Warning_no_x.append(index+2)
45
46     # Add a comma if it's not the last row
47     if index != num_rows - 1:
48         print(",")
49     else:
50         print("];") # End of the overall list
51
```

Figur 24 Pythonkod för Figur 23

Varningarna existerar för att tillåta användaren att använda mer än en tråd med hjälp av griparmen om så önskas, samt för att hantera situationer då ingen tråd finns. Detta var ett krav som tillkom under utveckling. Varningen fungerar genom att varje rad som uppfyller villkoren flera x eller inga x, sparas i en array för att senare i programmet skriva ut en varning.

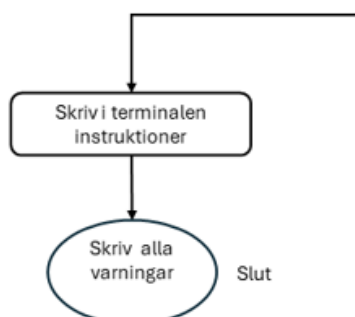
Som nämnt tidigare så har det bara satts ut kommatecken för de sju första då det finns ett specialfall för kolv\_8. Detta infaller då programmet är på sista raden i instruktionen. Detta illustreras Figur 23 ovan. Om programmet inte är på sista raden skrivs ett kommatecken, annars är programmet på sin sista rad och ska därmed skriva en hakparentes. Detta för att avsluta arrayen som ska sättas in i PLC programmet.

När arrayen är klar skrivs instruktioner ut för vart arrayen ska sättas in, hur lång arrayen är och platsen som variabeln har, se Figur 25 och Figur 26. Om allt har gjorts utan problem returneras noll och inga fel hanteras.

## 5. Programlogik

```
print(f'''  
Nu ska du koppla in datorn till PLC:t och starta ABBs Automation Builder 2.7 när du startat  
Tryck på öppna projektet och scanna efter plc, gå in på ventil_kontroll och sätt in informationen:  
  
Det mönster ovan ska sättas in Ventil_Array: ARRAY[0..12, 0..7] OF INT := HÄR SKA MÖNSTRET SÄTTAS IN (ARRAY)  
Antal rader ska sättas in Ventil_Array: ARRAY[0.. HÄR SKA ANTAL RADER IN, 0..7] OF INT :=  
ANTAL RADER är: {num_rows-1}  
  
EXCELL ARKET MÅSTE SPARAS EFTER VARJE FÖRÄNDRING  
''')  
  
for i in range(len(Warning_more_than_two_x)):  
    print(f'''WARNING MER ÄN EN KOLV AKTIVERAD PÅ RAD {Warning_more_than_two_x[i]} I EXCELLARKET''')  
  
for i in range(len(Warning_no_x)):  
    print(f'''WARNING INGEN KOLV AKTIVERAD PÅ RAD {Warning_no_x[i]} I EXCELLARKET''')  
  
return 0
```

Figur 25 Pythonkod av Figur 26



Figur 26 Del av  
Figur 16

## 6. Resultat

### 6. Resultat

Resultatet av rapporten beskriver en effektiv lösning för att digitalisera en vävmaskin och vägen dit. Hur uppsättningen och övergången från hålkorten till en modern dator har gjorts. Ytterligare beskrivning fokuserar på pneumatiken och målet att implementera ett system med tryckluft som styr trådarna till inslaget. Examensarbetet har resulterat i ett fungerande system som fungerar ihop med vävmaskinen.

#### 6.1 Proof of Concept

Som beskrivet i avsnitt 4.2 byggdes systemet och enligt avsnitt 3.4 har ett krav på 0,2 s från signal till att kolven är fullt ute. Detta kontrollerades och mättes med koden i avsnitt 5.1.1 och testades tre gånger med tre olika tryck: 2 bar, 4 bar och 6 bar. Detta gav då 100 resultat från varje försök, vid testet av 2 bar tog det 130 ms från signal till utslag. När 4 bar testades blev tiden 110 ms och 6 bar fick samma tid. Testet avbröts efter ca tio försök då krafterna blev för stora på kolven eftersom den inte hade någon belastning. Efter dessa försök utfördes samma försök med kolven på maskinen och resultaten blev: vid 2 bar 150-160 ms, vid 4 bar 120-130 ms och samma för 6 bar men vibrationer gjorde att delarna på testriggen lossna. Det gick då inte att slutföra alla 100 försök. Ett sista snabbt test med 20 tester vid sämsta möjliga position (då slangen som förflyttar rörelsen har störst motstånd) blev resultatet 180 ms vid 2 bar och 150 ms vid 4 bar se Tabell 2 tydligare visualisering. Detta visar ett bättre resultat än simulationerna i avsnitt 3.1, även vid suboptimala förhållanden. Kravet på 0,2 s har uppfyllts och det teoretiska kravet på 0,15 s har också uppfyllts. Resultaten visade att POC:en uppfyller kraven vilket gör att arbetet kunde gå vidare med att bygga hela lösningen enligt planen och göra en fullskalig montering på maskinen.

Tabell 2 Uppmätta tider vid olika tryck och monterings positioner

	2 bar	4 bar	6 bar
Utan belastning	130 ms	110 ms	110 ms
Monterad på maskin	150-160 ms	120 -130 ms	120-130 ms
Monterad med sämsta läge på maskin	180 ms	150 ms	Ingen data riggen gick sönder.

## 6. Resultat

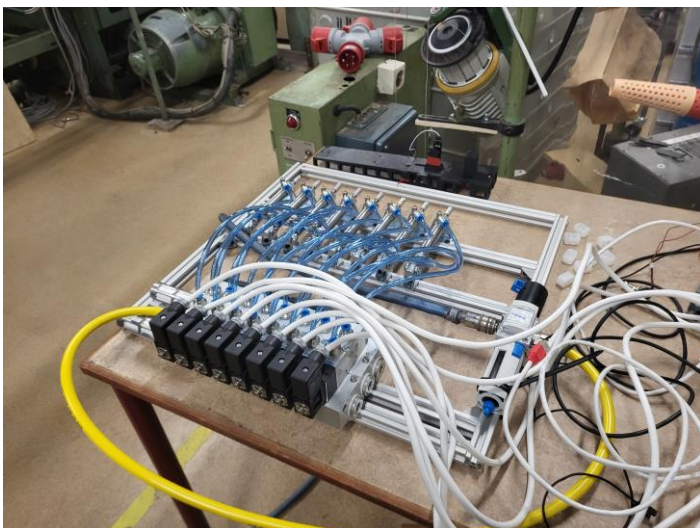
### 6.2 Fullskalig montering på maskinen

Efter att POC:en verifierats beställdes resterande delar samt material för att kunna fästa allt på maskinen. I väntan på att delarna skulle anlända testades sensorerna. Deras funktion verifierades och koden i avsnitt 5.1.2 skapades. Utöver koden och sensorerna gjordes även fästen för alla åtta slangar som för rörelsen från kolven till de öglor som kontrollerar trådarna. Fästen för sensorerna skapades också enligt Figur 27.



Figur 27 Optiskt öga, rotationsgivare och 3D-printat fäste.

Maskineriet till hålkorten avmonterades. En axel med en lämplig monteringsplats för sensorerna kontrollerades och bekräftades ha samma hastighet som huvudaxeln med hjälp av en takometer. En ram av aluminiumprofiler konstruerades och allt utom PLC:t fästes på ramen.



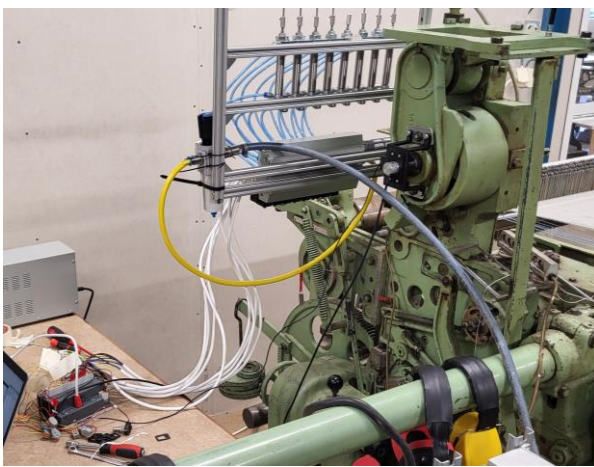
Figur 28 Ramen med pneumatiken monterad

## 6. Resultat

Därefter monterades ramen fast på maskinen och slangarna placerades i fästen som 3D-printats. Stängerna fästes i aluminiumfästena som monterats på kolvarna. Därefter monterades sensorerna på maskinen. Strypbackventilerna sattes till att låta fullt flöde genomgå för att få den högsta möjliga hastigheten. Montering av ramen och sensorerna är fotograferade enligt Figur 29 och Figur 30.



*Figur 29 Ramen med pneumatiken monterad på maskinen bakifrån*



*Figur 30 Ramen med pneumatiken monterad på maskinen framifrån*

När allt var monterat på maskinen och koden uppladdad på PLC:t skapades ett mönster med koden i avsnitt 5.2 och maskinen startades med lägsta hastighet. När det bekräftats att allt fungerade som det skulle ökades hastigheten lite i taget tills maskinen var uppe i full hastighet.

Nästa steg var att pröva med tråd. Två trådar sattes in och ett mönster skapades, vilket resulterade i att val av tråd stydes med maskinen. Det konstaterades att kolvarna arbetade i fel ordning. Detta rättades i koden och därmed fungerade allt efter alla krav och mål.

## 6. Resultat

Sist testades ett mönster med 2000 instruktioner för att se hur stor instruktionslistan kunde bli. Tråden som användes var 0,2 cm vilket innebar att längden på tyget skulle bli  $2000 \cdot 0,002 = 4$  m innan instruktionerna börjar om. Detta ansågs mer än tillräckligt. Därmed har maskinen utvecklats enligt de åtaganden som fastställdes och utifrån alla teorier och antaganden som formulerats genom arbetet.

## 7. Diskussion

### 7. Diskussion

I stycket nedan görs en utvärdering av systemet som byggts och de etiska aspekter som detta kan innebära. Utöver detta beskrivs vikten av ett hållbarhetstänk som varit en stor del av projektet.

Efter tester och beräkningar gjordes den praktiska delen. Maskinen visade sig fungera som planerat. De olika delarna arbetar tillsammans och det går att styra trådväljarna med hjälp av koden som implementerats. Detta har gjorts till en låg kostnad med egentillverkade delar och delar vars datablad noggrant lästs igenom för att allt ska fungera enligt plan.

Hålkorten har ersatts med en mer modern och effektiv lösning som tar mindre utrymme och kräver mindre arbetskraft. Hålkorten av plast och hela dess hölje har monterats bort för att få dit den nya lösningen. Moderniseringen av maskinen är långsiktigt en bra satsning främst ekonomiskt för företaget. Med det nya systemet finns en möjlighet till en uppgradering för Mallas Stenstugu AB. Om företaget i framtiden skulle växa och vara i behov av det, kan flera utav deras maskiner kopplas samman med PLC:t och styras från det.

Trots att vi har mött utmaningar vad gäller att hantera nya komponenter, avgöra rimligheten i komplexitet, komma på lösningar för att fästa olika delar med varandra, och utveckla en kod som fungerar fullt ut har projektet fortlöpt nästintill oproblematiskt.

Ett problem som upptäcktes i efterhand var att huvudaxelns nolläge inte verifierades samt att koden i avsnitt 5.1.2 säger att kolvarna ska åka ut mellan  $30^{\circ}$ - $200^{\circ}$  när de i själva verket ska förflyttas då huvudaxeln är mellan  $230^{\circ}$  och  $30^{\circ}$ . Trots att punkten för  $0^{\circ}$  inte hittades och att kolvarna rör sig på fel gradantal, fungerade maskinen. Detta beror sannolikt på en kombination av att kolven rörde sig i tid när huvudaxeln var vid  $180^{\circ}$  och att kolvarna rör sig snabbare, 0,13 s istället för den förväntade tiden på 0,2 s. Problemet med nollstället och tiden för kolvarnas rörelse går att rätta till i mjukvaran i efterhand. I brist på tid fanns inte möjligheten att kalibrera kolvarna med strypbackventilerna för att säkerställa att de rör sig med lägsta möjliga hastighet, vilket skulle minska stressen på delarna.

Att tänka långsiktigt och hållbart är inte något som enbart gynnar företagen utan även människor världen över. Textilindustrin är en industri där många människor runtom i världen utnyttjas och arbetar under extremt dåliga omständigheter [24]. I diverse länder finns oetiska sätt att hantera både människor och material på.

Genom att utföra detta examensarbete som gynnar svensk textilindustri är förhoppningen att få mer inhemsk textilproduktion och därmed flytta industrin från en mindre reglerad marknad till en mer reglerad marknad.

## 7. Diskussion

De flesta produkterna som beställdes var från lager i Sverige. Vid något enstaka tillfälle fraktades någon del från England vilket innebar en sämre inverkan på miljön. För att ha så låg negativ inverkan på miljön som möjligt gjordes valen att beställa allt som gick från återförsäljare i Sverige med ambitionen att minska fraktutsläpp.

I datorer finns det konfliktmetaller, på grund av detta har vi valt att beställa PLC:t från ett EU land som har reglering av de konfliktmetaller som finns i produkten.



## 8. Slutsats

### 8. Slutsats

De mest centrala delarna för projektet har innefattat en fördjupning i de olika komponenterna och dess tekniska specifikationer såsom det elektriska och det pneumatiska. Även ritningar i 3D-program för att få till rätt mått, byggandet och koden har varit essentiella.

För att uppnå ett resultat som uppfyllde samtliga krav har det varit avgörande att bryta ner arbetet i små steg och arbeta metodiskt. Första steget var att testa i teorin och säkerställa att beräkningar var rimliga och därefter övergå till att arbeta med det praktiska. På detta sätt var det enklare att felsöka och gå tillbaka för att korrigera. Konstruktionen har uppfyllt de krav som specificerats i början av projektet.

Konstruktionen av systemet kan användas för att styra trådväljarna till inslaget. För att göra en liknande lösning till varpen krävs mer tid och fler komponenter då det ingår många fler trådar i varpen. Detta skulle kunna vara ett möjligt projekt framtida projekt. Om projektet skulle vidareutvecklas bör dock även koden uppdateras efter detta.

## Litteraturförteckning

- [1] K.A. Almgren sidenväveri & museum, ”Vävsalen och vävning,” U.Å. [Online]. Available: <https://www.kasiden.se/vavsalen-och-vavning/>. [Använd 2024 05 30].
- [2] Filter regulator, ”Festo,” U.Å. [Online]. Available: <https://www.festo.com/se/en/a/download-document/datasheet/529144>. [Använd 2024 05 30].
- [3] Festo, ”Filter regulators MS-LFR, MS series,” 2023 05. [Online]. Available: <https://www.festo.com/media/pim/747/D15000100122747.PDF>. [Använd 2024 06 30].
- [4] RS Components AB, ”Solenoid valves VUVS/valve manifold VTUS,” u.d. [Online]. Available: <https://docs.rs-online.com/de77/0900766b814f9de4.pdf>. [Använd 2024 06 30].
- [5] Festo, ”Manifold block VABM-B10-25E-G12-8,” u.d.. [Online]. Available: <https://festo.com/se/en/a/download-document/datasheet/8026249>. [Använd 2024 06 30].
- [6] RS Components AB, ”Round cylinders DSNU/DSNUP/DSN/ESNU/ESN,” u.d. [Online]. Available: <https://docs.rs-online.com/1f17/0900766b814f0729.pdf>. [Använd 2024 06 30].
- [7] P.-Å. Jansson, R. Grahn och M. Enelund, *Mekanik statik och dynamik*, Lund, Studentlitteratur AB, 2019, pp. 13-14.
- [8] Panasonic industry, ”Amplifier Built-in U-shaped Micro Photoelectric Sensor,” 10 2017. [Online]. Available: [https://api.pim.na.industrial.panasonic.com/file\\_stream/main/fileversion/2113](https://api.pim.na.industrial.panasonic.com/file_stream/main/fileversion/2113). [Använd 2024 06 30].
- [9] kübler, ”Incremental encoders,” 01 2006. [Online]. Available: [https://media.distrelec.com/Web/Downloads/t\\_/en/vt2400\\_210160-65\\_dat\\_en.pdf](https://media.distrelec.com/Web/Downloads/t_/en/vt2400_210160-65_dat_en.pdf). [Använd 04 07 2024].
- [10] B.Thomas, *Modern Reglerteknik*, Liber, 2016 5:e upplagan, pp. 428-431.
- [11] ABB, ”Styrssystem i det mindre formatet,” U.Å. [Online]. Available: <https://new.abb.com/se/om-abb/teknik/sa-funkar-det/plc>. [Använd 2024 05 30].
- [12] ABB, ”PM5032-T-ETH,” U.Å. [Online]. Available: <https://new.abb.com/products/sv/1SAP123400R0172/pm5032-t-eth>. [Använd 2024 05 30].
- [13] F. Wennersten, Undervisningstekniker vid Textilhögskolan i Borås. [Intervju]. 21 02 2024.
- [14] Wikipedia, ”Stegmotor,” [Online]. Available: <https://sv.wikipedia.org/wiki/Stegmotor>. [Använd 2024 06 25].

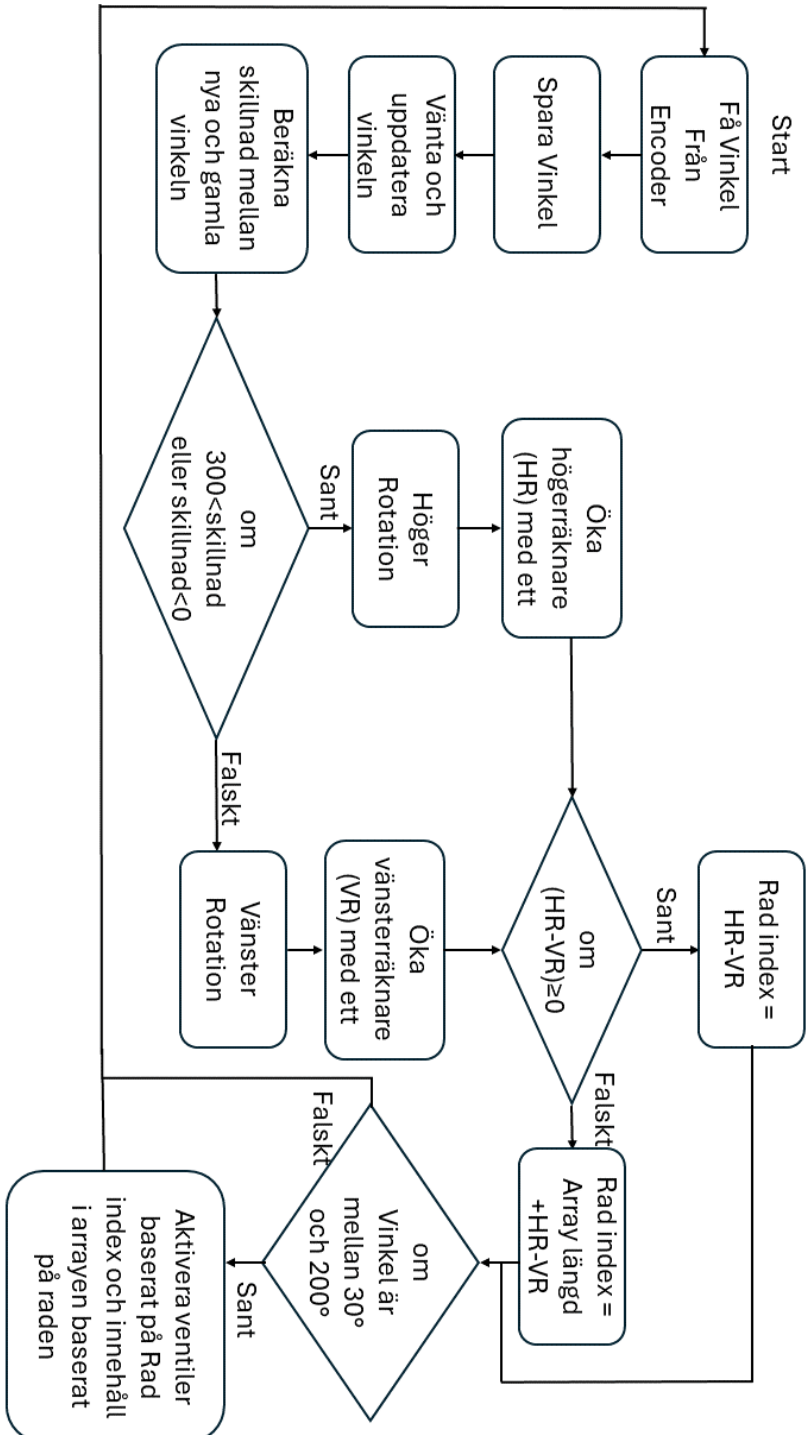
## Litteraturförteckning

- [15] Electricity and Magnetism, "How do solenoids work?," [Online]. Available: <https://www.electricity-magnetism.org/how-do-solenoids-work/>. [Använd 2024 06 30].
- [16] S. Held, "Choosing between absolute and incremental encoders: it involves more than simply deciding if you need to know how to get back home," *Control Solutions International*, 09 2003.
- [17] Glamox, "Ip-klasser-en guide," U.Å. [Online]. Available: <https://www.glamox.com/sv/pbs/kunskapsbank/ip-klasser/>. [Använd 2024 05 30].
- [18] Norwesco, "Vad är IP-klassning och vilka olika IP-klasser finns?," U.Å. [Online]. Available: <https://www.norwesco.se/sv/faq/vad-ar-ip-klassning/>. [Använd 2024 05 30].
- [19] J. Borg, S. Sagen och S. Tivell, "Att specificera ullgarn," Textilhögskolan, Borås, 2019.
- [20] EU, "Europaparlamentet," 21 03 2017. [Online]. Available: <https://www.europarl.europa.eu/topics/sv/article/20170314STO66681/konfliktmineraler-sanningen-bakom-din-smartphone>. [Använd 04 07 2024].
- [21] Festo, "one-way flow control valve GRLA-M5-QS-6-D," u.d. [Online]. Available: <https://www.festo.com/se/en/a/download-document/datasheet/193139>. [Använd 2024 07 04].
- [22] Festo, "ISO cylinder DSNU-16-60-P-A," u.d. [Online]. Available: <https://www.festo.com/us/en/a/download-document/datasheet/1908263>. [Använd 2024 07 04].
- [23] "PYPL PopularitY of Programming Language," 07 2024. [Online]. Available: <https://pypl.github.io/PYPL.html>. [Använd 05 07 2024].
- [24] European Parliament, "Textile workers in developing countries and the European fashion industry," [Online]. Available: [https://www.europarl.europa.eu/RegData/etudes/BRIE/2020/652025/EPRS\\_BRI\(2020\)652025\\_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/BRIE/2020/652025/EPRS_BRI(2020)652025_EN.pdf). [Använd 2024 07 08].
- [25] S. Ka'ka, D. Kambuno och A. Tangkemandu, "Damping transformation modeling on wheel suspension using pneumatic cylinder thrust force as a substitute for vehicle weight," *Extrica*, p. 14, 26 11 2022.

Bilagor

Bilagor

## Bilaga A PLC flöde



Bilagor

## Bilaga B PLC kod

```
//Checks Array Length
Array_Len:= ABS(UPPER_BOUND(Ventil_Array, 1) - LOWER_BOUND(Ventil_Array,
1))+1;

//***** This Segment checks wich way the encoder is rotating *****
Encoder();
T1 := GVL.MAIN_AXLE_ROTATION_DEGREE;

//A wait loop could possibly be decreased
FOR x:= 0 TO 1000 DO
    Encoder();
END_FOR

T2 := GVL.MAIN_AXLE_ROTATION_DEGREE;
T3 := T2-T1;
IF(T3<0 OR T3>300)THEN
    Right:=TRUE;
ELSE
    Right:=FALSE;
END_IF
```

Bilagor

```

//*****
*****

//Loop for the M part of the MxN array uses the optical sensor as the counter up and resets after
reaching the array length
IF(NOT RIGHT)THEN
    CounterUP_Right(CU:=Optisk_Givare_Ljus_Off ,RESET:=CounterUP_Right.Q,PV:=
DINT_TO_WORD(Array_Len));
END_IF
IF(Right)THEN
    CounterUP_Left(CU:=Optisk_Givare_Ljus_Off ,RESET:=CounterUP_Left.Q,PV:=
DINT_TO_WORD(Array_Len),);
END_IF

//IF the counter diff is larger then zero right-left works but when it is negative we need to
subtract from array_len (the legth of the punchcard)
IF(UINT_TO_INT(CounterUP_Right.CV - CounterUP_Left.CV)>=0)THEN
    Row_Index :=UINT_TO_INT(CounterUP_Right.CV - CounterUP_Left.CV);
ELSE
    Row_Index := DINT_TO_INT(Array_Len) + UINT_TO_INT(CounterUP_Right.CV -
CounterUP_Left.CV);
END_IF

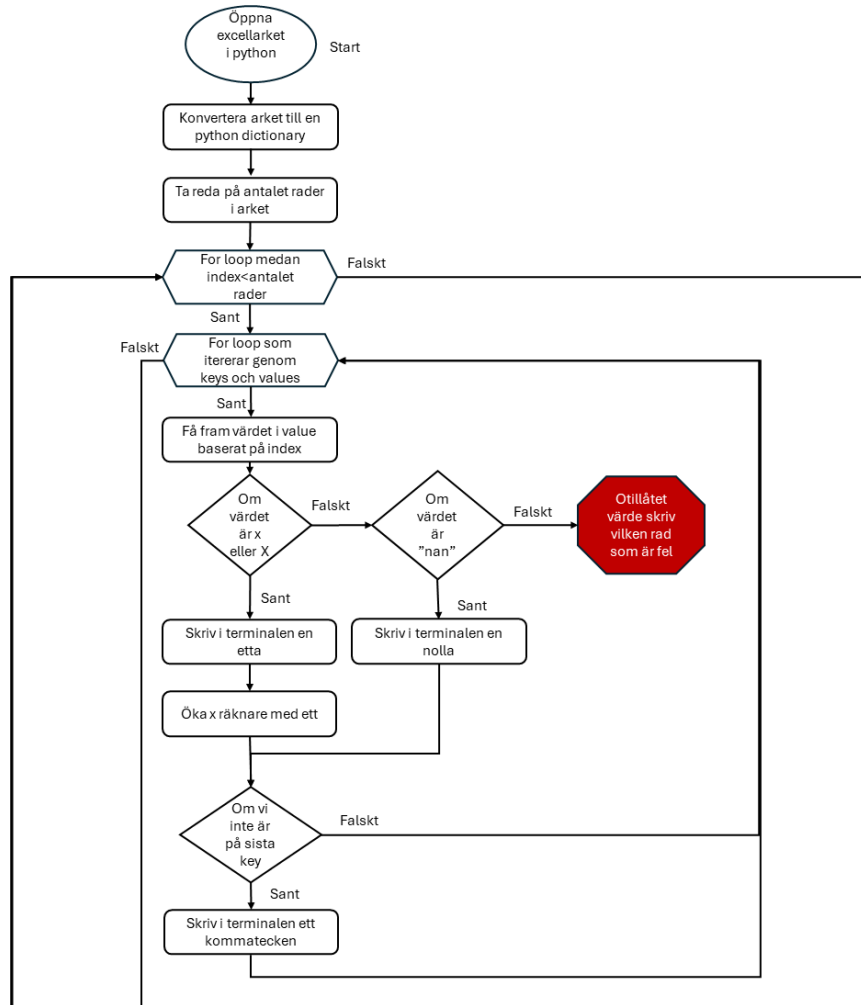
//The pneumatics should be set and reset when the main axle of the weaving machine is between
30 and 200 deg
IF(GVL.MAIN_AXLE_ROTATION_DEGREE > 30 AND
GVL.MAIN_AXLE_ROTATION_DEGREE <200) THEN
    FOR Index:= 0 TO 7 DO
        CASE Ventil_Array[Row_Index,Index] OF
            0:
                Reset_Ventil(Ventil:=Index);
            1:
                Set_Ventil(Ventil:=Index);

        END_CASE
    END_FOR
END_IF

```

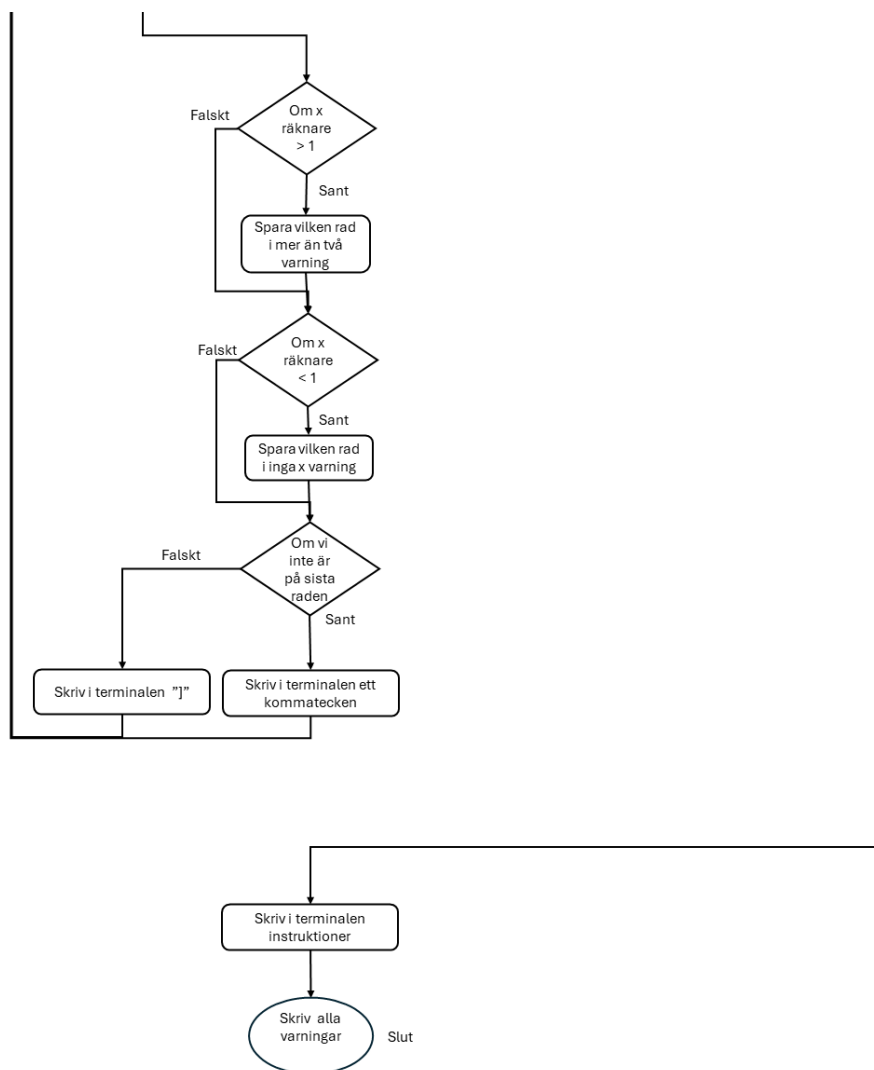
# Bilagor

## Bilaga C Python flöde





# Bilagor



## Bilaga D python kod

```
import pandas as pd

def main():
    excel_file = 'Pattern.xlsx'

    df = pd.read_excel(excel_file)

    new_dict = df.to_dict()

    print('MÖNSTRET:')
    print('[',end='')

    # Get the number of columns (number of "Kolv" keys)
    num_columns = len(new_dict)

    # Get the number of rows (length of values in any "Kolv" key)
    num_rows = len(next(iter(new_dict.values())))
    Warning_more_than_two_x = []
    Warning_no_x = []
    # Iterate over each index (0, 1, 2, ...)
    for index in range(num_rows):
        x_counter=0

        # Iterate over each "Kolv" key
        for key, value in new_dict.items():
            val = value[index]
            if val == 'x' or val == 'X':
                print("1", end='')
                x_counter+=1
            elif pd.isna(val):
                print("0", end='')
            else:
                return index

        # Add a comma if it's not the last "Kolv" key
        if key != list(new_dict.keys())[-1]:
            print(",", end='')
        if x_counter > 1:
```

## Bilagor

```
Warning_more_than_two_x.append(index+2)
if x_counter < 1:
    Warning_no_x.append(index+2)

# Add a comma if it's not the last row
if index != num_rows - 1:
    print(",")
else:
    print("];") # End of the overall list

print(f''
```

Nu ska du koppla in datorn till PLC:t och starta ABBs Automation Builder 2.7 när du startat

Tryck på öppna projektet och scanna efter plc, gå in på ventil\_kontroll och sätt in informationen:

Det mönster ovan ska sättas in Ventil\_Array: ARRAY[0..12, 0..7] OF INT :=  
HÄR SKA MÖNSTRET SÄTTAS IN (ARRAY)

Antal rader ska sättas in Ventil\_Array: ARRAY[0.. HÄR SKA ANTAL RADER IN,  
0..7] OF INT :=

ANTAL RADER är: {num\_rows-1}

EXCELL ARKET MÅSTE SPARAS EFTER VARJE FÖRÄNDRING  
''')

```
for i in range(len(Warning_more_than_two_x)):
    print(f''WARNING MER ÄN EN KOLV AKTIVERAD PÅ RAD
{Warning_more_than_two_x[i]} I EXCELLARKET''')
```

```
for i in range(len(Warning_no_x)):
    print(f''WARNING INGEN KOLV AKTIVERAD PÅ RAD {Warning_no_x[i]} I
EXCELLARKET''')
```

```
return 0
```

```
err = main()
```

```
if err != 0:
    print(f''Något är fel på rad {err+2}!
```

## Bilagor

Har du skrivit något annat än "x", "X" eller " "''')

INSTITUTIONEN FÖR RYMD-, GEO- OCH MILJÖVETENSKAP  
CHALMERS TEKNISKA HÖGSKOLA  
Göteborg, Sverige 2024  
[www.chalmers.se](http://www.chalmers.se)



**CHALMERS**