

# Simulering av maskin med hjälp av TwinCat3 TE1400 och Simulink



---

**Andreas Kanewoff**

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



# **Simulering av maskin**

## **Med hjälp av TwinCat3 TE1400 och Simulink**

**Simulering för att testa PLC kod i kontorsmiljö.**



**LUNDS  
UNIVERSITET**

Lunds Tekniska Högskola

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

Examensarbete:  
Andreas Kanewoff

© Copyright Andreas Kanewoff

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 2022

## Sammanfattning

Randek AB är ett företag som utvecklar och producerar maskiner för tillverkning av prefabricerade hus. De utvecklar automationslösningar för tillverkning av exempelvis vägg-, golv, takstolar. De använder sig idag utav Beckhoff Automation för sin PLC styrning. De önskade kunna testa sin PLC kod i kontorsmiljö för att underlätta felsökningen av befintliga anläggningar samt vid driftsättning av nya. Utöver det ville de även kunna testköra ny kod utan att ha den fysiska maskinen på plats.

Detta kan göras med hjälp av en simulering av systemet. Två olika sätt har undersökts för att se om det är möjligt att skapa en simulering. Den ena med hjälp av TwinCat3 TE1400, Target for Simulink som senare användes för att utveckla simuleringen och den andra metoden med hjälp av TwinCat3 TE1410, Interface for Matlab/Simulink. Utöver det har även ett HMI utvecklats för att kunna styra samt representera data från simuleringen.

Examensarbetet började med en utbildningskurs i TwinCat3 som företaget tillhandahöll. Detta lade grunden för den kunskap som krävdes för att kunna genomföra arbetet i TwinCat3. Efter det började arbetet med att utveckla både simuleringen och ett HMI för att styra den.

Nyckelord: TwinCat3, Simulink, PLC, HMI, IEC 61131-3

## **Abstract**

Randek AB is a company who develops and produces machines for production of prefabricated houses. They develop automation solutions for the manufacture of, for example, walls, floors and trusses. They wanted to be able to test their PLC code in an office environment to facilitate troubleshooting of existing facilities and when commissioning new ones. In addition to that they also wanted to be able to test new code without having the physical machine in place.

This can be done using a simulation of the system. Two different ways have been investigated, to see if it's possible to develop a simulation. One using TwinCat3 TE1400, Target for Simulink which was later used to develop the simulation and the second method using TE1410, Interface for Matlab/Simulink. In addition, an HMI has been developed to be able to control and represent data from the simulation.

The thesis started with a training course in TwinCat3 that the company provided. This laid the foundations of the knowledge required to be able to accomplish the work in TwinCat3. After that, the work began on developing both the simulation and the HMI to control it.

**Keywords:** TwinCat3, Simulink, PLC, HMI, IEC 61131-3

## Förord

Examensarbetet som genomförts är den avslutande delen av högskoleingenjörsutbildningen i elektroteknik och automation vid Lunds tekniska högskola på Campus Helsingborg. Arbetet har genomförts under hösten 2021 i samarbete med Randek AB.

Projektet har varit väldigt lärorikt och jag är väldigt tacksam för att jag fick möjligheten att genomföra mitt examensarbete hos Randek AB. Jag har blivit väl mottagen av både min handledare på företaget och de andra medarbetarna och vill tacka alla som har hjälpt mig under projektets gång.

Ett speciellt tack till Sebastian Blom, min handledare på Randek AB som har guidat och hjälpt mig under hela projektet.

Jag vill även tacka min handledare Bengt Simonsson och min examinator Mats Lilja för den hjälp och guidning de har bidragit med samt för deras undervisning under min tid på LTH.

Andreas Kanewoff

# Innehållsförteckning

<b>1 Inledning</b> .....	<b>1</b>
1.1 Bakgrund .....	1
1.2 Syfte .....	1
1.3 Målformulering .....	1
1.4 Problemformulering .....	2
1.5 Motivering av examensarbetet .....	2
1.6 Avgränsningar .....	2
<b>2 Teknisk bakgrund</b> .....	<b>3</b>
2.1 IEC 61131-3 .....	3
2.2 TwinCat3 .....	4
2.3 ADS -Automation Device Specification .....	4
2.4 TE1400 TwinCat3 Target for Simulink .....	5
2.5 TE1410 TC3, Interface for Matlab/Simulink .....	5
2.6 Simulink Coder .....	6
2.7 TC3 NC PTP .....	6
2.8 TwinCat PLC Control visualization Editor .....	7
2.9 Testcertifikat .....	7
<b>3 Metod</b> .....	<b>8</b>
3.1 Förstudier .....	8
3.2 Arbetsmetod .....	9
3.3 Inhämtning av information .....	9
3.4 Inledning .....	10
3.5 Förståelse för PLC-programmet .....	10
3.6 Utveckling av Simulink-modell .....	11
3.7 Utveckling av HMI och PLC kod .....	12
3.8 Test i TwinCat3 .....	12
3.9 Källkritik .....	13
<b>4 Analys</b> .....	<b>14</b>
4.1 Simulink .....	14
4.2 TE1410 TC3, Interface for Matlab/simulink .....	14
4.3 TE1400 TC3, Target for Simulink .....	15
4.4 Modeller i TwinCat3 .....	15
4.5 HMI .....	16
4.6 Simulering av servo .....	18
4.7 Testcertifikat .....	20
<b>5 Resultat</b> .....	<b>22</b>
5.1 Simulering .....	22
5.2 HMI .....	23
5.3 Simulink-modeller .....	24
5.3.1 Frekvensomriktare och transportband .....	24



5.3.2 Modell för Breddomställning.....	26
5.3.3 Spikportal .....	27
6 Slutsats.....	29
7 Framtida utvecklingsmöjligheter .....	30
8 Terminologi .....	31
9 Källförteckning.....	32



# 1 Inledning

## 1.1 Bakgrund

Företaget där jag har genomfört mitt examensarbete heter Radek AB. De utvecklar och producerar maskiner för prefabricerad hustillverkning. De producerar automationslösningar för tillverkning av exempelvis vägg-, golv- och takstolar samt taklinjer.

Efter samtal med företaget framgick det att de gärna vill ha möjlighet att kunna testa sin kod utan att ha den fysiska maskinen på plats. Detta kan lösas genom att simulera maskinen för att sedan testköra koden i simuleringsmiljön. Däremot kan simuleringsmiljön utvecklas på olika sätt. Därför genomfördes en förstudie för att ta reda på vilket sätt att skapa simuleringsmiljön som är bäst anpassat för Radeks syfte.

## 1.2 Syfte

Syftet med examensarbetet har varit att utveckla en simuleringsmiljö för en eller flera av Radeks maskiner så att det blir enklare att testa sin kod utan att ha den verkliga maskinen på plats. Det förväntade resultatet är att utveckla en simuleringsmiljö som gör att Radek kan köra sin kod i kontorsmiljö vilket kan spara mycket tid framför allt vid driftsättning av nya anläggningar men även vid felsökning av befintliga anläggningar.

## 1.3 Målformulering

Examensarbetet ska undersöka på vilka sätt det går att simulera maskinen och redogöra för fördelar och nackdelar med de olika metoderna.

De program som kommer att användas är TE1410 TwinCat3 Interface for Matlab/Simulink och TE 1400 TwinCAT3 Target for Simulink.

Målet är att skapa en färdig simuleringsmiljö för minst en av Radeks maskiner med det mest lämpliga alternativet ovan.

## 1.4 Problemformulering

Examensarbete kommer att omfatta utvecklingen av en simuleringsmiljö där följande problemfrågor ska besvaras:

- Hur ska simuleringsmiljön fungera? Grafiskt?
- Vilken programvara ska användas för att utveckla simuleringsmiljön?
- Vilka kriterier ska användas för att bestämma vilken programvara som ska användas?
- Hur fungerar maskinen?

## 1.5 Motivering av examensarbetet

Då jag är bosatt i Falkenberg kändes det självklart att försöka hitta ett företag i närområdet som jag kan göra mitt examensarbete tillsammans med. Randek är ett spännande företag med egen utveckling och produktion som kändes väldigt relevant för min utbildning. Efter förfrågan fick jag tre olika projekt att välja mellan men då jag tycker automationsteknik och PLC-programmering är spännande landade jag i att detta projekt passade mig bäst.

## 1.6 Avgränsningar

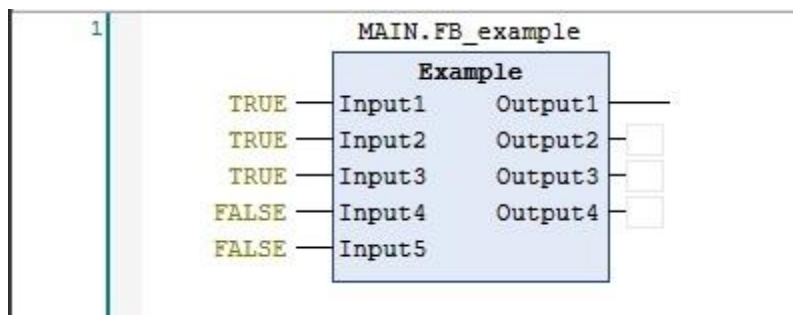
Projektet kommer att begränsas till att undersöka om det går att genomföra simuleringen med hjälp av TE1400 TwinCat3 Target for Simulink och TE1410 TC3 Interface for Matlab/Simulink. Utöver det ska även en simuleringsmiljö utvecklas för minst en av Randeks maskiner.

## 2 Teknisk bakgrund

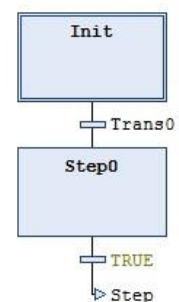
### 2.1 IEC 61131-3

Det finns flera olika programspråk som kan användas för att programmera en PLC. Av den anledningen har de vanligaste språken standardiserats i IEC 61131-3. Detta är en standard som främst används inom industriell automation. I standarden anges vilka datatyper, typer av variabler samt vilka programmeringsspråk som ingår. De finns både textbaserade språk som IL och ST samt grafiska programmeringsspråk som exempelvis SFC och LD. De programmeringsspråk som används i IEC 61131-3 är:

1. Sequential Function Charts (SFC)
2. Ladder Diagram (LD)
3. Instruction List (IL)
4. Function Block Diagram (FBD)



Figur 2. Exempel på ett enkelt FBD.



Figur 1. Exempel på SFC.

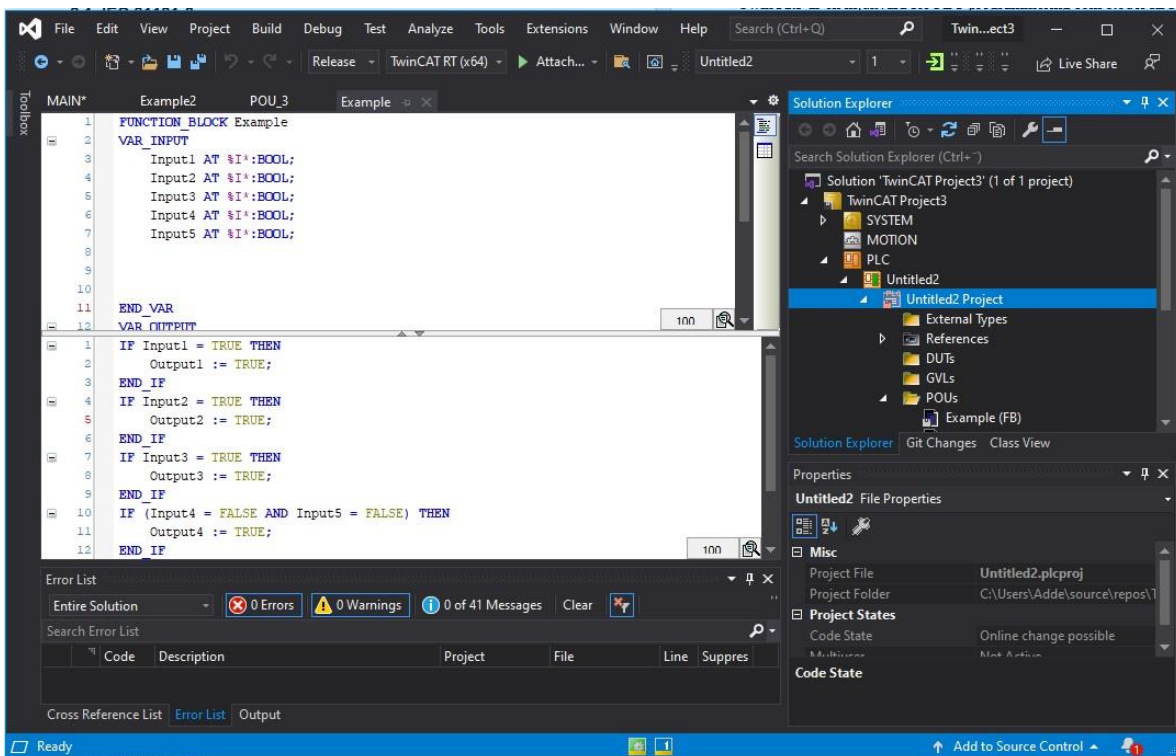
### 5. Structured Text (ST)

```
IF Input1 = TRUE THEN
    Output1 := TRUE;
END_IF
IF Input2 = TRUE THEN
    Output2 := TRUE;
END_IF
IF Input3 = TRUE THEN
    Output3 := TRUE;
END_IF
IF (Input4 = FALSE AND Input5 = FALSE) THEN
    Output4 := TRUE;
END_IF
```

Figur 3. Exempel på ST kod.

## 2.2 TwinCat3

TwinCat3 är en mjukvara för PLC-programmering som stöder *IEC 61131-3* och är utvecklad av Beckhoff Automation. Det är en utvecklingsmiljö, där det även går att konfigurera PLC:n via plattformen. Kommunikation mellan enheter sker med EtherCat, som fungerar ungefär som Ethernet. TwinCat står för, The Windows Control and Automation Technology. Utvecklingsmiljön är baserad på Visual Studio och kan även fullständigt integreras med Visual Studio. Tillsammans med Visual Studio kan man i TwinCat3 kommunicera mellan de vanliga programspråken som finns i *IEC 61131-3* med exempelvis projekt skapade med C/C++ eller exempelvis Simulink-modeller. Detta skapar en kraftfull utvecklingsmiljö där man kan utveckla stora delar av projektet på samma plattform.



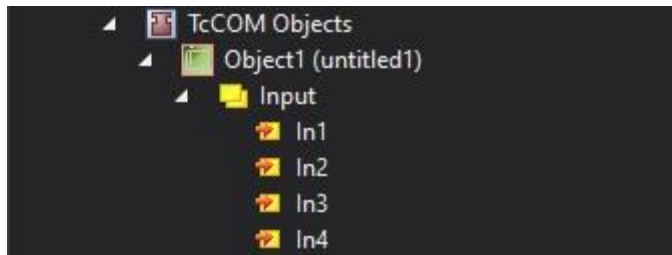
Figur 4. Utvecklingsmiljö för TwinCat3 integrerad med Visual Studio.

## 2.3 ADS -Automation Device Specification

ADS är ett kommunikationsprotokoll utvecklat av Beckhoff som fungerar som transportlagret i OSI-modellen, det vill säga hantering av sändning och mottagning samt ankomstkontroll för data. ADS använder sig av TCP/IP eller UDP/IP protokollen, vilket gör det möjligt att kommunicera med andra enheter eller mjukvara. I fallet med TE1410 används ADS för kommunikation mellan Simulink och TwinCat3-miljön, däremot för TE1400 så är modellerna integrerade i TwinCat3-miljön och det finns därför inget behov av ADS.

## 2.4 TE1400 TwinCat3 Target for Simulink

TE1400 TC3 Target for Simulink är ett tilläggsprogram för TwinCat3. Detta tillägg används för att integrera Simulink-modeller till TwinCat3-miljön genom att skapa TwinCat objekt från den C++ genererade koden som skapats med hjälp av Simulink Coder. TE1400 är inte begränsat till endast simulink utan även modeller skapade i exempelvis SimScape kan integreras till TwinCat3. Dessa objekt kan sedan kopplas ihop med resten av PLC projektet.

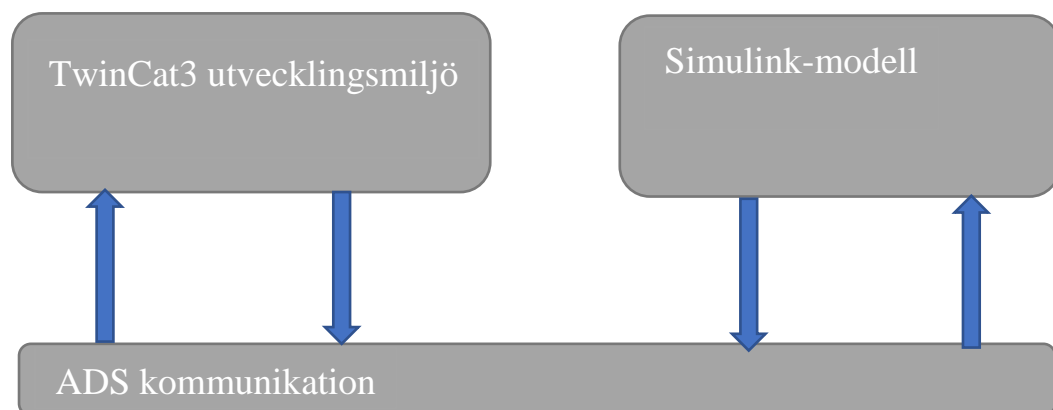


Figur 5. Integrerad modell även kallat TcCom object, med insignaler och utsignaler i solution explorer.

Dessa insignaler och utsignaler kan kopplas ihop mellan olika Simulink-modeller eller till befintliga I/Os i PLC-programmet. Modellerna kan sedan köras i realtid tillsammans med resten av programmet. I den integrerade modellen kan man även utläsa realtidsvärden för olika signaler.

## 2.5 TE1410 TC3, Interface for Matlab/Simulink

TE1410 TC3 Interface for Matlab/Simulink är en mjukvara från Beckhoff Automation och är ett tilläggsprogram för TwinCat3. Med TE1410 kan man kommunicera mellan Matlab/Simulink och TwinCat3 med hjälp av Beckhoffs ADS-protokoll. Det är möjligt att knyta I/Os från en PLC till I/Os i Simulink-modellen samt att köra script från Matlab och ge kommandon till TwinCat3-miljön. Själva modellen körs sedan via Simulink och PLC-programmet i TwinCat3.



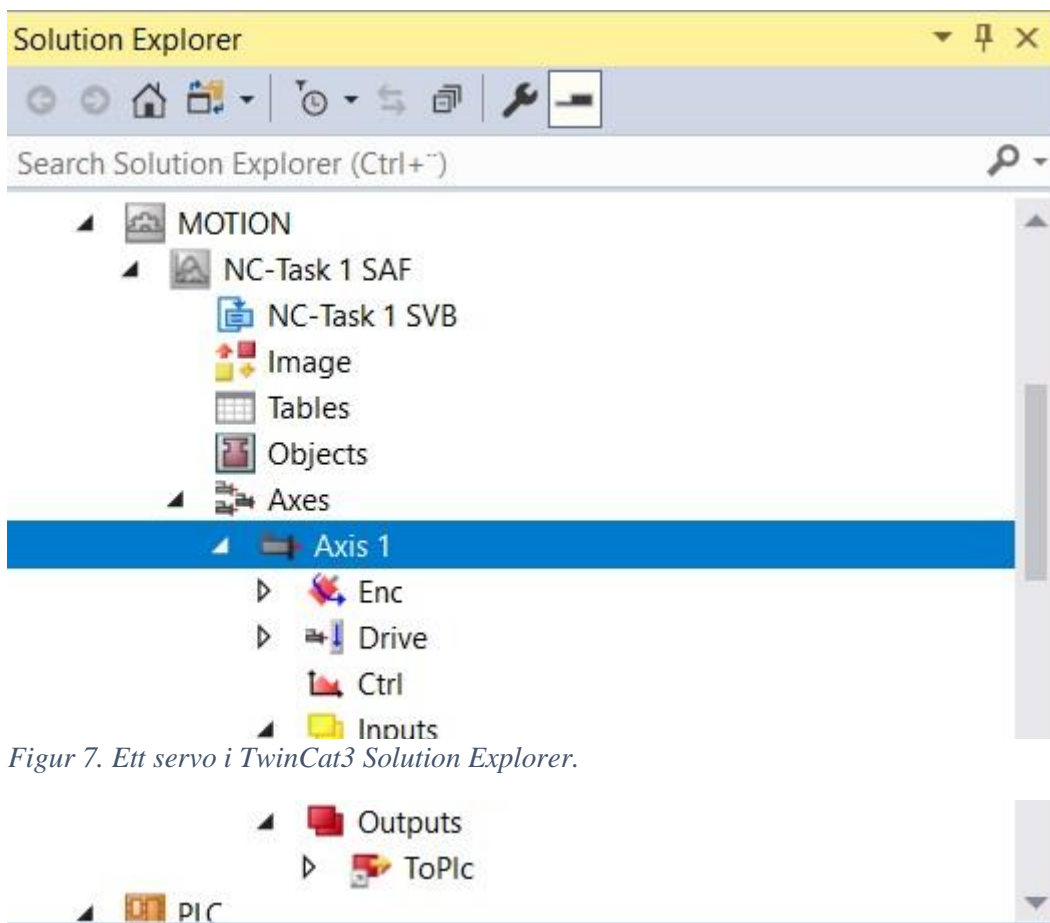
Figur 6. Kommunikation mellan Twincat3 och Simulink via ADS.

## 2.6 Simulink Coder

Simulink Coder är en addOn till Matlab/Simulink som genererar C/C++ kod från exempelvis Simulink-modeller eller Matlab funktioner som sedan kan användas utanför Matlab/Simulink.

## 2.7 TC3 NC PTP

Är en mjukvara från Beckhoff Automation som gör det möjligt att implementera en eller flera servon för att kunna kontrollera rörelser i X, Y eller Z-led. NC står för Numerical Control, även kallat CNC och PTP för Point to Point. Där en axel motsvarar ett servo. Insignaler och utsignaler knyts sedan till det fysiska servots insignaler och utsignaler. Men här är det även möjligt att knyta dessa I/Os till variabler för att simulera servot. Där är det möjligt att antingen köra servot manuellt alternativt att servot styrs via PLC kod.



Figur 7. Ett servo i TwinCat3 Solution Explorer.



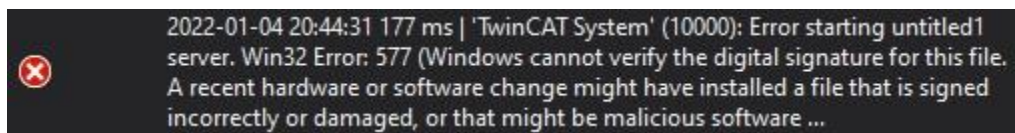
## 2.8 TwinCat PLC Control visualization Editor

TwinCat visualization editor är en inbyggd funktion i TwinCat3 för att skapa enklare visualiseringar och objekt. Dessa objekt kan sedan kopplas till variabler i projektet för att exempelvis representera data eller skriva data till variabler.

## 2.9 Testcertifikat

Testcertifikat fungerar ungefär som ett vanligt certifikat. Det krävs för att kunna exempelvis autentisera ett dokument eller en programvaras ursprung. Dessa certifikat utfärdas av en certifikatutfärdare.

Testcertifikaten däremot kan skapas på Windows PC:n och kan användas för digitala signaturer. Detta krävs för att en Simulink-modell ska kunna köras i TwinCat3. Utan testcertifikatet får man felmeddelandet i figur 8 när man försöker köra programmet.



Figur 8. Felmeddelande i TwinCat3.

## 3 Metod

Arbetet inleddes efter kontakt med företaget med att införskaffa information om främst behovet och problemformuleringen, men även kunskap om hur det kan genomföras. Vilken hårdvara och mjukvara som används var också relevant att införskaffa kunskap om för att kunna utveckla simuleringsmiljön.

### 3.1 Förstudier

I början av projektet inhämtades förståelse om TwinCat3 och kommunikationen med PLC:n med hjälp av en utbildningskurs online som Randek tillhandahöll. I utbildningskursen användes en PLC vars insignaler och utsignaler var kopplade till olika knappar och indikatorer. Under kursen programmerades PLC:n på olika sätt för att förstå hur både PLC:n och Twincat3-miljön fungerade. Utbildningskursen samt information från Beckhoffs hemsida gav en bra grund att arbeta vidare på under projektets gång.

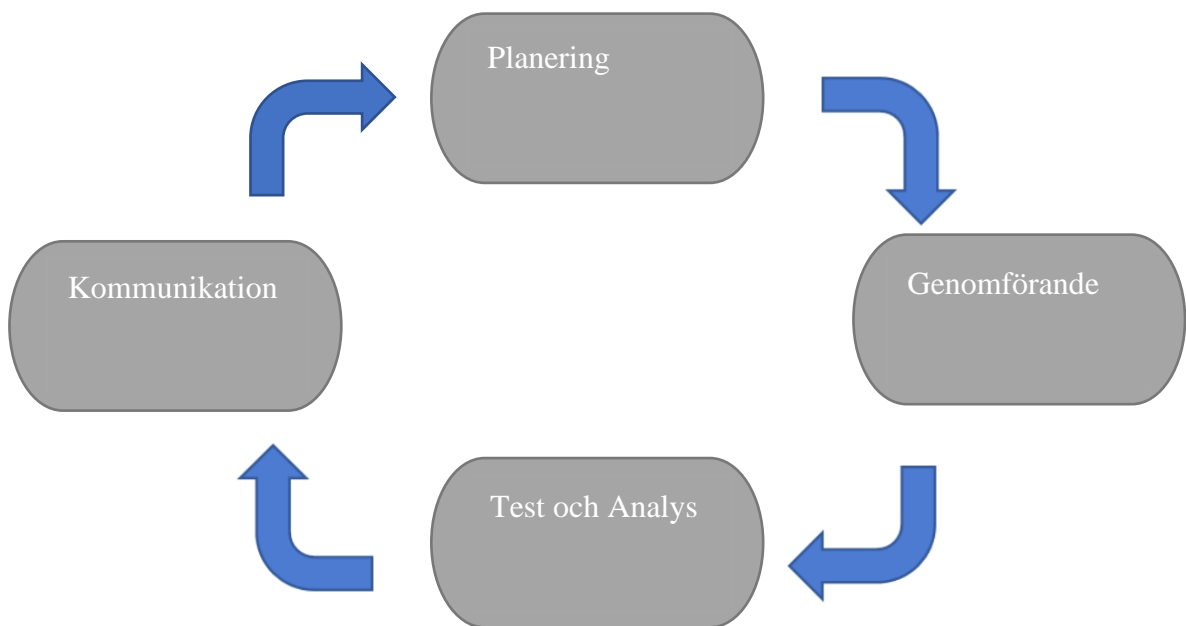
För att kunna integrera modellerna i TwinCat3 behöves även kunskap om TE1400 samt om hur man exporterar modellerna från Matlab/Simulink. Då det inte fanns någon utbildningskurs för TE1400 användes enbart Beckhoffs dokumentation om mjukvaran för att skaffa tillräcklig kunskap.

Detsamma gäller för Matlab/Simulink, kunskapen för att kunna utveckla modellerna och generera koden till TcCom objekten införskaffades via Matlabs hemsida.

Med kunskapen om TwinCat3, Matlab/Simulink och TE1400 fortsatte projektet med att skapa en enklare ”proof of concept” modell i Simulink för att försöka integrera den i TwinCat3-miljön. När en fungerande modell fanns på plats fortsatte arbetet med att utveckla de modeller som användes i simuleringen.

### 3.2 Arbetsmetod

Arbetet utfördes på plats hos företaget vilket underlättade kommunikationen. Planeringen genomfördes framför allt muntligt men även med hjälp av skisser och blockscheman. Detta gjordes tillsammans med handledaren på företaget. Arbetet delades upp i delmål, vilket var naturligt då en del av maskinen refererade till en Simulink-modell. Detta för att undvika en alltför stor och svårtolkad modell. Genomförandet gick till enligt figur 9 och upprepades efter varje färdig modell.



Figur 9. Arbetsmetod

### 3.3 Inhämtning av information

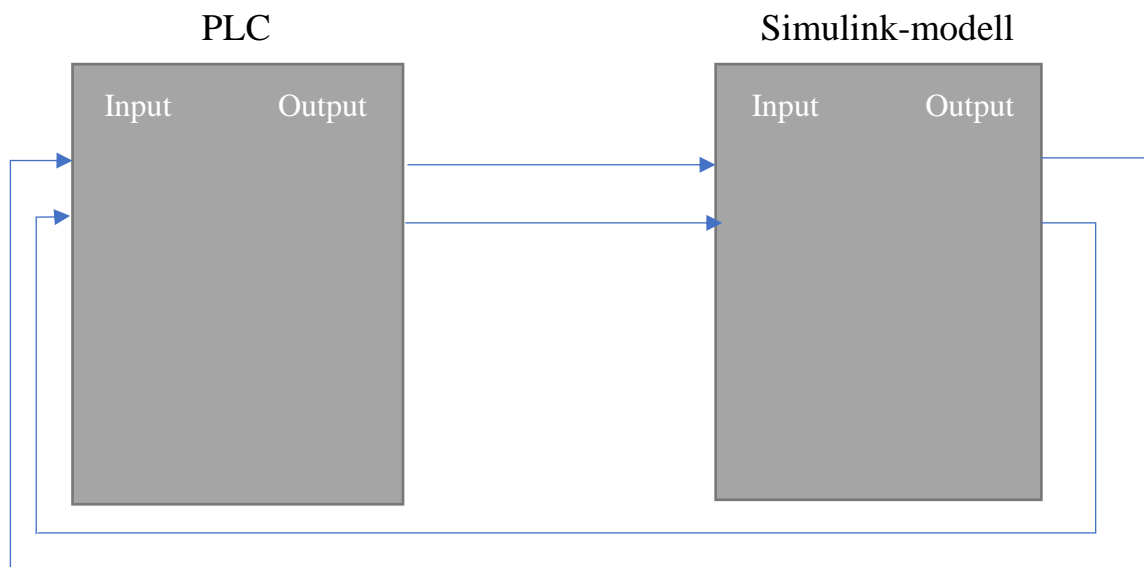
Utöver den utbildningskurs som genomfördes i början av projektet inhämtades även information löpande under projektets gång. Detta var framför allt ifrån Beckhoffs respektive Matlabs hemsidor. Då projektet gjordes på plats hos företaget så inhämtades även information och kunskap via handledaren på Randek och andra medarbetare på företaget.

### 3.4 Inledning

Inledningen bestod av ett möte med handledaren. Under mötet togs det upp vad som skulle ingå i simuleringsmiljön. Detta gjordes muntligt tillsammans med skisser och blockschema för att enklare få en förståelse för maskinen. Under mötet framgick vilka parametervärden som var relevanta samt önskemål på hur de skulle representeras.

### 3.5 Förståelse för PLC-programmet

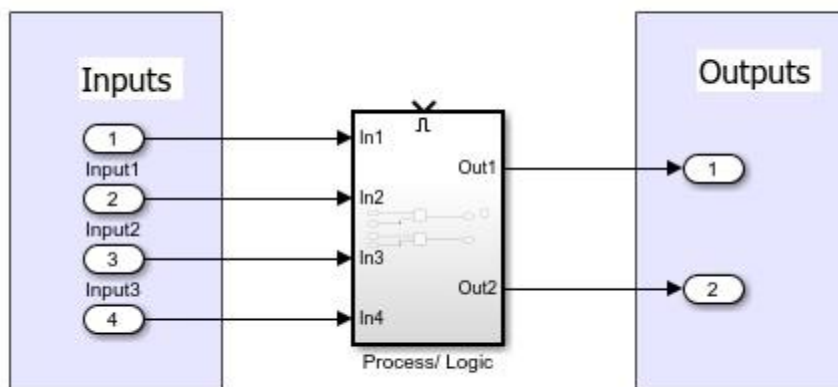
Innan arbetet med att utveckla Simulink-modellen kunde påbörjas behövdes även förståelse för hur företagets PLC kod till maskinen var uppbyggt. Detta gjordes i TwinCat3-miljön utan att vara kopplad till PLC genom att helt enkelt köra programmet. Det är där möjligt att skriva till eller forcera olika variabler för att se hur programmet beter sig. Detta underlättade arbetet med att förstå hur det var uppbyggt och vilka datatyper som skulle skickas samt simuleras tillbaka.



Figur 10. Kommunikation mellan PLC och Simulink-modell.

### 3.6 Utveckling av Simulink-modell

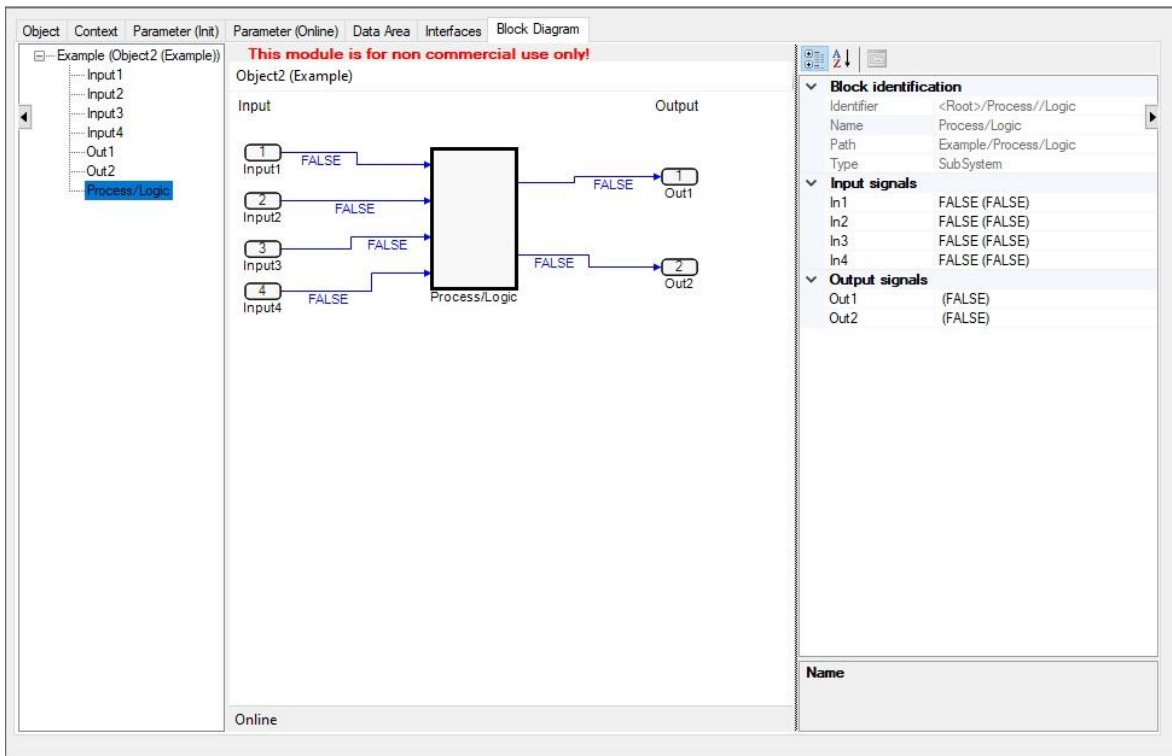
Efter att ha fått en förståelse för hur PLC-programmet fungerade var det möjligt att påbörja arbetet med att utveckla Simulink-modellen. Med information och förståelse om både maskinen och PLC-programmet arbetades en modell fram med de insignaler och datatyper som krävdes samt ”logiken” för att simulera olika utsignaler baserat på vilka insignaler som skickats till Simulink-modellen.



Figur 11. Exempel på en enkel modell i Simulink. Där insignaler kommer från PLC:n och utsignaler skickas tillbaka.

### 3.7 Utveckling av HMI och PLC kod

Under arbetets gång så utvecklades både HMI och PLC kod för att kunna testa modellerna och se så att de simulerade rätt värden baserat på insignalen. Till en början var HMI:et väldigt simpel, men för varje ny del i simuleringen så byggdes även HMI:et större. Till detta skrevs även PLC kod för att kunna testa de olika Simulink-modellerna när de hade importerats till TwinCat3-miljön.



Figur 12. Integrerad modell i TwinCat3.

### 3.8 Test i TwinCat3

När en ny Simulink-modell utvecklats samt att HMI och PLC kod för att köra modellen var klart testades den integrerade modell i TwinCat3-miljön tillsammans med HMI:et. Detta upprepades för varje Simulink-modell, tills alla delar av maskinen var klara.

### 3.9 Källkritik

Det mesta av informationen är hämtad ifrån Beckhoff's egna hemsida samt deras informationssida [infosys.beckhoff.com](http://infosys.beckhoff.com). Det är en sida med instruktioner, manualer och exempel på alla Beckhoffs produkter.

Information har även hämtats från MathWorks hemsida samt deras dokumentation i Matlab/Simulink.

- [1][2][3][4][5] Kan anses vara tillförlitligt då det är Beckhoff som har utvecklat mjukvaran och skrivit manualerna till [Infosys.beckhoff.com](http://Infosys.beckhoff.com).
- [6] Även detta är en instruktion ifrån Beckhoff's informationssida om deras produkter. Därför kan den anses tillförlitlig.
- [7] Matlab/Simulink är en väl känd mjukvara som bland annat används i undervisningen på Lunds universitet. Deras hemsida kan anses vara tillförlitlig.

## 4 Analys

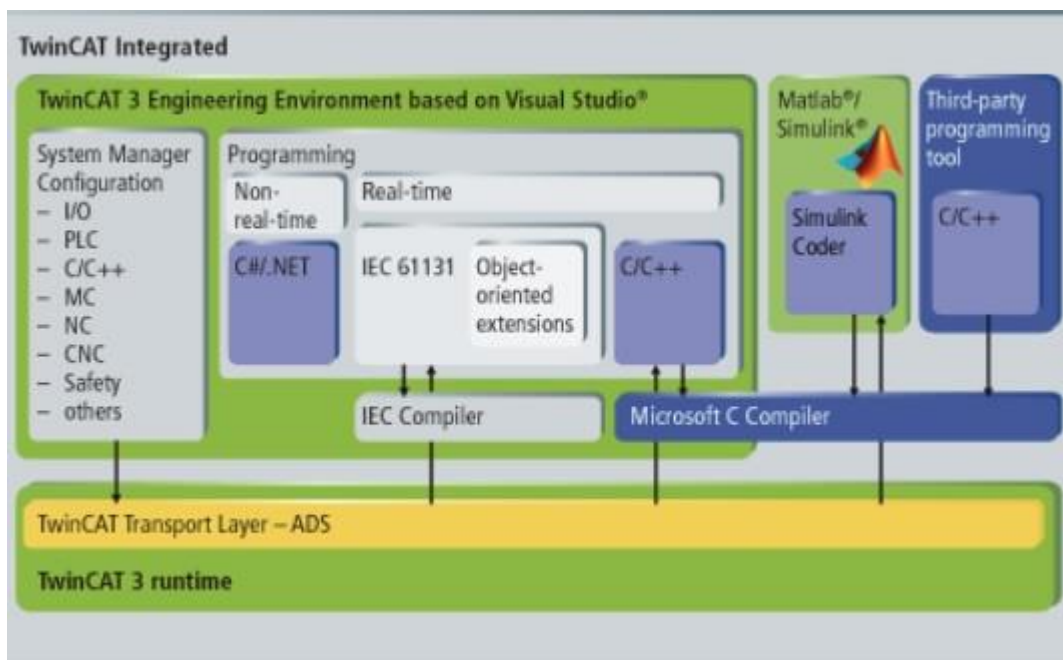
I detta kapitel redogörs vilka val som gjorts under projektets gång samt problem som uppstått på vägen.

### 4.1 Simulink

Valet av att använda Simulink som plattform för att utveckla simuleringsmiljön baserades främst på att Beckhoff har utvecklat mjukvara som gör det möjligt att kunna kommunicera med Matlab/Simulink.

### 4.2 TE1410 TC3, Interface for Matlab/simulink

Till en början gällde det att skaffa en överblick över problemställningen för att kunna inhämta relevant information om hur projektet skulle genomföras. Det fanns exempelvis olika möjligheter att göra simuleringen på. De metoder som kändes mest relevanta var med hjälp av TE1400 TC3, Target for Simulink samt TE1410 TC3, Interface for Matlab/Simulink. De var båda intressanta då de kunde skriva till och läsa data från TwinCat3 i realtid. TE1410 TC3, Interface for Matlab/Simulink körs i Matlab/Simulink i stället för i TwinCat3-miljön. Via TE1410 kan man knyta insignaler och utsignaler mellan TwinCat och Matlab/Simulink-modellen samt att de är möjligt att diagnostisera och manipulera data som vanligt i Matlab/Simulink. Kommunikationen mellan TE1410 och Simulink sker med hjälp av Beckhoffs ADS-protokoll enligt figur 13. TE1410 tillsammans med Matlab/simulink skapar en kraftfull plattform för simuleringar.



Figur 13. Överblick över hur utvecklingsmiljön i TwinCat3 kommunicerar med extern programvara eller andra enheter som också är kopplade till ADS.



### **4.3 TE1400 TC3, Target for Simulink**

TE1400 körs till skillnad från TE1410 i TwinCat3-miljön som en integrerad modell. Det vill säga att modellerna kommunicerar med TwinCat3-miljön utan ADS-protokollet. Dessa modeller skapas först i Simulink och integreras sedan i TwinCat3-miljön som ett C/C++ objekt, se figur 13. Även i detta fall är det möjligt att knyta I/Os mellan andra delar av PLC-programmet till Simulink-modellen, samt att det är möjligt att koppla samman flera olika Simulink-modellers I/Os med varandra. Efter att ha jämfört de båda metoderna så har TE1400 en stor fördel sett till företagets behov. Först och främst så skapas en modell som sedan kan integreras i TwinCat3 utan det behövs någon annan mjukvara.

Behovet av att kunna testa PLC kod i kontorsmiljö underlättas av att det varken behövs Matlab/Simulink licenser eller någon annan mjukvara utöver TwinCat3 för att kunna köra simuleringen. Därav valdes TE1400 för att utveckla simuleringen.

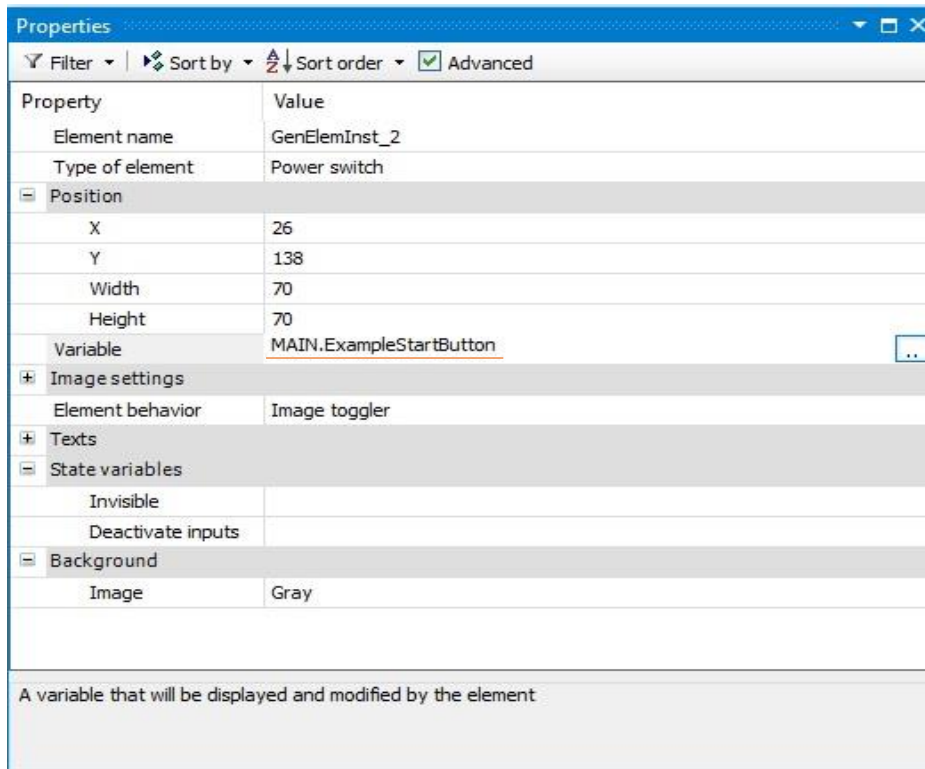
### **4.4 Modeller i TwinCat3**

Innan TcCom objekten kunde kopplas ihop med det befintliga PLC-programmet var modellerna tvungna att testas för att se om de fungerade som det var tänkt. För att testa modellerna skrevs ett enklare PLC-program för att koppla ihop med modellen för att se om den returnerade rätt parametervärden. Detta PLC-programmet skrevs i ST-kod för att knyta I/Os samt att en visualisering utvecklades. Med den integrerade modellen kunde man enkelt se om en insignal eller utsignal hade fel parametervärde. Visualiseringen tillsammans med den integrerade modellen gjorde felsökningen betydligt enklare under utvecklingsarbetet.

## 4.5 HMI

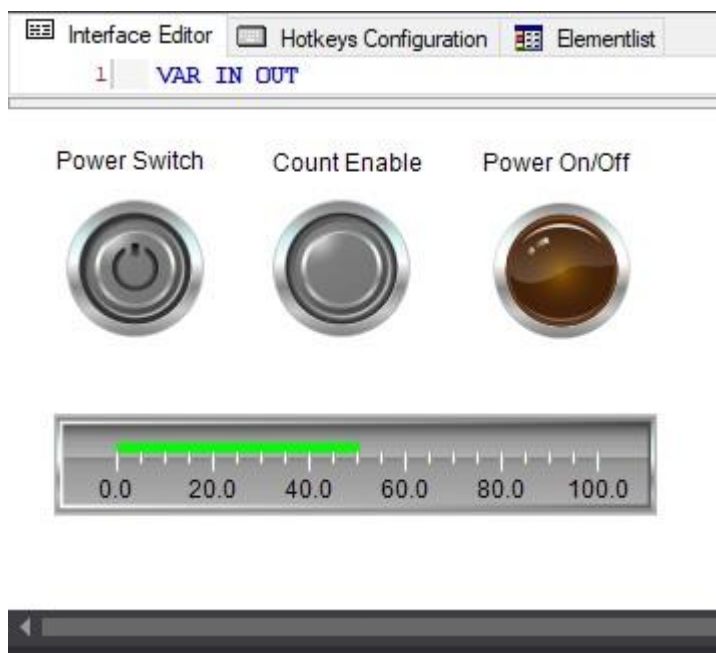
Ett annat avvägande som gjordes var hur simuleringen skulle representeras utöver att manuellt läsa de simulerade insignalerna, skulle det ske grafiskt eller enbart numerisk, samt vilken mjukvara som bäst uppfyller kraven för att kunna representera simuleringen på ett smidigt sätt. För att tydligt visualisera simuleringen användes både grafisk och numerisk representation av data. Detta gjordes med hjälp av TwinCat PLC Control Visualization Editor som är ett inbyggt verktyg i TwinCat3.

Med visualization Editor är det möjligt att representera data på båda sätt samt att det går att koppla variabler som är knutna till I/Os med olika objekt i visualiseringen.



Figur 14. En variabel knuten till ett objekt i Visualization Editor.

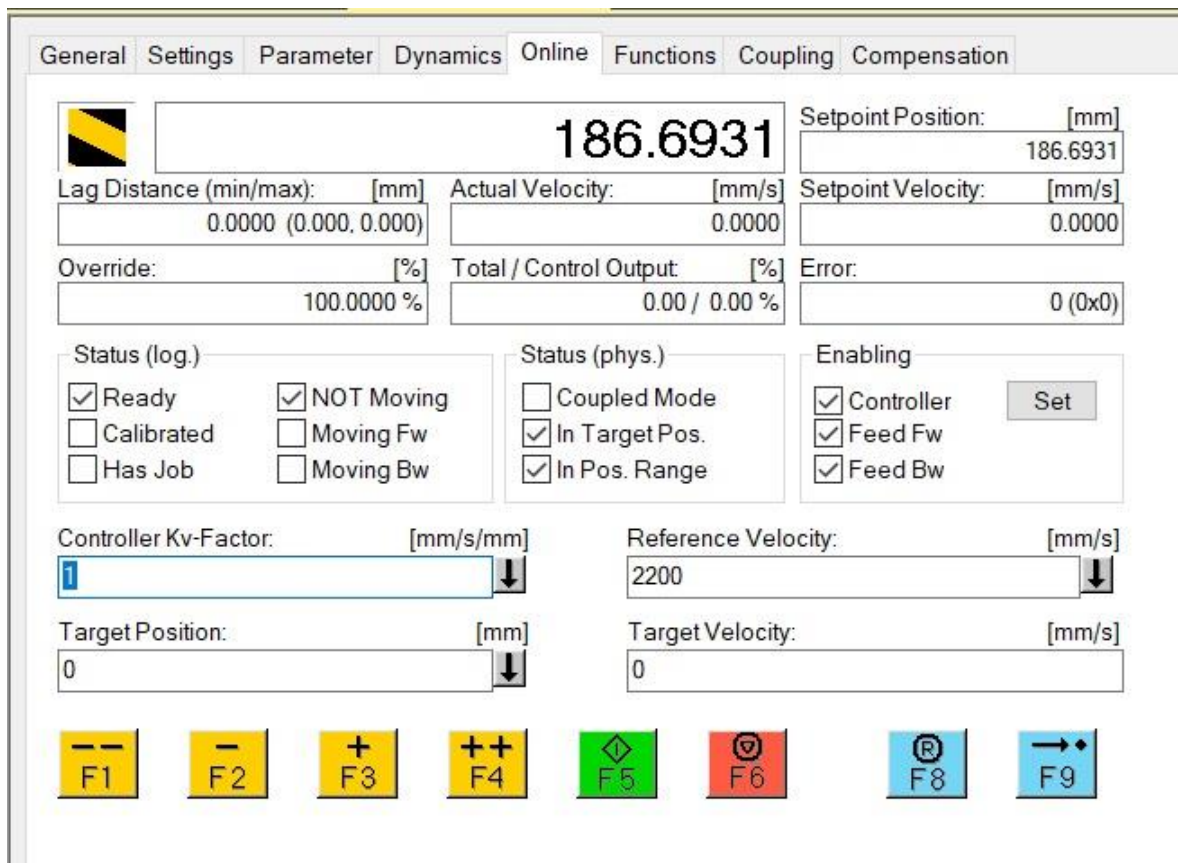
Detta gjorde att simuleringen kunde representeras med exempelvis lampor som motsvarade givarsignaler eller en förloppsindikator för att representera en position.



Figur 15. Exempel på objekt i Visualization Editor.

## 4.6 Simulering av servo

För att simulera rörelsen för ett servo användes dock inte en Simulink-modell. I stället användes TC3 NC PTP som är en mjukvara utvecklad av Beckhoff. Denna metod användes i stället för en Simulink-modell eftersom det var enkelt att ändra mellan att köra den manuellt eller automatiskt via PLC-koden. Möjligheten att kunna köra servot manuellt underlättar vid felsökning men även då enbart delar av koden behöver testas.



Figur 16. Servo i onlineläge. I detta fall styrs servot via F1-F9.

I figur 16 körs servot antingen manuellt med hjälp av F1-F9 knapparna eller via PLC kod. Servot rör sig enbart i antingen X, Y eller Z-led. För rörelse i flera led behövs fler servon.

I figur 17 kan man välja en exakt position som servot ska förflytta sig till samt andra parametrar som exempelvis hastighet. Detta kan vara användbart om det handlar om exakta rörelser som behöver testas.

General	Settings	Parameter	Dynamics	Online	Functions	Coupling	Compensation
		<b>186.6931</b>			Setpoint Position: [mm]		186.6931
<b>Extended Start</b>							
Start Mode:	Absolute				Start		
Target Position:	0	[mm]			Stop		
Target Velocity:	0	[mm/s]					
<input type="checkbox"/> Acceleration:	0	[mm/s <sup>2</sup> ]					
<input type="checkbox"/> Deceleration:	0	[mm/s <sup>2</sup> ]			Last Time:	[s]	
<input type="checkbox"/> Jerk:	0	[mm/s <sup>3</sup> ]					0.86400
<b>Raw Drive Output</b>							
Output Mode:	Percent				Start		
Output Value:	0	[%]			Stop		
<b>Set Actual Position</b>							
Absolute		0			Set		
<b>Set Target Position</b>							
Absolute		0			Set		

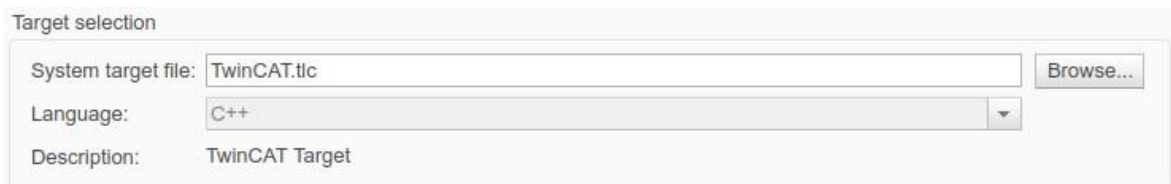
Figur 17. Servo i onlineläge. Servot styrs manuellt.

Den här lösningen var mer flexibel och noggrann därför valdes den i stället för att använda en Simulink-modell.

## 4.7 Testcertifikat

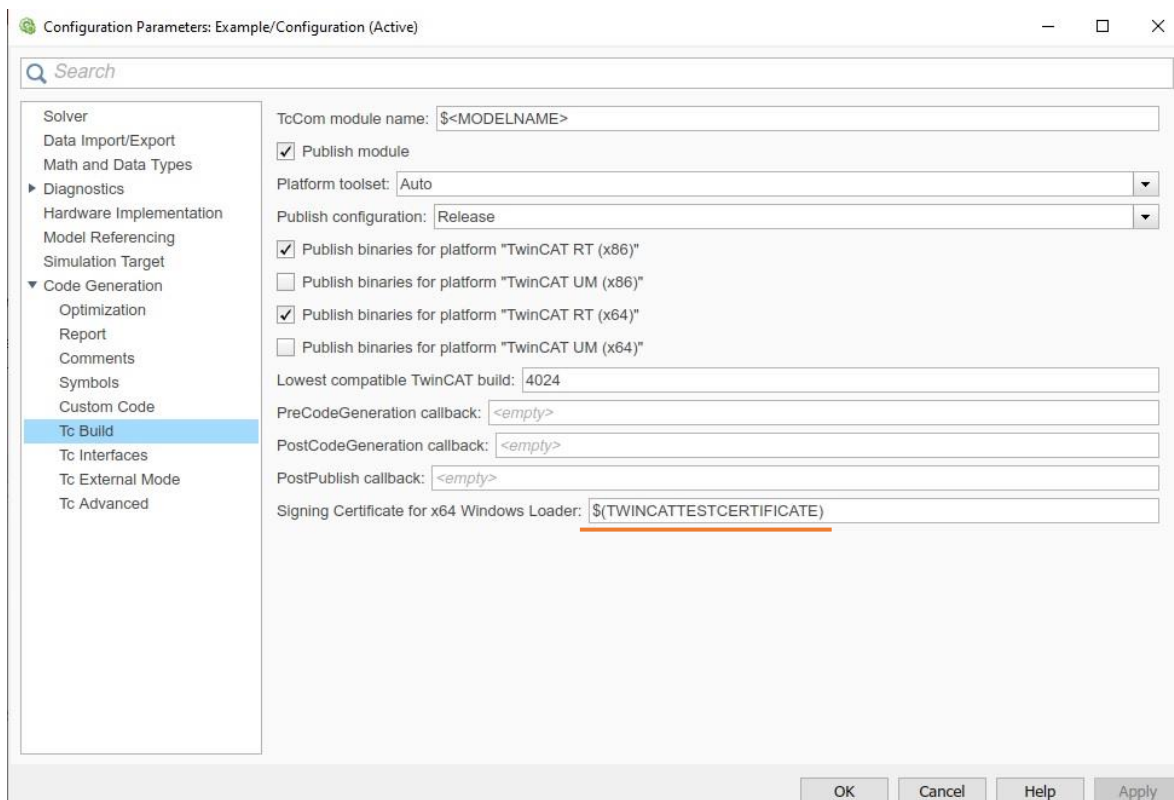
Den mjukvara som krävdes för att kunna utveckla och integrera modellerna var Visual Studio, TwinCat3, Matlab/Simulink, TE1400 Target for Simulink samt om det körs på Windows 64-bitars operativsystem så krävs ett testcertifikat för att kunna importera och köra Simulink-modellerna i TwinCat3.

När C++ koden ska genereras med hjälp av Simulink Coder väljs först vilket system man riktar sig mot, i detta fall i figur 18 är det TwinCat.



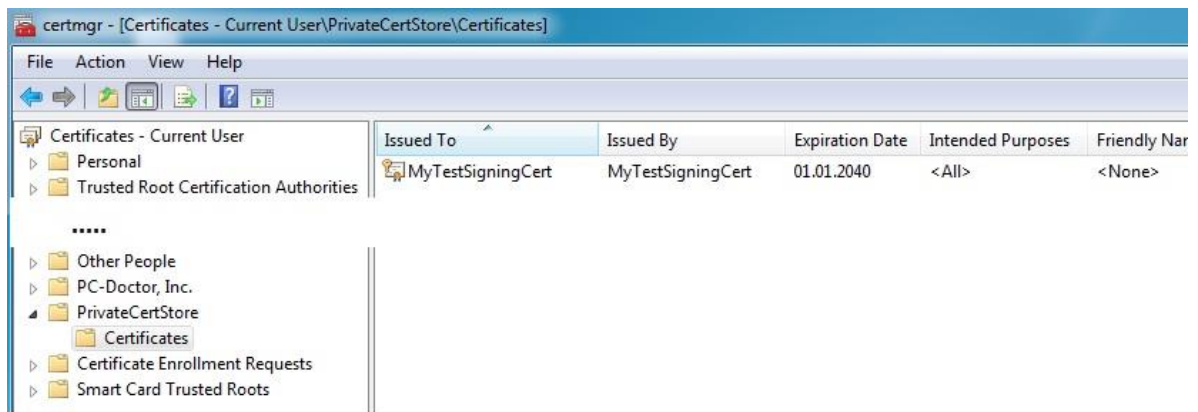
Figur 18. TwinCat valt som System.

Det certifikat som ska användas för att signera den genererade koden och Simulink-modellen väljs också här. Default för TwinCAT Target går att utläsa i figur 19, och är certifikatet "TWINCATTESTCERTIFICATE".



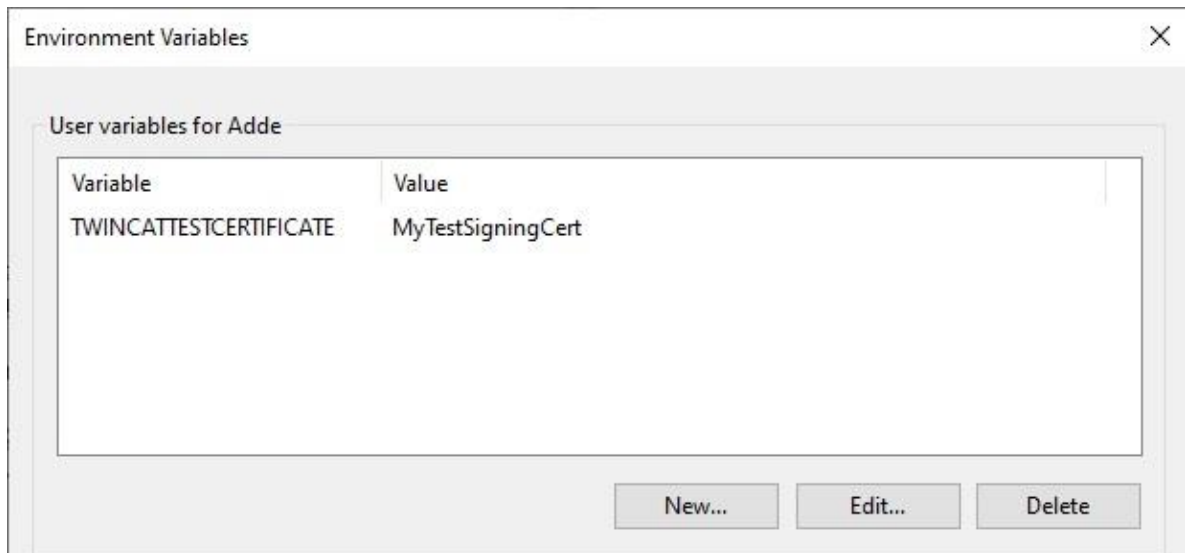
Figur 19. Default för TwinCat som target system.

Certifikatet skapades för att sedan kunna köra modellerna i TwinCat3.



Figur 20. Testcertifikat. Detta kopplas sedan till variabeln TWINCATTESTCERTIFICATE..

MyTestSigningCert i figur 20 kopplas sedan till TWINCATCERTIFICATE som en miljövariabel i Windows enligt figur 21.



Figur 21. Certifikat för att signera Simulink-modellerna i TwinCat3.

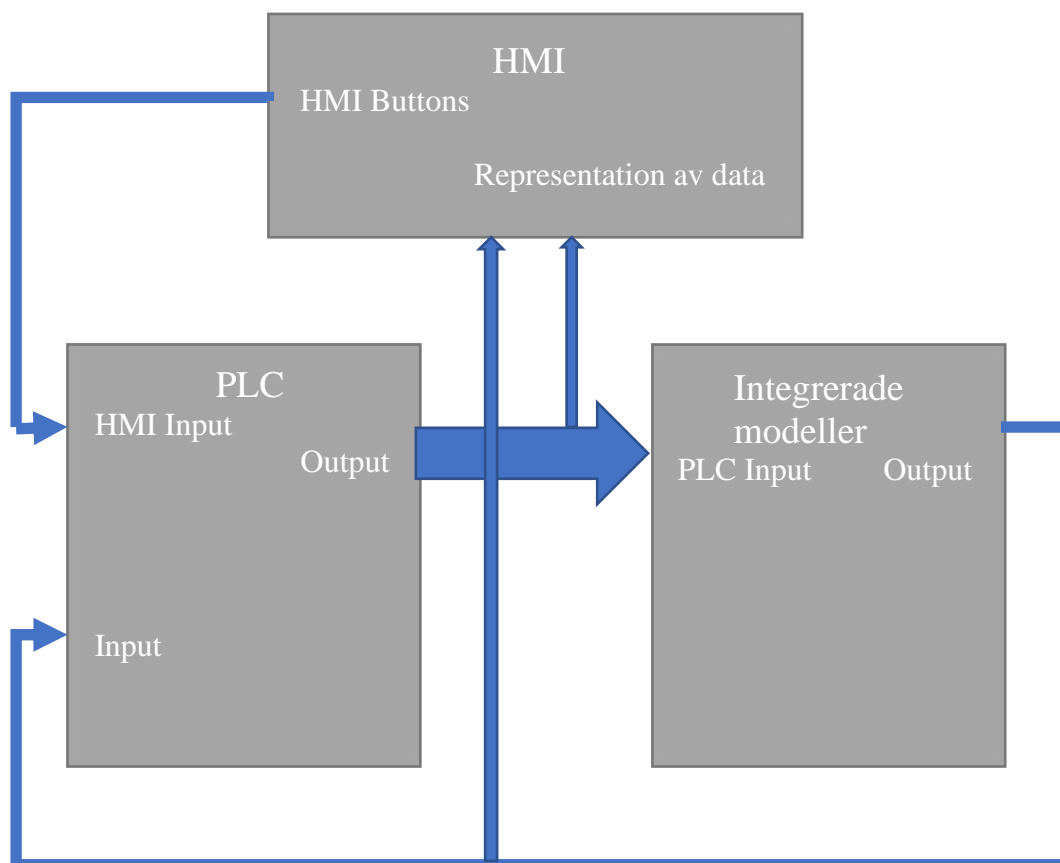
## 5 Resultat

Detta avsnitt avser att presentera resultatet av projektet. Med delar av de modeller som utvecklas samt HMI:et. Då arbetet är gjort baserat på företagets mjukvara utelämnas vissa delar på grund av sekretess.

Arbetet resulterade i en simulering utav två av företagets maskiner. Den första maskinen är en arbetsbänk som kan förflytta väggelement i olika storlekar via ett transportband. Vid bänken utförs arbete av andra maskiner som exempelvis att spika, såga eller limma. Den andra maskinen som simulerats under arbetet är ett exempel på en sådan, vilket är en spikportal som rör sig över arbetsbänken. Portalen förflyttas med hjälp av olika servon till rätt position.

### 5.1 Simulering

Simuleringen består av de olika integrerade Simulink-modellerna samt ett HMI för att styra och representera data. Till detta finns även en global variabel lista för att enklare kunna koppla samman de olika delarna till PLC:n.



Figur 22. Överblick av simuleringsmodell av verklig maskin. Där ett PLC-program testas i den simulerade verkliga maskinen.



Simulink-modellerna ersätter den fysiska maskinen, där utsignaler från PLC-programmet skickas som insignaler till olika Simulink-modeller. Beroende på vilken insignaler som skickas kommer modellen att simulera tillbaka olika värden till PLC:n.

## 5.2 HMI

Simuleringsmiljön styrs med hjälp av ett HMI. Simuleringen kan antingen köras manuellt där man exempelvis kan flytta väggelementet framåt eller bakåt samt justera bredden på maskinen beroende på väggelementets storlek. Det är även möjligt att köra servot manuellt, se figur 16. Positionen på elementet skickas till både HMI och PLC, detta gör att man med hjälp av HMI:et direkt kan utläsa elementets position samtidigt som man förflyttar det. Det finns även olika indikatorer i form av lampor, dessa motsvarar olika givare. De indikerar exempelvis att ett element har nått en specifik position.



Figur 23. HMI för arbetsbänken.



Figur 24. HMI för spikportalen.

## 5.3 Simulink-modeller

Här redogörs för några av de större Simulink-modellerna som använts i simuleringen.

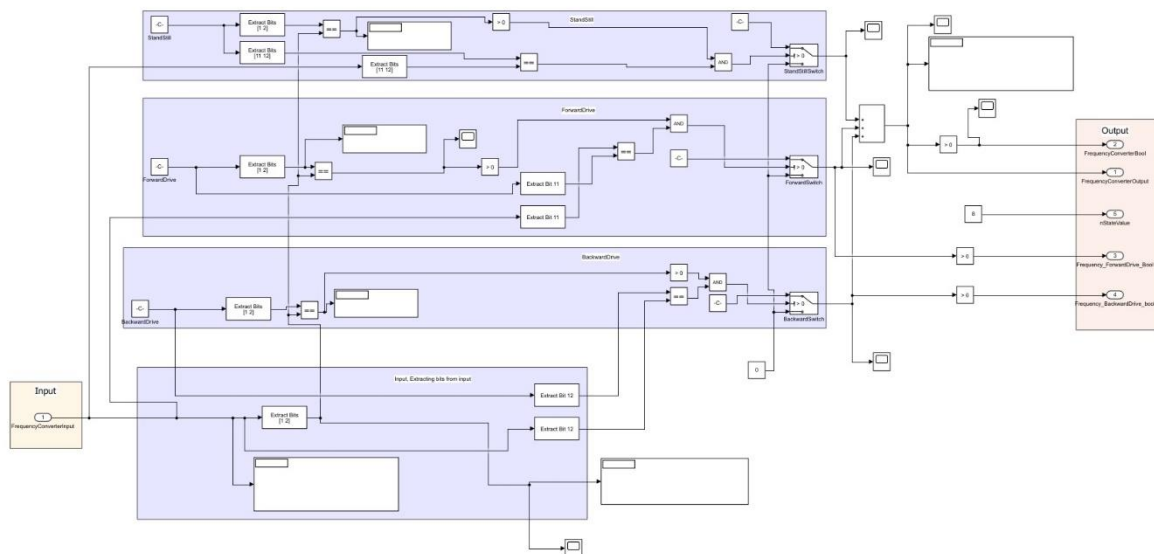
### 5.3.1 Frekvensomriktare och transportband

#### 5.3.1.1 Modell för frekvensomriktare

Elmotorn som kör transportbandet styrs med hjälp av en frekvensomriktare. Den här modellen ska simulera frekvensomriktarens ut signaler tillbaka till PLC-programmet. I modellen nedan läses insignaler in från PLC-programmet. Värdet som skickas till modellen jämförs sedan med värden från tidigare kända värden för olika driftläge. Beroende på insignalen kommer modellens utsignal att variera.

Utsignalerna från modellen består av olika booleska variabler samt två uint, där Forward och Backward drive är kopplade till modellen i figur 26 medans FrequencyConverterOutput är kopplad tillbaka till PLC:n.

FrequencyConverterOutput är värdet som den verkliga frekvensomriktaren hade skickat tillbaka till PLC:n

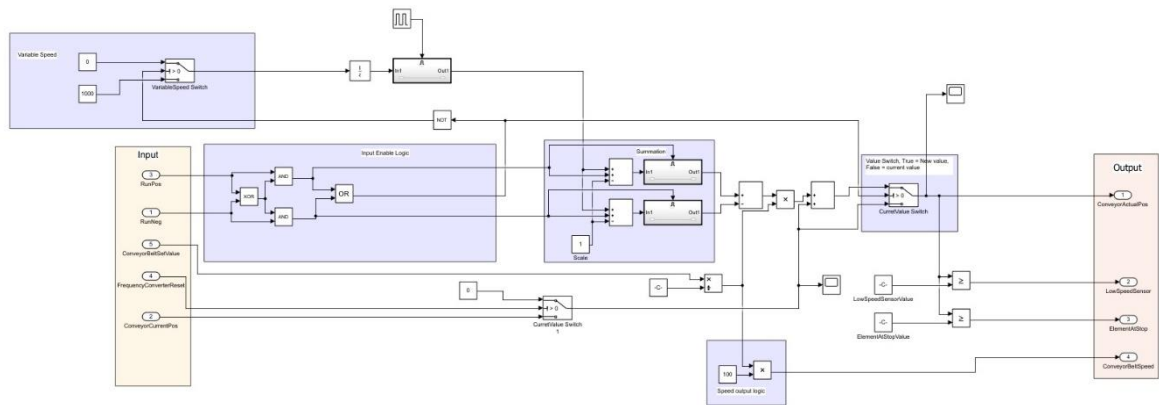


Figur 25. Modell för att simulera frekvensomriktaren.

### 5.3.1.2 Modell för transportbandet

Modellen i figur 26 simulerar transportbandet. Den får insignaler från både PLC-koden och från modellen i figur 25. Den insignalen som kommer från PLC:n styr hastigheten på transportbandet samt att det är möjligt att återställa transportbandet till startpositionen. Insignalen från figur 25 bestämmer om transportbandet ska vara stillastående, röra sig framåt eller bakåt.

Utsignalerna består av två booleska värden som skickas till PLC:n samt två utsignaler som skickas till HMI:et för att representera positionen för elementet. De två booleska värdena representerar två givare som återkopplas till PLC:n. De två givarnas värde skickas sedan ifrån PLC:n till HMI:et för att kunna indikera om elementet nått en viss givare

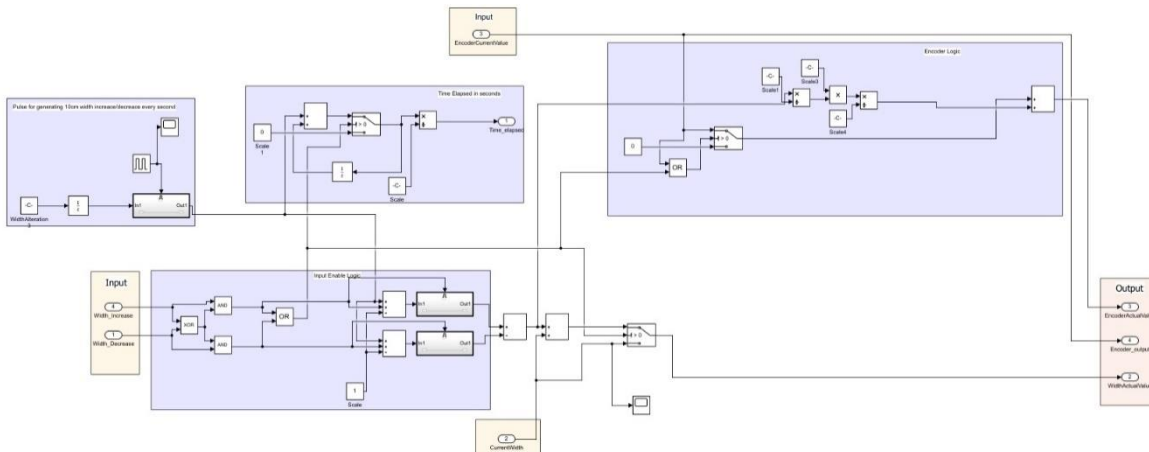


Figur 26. Modell för att simulera transportbandet.

### 5.3.2 Modell för Breddomställning

För att kunna hantera olika storlekar på elementen behöver även bredden kunna justeras. För att kunna utläsa bredden används en encoder.

Breddomställningen sker alltid med samma hastighet. Omställningen sker antingen manuellt via HMI:et eller automatiskt via PLC-koden. Encoder värdet simuleras och skickas tillbaka till PLC:n. En av utsignalerna ifrån figur



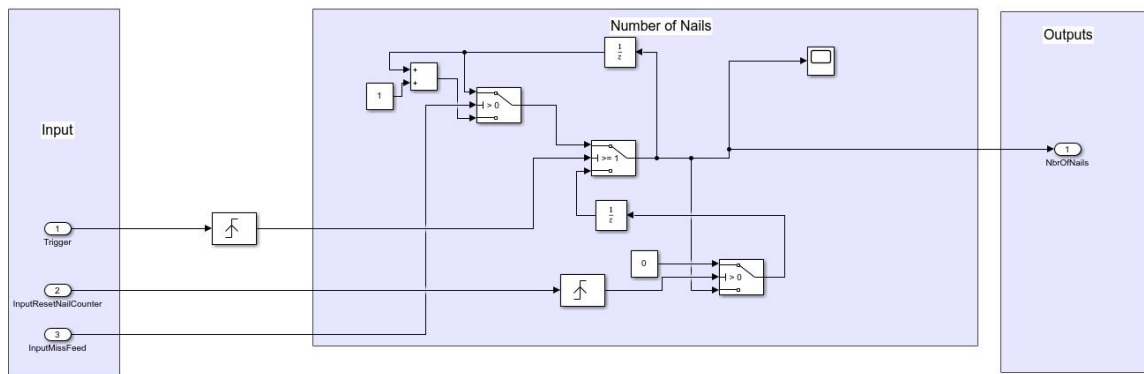
Figur 27. Modell för att simulera breddomställningen.

27 skickas även till en mindre modell som gör en jämförelse mellan tidigare bestämda största och minsta värden och det faktiska värdet på breddomställningen.

Utsignalen från den modellen skickas som simulerade givarsignaler tillbaka till PLC:n. Detta indikeras i HMI:et i form av lampor som tänds om breddomställningen når ett max eller minsta värde samt att den aktuella bredden kan utläsas, se figur 23.

### 5.3.3 Spikportal

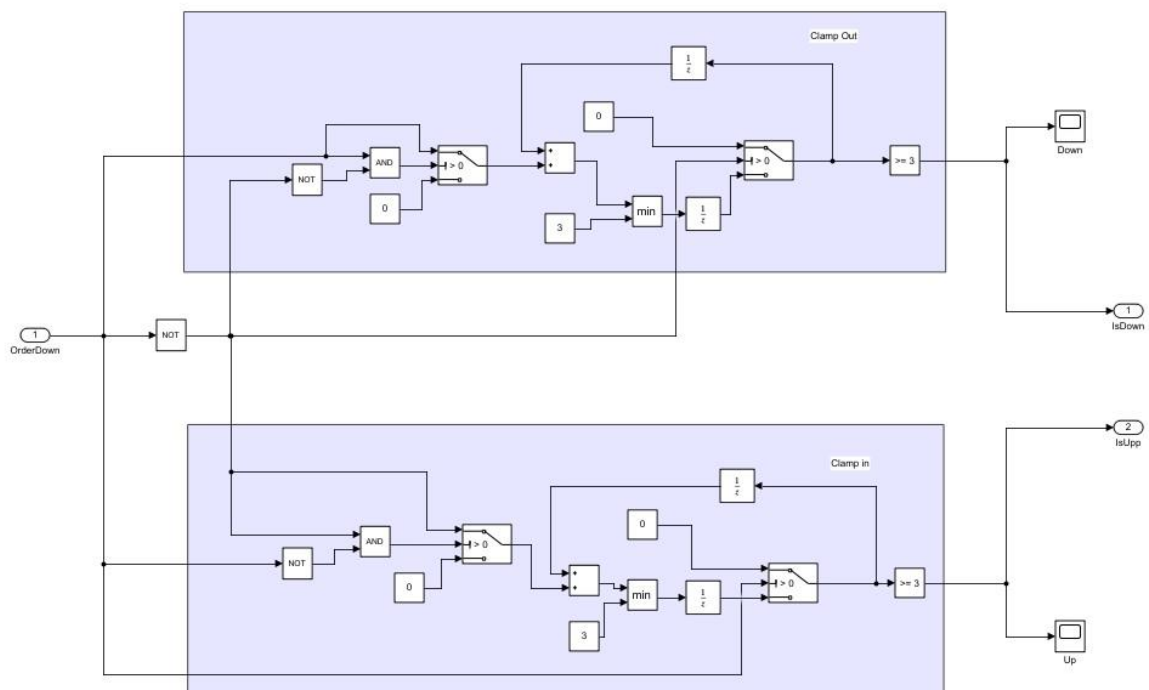
Portalen flyttar sig över arbetsbänken och utför arbete, den här förflyttningen sker med hjälp av servon som beskrivs i avsnitt 4,7. Spikpositioner skickas till servot, som sedan förflyttar sig till en angiven position. Väl där skickas en triggersignal från PLC-koden till modellen som registrerar och räknar de olika spikpositionerna.



Figur 28. Modell för att simulera spikpistol.

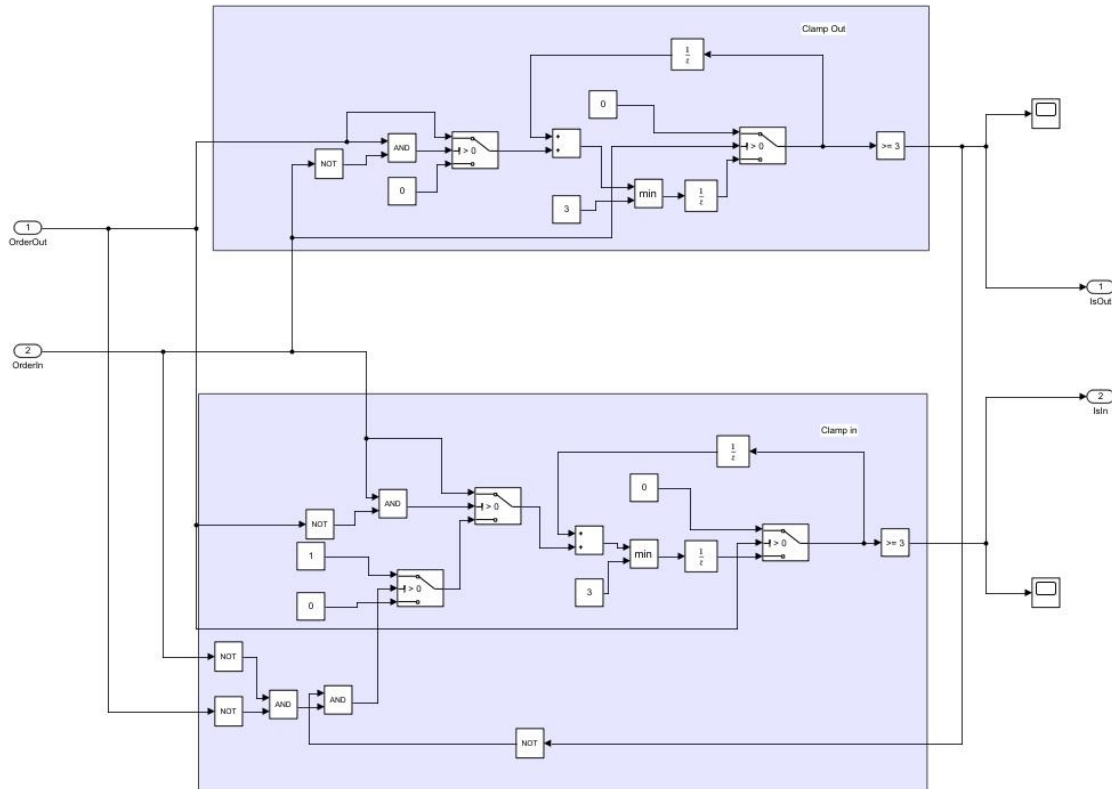
Innan spikpistolen skjuter en spik måste dock spikpistolen vara i rätt position, det vill säga att spikpistolen måste ha rätt vinkel samt att den är tryckt mot materialet. Detta simuleras med hjälp av modellen i figur 29.

En insignal skickad från PLC:n beordrar verktyget ner som efter en fördröjning sedan håller sig i den positionen tills den blir beordrad upp. Utsignalerna från modellen är helt enkelt vilket läge verktyget befinner sig i.



Figur 29. Simulering av spikpistolens position.

En liknande modell används för att justera vinkeln. Även den här modellen får insignaler ifrån PLC-koden och efter en fördröjning simulerar tillbaka vilken position den befinner sig i.



Figur 30. Modell för att simulera justeringen av spikpistolens vinkel.

## 6 Slutsats

En Simulering för två av Randeks maskiner har utvecklats samt ett HMI som knyter ihop simuleringen med PLC-programmet. Simuleringen går att köra antingen manuellt med hjälp av HMI:et eller att den körs automatiskt via PLC-programmet.

Simuleringsmiljön kommer att användas för att spara tid då företaget ska testa PLC kod samt underlätta vid driftsättningar av nya anläggningar och felsökning av befintliga anläggningar.

En av problemformuleringarna i avsnitt 1.4 var vilken programvara som skulle användas och vilka kriterier som ska användas för att avgöra vilken programvara som var mest lämplig. Det går att använda sig utav både TwinCat3 TE1400, Target for Simulink och TwinCat3 TE1410, Interface for Matlab/Simulink tillsammans med Simulink för att skapa en simulering av en maskin. Den stora skillnaden är att med TE1410 körs modellerna i Simulink och kommunicerar med TwinCat3-miljön medan för TE1400 integreras modellerna in i TwinCat3-miljön. Enkelheten i att det räcker med att importera dessa Simulink-modeller till ett existerande PLC-program samt att det inte krävs någon licens för Matlab/Simulink för att köra simuleringen gör dock TwinCat3 TE1400, Target for Simulink till det alternativ som passade företaget bäst. Däremot skulle TE1410 passa bättre i de fall man behöver utveckla en mer komplicerad simulering där man exempelvis återskapar de relevanta fysiska delarna av maskinen i exempelvis Simscape. TE1410 hade varit ett mer kraftfullt verktyg i detta fall eftersom man då har tillgång till diagnostiseringsverktyg och script i Matlab/Simulink.

Möjligheten att utveckla ett HMI i TwinCat3 var är också till stor hjälp när det gäller att representera och tolka data från simuleringen. Det är en stor fördel att kunna samla allt i samma utvecklingsmiljö.

## 7 Framtida utvecklingsmöjligheter

Det finns flera framtida utvecklingsmöjligheter för det här projektet. Först och främst är simuleringen enbart en liten del av en stor produktionslinje. Det finns många andra maskiner som skulle kunna implementeras. Detta skulle kunna göras genom antingen samma metod som projektet genomfördes med eller med hjälp av TwinCat3 TE1410, Interface for Matlab/Simulink.

Utöver det finns det mycket utrymme för att vidareutveckla HMI:et mer. Det hade varit spännande att exempelvis använda sig av en mer grafisk representation för de väggelement som ”tillverkades”.

Det gjordes ett försök att skapa en bild av simuleringens olika spikpositioner, men det visade sig vara för tidskrävande för att hinna med innan projektets slut.



## 8 Terminologi

PLC	Programmable Logic Controller
HMI	Human Machine Interface
I/O	Input/Output
TwinCat3	Är en mjukvara från Beckhoff
TE1400	Tillägg till TwinCat3
TE1410	Tillägg till TwinCat3

## 9 Källförteckning

- [1] TwinCat3 XAE [Hämtad: 2022-01-03]  
URL:<https://infosys.beckhoff.com/english.php?content=../content/1033/tcinfosys3/108086395384195723.html&id=>
  
- [2] TwinCat3 TE1400, Target for Simulink [Hämtad: 2022-01-03]  
URL:[https://infosys.beckhoff.com/english.php?content=../content/1033/te1400\\_tc3\\_target\\_matlab/4211215243.html&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/te1400_tc3_target_matlab/4211215243.html&id=)
  
- [3] TwinCat3 TE1410, Interface for Matlab/Simulink [Hämtad: 2022-01-03]  
URL:[https://infosys.beckhoff.com/english.php?content=../content/1033/te1410\\_tc3\\_interface\\_matlab/3298344843.html&id=1673005873580144126](https://infosys.beckhoff.com/english.php?content=../content/1033/te1410_tc3_interface_matlab/3298344843.html&id=1673005873580144126)
  
- [4] TwinCat3 TC3 NC PTP [Hämtad: 2022-01-03]  
URL:[https://infosys.beckhoff.com/english.php?content=../content/1033/tf50x0\\_tc3\\_nc\\_ptp/3406538635.html&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tf50x0_tc3_nc_ptp/3406538635.html&id=)
  
- [5] TwinCat3 Visualization Editor [Hämtad: 2022-01-03]  
URL:[https://infosys.beckhoff.com/english.php?content=../content/1033/tc3\\_plc\\_intro/3605416331.html&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/3605416331.html&id=)
  
- [6] Driver Singning [Hämtad: 2022-01-03]  
URL:[https://infosys.beckhoff.com/english.php?content=../content/1033/tc3\\_c/6778768523.html&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_c/6778768523.html&id=)
  
- [7] MathWorks support [Hämtad: 2022-01-03]  
URL:<https://ch.mathworks.com/help/>