

Demoapplikation av EcoStruxure Automation Expert



Johannes Karlsson

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

Demoapplikation av EcoStruxure Automation Expert



LUNDS
UNIVERSITET

Lunds Tekniska Högskola

LTH Ingenjörshögskolan vid Campus Helsingborg
Avdelning för Industriell elektronik och automation

Examensarbete:
Johannes Karlsson

© Copyright Johannes Karlsson

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

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

Tryckt i Sverige
Media-Tryck
Biblioteksdirektionen
Lunds universitet
Lund 2022

Sammanfattning

Schneider Electric är ett företag som utvecklar produkter allt från hemelektronik så som lampknappar och vägguttag, till industrielektronik i form av PLC, frekvensomriktare, utvecklingsmjukvara och mycket mer. EcoStruxure Automation Expert, EAE, är en av Schneider Electric nya utvecklingsmjukvaror och är vad det här examensarbete handlar om, implementering av demoapplikation för EAE.

EAE är en mjukvara som bygger på standarden IEC 61499 och är ett verktyg som möjliggör sömlös integration och portabilitet. Det frigör mjukvaran från hårdvaran och möjliggör ett mer flexibelt sätt att arbeta. Detta då Automation Expert är en miljö där den hårdvara som stödjer firmwären Universal Automation samt Linux och Windows kan användas.

Det önskemål som Schneider Electric hade var att en demoapplikation skulle programmeras och designas i deras nya EcoStruxure Automation Expert. En demoapplikation som skulle belysa just mjukvarans öppenhet mot all typ av hårdvara och inte bara hårdvara från Schneider Electric själv. Demoapplikationen skulle presenteras i mjukvarans egna Human Machine Interface (HMI), där kommunikationen mellan de olika komponenterna skulle åskådliggöras enkelt och pedagogiskt. Installation av mjukvara samt spänningssättning och konfiguration av all hårdvara skulle utföras.

Nyckelord: EAE, PLC, HMI, IEC 61499

Abstract

Schneider Electric is a company that develops products ranging from home electronics such as lamp buttons and wall sockets, to industrial electronics in the form of PLCs, frequency converters, development software and much more.

EcoStruxure Automation Expert, EAE, is one of Schneider Electric's new development software and is what this thesis is about, implementing a demo application for EAE.

EAE is a software based on the standard IEC 61499 and is a tool that enables seamless integration and portability. It frees the software from the hardware and enables a more flexible way of working as EcoStruxure is an environment that supports all hardware.

The wish of Schneider Electric was that a demo application be programmed and designed in their new EcoStruxure Automation Expert. A demo application that would radiate the openness of the software to all types of hardware and not just hardware from Schneider Electric itself. The application would be presented in the software's own Human Machine Interface (HMI), where the communication between the various components would be illustrated simply and pedagogically.

Software installation as well as energization and configuration of all hardware would be performed.

Keywords: EAE, PLC, HMI, IEC 61499

Förord

Det här examensarbetet som jag utfört på Schneider Electric under sen vår och tidig sommar år 2022 har varit den sista del av min utbildning på Lunds tekniska högskola på Campus Helsingborg.

Det har gett mig en väldigt god insikt av hur en bra arbetsplats på ett stort företag kan fungera.

Jag har blivit otroligt bra omhändertagen av Schneider Electric och är väldigt tacksam för att ha fått vara en del av ett helt nytt och utforskat område inom deras nyutvecklade mjukvara.

Ett speciellt stort tack till Josef Christoffersson som tog in mig och gav mig chansen att skriva mitt examensarbete hos dem. Han har även sett till att jag fått den hårdvara och mjukvara jag behövt för att kunna utföra examensarbetet.

Ett annat speciellt stort tack till Martin Arildsson-Wahlström som varit min handledare under min period på Schneider Electric. Som alltid varit kontaktbar och hjälpt mig med allt mellan himmel och jord vad som gäller projektet.

Vill även passa på att tacka Lars Celano som tillsammans med Josef Christoffersson gav mig chansen att få skriva mitt examensarbete hos Schneider Electric.

Vill också tacka Thomas Gillblad som ställde upp att vara min handledare även under semestertid. Samma gäller min examinator Jörgen Svensson på LTH.

Innehållsförteckning

1 Inledning	1
1.1 Bakgrund	1
1.2 Syfte	2
1.3 Målformulering	2
1.4 Problemformulering	2
1.5 Motivering av examensarbete	2
1.6 Avgränsningar	3
2 Teknisk bakgrund	3
2.1 IEC 61131-3	3
2.2 IEC 61499	4
2.2.1 Basic Function Block (BFB).....	4
2.2.2 Composite Function Block (CFB)	5
2.2.3 Service Function Block (SFB)	5
2.3 Mjukvara	5
2.3.1 EcoStruxure Automation Expert (EAE).....	5
2.3.2 SoMove.....	6
2.3.3 Automation Device Maintenance.....	6
2.3.4 TM3BC IO Configurator	7
2.4 Human Machine Interface (HMI)	7
2.5 Hårdvara	7
2.5.1 IPC - Harmony P6 (Schneider Electric, HMIP67GA0N30BN7N00).....	7
2.5.2 IPC - Simatic (Siemens, IPC 6AG4141-5AB17-0FA0).....	7
2.5.3 Ethernet TCP/IP managed switch - MCSESM163F23F0 (Schneider Electric).....	7
2.5.4 Nätaggregat - ABL7RE2405 (Schneider Electric).....	8
2.5.5 PLC - TM251MDESE (Schneider Electric)	8
2.5.6 Fältbuskopplare - Modbus TCP 750-362 Ethernet (WAGO)8	
2.5.7 Bus Coupler – TM3BCEIP (Schneider Electric).....	9
2.5.8 Frekvensomriktare - ATV340U07N4E (Schneider Electric) ..	9
2.5.9 Energimätare – METSEPM8240 (Schneider Electric)	9
2.5.10 WIFI Accesspunkt – TCSEGWB13FA0 (Schneider Electric)	9
3 Metod	10
3.1 Förberedelse och förundersökning	10
3.2 Installation av hårdvara och mjukvara	12
3.3 Nätverkskonfiguration	14
3.3.1 PLC - M251	14
3.3.2 Buskopplare - TM3BCEIP	14
3.3.3 Fältbuskopplare - WAGO 750-362	14
3.3.4 WIFI Access Point - Wifer	14

3.3.5	Energimätare - PM8240	14
3.3.6	Frekvensomriktare - ATV340E	14
3.4	Utbildning	15
3.4.1	Första dagen	15
3.4.2	Andra dagen.....	15
3.4.3	Tredje dagen	15
3.4.4	Fjärde dagen	15
3.5	Programmering	16
3.5.1	Idéutveckling	16
3.5.2	Lampsekvens	16
3.5.3	Energimätning	17
3.5.4	Motorstyrning	17
3.5.5	HMI	17
3.6	Källkritik.....	18
4	Analys	19
4.1	Hårdvara	19
4.2	Mjukvara	20
4.3	HMI	20
4.4	Problem	21
4.4.1	Uppkoppling till internet	21
4.4.2	Kommunikation mellan komponenterna.....	21
4.4.3	Fel sort av frekvensomriktare	21
4.4.4	Lång leveranstid	21
4.4.5	Konfigurationshjälp av ICE-Block	21
5	Resultat	22
5.1	Hårdvara	22
5.2	Lampsekvens - Lights_M251_WAGO_BC	24
5.2.1	CAT - Start_Stop_Btn.....	25
5.2.2	CAT - M251_lights03.....	26
5.2.3	CAT - WAGO_lights3	30
5.2.4	CAT - BusCoupler_lights3.....	30
5.3	Motorstyrning med frekvensomriktare - ATV340_Controll ..	32
5.4	Datahämtning av energimätare - PM8240	36
5.5	Grafiska gränssnittet - HMI	37
5.5.1	Canvas - Main Menu	37
5.5.2	Canvas - Siemens IPC	38
5.5.3	Canvas - M251 dPAC.....	39
5.5.4	Canvas - Application	41
5.5.5	Canvas - Connection overlay.....	42
6	Slutsats	43
6.1	Reflektion kring etiska aspekter	43
6.2	Framtida utvecklingsmöjligheter	44

7 Terminologi	45
8 Källförteckning	46

1 Inledning

1.1 Bakgrund

Schneider Electric Sverige AB är ett svenskt bolag med uppgift att sälja produkter och tjänster från moderbolaget i Frankrike. Detta är i sin tur ett väletablerat företag inom den europeiska elektronikkoncernen. För allmänheten är de väl kända för elkontakter såsom lampknappar och vägguttag med mera, medan inom industrin är de ett företag som producerar PLC:s, frekvensomriktare och utvecklingsmjukvaror, mer eller mindre de komponenter som behövs inom automationsindustrin.

Mailkonversation mellan student och företag ledde till ett möte där möjligheten gavs om att få utforma en demoapplikation för företagets nya mjukvara EcoStruxure Automation Expert. En demoapplikation som skulle omfatta elkonstruktion, nätverkskonfiguration, PLC-programmering och utformning av ett enkelt och pedagogiskt Human Machine Interface (HMI).

EcoStruxure Automation Expert eller EAE ska göra det möjligt att styra och övervaka maskiner samt processer på ett mer IT-orienterat sätt. För den här typen av aktiviteter används vanligtvis PLC-programmering som är baserat på standarden IEC 61131-3. Ett standardspråk för programmerbara styrsystem (PLC). EAE är baserat på standarden IEC 61499 som i sin tur är baserad på IEC 61131-3. Det innebär att språken är lika men att EAE blir mer likt ett högnivåspråk som ger systemet en högre transparens mellan IT (information technology) och OT (operational technology). I och med detta öppnas möjligheten för programmerare med en mer IT-orienterad kompetens att kunna arbeta med industriell automation.

Schneider Electric har utformat EAE så att mjukvaran inte ska vara beroende av hårdvaran. Att frikoppla hårdvara från mjukvara innebär att hårdvaruplattformen kan uppgraderas efter behov, och kan ändras från plattform till plattform utan att påverka applikationen som programmerats i mjukvaran. Att frikoppla ett "virtuellt PLC-gränssnitt" innebär att systemintegratörer kan se systemhårdvaruplattformar som både oberoende och agnostiska, och därför blir uppgifter som att uppgradera eller byta ut styrenheter(HW) mycket enklare.

1.2 Syfte

Syftet med examensarbetet är att bygga och programmera en demoapplikation som på ett pedagogiskt sätt visar hur kommunikationen mellan de ingående delarna fungerar. Ett lätthanterligt HMI ska gå att styra, samt visa data. Detta för att Schneider Electric senare ska kunna visa upp denna demo för kunder. En demo som visar att EAE är en mjukvara och ett verktyg som är lätthanterligt och smidigt att använda sig av inom automationsindustrin.

1.3 Målformulering

Examensarbetet ska utföras så att en demoapplikation programmerad med en ny mjukvara EAE kan demonstrera och kontrollera olika komponenter och dess data. Komponenter som är Schneider Electrics egna samt produkter från andra tillverkare.

Ett HMI ska byggas som på ett enkelt och översiktligt sätt åskådliggör data. HMI:et ska visa data såsom energiförbrukning för hela anläggningen. Det ska även kunna starta och stoppa en motor samt visa olika data från motorn, såsom spänningsnivå, RPM (varvtal) och strömförbrukning.

Kommunikation mellan de olika komponenterna ska visas i form av att olika utgångar blinkar.

1.4 Problemformulering

1. Kommer det fungera att implementera EAE på alla komponenter?
2. Vilka komponenter kommer att väljas?
3. Hur ska anläggningens elkonstruktion se ut?
4. Hur ska HMI:et utformas, finns där färdiga objekt att använda sig av eller behövs egna göras?

1.5 Motivering av examensarbete

Ett helt nytt projekt där det flesta beståndsdelar av utbildningen fanns med, programmering, elkonstruktion och nätverkskonfiguration, gjorde att valet av examensarbete landade hos Schneider Electric. En behaglig kontorsmiljö samt ett framtidsstänk som lockade.

Idén om att projektet sedan ska kunna visas för potentiella kunder gav en känsla av att projektet är lite extra seriöst och viktigt. Detta samtidigt som mjukvaran är en helt ny produkt som kan komma att gynna industriautomationen i framtiden.

1.6 Avgränsningar

Hårdvara är begränsad till ett visst antal komponenter som i sin tur kunde minskas om tidsbrist uppstod. Komponenter fick prioritet från ett till sju.

Applikationens komponenter ska bara kunna kommunicera med varandra i form av att kunna blinka med lampor på in- och utgångar samt att HMI ska enbart täcka de data och funktioner som är önskade.

2 Teknisk bakgrund

Följande kapitel kommer behandla den tekniska bakgrunden för hårdvara och mjukvara som använts under projektets gång. Hårdvaran är de olika komponenterna som är inblandade och mjukvaran representerar de olika program som använts vid konfiguration, programmering samt design av det grafiska gränssnittet.

2.1 IEC 61131-3

Programmable Logic Controller (PLC) har blivit en väldigt central punkt inom automationsindustrin och så tidigt som slutet av 1960-talet föddes den första PLC:n. Richard E. Morley var en av huvudpersonerna i utvecklandet av denna komponent. PLC-programmering har därefter växt fram och kan appliceras med en mängd olika språk. Därför har det blivit en standardisering utav de vanligaste språken man använder sig av och denna standard har fått namnet IEC 61131. Därefter har det tillkommit uppdateringar av denna standard en av dessa är IEC 61131-3 [1]. Denna standard innehåller fem olika delar:

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

2.2 IEC 61499

Det är ett händelsestyrt standardspråk som är vidareutvecklat från standard IEC 61131-3:s funktionsblocksdiagram (FBD). Där funktionsblock har fått ett huvud och en kropp. Event-ingångar/utgångar som utgör huvudet och styr hur kroppens data-ingångar/utgångar ska hantera sin data.

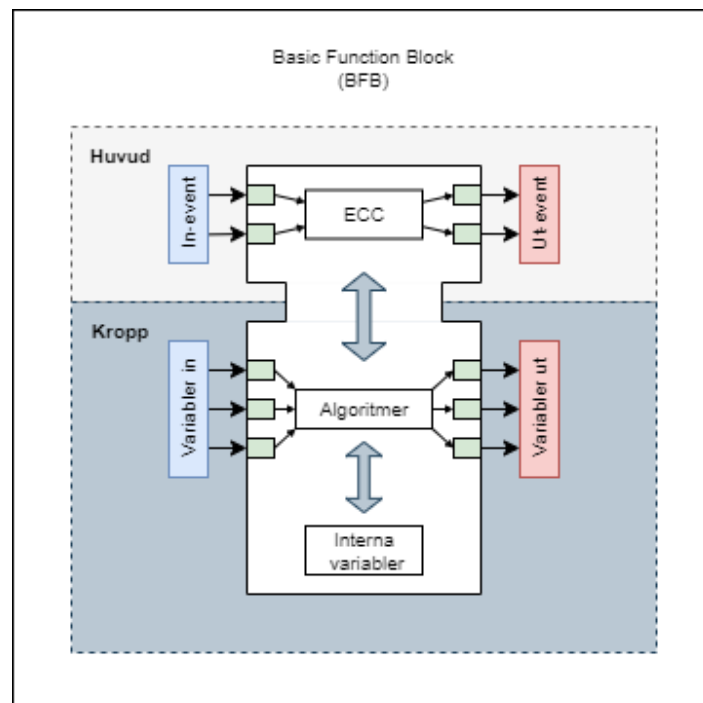
Där finns tre olika funktionsblock som ingår i IEC 61499 [2] och är:

- Basic Function Block (BFB)
- Composite Function Block (CFB)
- Service Function Block (SFB)

2.2.1 Basic Function Block (BFB)

BFB är ett funktionsblock som består av tillståndsmaskin och algoritm.

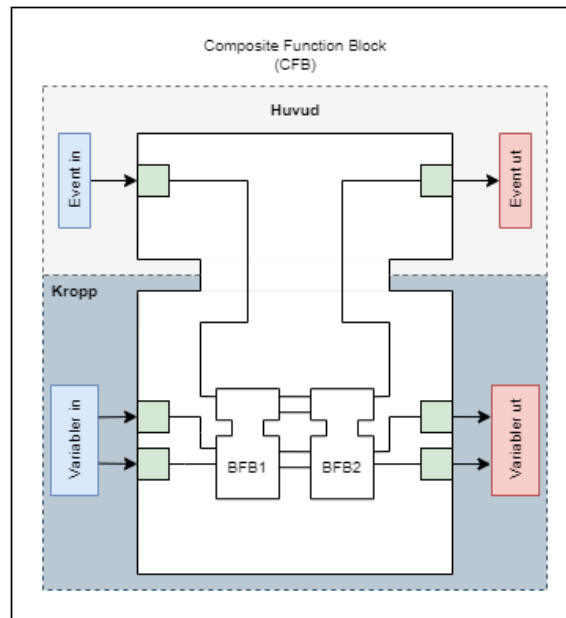
Tillståndsmaskinen byggs upp via funktionsblockets Execution Control Chart (ECC) där olika tillstånd blir kopplade till olika event och därefter exekverar de algoritmer som önskas. Algoritmerna är den del som innehåller kod i form av strukturerad text. Antal algoritmer är så många som behövs för att utföra det arbete funktionsblocket är försedd till att göra.



Figur 1. Bild på ett exempel av Basic Function Block

2.2.2 Composite Function Block (CFB)

CFB är ett funktionsblock som kan innehålla flera olika funktionsblock ihopkopplade med varandra i ett varierande stort nät. Till exempel kan ett CFB innehålla flera olika BFB med olika funktioner som utför olika sorters algoritmer när de ansluts till varandra.



Figur 2. Bild på ett exempel av Composite Function Block

2.2.3 Service Function Block (SFB)

SFB är ett funktionsblock som kan representera gränssnittet till hårdvaran, det kan även fungera som en mjukvarukomponent som implementerar tjänster. Det är tjänster så som sensorers värden, utökning av minnet som körs av en resurs och skicka meddelanden. Den del som använts i examensarbetet är i form av Symlink som länkar data mellan mjukvara och hårdvara.

2.3 Mjukvara

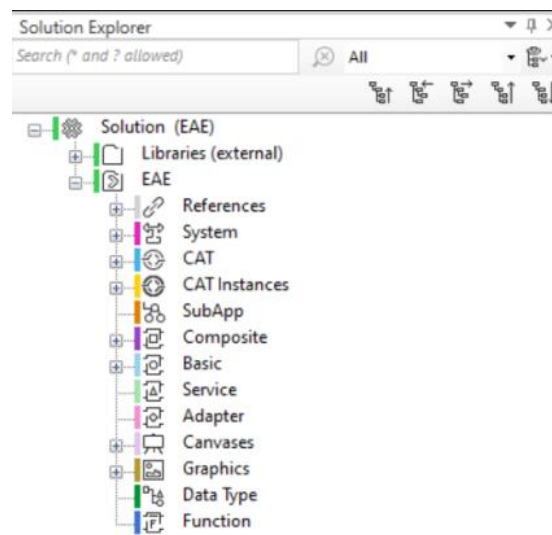
Under projektets gång har det använts fyra olika mjukvaror. Det är självaste mjukvaran som demoapplikationen har blivit programmerad i och även konfigurationsprogram till de olika komponenterna.

2.3.1 EcoStruxure Automation Expert (EAE)

EcoStruxure Automation Expert är en mjukvara som är självutvecklad av Schneider Electric. Den är utvecklad utifrån standarden IEC 61499 och har förmågan att kunna arbeta i en öppen systemmiljö. Vad som innebär med det är att man kan frigöra mjukvaran från hårdvaran. Med den funktionen innebär

det att den är kompatibel med all hårdvara som i sin tur är kompatibel med firmwären Universal Automation.

EAE innehåller färdiga bibliotek att installera med olika sorters funktionsblock. De olika funktionsblocken kan få en egendesignad struktur så att den önskade funktionen uppnås. I och med att EAE är utvecklad från IEC 61499 medför det att programmeringen utförs med funktionsblock och har tillgång till de standardiserade funktionsblock som finns i IEC 61499. Ett tillägg av funktionsblock i EAE som inte är med i standarden IEC 61499 är Composite Automation Type (CAT) och fungerar liknande som Composite Function Block gör. Det som skiljer de båda åt är att CAT har möjlighet att presenteras som ett grafiskt objekt i ett canvas. Ett canvas som är det inbyggda Human Machine Interface (HMI) i EAE och är det som tillåter grafisk presentation och interaktion av koden för användaren [3].



Figur 3. Bild på Solution Explorer i EAE

2.3.2 SoMove

Är ett konfigurationsprogram för olika hårdvara i form av styrenheter. Här konfigureras Schneider Electrics egna produkter som i det här fallet var frekvensomriktaren ATV340 [4].

2.3.3 Automation Device Maintenance

Automation Device Maintenance är ett program som uppdaterar programvaran i komponenten så att den är kompatibel med annan mjukvara. Den kan även konfigurera själva komponentens kommunikationsdel så som IP-adress och subnät mask [5].

2.3.4 TM3BC IO Configurator

Är Schneider Electrics egna konfigurationsverktyg för deras TM3 Bus Couplers. Där konfiguration av IO görs[6].

2.4 Human Machine Interface (HMI)

Är ett grafiskt gränssnitt som är utvecklat för att en människa och maskin ska kunna kommunicera på ett enkelt sätt. Refererande data om maskinen presenteras för människan samtidigt som styrning av maskin kan utföras av människan.

2.5 Hårdvara

Det var en mängd olika hårdvara som användes i projektet och detta avsnitt kommer förse information om den använda hårdvaran. Det är hårdvara som är implementerad i applikationen och är fabrikat så som Schneider Electric, WAGO och Siemens.

2.5.1 IPC - Harmony P6 (Schneider Electric, HMIP67GA0N30BN7N00)

Är en industriell PC (IPC) med Windows som operativsystem och har en 15'' touchdisplay installerad som skärm.

Det är PC som är designad för arbete inom industrin och som i det här fallet kommer vara den del som presenterar självaste applikationen i ett HMI [7].

2.5.2 IPC - Simatic (Siemens, IPC 6AG4141-5AB17-0FA0)

Är även detta en industriell PC med Windows som operativsystem men har ingen skärm installerad. Den kommer i stället att vara en av de två komponenterna som körs som en kontrollerenhet. Dess huvuduppgift kommer att vara att distribuera ut den programmerade kod till valda komponenter i applikationen [8].

2.5.3 Ethernet TCP/IP managed switch - MCSESM163F23F0 (Schneider Electric)

Är den produkt som länkar ihop alla olika komponenter i applikationsnätet. Hårdvarukonfigurationen av switchen är 16 Ethernetkontakter som kan kommunicera med en mängd olika port-protokoll, så som Ethernet TCP/IP, SSH, HTTP, HTTPS, SNMP, RS232, SFTP, SCP, LLDP, Telnet, SNTP och MRP. Detta för att täcka ett så stort område av antal komponenter att kunna kommunicera med [9].

2.5.4 Nätaggregat - ABL7RE2405 (Schneider Electric)

Är en äldre modell av strömförsörjningskomponent från Schneider Electric. Det är mer eller mindre ett nätaggregat som ser till att transformera och likrikta vägguttagets 230V växelspänning till 24V likspänning. Detta då vissa av komponenterna inte har ett eget internt nätaggregat som klarar av att göra det [10].

2.5.5 PLC - TM251MDESE (Schneider Electric)

Förkortningen PLC står för engelskans Programmable Logic Controller som översatt till svenska står för programmerbart styrsystem. Namnet för komponenten är M251 och är en controller för industrimaskiner. Den innehåller en processor, RAM och Flash-minne och Ethernetportar. In- och utgångar hanteras av två externa moduler som man klickar ihop med själva PLC:n [11].

2.5.5.1 Ingångskort - TM3DI16G (Schneider Electric)

Är modulen som har 16 ingångskontakter som är länkade till varsin LED-lampa [12].

2.5.5.2 Utgångskort - TM3DQ16TG (Schneider Electric)

Är den andra modulen som har 16 utgångskontakter som också är länkade till varsin led-lampa [13].



Figur 4. Bild på PLC:n M251 med in- och utgångsmoduler

2.5.6 Fältbuskopplare - Modbus TCP 750-362 Ethernet (WAGO)

ModBus är ett protokoll som är ett av de vanligaste kommunikationssätten inom automationsindustrin. Den här komponenten tar in signaler via Ethernet som i sin tur kan kommunicerar via ett antal ETHERNET protokoll så som, HTTPS, BootP, DHCP, DNS, SNTP, FTPS, SNMP.

750-362 ETHERNET fungerar som en fältbuskopplare och är kompatibel med att ansluta det modulära WAGO-I/O-systemet till ETHERNET [14].

2.5.7 Bus Coupler – TM3BCEIP (Schneider Electric)

Är en komponent som anpassar storleken på in och ut dataarea beroende av vad den är konfigurerad för där det minsta är ett 16-bitars register, där varje bit i registret motsvarar en kanal på utgångskortet. TM3BCEIP styrs av en kontroller och dirigerar sedan ut signaler via utgångskortets I/O:s [15].

2.5.7.1 Utgångskort - TM3DQ8TG (Schneider Electric)

Är en utgångsmodul som har åtta utgångar med länkade LED-lampor, detta är den utgångsmodul som används till TM3BCEIP [16].

2.5.8 Frekvensomriktare - ATV340U07N4E (Schneider Electric)

Denna komponent används för att omvandla frekvensen på inkommande växelspänning till önskad frekvens på den utgående växelspänningen. Detta för att till exempel kunna styra en elektrisk motors varvtal väldigt precist, där låg frekvens ger lågt varvtal och hög frekvens ger högt varvtal.

Den här modellen på frekvensomriktare är dock gjord för att operera i en trefasmiljö, men med en specialkonfiguration kan den även operera i enfasmiljö [17].

2.5.9 Energimätare – METSEPM8240 (Schneider Electric)

Är den komponent som mäter energiförbrukningen av hela applikationen. Har flera olika funktionaliteter och kan mäta och presentera flera olika elenergidata. Det är en produkt likt ATV340 som är gjord för att arbeta i en trefasmiljö. Specialkonfiguration kan göras för att mätning av enfas ska vara möjligt [18].

2.5.10 WIFI Accesspunkt – TCSEGWB13FA0 (Schneider Electric)

En produkt som fungerar som en WIFI accesspunkt, den kopplas till önskad produkt och gör därefter kommunikationen till denna produkt trådlös.

Underlättar uppladdning av kod till hårdvara då man annars hade behövt vara ansluten via kabel [19].

3 Metod

I det kommande avsnittet kommer examensarbetets olika faser att beskrivas och motivation av de källor som använts vid informationssökningen att presenteras.

Första fasen av arbetet ägnades åt planering och inhämtning av information om de olika produkterna. Det följdes av inkoppling, installation och nätverkskonfiguration av alla komponenter. Detta så att de var redo för implementation när det var deras tur att implementeras.

Fasen därefter omfattade utbildning inom programmeringsmjukvaran, EcoStruxure Automation Expert.

Tredje fasen av arbetet innehöll programmeringen av demo-applikationen. Det innebar test av nya funktioner inom mjukvara samt utförande av den planerade lösning som skulle programmeras.

Den sista fasen av arbetet har varit den digitala designen av själva demo-applikationen. Hur självaste gränssnittet ska se ut och vilka funktioner som ska vara tillgängliga och inte. Allt utfört i EAE:s egna HMI.

3.1 Förberedelse och förundersökning

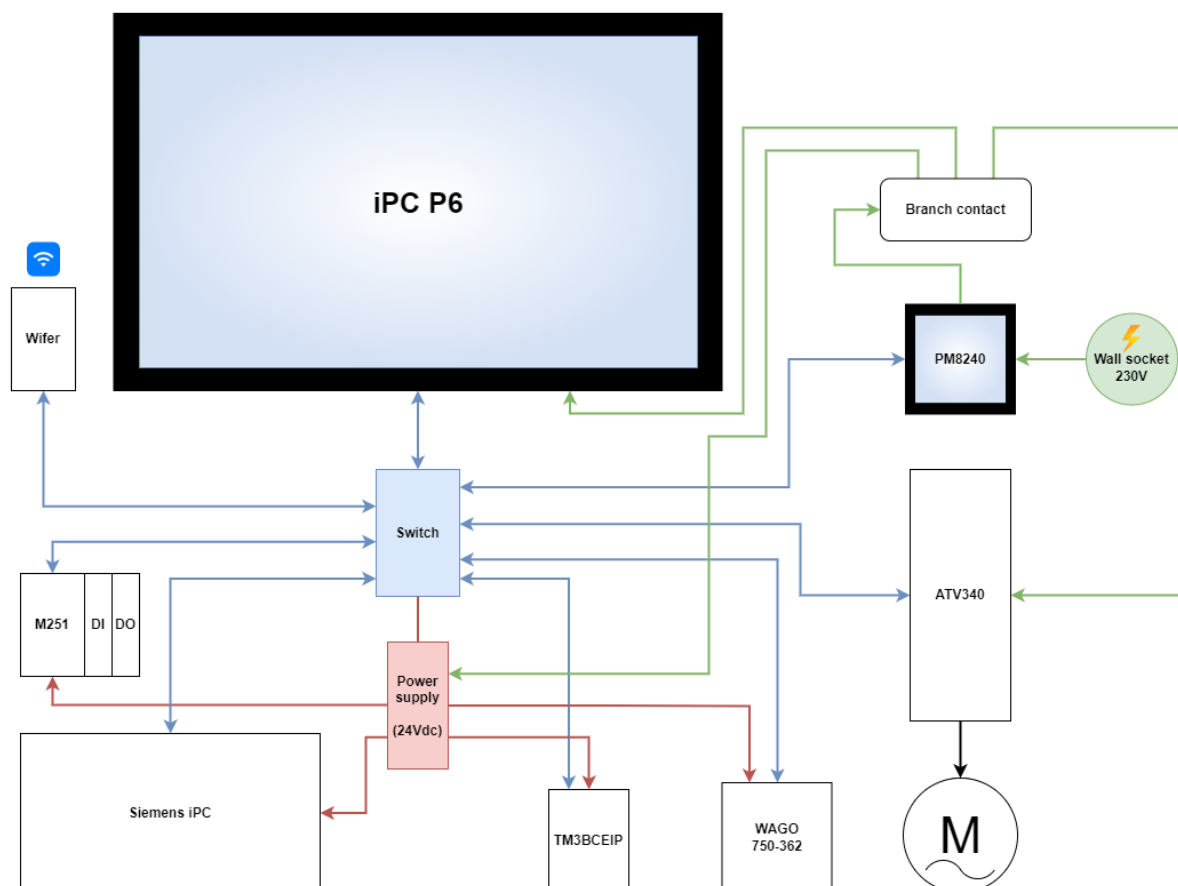
Examensarbetets inledning började med att ett uppstartsmöte planerades in. Det omfattade vad företaget vill uppnå med arbetet och vilka komponenter som skulle vara inblandade. För ett så effektivt arbete som möjligt lades en plan upp tillsammans med handledare på företaget där implementering av de olika komponenterna fick olika prioritet. Handledare förespråkade att en komponent i taget skulle implementeras, vilket medförde att arbetsättet och planeringen utformades därefter.

Prioriteringen gjordes dels för den rådande ovisshet som fanns angående leverans av de olika komponenterna som projektet skulle innehålla.

Tidsuppfattningen angående implementation av de olika komponenterna var relativt oviss och togs även den in i prioriteringen.

När initiering av projektet var klar startades informationsinhämtning angående de olika komponenter som fanns på plats på kontoret. Information så som vilken spänning de olika komponenterna ska ha, hur kommunikation mellan de ingående delar ska fungera samt hur och var monteringen skulle äga rum. Information för Schneider Electrics egna produkter och komponenter togs fram från Schneider Electrics egen hemsida. Produktblad och installationsmanualer för varje Schneider Electric-produkt fanns att ta del av där.

De komponenter som inte var från Schneider Electric var från WAGO och Siemens, där de officiella hemsidorna för vardera varumärke hade tillgängliga produktblad och installationsmanualer att ta del av.



Figur 5. Översiktsskiss för hela kopplingschemat, med gröna linjer som 230V, blå är Ethernet och röda är 24V.

Tabell 1. Sammanställning av figur 5

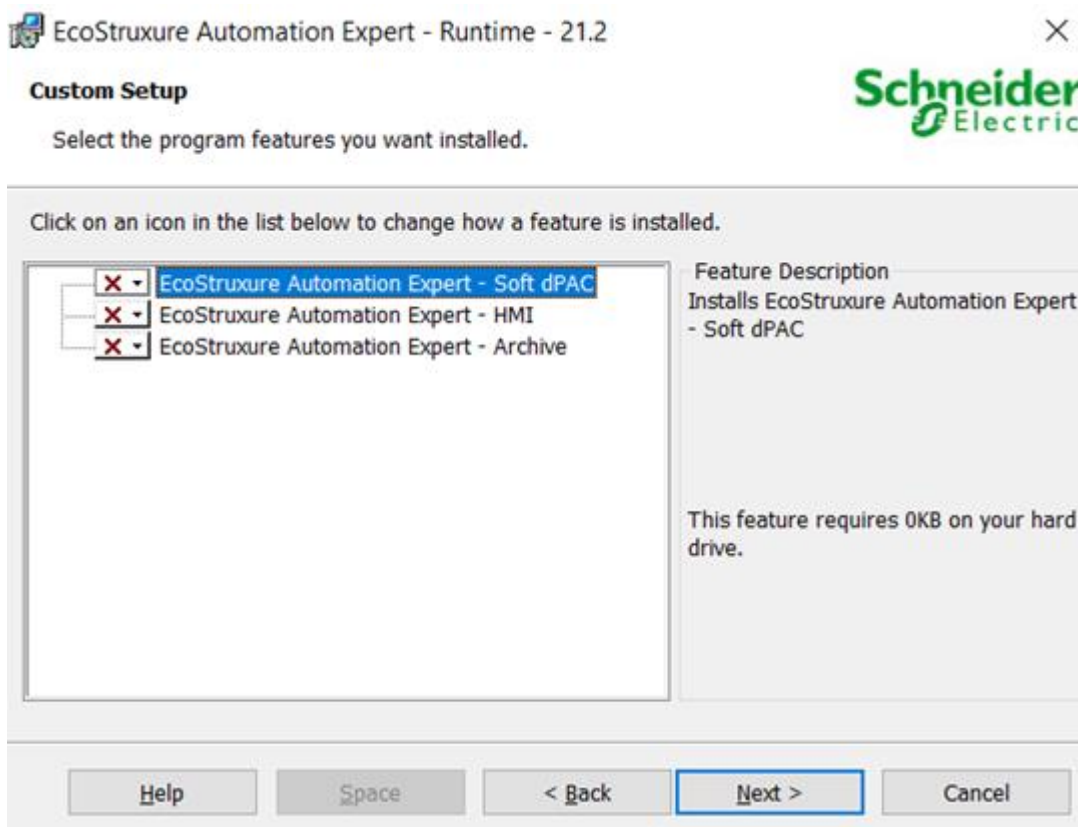
Benämning	Komponent
Wifer	WIFI Access Point
M251	PLC
Siemens IPC	Industri-PC
TM3BCEIP	Bus Coupler
WAGO 750-362	Fältbuskopplare
ATV340	Frekvensomriktare
PM8240	Energimätare
IPC P6	Industri-PC

3.2 Installation av hårdvara och mjukvara

Efter att planering och förundersökning utförts påbörjades inkoppling av elförsörjningen för samtliga komponenter förutom PM8240 och ATV340. Inkoppling av de senare nämnda komponenter inväntades med fram tills att information angående specialkoppling och specialkonfiguration inkommit från kunnig personal hos Schneider Electric.

Nätaggregatet var det första som blev inkopplat tillsammans med switchen och IPC:na, Harmony P6 och Simatic. Harmony P6 kopplades rakt in i vägguttaget med 230V och Simatic kopplades till nätaggregatets 24V. När elförsörjningen till P6:an och Simatic var på plats behövdes en installation och konfiguration av operativsystemet Windows 10 på de båda IPC:na. Därefter tilldelades de varsin statisk IP-adress och installationsfiler för mjukvaran fördes över. Installationsfilerna bestod av olika alternativ beroende av vad komponenten skulle ha för uppgift i applikationen.

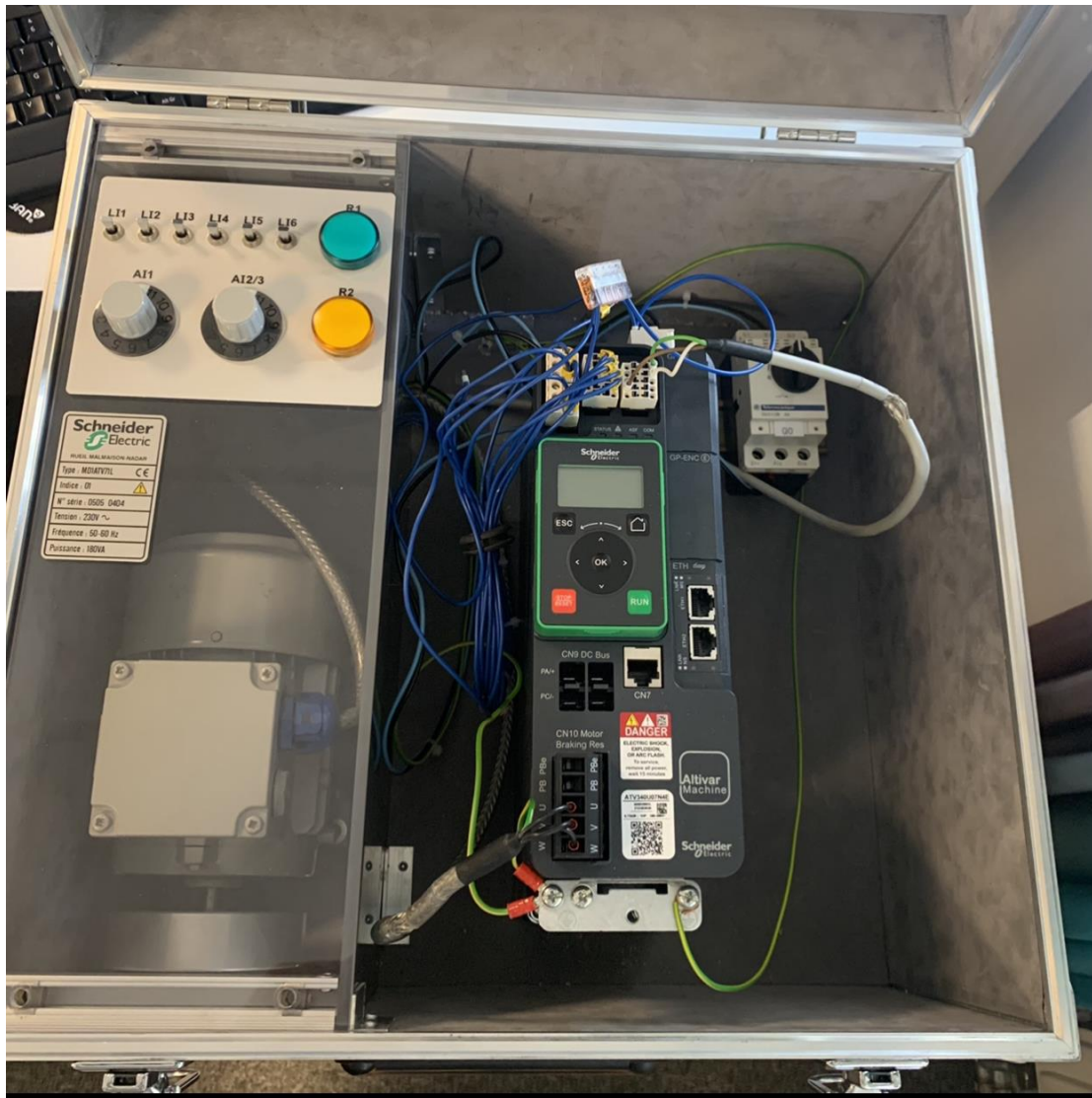
EAE:s Soft dPAC installerades på Simatic och HMI-manager på Harmony P6.



Figur 6. Vilka installationsval som var möjliga att välja vid installation

Alla produkter var fabriksnya förutom motorn som frekvensomriktaren skulle styra. Motorn och den gamla frekvensomriktaren satt monterade i en väska. Den gamla frekvensomriktaren var en Altivar 71 och var Schneider Electrics

äldre version av frekvensomriktare. Ett byte av denna var nödvändig då den inte var kompatibel med nya mjukvaran och dess nya funktioner. Inkopplingen och monteringen av den nya frekvensomriktaren ATV340 visas i *figur 7* och gjordes med hjälp av handledare. Kopplingarna numrerades och dokumenterades för framtida utbyte av frekvensomriktare.



Figur 7 Motorväska med ATV340.

3.3 Nätverkskonfiguration

För att kommunikation mellan komponenterna skulle vara möjlig var det nödvändigt att sätta upp olika statiska IP-adresser och subnätmasker för varje komponent som skulle ingå i demo-applikationen. På IPC:erna sattes IP-adresserna och subnätmaskerna via inställningar i Windows som på en vanlig PC. Därefter var det varierande för nästan alla komponenter.

3.3.1 PLC - M251

PLC:n M251 fick sin IP-adress och subnätmask via Schneiders egen konfigurationsprogramvaran Automation Device Maintenance. Där fanns även tillgång till rätt version av programvara för själva komponenten.

3.3.2 Buskopplare - TM3BCEIP

TM3CEIP fick sin konfiguration via programmet TM3BC IO Configurator som även det är en mjukvara utvecklad av Schneider Electric.

3.3.3 Fältbuskopplare - WAGO 750-362

Öppnades via webbläsaren med hjälp av dess default IP-adress och blev därefter tilldelad önskad konfiguration av IP-adress och Subnätmask.

3.3.4 WIFI Access Point - Wifer

Konfigurerades på samma sätt som WAGO genom webbläsare och med hjälp av dess default IP-adress.

3.3.5 Energimätare - PM8240

Hade interna inställningar där konfiguration av olikas saker kunde göras inklusive tilldelning av statisk IP-adress och subnätmask.

3.3.6 Frekvensomriktare - ATV340E

Behövde programvaran SoMove som konfigurationsprogram, det är som tidigare nämnt en programvara från Schneider Electric.

3.4 Utbildning

Informationssökande på Schneider Electric's egen hemsida samt deras videos på Youtube lärde lite om grunderna för EcoStruxure Automation Expert. Det behövdes dock en grundläggande utbildning om hur mjukvaran EAE är uppbyggd och fungerar för att projektet skulle kunna genomföras.

Utbildningen bokades in och var en digital fyra-dagars kurs som Schneider Electric erbjuder sina kunder samt anställda som behöver utbildas inom mjukvaran.

3.4.1 Första dagen

Första dagen omfattade:

- Introducering av IEC 61499
- Skapandet av solution
- Skapa FB och koppling mellan FB
- Mappning till simulationskomponenter
- Skapande av canvas
- Skapande av alarmgränssnitt
- Interlocks och Permissives

3.4.2 Andra dagen

Andra dagen av utbildningen innehöll:

- Hantering av en solution
- Hantering av bibliotek
- Skapande av liten applikation

3.4.3 Tredje dagen

Tredje dagens agenda gick igenom:

- Composite Function Block (CFB)
- Adapters
- Composite Automation Type (CAT)

3.4.4 Fjärde dagen

Fjärde och sista dagen av utbildningen omfattade:

- Basic Function Block (BFB)
- Enheter och distribution
- Fysisk topologi
- ModBus-konfiguration

3.5 Programmering

Efter att utbildningen avslutats var de två första veckorna någon form av testprogrammering, där programmeringstips som getts på utbildningen testades och utvecklades. Handledaren på företaget avsatte även tid för frågor och tips under hela programmeringsfasen.

3.5.1 Idéutveckling

Programmeringen påbörjades och den första delen innehöll idéutveckling för hur självaste demo-applikationen skulle fungera och se ut. Vilka komponenter som skulle göra vad och vilken som skulle kommunicera med vilken.

Vetskapen om att två kontrollerenheter skulle styra och distribuera ut kod till fyra andra komponenter gjorde att applikationen kunde utformas utifrån det. Olika idéer tänktes igenom och vissa sattes åt sidan då de tenderade till att vara för tidskrävande. När en någorlunda funktionell idé utformats påbörjades den riktiga programmeringen i mjukvaran. Utformningen av idén såg ut på följande sätt, två controllers, PLC:n M251 och Siemens IPC. En första funktion i form av en lampsekvens för utgångslamporna på komponenterna M251, WAGO 750-362 och TM3BCEIP. En andra funktion i form av att styra motorn som var ansluten till frekvensomriktaren ATV340. Sen en tredje funktion som skulle visa elenergidata för hela systemet med hjälp av PM8240. Programmeringen av Human Machine Interface (HMI) vävdes ihop med all programmering, detta då knappar och andra objekt testades och indikerade ifall om implementationen fungerade korrekt eller inte.

3.5.2 Lampsekvens

Programmeringen av lampsekvensen påbörjades och inleddes med att skapa egna funktionsblock. Ett Basic Function Block (BFB) som skapades från grunden med olika algoritmer skrivna i strukturerad text. En algoritm som räknade upp och en som räknade ner ett värde. Detta länkades senare med ett antal befintliga funktionsblock från installerade bibliotek i mjukvaran. Alla block tillsammans skapade ett funktionsblocks-nät som i sin tur skapades inuti en CAT. Denna Composite Automation Type kunde därefter användas för både M251 och WAGO 750-362 som båda två hade 16 utgångar i utgångsmodulen.

För TM3BCEIP gjordes ett ny Basic Function Block, funktionen som efterfrågades var densamma som i för föregående BFB. Skillnaden var att utgångsmodulen för TM3BCEIP bara hade 8 utgångar. Koden ändrades därefter till att behandla 8 bitar i stället för 16 bitar. Samma uppbyggnad som för föregående CAT bara att denna behandlade 8 utgångslampor i stället för 16 utgångslampor.

3.5.3 Energimätning

PM8240 är komponenten som blev implementerad efter lampsekvensens olika komponenter. Den implementerades som en ModBus TCP och olika önskade register med mätdata länkades till symboler skapade i HMI. Dessa symboler presenterade mätvärdena live i gränssnittet. Detta byttes sedan ut mot en färdig hårvaru-CAT som presenterade alla mätvärden för PM8240. Denna hårvaru-CAT blev implementerad i en uppdatering av EAE under programmeringsfasen och fanns inte i biblioteket från början.

3.5.4 Motorstyrning

ATV340 deklarerades som en ModBus TCP i EAE och hade en färdig hårvaru-CAT installerad. Den färdiga hårvaru-CAT kunde styra motorn genom forcerade värden från början. En mer intuitiv styrning ville uppnås så implementering av styrning och feedbackdata utan forcerade värden påbörjades. En ny CAT skapades där ett nytt Basic Function Block lades in. BFB:et fick implementerat kommandon så som framåtkörning, bakåtkörning, stopp och börvärdesinmatning av hastighet. All styrningsdata länkades senare till HMI-delen och knappar designades för intuitiv styrning av motorn. För länken mellan hårvara och mjukvara användes Symlinkmultivars som i sin tur är väldigt intressanta då de skapar kopplingen mellan signalerna i applikationen till I/O-gränssnittet i hårvaran.

3.5.5 HMI

Programmering av HMI:et pågick under hela fasen av examensarbetets programmeringsdel där grafiken i början inte var bearbetad utan testkördes i många gånger om. Detta för att senare utvecklas till ett mer detaljrikt men lättförståeligt grafiskt gränssnitt.

Färdiga CAT-instanser för olika hårvara fanns i de installerade biblioteken och användes samtidigt som egendesignade objekt programmerades och skapades.

3.6 Källkritik

Den inledande delen av teknisk bakgrund är de standardiserade språken IEC 61131-3 med referens [1] och IEC 61499 med referens [2]. Dessa är båda två publicerade artiklar på IEEE vilket betyder att innehållet är granskat och kan därför anses vara pålitliga källor.

De tekniska specifikationer angående de hårdvaror och mjukvaror som är utvecklade av Schneider Electric har tagits från Schneider Electrics egna manualer, referenser [3][4][5][6][7][9][10][11][12][13][15][16][17][18][19]. Där företaget står bakom informationen som ges ut i dessa manualer samt att det är information som ges till kunder och användare av produkterna, därför anses dessa vara mycket pålitliga.

De andra tekniska specifikationer som är produkter från WAGO referens [14] och Siemens referens [8] har information hämtats från respektive hemsida och anses även de vara pålitliga källor av samma anledning som för Schneider Electrics egna produkter.

4 Analys

Avsnittet här nedan kommer att ta upp vilka alternativ som valts för att möjliggöra genomförandet av examensarbetet. Samtidigt kommer de problem som uppkommit samt lösningar till dem att presenteras och diskuteras.

4.1 Hårdvara

Vilken hårdvara som skulle användas i arbetet bestämdes tillsammans med handledare. En prioritering för alla tillgängliga komponenter gjordes där krav sattes så som tid och funktionalitet. Antalet tillgängliga komponenter var fler än de som var med i slutversionen av applikationen och hade att göra med att något av kraven inte kunde uppfyllas.

Det var två produkter som inte blev implementerade och var M262 och ICE-block och anledningen till detta var i största del på grund av tidsbrist.

Inkoppling av all hårdvara gjordes gradvis och förenklades av anslutningsklämmor. Dessa klämmor gav möjlighet att koppla flera komponenter från samma utgång från nätaggregatet. Samtidigt som fränkoppling av komponenter som inte användes var enkelt att utföra.

Informationsinhämtningen av olika komponenters data fungerade smidigt dels för att de flesta komponenter hade medskickat ett fysiskt datablad, samtidigt som alla datablad till alla Schneider Electrics komponenter fanns tillgänglig på deras hemsida. Produkterna som var utifrån Schneider Electric hade liknande system och var även de lättillgängliga på respektives hemsida.

Uppgradering av hårdvara var enbart nödvändig för frekvensomriktaren Altivar 71 som ej var kompatibel med nya mjukvaran. Den byttes ut mot en nyare version i form av ATV340. Som i sin tur var kompatibel med de nya funktioner som fanns i EcoStruxure Automation Expert. Det här utbytet av frekvensomriktare var planerat från start, det var inget som kom upp under arbetets gång. De nya inkopplingarna numrerades och det gjordes för att framtida uppgraderingar av hårdvara ska kunna göras smidigt.

I och med att arbetsmiljön var i enfas så var specialkonfigurationer nödvändiga för de produkter som egentligen är skapade för att arbeta i ett trefas-system. De två produkter som behövde specialkonfigurationer var PM8240 och ATV340. Med hjälp av handledare kunde konfigureringen av PM8240 göras genom en specialkoppling av komponenten samt inställningar som gjordes på självaste produkten. Detta hjälpte handledaren till med så att inkopplingen och konfiguration blev rätt.

Frekvensomriktaren var aningen mer komplicerad och en SoMove-fil skapad av en anställd på Schneider Electric var tvungen att köras i mjukvaran SoMove. Detta innebar uppdateringar av parametrar inne i konfigurationsprogrammet som var nödvändiga för drift i enfas i stället för trefas.

4.2 Mjukvara

Där har använts en mängd olika mjukvaror för att konfigurera de olika komponenterna. Navigering i dessa olika har varit en utmaning och med hjälp av handledare och guider online kunde det genomföras relativt smidigt. Information angående vilken mjukvara som skulle användas till vilken komponent var också en utmaning och assistans från handledare erbjöds.

EcoStruxure Automation Expert som var programmeringsmjukvaran för demoapplikationen användes till mesta dels under programmeringsfasen av arbetet. Mjukvaran innehöll en mängd olika funktioner och funktionsblock. Valen av använda funktionsblock gjordes efter att den önskade idé fastställdes. Funktionsblock som tagits upp under utbildningsfasen användes i första hand och var funktionsblocken BFB, CAT, CFB men med input av handledare kunde även funktionsblock som inte behandlats i utbildningen användas. Därför finns där inga strikta gränser för utbildningsfasen under arbetet utan den fortlöpte under hela arbetets gång, då utbildningen på fyra dagar omöjligt kunde täcka hela mjukvarans olika funktioner.

4.3 HMI

Syftet med Human Machine Interface var att få ett begripligt och lätthanerat gränssnitt för användaren. Där alla funktioner skulle vara tillgängliga samtidigt som en överblick skulle åskådliggöra hela konstruktionen. För skapande av egna grafiska objekt kunde programmering i C-Sharp utföras. Detta behövdes aldrig användas då de färdiga CAT-instanser och funktioner som fanns tillgängliga i canvas var tillräckliga för det önskade ändamål.

4.4 Problem

4.4.1 Uppkoppling till internet

Uppdatering av operativsystem och nerladdning av större filer var svårt att ladda ner med internetdelning från telefonen. Ingen internetuppkoppling på kontoret var möjlig, då en förhöjd säkerhet inom företaget förhindrade detta. Problemet löstes genom att de komponenter som behövde större uppdateringar togs hem och uppdaterades hemifrån.

4.4.2 Kommunikation mellan komponenterna

Vid inledande inkopplingar av komponenterna var kommunikationen mellan dem svår att upprätthålla. Problemet var att brandväggarna hos IPCerna och datorn som användes som arbetsstation, blockerad kommunikationen och löstes genom att brandväggen för den porten stängdes av. Därefter fungerade kommunikationen och problemet var löst.

4.4.3 Fel sort av frekvensomriktare

Vid implementation av ATV340 insågs det att beställning av fel sorts ATV340 gjorts. Meningen från början var att ATV340 skulle ha ett dPAC-kort som skulle installeras på frekvensomriktaren. Det vill säga att den skulle fungera som en kontroller, så som M251:an och Siemens IPC skulle göra. Problemet löstes genom att använda den befintliga frekvensomriktaren som en ModBus TCP i stället. De grundläggande funktionerna som önskades kunde därefter uppfyllas och problemet var löst.

4.4.4 Lång leveranstid

WAGO 750-362 levererades inte förrän i mitten av projektet, långa leveranstider för olika produkter var på grund av rådande komponentbrist runt om i Europa och resten av världen. Vid leverans kunde produkten implementeras, i och med att en ovisshet angående leveransen sattes prioritet på produktens implementation till väldigt låg.

4.4.5 Konfigurationshjälp av ICE-Block

Hjälp angående konfigureringen av en produkt vid namn ICE-block skulle göras med Luleås Universitet. Möte bokades in med en person som inte anslöt under utsatt tid. Det visade sig att personen råkat ut för komplikationer under en resa och på grund av det inte kunde ansluta till mötet. Problemet löstes genom att implementation av komponenten lades ned detta då ingen tid för ytterligare ett konfigurationsmöte fanns.

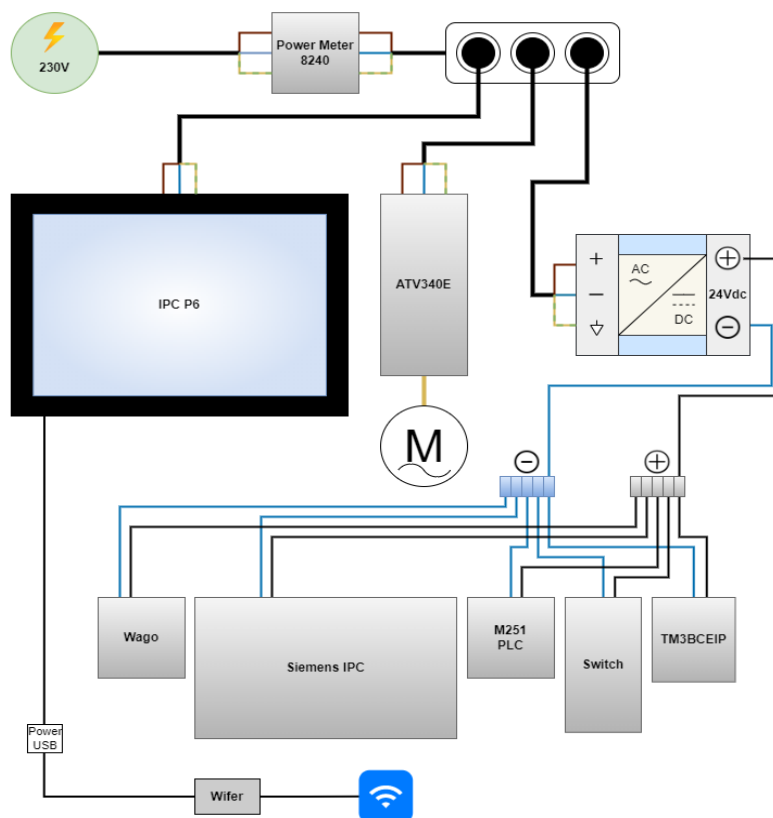
5 Resultat

Detta avsnitt är avsatt för att presentera resultatet av projektet. Ett projekt innehållande ett skapande av en demoapplikation för Schneider Electrics nya mjukvara EcoStruxure Automation Expert.

Arbetet resulterade i en demoapplikation programmerad i form av funktionsblock och en del strukturerad text, samt ett interagerande och presenterande grafiskt gränssnitt. Applikationen omfattar ett nätverk på nio interagerande komponenter som alla är länkade och programmerade via EAE.

5.1 Hårdvara

Den hårdvara som projektet till slut består av är till mestadels ny utrustning som blivit konfigurerade och installerade från grunden. Den utrustning som inte är ny är nätaggregatet och motorn, resterande är fabriksnya produkter beställda för ändamålet att ingå i demoapplikationen. Uppbyggnad av konstruktionen visas i *figur 8* och består av PM8240 som är i mitten av en seriekoppling mellan vägguttaget och förgreningskontakten. Den har i uppgift att mäta energiförbrukningen för hela systemet. Därefter är nätaggregatet, Harmony IPC och ATV340 kopplad till förgreningskontakten som är kopplad till PM8240. Nätaggregatet förser 24Vdc till de resterande komponenterna i konstruktionen så som WAGO 750-362, Siemens IPC, PLC:n M251, Switch och Bus Couplern.

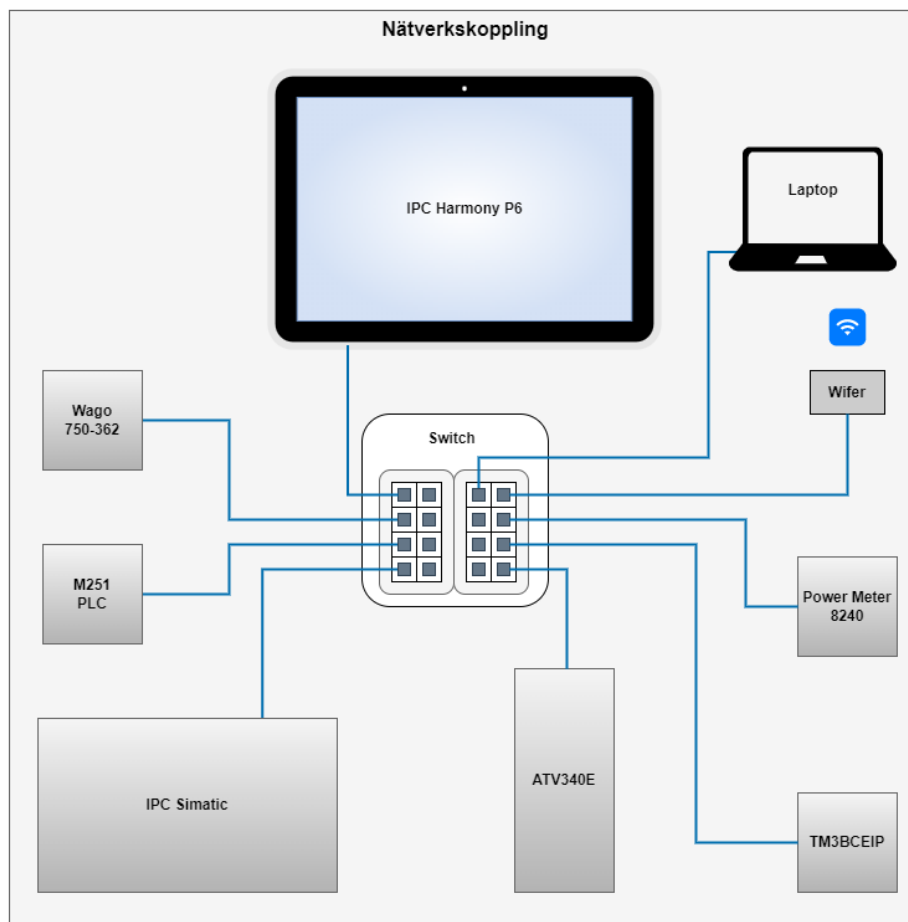


Figur 8 Elkoppling för hela applikationens olika komponenter

All kommunikation går via switchen och är en av komponenterna som blev matade med 24Vdc. Alla komponenter förutom nätaggregatet är kopplade till switchen via Ethernetkabel enligt *figur 9*. Detta möjliggör etablering av kommunikation mellan de olika produkterna. Varje komponent har blivit tilldelade en statisk IP-adress och subnät-mask som presenteras i *tabell 2*.

Tabell 2 Nätverkskonfiguration för de olika komponenterna

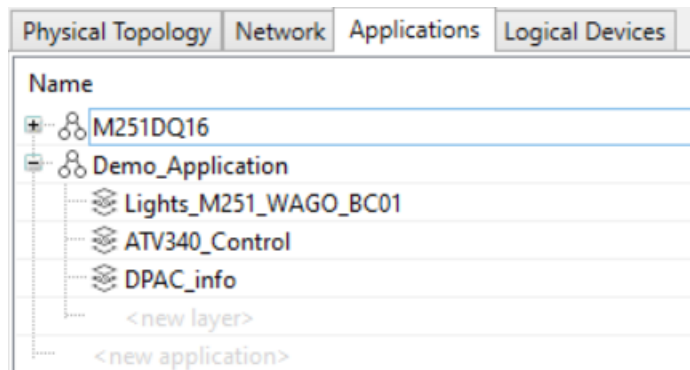
Produkt	IP-adress	Subnätmask
Wifer	192.168.1.1	255.255.255.0
Laptop (Arbetsstation)	192.168.1.10	255.255.255.0
Harmony P6 IPC	192.168.1.11	255.255.255.0
Switch	192.168.1.12	255.255.255.0
M251 PLC	192.168.1.13	255.255.255.0
Simatic IPC	192.168.1.14	255.255.255.0
Power Meter 8240	192.168.1.15	255.255.255.0
WAGO 750-362	192.168.1.16	255.255.255.0
ATV340E	192.168.1.18	255.255.255.0
TM3BCEIP	192.168.1.19	255.255.255.0



Figur 9 Nätverkskoppling för alla ingående komponenter

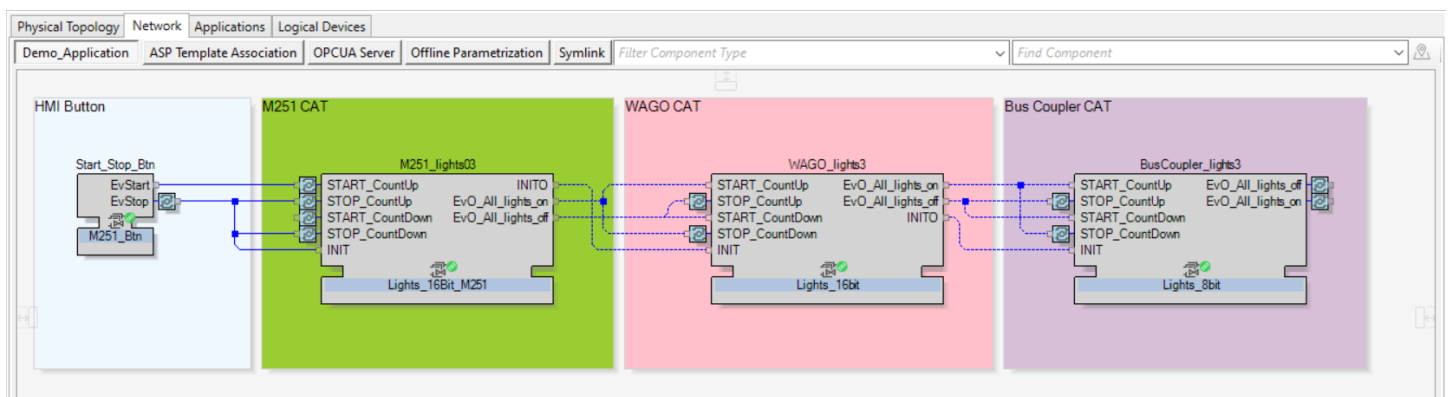
5.2 Lampsekvens - Lights_M251_WAGO_BC

För funktionen som blinkar med utgångarna i en viss lampsekvens programmerades i lagret Lights_M251_WAGO_BC och är ett lager under applikationen Demo_Application.



Figur 10. Applikationsfliken med applikation och lager i EAE

Inne i lagret skrevs koden och innehåller fyra olika CAT-block som visas i figur 11 nedan, läst från vänster till höger. Första CAT-blocket är det som styr den egendesignade knappen för start och stopp i HMI:et. De tre som kommer därefter styr uppräknandet av lampsekvenserna för de olika komponenterna M251, WAGO 750-362 och TM3BCEIP.



Figur 11 Koden i Lights_M251_WAGO_BC med fyra CAT-objekt, Start_Stop_Btn, M251_lights03, WAGO_lights3 och BusCoupler_lights3

Varje block behöver mappas till en logisk enhet vilket innebär att det är den logiska enhetens länkade fysiska enhets CPU som utför exekvering av koden inuti funktionsblocken. Projektet har tre logiska enheter som är länkade till tre olika fysiska komponenter. Dessa har i tidigare skede beskrivits som kontroller och HMI-hanterare och är Siemens IPC, M251 som kontrollers och P6 Harmony som HMI-hanterare.

Logical Device Name	Device Type	Info	Physical Device
EcoRT_0	SE.DPAC.Soft_dPAC		Siemens_iPC (RT_Siemens)
M251_dPAC	SE.DPAC.M251_dPAC		M251dPAC_1
HMI_P6	SE.Standard.HMI_NET		HarmonyIPC_1

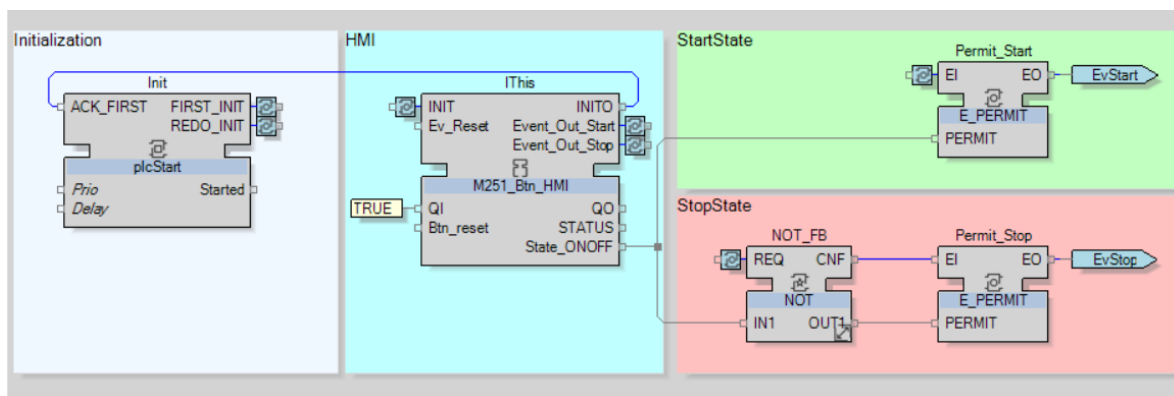
Figur 12 De tre olika logiska enheterna i projektet länkade till de fysiska produkterna.

Start_Stop_Btn, M251_lights03 och BusCoupler_lights3 är länkade till M251_dPAC medan WAGO_lights3 är länkad till EcoRT_0.

5.2.1 CAT - Start_Stop_Btn

Ett steg in i CAT-blocket som illustreras i *figur 13* visas koden för hur knappen och dess funktion är kodad. Det börjar med ett initieringsblock som initierar HMI-blocket M251_Btn_HMI och är det block som länkar data mellan koden och gränssnittet. Knappen är två olika tillstånd, på eller av. Beroende av vilket tillstånd knappen är i, ger den olika värde till utgången State_ONOFF.

Är knappen till blir värdet true och är den från blir värdet false. När utgången får nytt värde triggas både Event_Out_Start och Event_Out_Stop. Beroende av vilket värde State_ONOFF får släpps ett av eventen, EvStart och EvStop vidare ut ur CAT-blocket Start_Stop_Btn. E_PERMIT-blocken fungerar så att när ingången PERMIT får in värdet true släpper den vidare inkommande event.



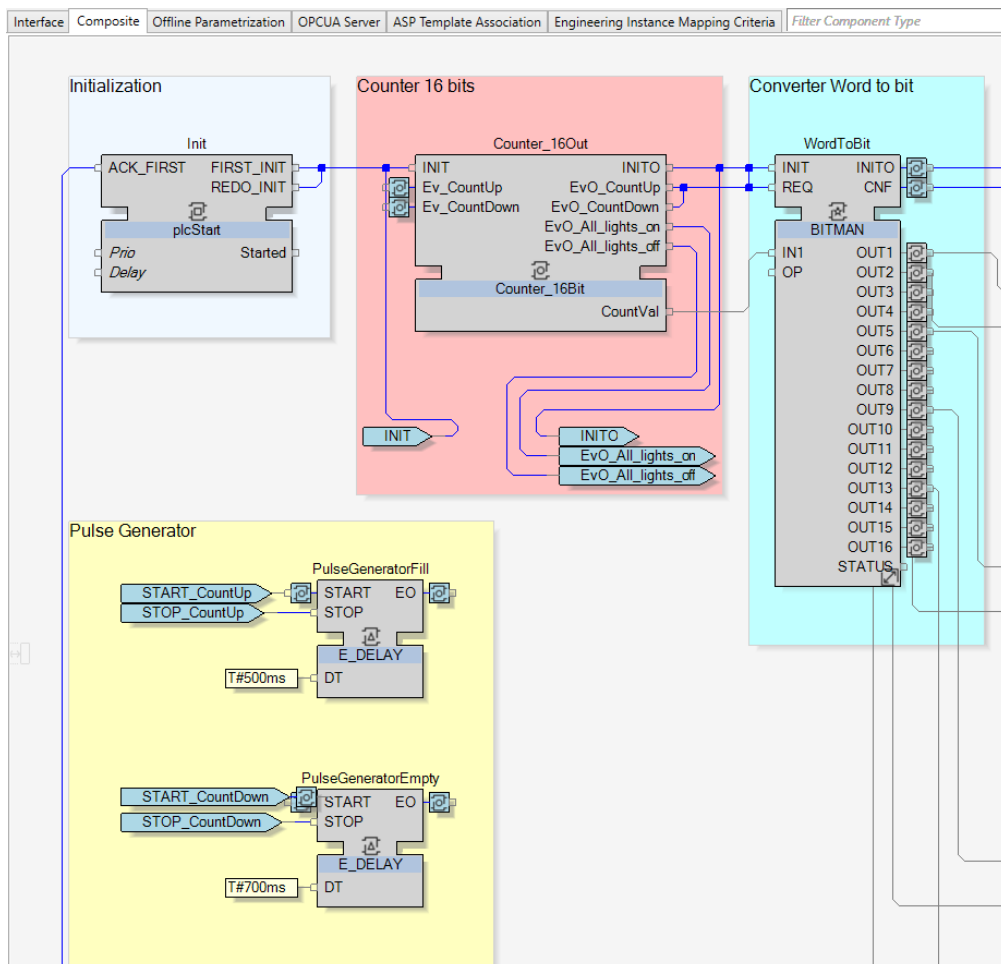
Figur 13 inuti Start_Stop_Btn-blocket

5.2.2 CAT - M251_lights03

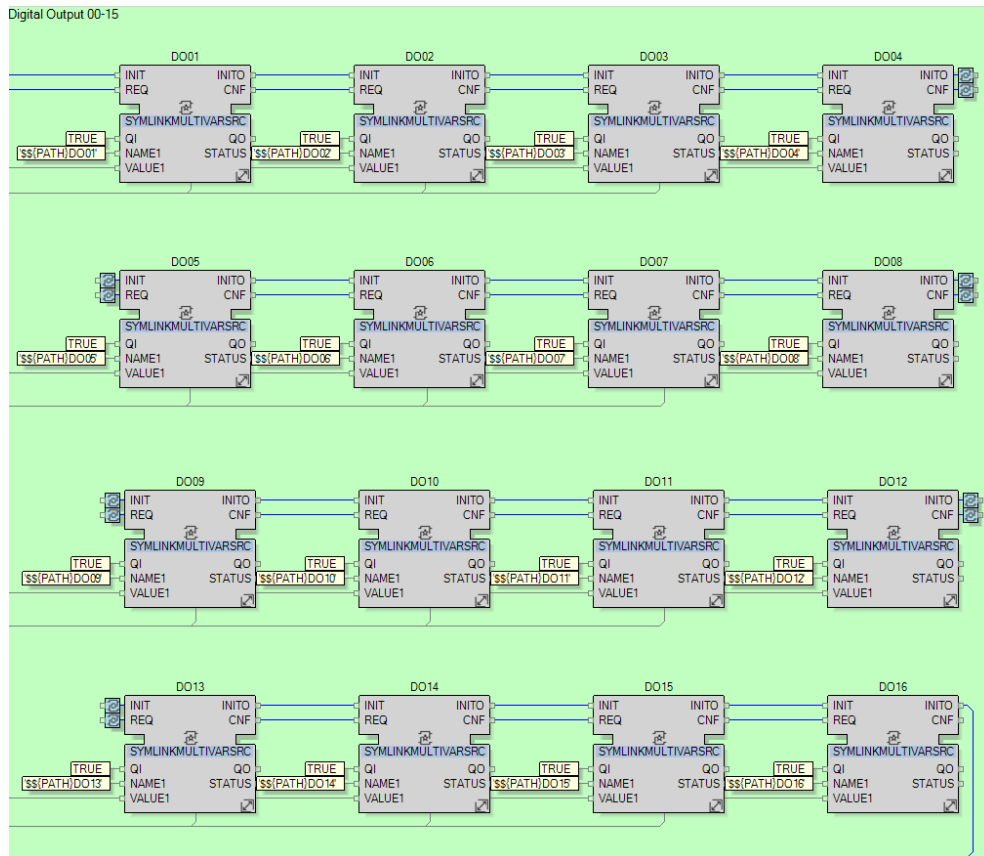
Ett steg in i M251_lights03 visar *figur 14* och *figur 15* nedan att beroende av vilket det inkommande eventet är, triggas antingen E_DELAY-blocken PulseGeneratorFill, PulseGeneratorEmpty eller basic-blocket Counter_16Out. De båda E_DELAY-blocken triggas olika in-event på Counter_16Out för att antingen starta uppräknningen eller nedräkningen.

De är pulsgeneratorer med en fixerad tid för hur ofta en puls ska skickas ut och triggas Counter_16Out:s inevent. PulseGeneratorFill triggas Ev_CountUp och PulseGeneratorEmpty triggas Ev_CountDown.

Det uträknade värdet som CountVal blir tilldelat skickas sen vidare till funktionsblocket WordToBit som omvandlar värdet från datatypen LWORD till BIT. WordToBit har 16 utgångar med datatypen BOOL och som alla är kopplade till vars ett SRC symlink-block som är länken mellan koden och hårdvaran.



Figur 14 Inledande del av koden i M251_lights03



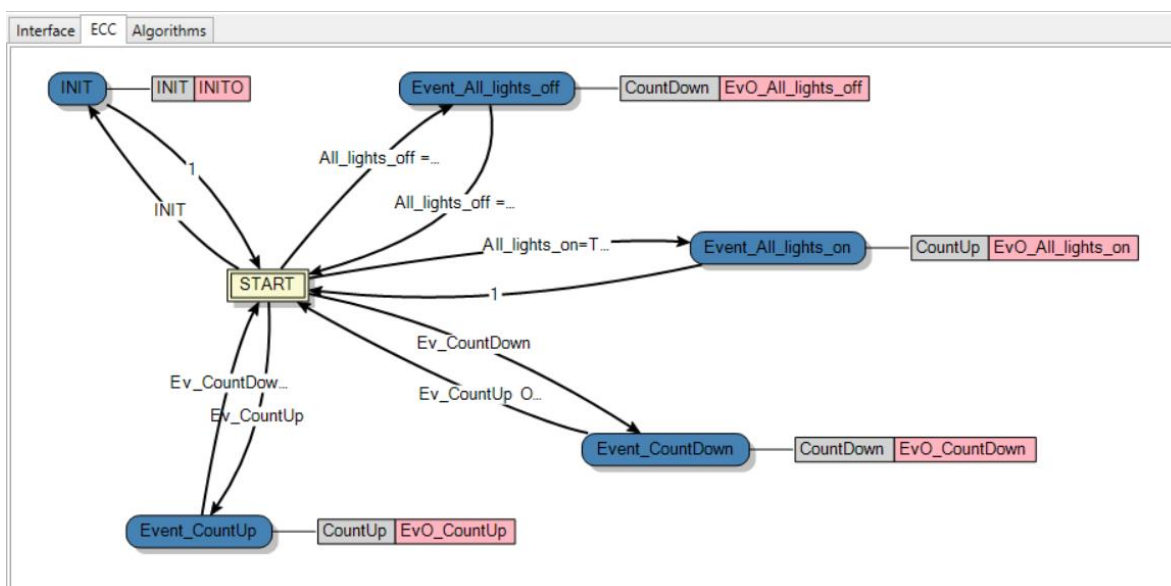
Figur 15 Kopplingarna från utgångarna OUT1-OUT16 på WordToBit av koden i M251_lights03

5.2.2.1 Basic function block - Counter_16Out

Ännu ett steg in i koden kommer vi in i Counter_16Out-blocket där algoritmerna för uppräkning, nedräkning och nollställning finns. Uträkningen sker i datatypen LWORD och är CountVal som blir tilldelat de olika värdena mellan de olika pulsarna. Beroende av vilket in-event som blir triggat kommer olika algoritmer att exekveras.

ECC-delen av Counter_16Out som visas i *figur 16* bestämmer vilken algoritm som ska exekveras och vilket ut-event som ska bli aktivt.

De tre olika in-eventen styr ut-eventen med liknande namn, det vill säga INIT styr INITO, Ev_CountUp styr EvO_CountUp och Ev_CountDown styr EvO_CountDown.



Figur 16 Execution Control Chart (ECC) för Counter_16Out. Ett tillståndsdiagram för funktionsblockets beteende.

Det första som händer för funktionsblocket är att INIT blir triggat och utför en initiering av alla värden, vilket betyder att CountVal får värdet 0, All_lights_on får false och All_lights_off får false.

```

Interface  ECC  Algorithms
INIT      CountUp  CountDown
1  ALGORITHM INIT IN ST:
2  (* Add your comment (as per IEC 61131-3) here
3  Initialization algorithm
4  *)
5  CountVal := 0;
6  All_lights_on := FALSE;
7  All_lights_off := FALSE;
8  END_ALGORITHM

```

Figur 17 Algoritmen för INIT

Därefter kan CountUp bli triggat och utföra en uppräknig av värdet på CountVal tills att det når 65535 eller blir nollställd av INIT. När CountVal får värdet 65535 lyser alla lampor och interna variabeln All_lights_on sätts till true. Det medför att tillståndsmaskinen hoppar in i tillståndet Event_All_lights_on och gör ut-eventet EvO_All_lights_on aktivt.


```

Interface  ECC  Algorithms
INIT  CountUp  Countdown
1  ALGORITHM CountUp IN ST:
2  (* Add your comment (as per IEC 61131-3) here
3
4  *)
5  IF CountVal = 0 THEN
6      CountVal := 1;
7      x := 1;
8      All_lights_off := FALSE;
9  ELSIF CountVal <> 0 AND NOT CountVal > 65535 THEN
10     CountVal := (x*2)-1;
11 END_IF;
12
13 IF CountVal <= 65535 THEN
14     x := x*2;
15 END_IF;
16
17 IF CountVal = 65535 THEN
18     All_lights_on := TRUE;
19 END_IF;
20
21 END_ALGORITHM

```

Figur 18 Algoritmen för CountUp-eventet

CountDown-algoritmen kommer inte aktiveras förrän alla lampor hos resterande komponenter lyser. När det gått så långt att alla lampor lyser hos alla komponenter triggas Ev_CountDown och påbörjar nedräkning av värdet på CountVal. Så fort CountVals värde inte längre är 65535 sätts den interna variabeln All_lights_on till false. När CountVal nått 0 sätts den andra interna variabeln All_lights_off till true och tillståndsmaskinen hoppar in i tillståndet Event_All_lights_off och gör ut-eventet EvO_All_lights_off aktivt.

```

Interface  ECC  Algorithms
INIT  CountUp  Countdown
1  ALGORITHM Countdown IN ST:
2  (* Add your comment (as per IEC 61131-3) here
3
4  *)
5  IF CountVal > 0 THEN
6      CountVal := (x/2)-1;
7  END_IF;
8
9  IF CountVal < 65535 THEN
10     All_lights_on := FALSE;
11 END_IF;
12
13 IF CountVal = 0 THEN
14     All_lights_off := TRUE;
15 ELSIF CountVal > 0 THEN
16     All_lights_off := FALSE;
17 END_IF;
18
19
20 x := x/2;
21
22 END_ALGORITHM

```

Figur 19 Algoritmen för Countdown-eventet

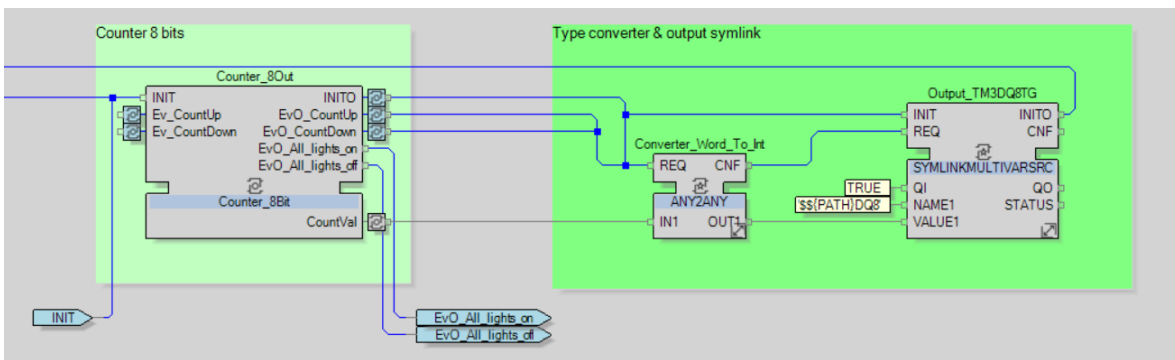
5.2.3 CAT - WAGO_lights3

Har samma uppbyggnad som M251_lights03 det enda som skiljer de åt är att de har 16 nya SRC Symlink-block för att länken mellan kod och hårdvara inte ska krocka. Annars fungerar koden likadant som i M251_lights03.

5.2.4 CAT - BusCoupler_lights3

Skiljer sig lite från de två föregående CAT-blocken i avsnitten 5.2.2 och 5.2.3. Utgångsmodulen för TM3BCEIP har endast 8 utgångar och koden i basic-blocket har därför utformats efter det och visas i *figur 21* och *figur 22*.

Det finns inte heller någon WordToBit utan den har blivit ersatt av Converter_Word_to_int som visas i *figur 20*. Detta block konverterar det framräknade värdet till datatypen INT i stället för BIT. Detta för att registret som behandlar värdet för de olika utgångarna bara är kompatibelt med INT. I och med att det är ett register behövs bara ett Symlink-block som länkar koden till hårdvaran som beroende av värdet tänder rätt antal lampor.



Figur 20 Inne i BusCoupler_lights3

```
INIT  CountUp  CountDown
1  ALGORITHM CountUp IN ST:
2  (* Add your comment (as per IEC 61131-3) here
3  Normally executed algorithm
4  *)
5  IF CountVal = 0 THEN
6    CountVal := 1;
7    x:=1;
8    All_lights_off := FALSE;
9  ELSIF CountVal <> 0 AND NOT CountVal > 255 THEN
10   CountVal := (x*2)-1;
11  END_IF;
12
13  IF CountVal <= 255 THEN
14   x := x*2;
15  END_IF;
16
17  IF CountVal = 255 THEN
18   All_lights_on := TRUE;
19  END_IF;
20
21  END_ALGORITHM
```

Figur 21 Algoritmen för CountUp inne i Counter_8Out

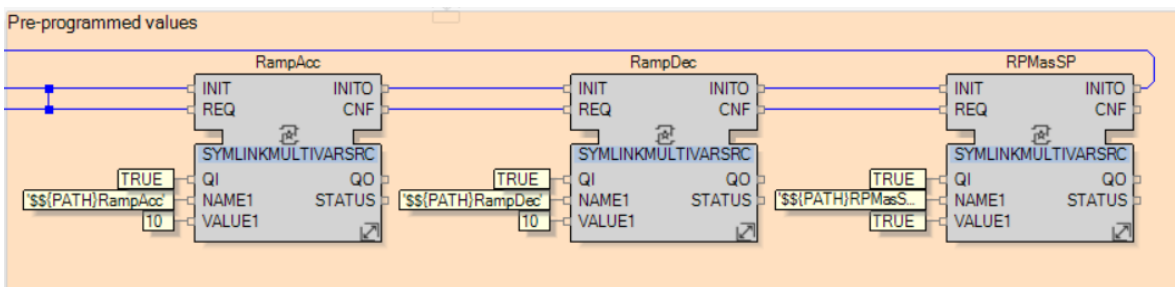
INIT	CountUp	CountDown
1		ALGORITHM CountDown IN ST:
2		(* Add your comment (as per IEC 61131-3) here
3		*)
4		IF CountVal > 0 THEN
5		CountVal := (x/2)-1;
6		END_IF ;
7		
8		IF CountVal < 255 THEN
9		All_lights_on := FALSE ;
10		END_IF ;
11		
12		IF CountVal = 0 THEN
13		All_lights_off := TRUE ;
14		ELSIF CountVal > 0 THEN
15		All_lights_off := FALSE ;
16		END_IF ;
17		
18		x:= x/2;
19		END_ALGORITHM
20		

Figur 22 Algoritmen för CountDown inne i Counter_8Out

5.3 Motorstyrning med frekvensomriktare - ATV340_Controll

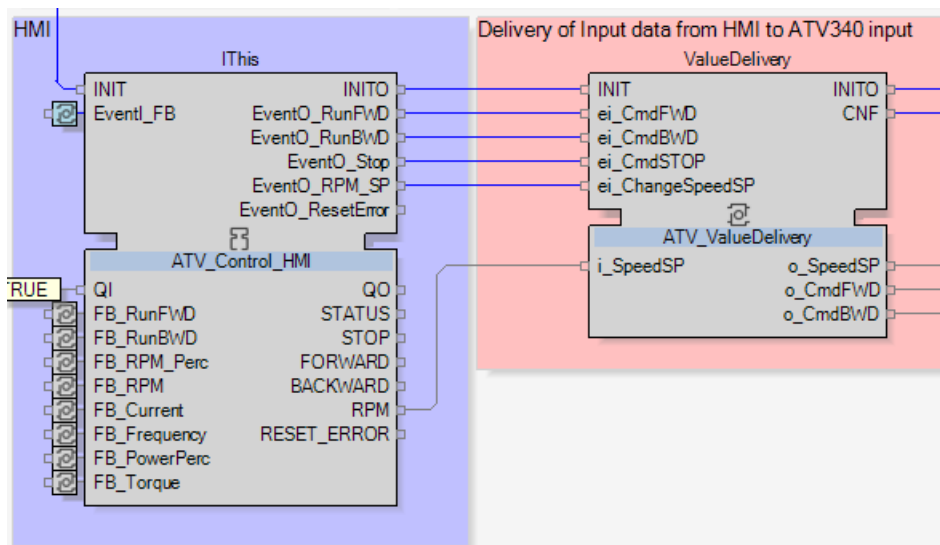
Är det lager i applikationen som programmerar frekvensomriktaren som i sin tur kör motorn. ATV340 är definierad som en ModBus TCP och har en färdig Hårdvaru-CAT installerat i ett bibliotek från Schneider Electric.

Första delen av koden visas i *figur 23* och är att fixerade värden matas in till frekvensomriktaren genom SRC symlink-block. Det är värden så som accelerationsramp och decelerationramp och vilken enhet den inmatade hastigheten ska matas in med.

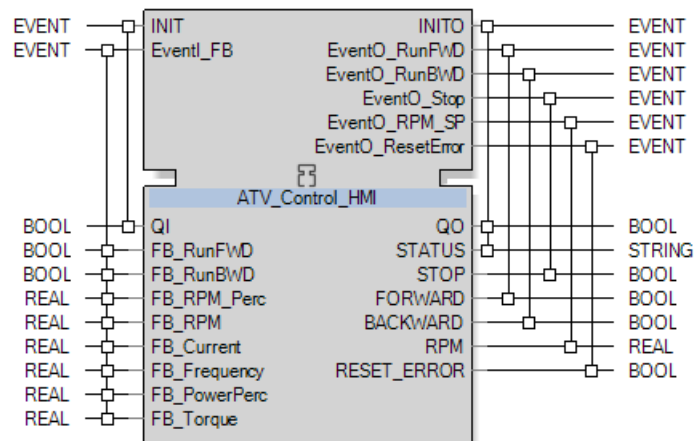


Figur 23 De förbestämda värdena för frekvensomriktaren

Därefter skickas den data som blir inmatad i HMI:et till basic-blocket ValueDelivery som visas i *figur 24* samt vilket av in-eventen som ska triggas. Beroende av vilket in-event som blir triggat hoppar blocket in i respektive tillstånd och algoritmen för detta tillstånd körs. Utgångsdata skickas sedan vidare till tre olika SRC symlink-block som är länkat till ATV340:s hårdvaru-CAT.

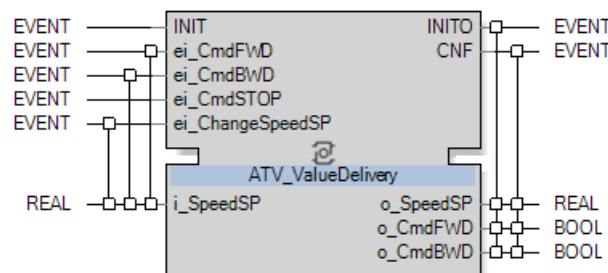


Figur 24 HMI-blockets koppling till ValueDelivery



Figur 25 Vilka ut-event som är länkade till vilka outputs på HMI-blocket

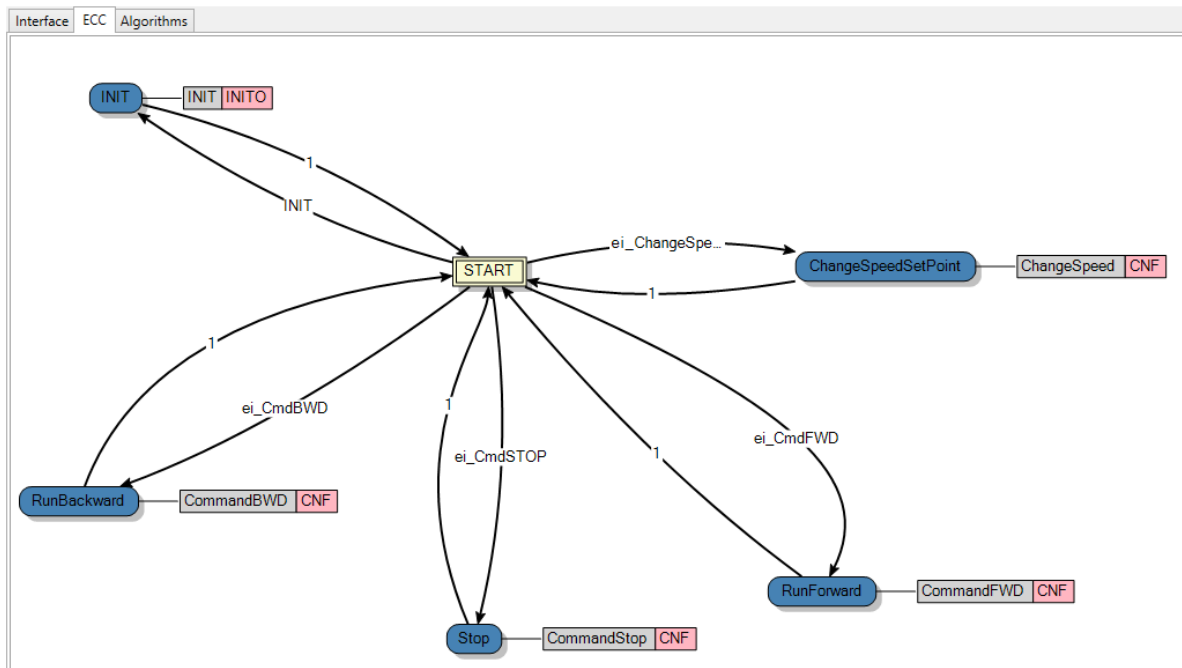
Enligt figur 25 ovan ser vi vilka utdata som är tillgängliga som knappar i HMI:et och som i sin tur är länkade till vars ett ut-event. Det vill säga om knappen för FORWARD slås till så triggas ut-eventet EventO_RunFWD.



Figur 26 ValueDeliverys olika event som är länkade till vilken data

Denna signal går sedan vidare till ValueDelivery och triggas dess in-event ei_CmdFWD, som i sin tur sätter utgångsdata o_CmdFWD till true. Både utgångsdata och event skickas sen vidare till ett SRC Symlink-block som skickar vidare informationen till hårdvaru-CAT.

De olika algortimer som körs i basic-blocket visas i figur 27-31.



Figur 27 ECC för ValueDelivery

```

Interface ECC Algorithms
INIT CommandFWD CommandBWD CommandStop ChangeSpeed
1 ALGORITHM INIT IN ST:
2 (* Add your comment (as per IEC 61131-3) here
3 Initialization algorithm
4 *)
5 o_CmdBWD := FALSE;
6 o_CmdFWD := FALSE;
7 o_SpeedSP := 0.0;
8
9 END_ALGORITHM

```

Figur 28 Algoritmen för INIT-tillståndet

```

Interface ECC Algorithms
INIT CommandFWD CommandBWD CommandStop ChangeSpeed
1 ALGORITHM CommandFWD IN ST:
2 (* Add your comment (as per IEC 61131-3) here
3
4 *)
5 o_CmdBWD := FALSE;
6 o_CmdFWD := TRUE;
7 o_SpeedSP := i_SpeedSP;
8
9 END_ALGORITHM

```

Figur 29 Algoritmen för CommandFWD-tillståndet

```

Interface  ECC  Algorithms
INIT  CommandFWD  CommandBWD  CommandStop  ChangeSpeed
1  ALGORITHM CommandBWD IN ST:
2  (* Add your comment (as per IEC 61131-3) here
3
4  *)
5  o_CmdBWD := TRUE;
6  o_CmdFWD := FALSE;
7  o_SpeedSP := i_SpeedSP;
8
9  END_ALGORITHM

```

Figur 30 Algoritmen för CommandBWD-tillståndet

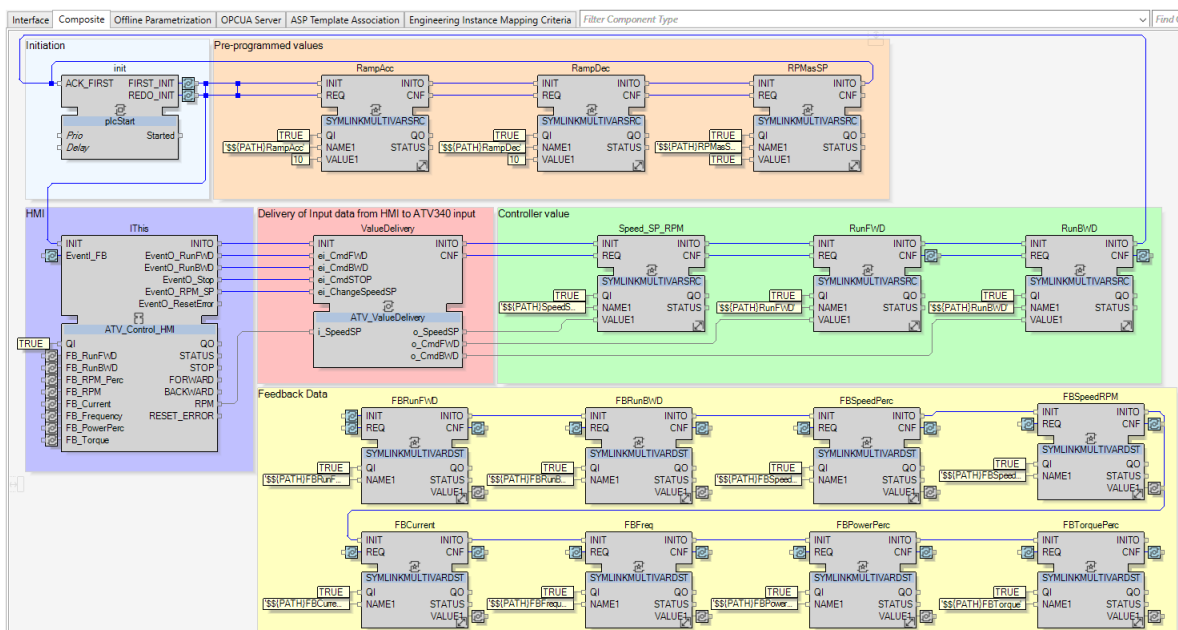
```

Interface  ECC  Algorithms
INIT  CommandFWD  CommandBWD  CommandStop  ChangeSpeed
1  ALGORITHM ChangeSpeed IN ST:
2  (* Add your comment (as per IEC 61131-3) here
3
4  *)
5  o_SpeedSP := i_SpeedSP;
6  END_ALGORITHM

```

Figur 31 Algoritmen för ChangeSpeed-tillståndet

Så fort ett kommando blivit triggat finns där feedback-signaler som skickas från hårdvaru-CAT till ett DST Symlink-block. Därefter skickas den data in i HMI-blocket och presenteras som feedbacksignaler i HMI:et. Hela kodflödet visas i figur 32.

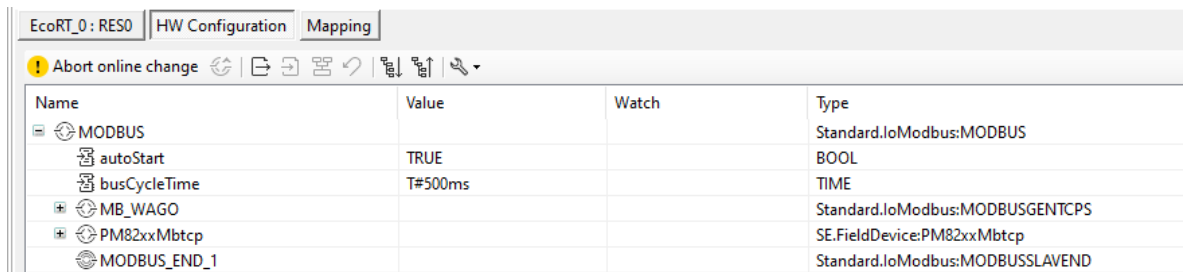


Figur 32 koden inne i ATV_Control

5.4 Datahämtning av energimätare - PM8240

Energiförbrukningsmätaren PM8240 har en färdig HW-CAT och innehåller ingen kod. Den använder i stället den redan implementerade CAT-instansen som finns i ett färdigt bibliotek i mjukvaran.

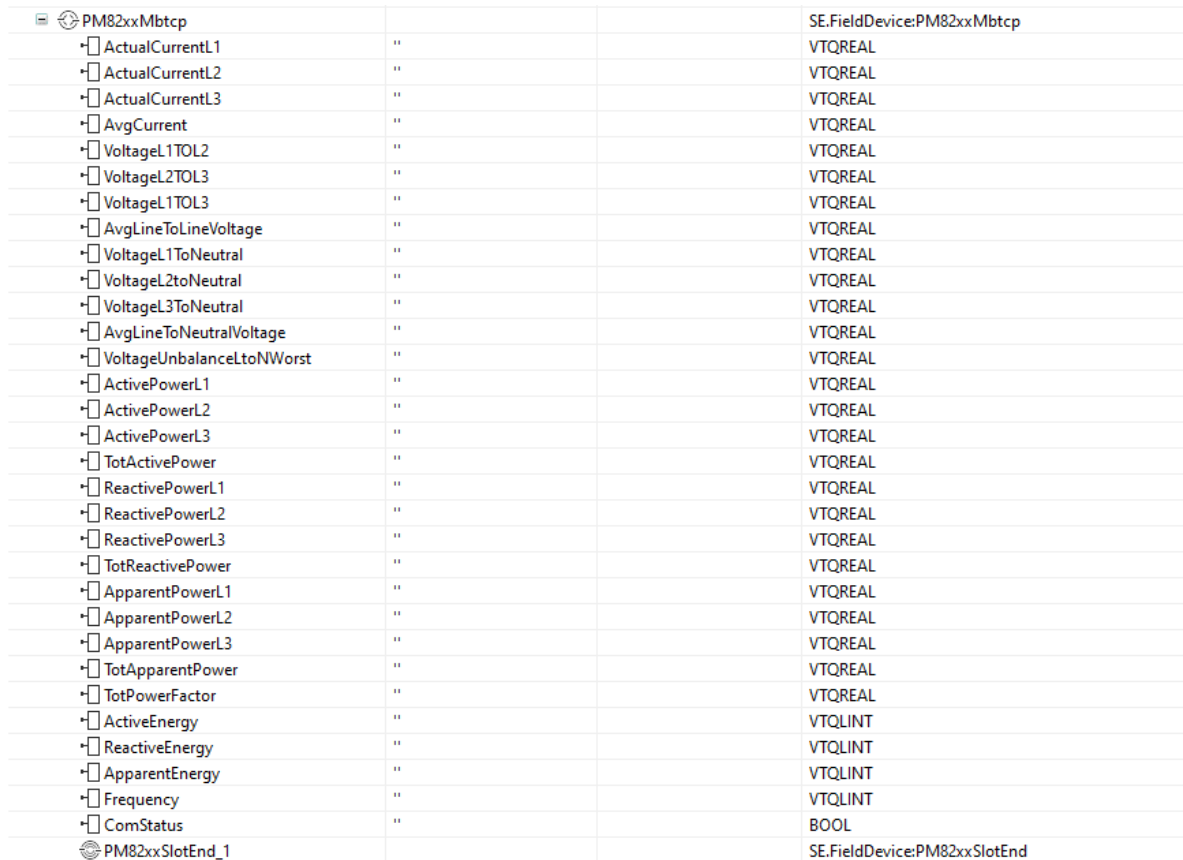
Den är konfigurerad i EcoRT_0 tillsammans med WAGO 750-362. EcoRT_0 är den runtime som körs på Siemens IPC.



Name	Value	Watch	Type
MODBUS			Standard.IoModbus:MODBUS
autoStart	TRUE		BOOL
busCycleTime	T#500ms		TIME
MB_WAGO			Standard.IoModbus:MODBUSGENTCPS
PM82xxMbtcp			SE.FieldDevice:PM82xxMbtcp
MODBUS_END_1			Standard.IoModbus:MODBUSSLAVEND

Figur 33 Siemens IPC hårdvarukonfiguration

I och med att PM8240 är en produkt som arbetar i trefasmiljö finns där många värden som inte är relevanta för denna demoapplikation. Möjliga värden att läsa av från PowerMeter visas i *figur 33* här under.



PM82xxMbtcp			SE.FieldDevice:PM82xxMbtcp
ActualCurrentL1	"		VTQREAL
ActualCurrentL2	"		VTQREAL
ActualCurrentL3	"		VTQREAL
AvgCurrent	"		VTQREAL
VoltageL1TOL2	"		VTQREAL
VoltageL2TOL3	"		VTQREAL
VoltageL1TOL3	"		VTQREAL
AvgLineToLineVoltage	"		VTQREAL
VoltageL1ToNeutral	"		VTQREAL
VoltageL2toNeutral	"		VTQREAL
VoltageL3ToNeutral	"		VTQREAL
AvgLineToNeutralVoltage	"		VTQREAL
VoltageUnbalanceLtoNWorst	"		VTQREAL
ActivePowerL1	"		VTQREAL
ActivePowerL2	"		VTQREAL
ActivePowerL3	"		VTQREAL
TotActivePower	"		VTQREAL
ReactivePowerL1	"		VTQREAL
ReactivePowerL2	"		VTQREAL
ReactivePowerL3	"		VTQREAL
TotReactivePower	"		VTQREAL
ApparentPowerL1	"		VTQREAL
ApparentPowerL2	"		VTQREAL
ApparentPowerL3	"		VTQREAL
TotApparentPower	"		VTQREAL
TotPowerFactor	"		VTQREAL
ActiveEnergy	"		VTQLINT
ReactiveEnergy	"		VTQLINT
ApparentEnergy	"		VTQLINT
Frequency	"		VTQLINT
ComStatus	"		BOOL
PM82xxSlotEnd_1			SE.FieldDevice:PM82xxSlotEnd

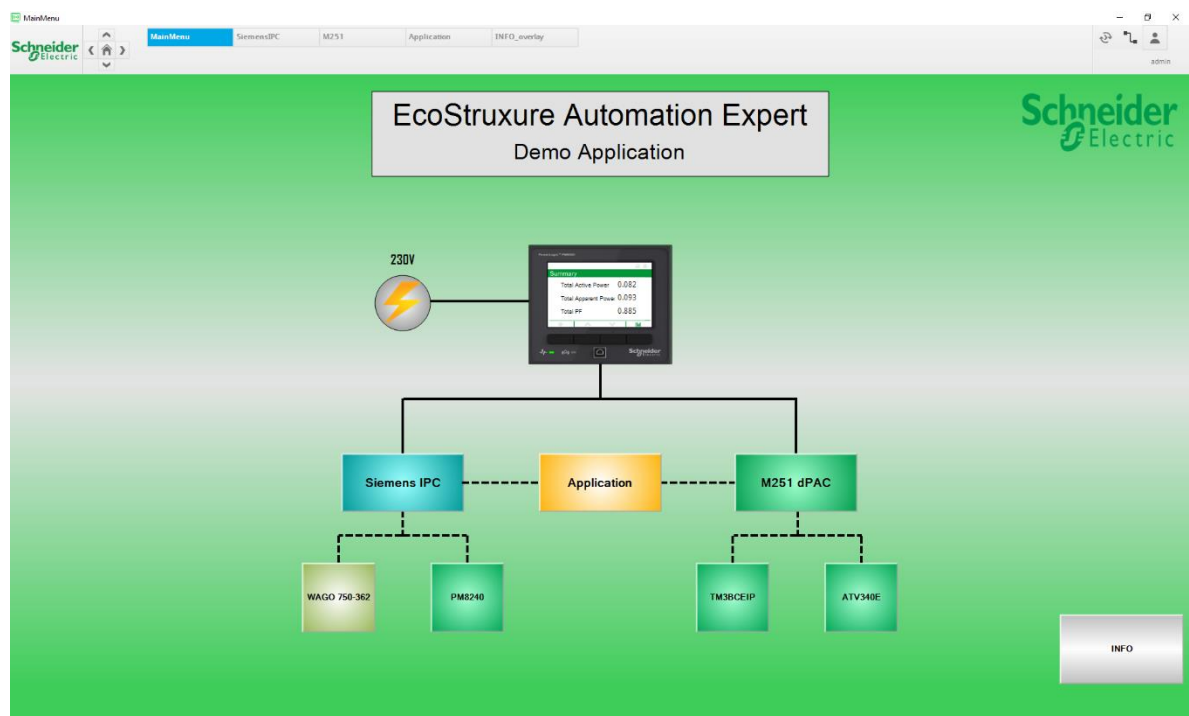
Figur 34 HW-Configuration av PM8240

5.5 Grafiska gränssnittet - HMI

Designen av Human Machine Interface (HMI), det grafiska gränssnittet, innehåller fem olika canvas, Main Menu, Siemens IPC, M251 dPAC, Application och Connection overlay. Alla de olika canvas är designade efter samma struktur, att på ett så enkelt sett visa vilka komponenter som styrs av vilka och hur kommunikationen mellan de olika fungerar.

5.5.1 Canvas - Main Menu

För att göra en så användarvänlig design som möjligt utformades huvudmenyn till att vara översiktlig och intuitiv. Grafiken för huvudmenyn visas i *figur 35* och är det första som visas. Det presenterar samtliga ingående delar, vilka produkter som används, mätning av energiförbrukningen och väldigt förenklat hur de olika komponenterna kommunicerar med varandra. Tillgång till en mer detaljerad översikt finns att se vid klick på INFO nere till höger.

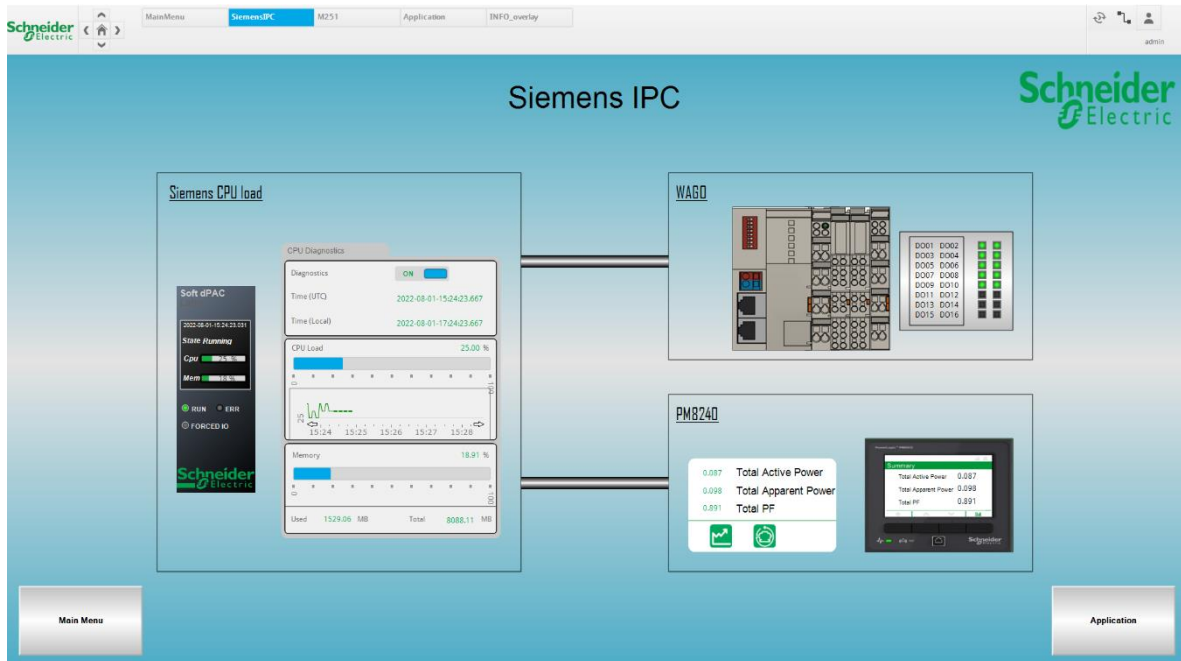


Figur 35 Presentation av huvudmenyn i det grafiska gränssnittet.

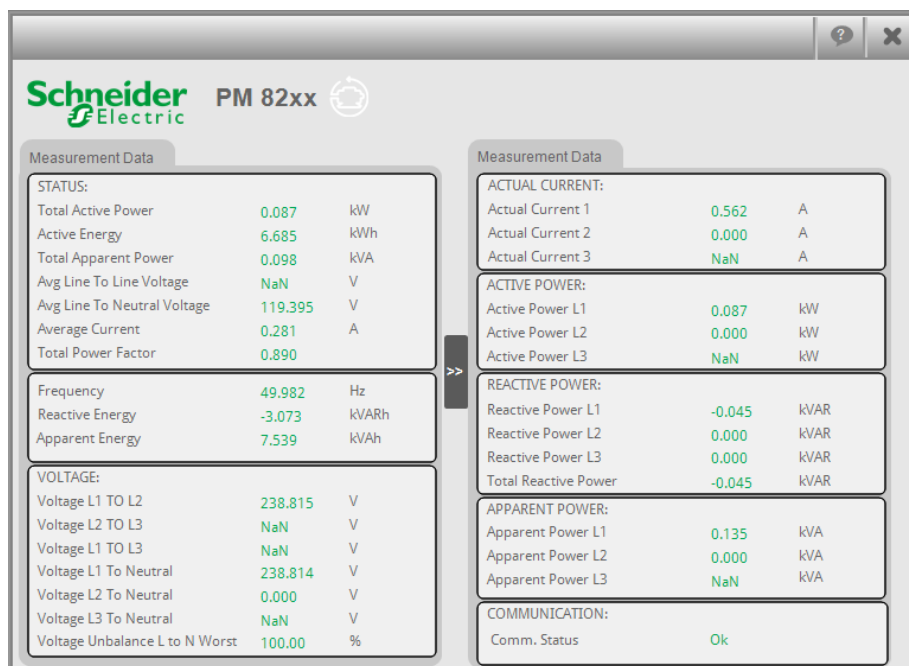
5.5.2 Canvas - Siemens IPC

Siemens IPC är den runtime som kör WAGO:n och PM8240. En presentation av vilken CPU-belastning som används hos Siemensmaskinen och hur de olika komponenterna ser ut i verkligheten presenteras i *figur 36*.

Navigation tillbaka till huvudmenyn är möjlig via knappen ner till vänster och navigation vidare till applikationens canvas görs via knappen ner till höger. El och energiförbrukningen presenteras via en befintlig CAT-instans och visas i *figur 37*.



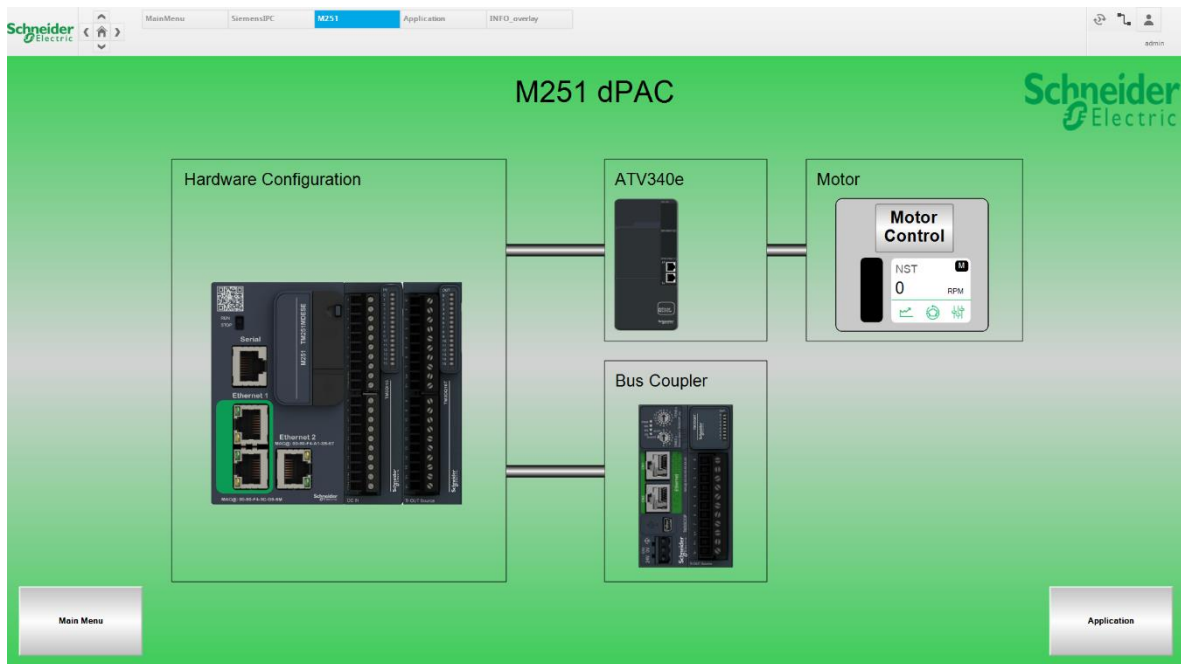
Figur 36 Skärmen som visar vilka komponenter och funktioner som går under Siemens IPC



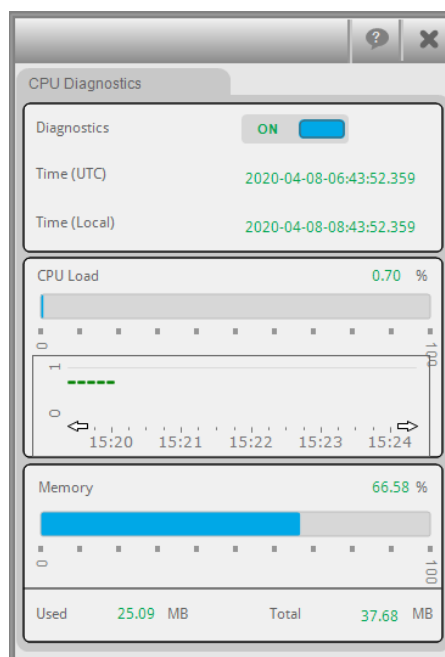
Figur 37 Faceplate för PM8240s HW-CAT.

5.5.3 Canvas - M251 dPAC

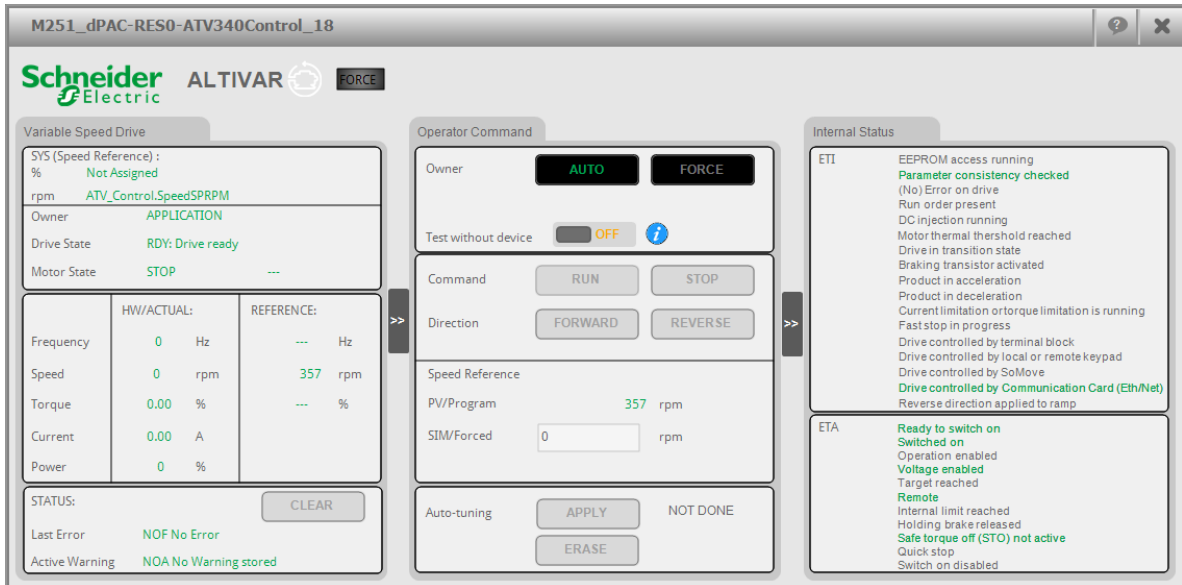
M251 dPAC är den andra runtime som kör resterande komponenter i applikationen. *Figur 38* visar att dessa komponenter är Bus Coupler och ATV340. ATV340 som i sin tur styr hur motorn ska köras. *Figur 39* visar data om CPU diagnostiseringen av PLC:n M251 och fås upp vid tryck på komponenten i Hardware Configuration. Styrning av motorn kan utföras via Schneider Electric's egna CAT-instans som visas i *figur 41* eller av ett egendesignat faceplate som visas i *figur 40*.



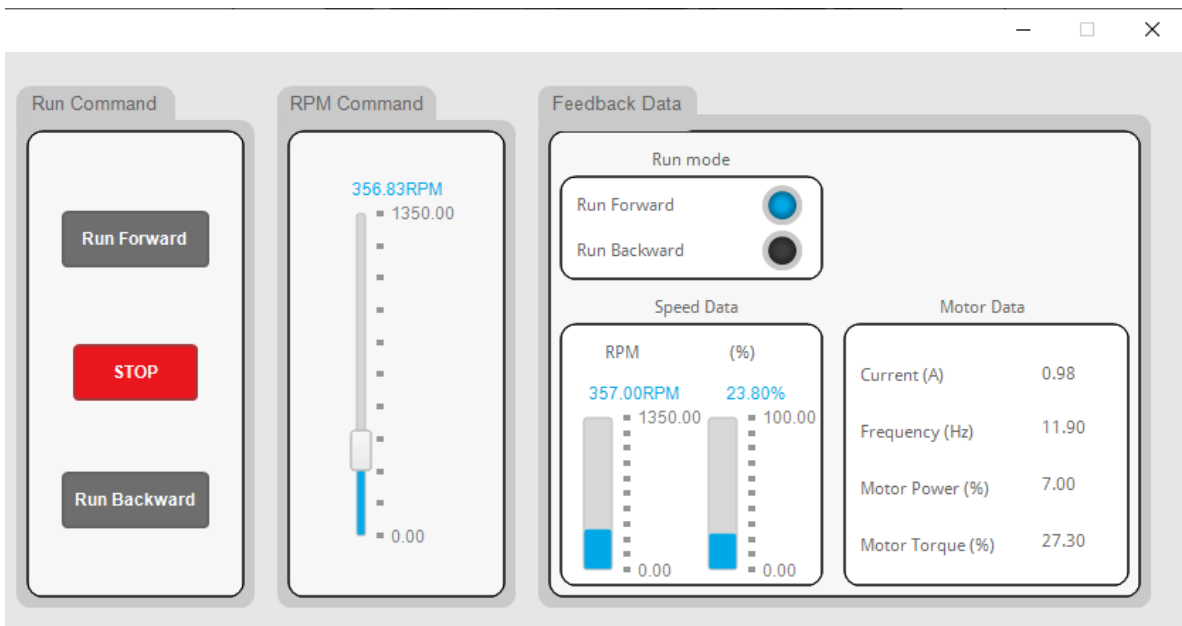
Figur 38 Skärmen som visar vilka komponenter och funktioner som går under M251 dPAC



Figur 39 CPU-diagnostisering av M251



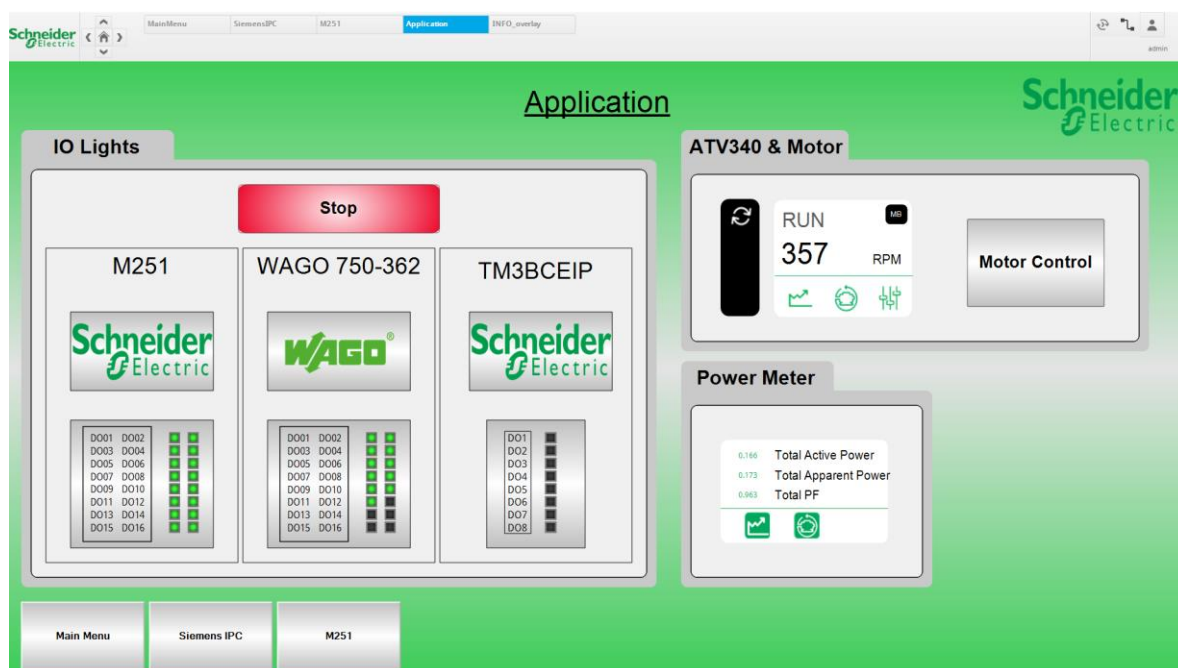
Figur 41 EAE:s egna CAT-instans för ATV340.



Figur 40 Hur den egendesignade faceplaten för ATV340 ser ut.

5.5.4 Canvas - Application

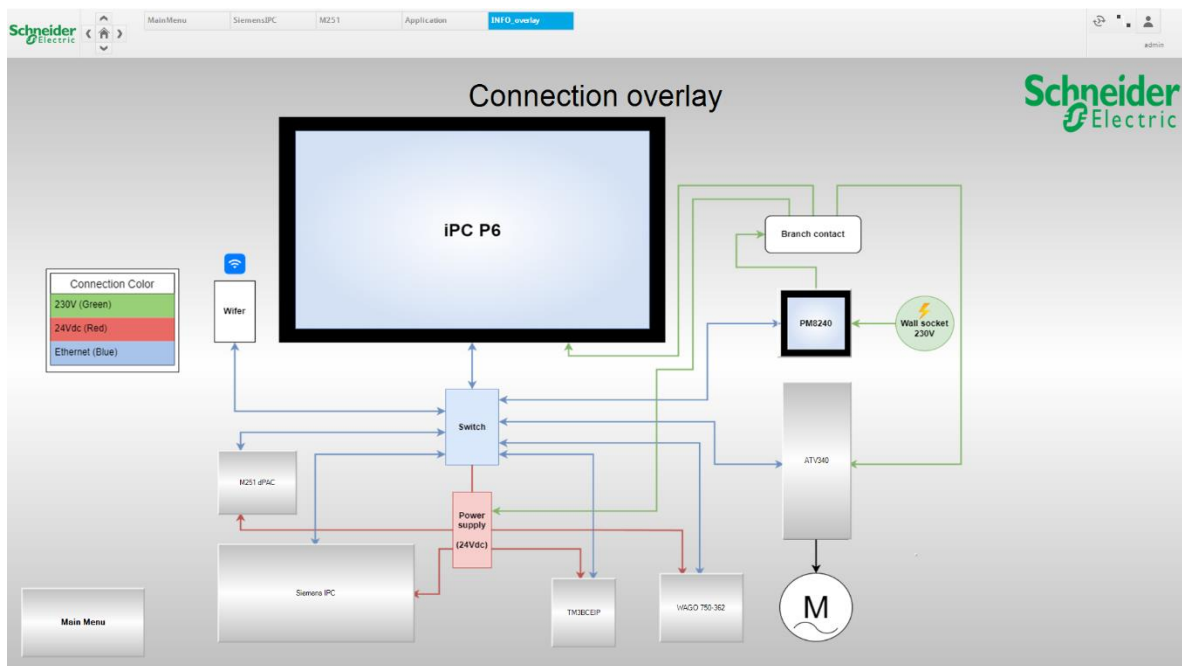
Applikationens alla olika funktioner är samlade i ett och samma canvas och visas i *figur 42*. Den programmerade lampsekvensen för utgångslamporna på de olika komponenterna presenteras i boxen IO Lights. ATV340 & Motor-boxen visar styrning av frekvensomriktaren som i sin tur kör motorn. Styrning kan göras antingen genom en egendesignad faceplate eller en redan befintlig CAT-instans. Presentation av el- och energiförbrukningen i form av realtidsdata nås genom tryck på någon av symbolerna i boxen Power Meter.



Figur 42 Applikationsskärmen som visar när alla funktioner körs

5.5.5 Canvas - Connection overlay

I canvasen Connection overlay visas en översikt på hur de olika kopplingarna mellan komponenterna är utformade. Detta för att få en klar bild av vilka komponenter som ingår samt hur de är sammankopplade. Möjlighet att gå till det canvas som presenterar den runtime som styr komponenten är möjligt från Connection overlay. Resultatet av grafiken visas i *figur 43*.



Figur 43 Skärmen för Connection overlay

6 Slutsats

I detta avsnitt kommer hela slutsatsen för examensarbetet att presenteras där problemformuleringen från avsnitt 1.4 kommer att besvaras. Presentation av hur syftet uppfyllts av resultatet och hur resultatet av examensarbetet kommer till att användas. Reflektion kring de etiska aspekterna kommer att tas upp här samt möjligheten av framtida utveckling av produkten.

En demoapplikation för den nya mjukvaran EcoStruxure Automation Expert (EAE) har programmerats. Funktioner så som motorstyrning med hjälp av en frekvensomriktare, blinkande med utgångslampor i ett visst mönster för tre ingående komponenter och mätning av hela systemets energi- och elförbrukning. Resultatet presenteras i ett egendesignat Human Machine Interface (HMI), som är utformat för att kunna navigeras genom på ett enkelt och översiktligt sätt.

En av problemformuleringarna var att om det kommer till att fungera att implementera EAE på alla komponenter. Något som resultatet visar på att alla komponenter var kompatibla till att implementeras, det enda som stoppade de två uteblivna komponenter var tidsbrist. Vilket leder till nästa problemformulering, vilka komponenter kommer till att väljas. Där stycket i resultatdelen 5.1 går igenom vilken hårdvara som använts vid implementation av demo-applikationen. Ett stycke som även går igenom hur anläggningens elkonstruktion ser ut och blir presenterad i *figur 8* på sidan 22.

Sista problemformuleringen var angående hur HMI:et skulle utformas, om där fanns färdiga objekt att använda sig av eller om egna var tvungna att skapas. Där var det både och, användning av existerande objekt för de olika komponenterna samtidigt som skapande av egna gjordes och täcks i resultatdelens avsnitt 5.5.

Med den informationen är alla fyra olika problemformuleringar besvarade och uppnådda.

6.1 Reflektion kring etiska aspekter

En färdig demoapplikation finns på plats efter att examensarbetet utförts. En demoapplikation som företaget kan använda vid utbildningar och uppvisning för kunder. Detta för att väcka intresse för en mer öppen automatiseringsindustri då mjukvaran kan kommunicera med i princip alla automationskomponenter. Möjligheten att kunna välja produkter från mer än ett företag bidrar till utvecklingen inom automationsindustrin.

Ett program som även kommer att spara tid och pengar för företag som senare kan läggas på nyttigare saker än på fysisk felsökning och omkonstruktion av system.

6.2 Framtida utvecklingsmöjligheter

Utvecklingspotentialen för detta projekt är väldigt stor då fler komponenter kan läggas till enkelt. Koden har beskrivande boxar som samlar de olika funktionsblocken för en viss funktion.

EcoStruxure Automation Expert är i den första utvecklingsfasen det vill säga att fler funktioner kommer att implementeras, något som gör att utvecklingsmöjligheterna är väldigt stora för framtida utveckling. Digital tvilling var något som kom på tal under tiden examensarbetet utfördes och är ännu en implementation som kan göras med hjälp av EcoStruxure Automation Expert.

7 Terminologi

EAE	EcoStruxure Automation Expert
IPC	Industrial PC
HMI	Human Machine Interface
PLC	Programmable Logic Controller
dPAC	Distributed Programmable Automation Controllers
CPU	Central Processing Unit
I/O	Input/output
BFB	Basic Function Block
ECC	Execution Control Chart
CFB	Composite Function Block
SFB	Service Function Block
CAT	Composite Automation Type
ST	Struktureraad Text

8 Källförteckning

[1] E. Tisserant, L. Bessard and M. de Sousa, "An Open Source IEC 61131-3 Integrated Development Environment," 2007 5th IEEE International Conference on Industrial Informatics, 2007, pp. 183-187, doi: 10.1109/INDIN.2007.4384753.

[2] V. Vyatkin "The IEC Standard and its Semantics" 2010 IEEE Industrial Electronics Magazine 3(4), 2010, pp. 40-48, doi: 10.1109/MIE.2009.934796

[3]Schneider Electric(2022), *EcoStruxure™ Automation Expert*.
URL: <https://www.se.com/se/sv/product-range/23643079-ecostruxure-automation-expert/#documents>
(Hämtad: 2022-07-18)

[4]Schneider Electric(2022), *SoMove*.
URL: <https://www.se.com/se/sv/product-range/2714-somove/#overview>
(Hämtad: 2022-07-18)

[5]Schneider Electric(2022), *Automation Device Maintenance*.
URL: <https://www.se.com/se/sv/download/document/EADM/>
(Hämtad: 2022-07-18)

[6]Schneider Electric(2022), *TM3BC IO Configurator*.
URL: <https://www.se.com/se/sv/download/document/EIO0000004112/>
(Hämtad: 2022-07-18)

[7]Schneider Electric(2022), *Harmony P6 IPC* .
URL: <https://www.se.com/se/sv/product-range/22953172-harmony-p6/#documents>
(Hämtad: 2022-07-18)

[8]Siemens Industry Mall(2022), *Simatic IPC 6AG4141-5AB17-0FA0*.
URL:
<https://mall.industry.siemens.com/mall/sv/se/Catalog/Products/10290372?activeTab=ProductInformation>
(Hämtad: 2022-07-18)

[9]Schneider Electric(2022), *Modicon Manage Switch*.
URL: <https://www.se.com/se/sv/product/MCSESM163F23F0/modicon-manage-switch-16-portar-f%C3%B6r-koppar/>
(Hämtad: 2022-07-18)

- [10]Schneider Electric(2022), *Nätaggregat*
URL: https://www.se.com/se/sv/download/document/1569140_01A55/
(Hämtad: 2022-07-18)
- [11]Schneider Electric(2022), *Modicon M251 Logic Controller*.
URL: <https://www.se.com/se/sv/product/TM251MDESE/modicon-m251-dpac/>
(Hämtad: 2022-07-18)
- [12]Schneider Electric(2022), *Modicon PLC IO-module TM3 16 DI*.
URL: <https://www.se.com/se/sv/product/TM3DI16/plc-iomodul-modicon-tm3-16-di-24-v-dc-skruvanslutning/>
(Hämtad: 2022-07-18)
- [13]Schneider Electric(2022), *Modicon PLC IO-module TM3 16 DO*.
URL: <https://www.se.com/se/sv/product/TM3DQ16T/plc-iomodul-modicon-tm3-16-do-trans-source-skruvanslutning/>
(Hämtad: 2022-07-18)
- [14]WAGO(2022), *Fieldbus Coupler ModBus TCP; 4th generation*
URL: <https://www.wago.com/se/i-o-system/faeltbuskopplare-modbus-tcp/p/750-362>
(Hämtad: 2022-07-18)
- [15]Schneider Electric(2022), *PLC IO-module, Modicon, Tm3, Ethernet TM3 Bus Coupler Module*
URL: <https://www.se.com/se/sv/product/TM3BCEIP/plc-iomodul-modicon-tm3-ethernet-tm3-bus-coupler-module/>
(Hämtad: 2022-07-18)
- [16]Schneider Electric(2022) *PLC IO-module, Modicon, TM3, 8 DO Trans Source*.
URL: <https://www.se.com/se/sv/product/TM3DQ8TG/plc-iomodul-modicon-tm3-8-do-trans-source-fj%C3%A4deranslutning/>
(Hämtad: 2022-07-18)
- [17]Schneider Electric(2022), *Variable speed drive, Altivar Machine ATV340, 0.75kW Heavy Duty, 400V, 3 phases, Ethernet*.
URL: <https://www.se.com/se/sv/product/ATV340U07N4E/frekvensomriktare-altivar-340-0-75-kw-2-2-amp-380-480-v-3-fas-ethernet-ip20/>
(Hämtad: 2022-07-18)

[18]Schneider Electric(2022) *PowerLogic PM8000- PM8240 Panel mount meter – intermediate metering.*

URL:

<https://www.se.com/se/sv/product/METSEPM8240/kombinationsinstrument-powerlogic-pm8240-med-35-f%C3%A4rgdisplay-sag-swellregistrering-med-riktningsavk%C3%A4nning/>

(Hämtad: 2022-07-18)

[19]Schneider Electric(2022), *Universal automation Wifi Interface- IP20 – RJ45 and USB connectors.*

URL: <https://www.se.com/se/sv/product/TCSEGWB13FA0/universell-tr%C3%A5dl%C3%B6s-wifi-accesspunkt-ip20-med-rj45-och-usb-anlutning/>

(Hämtad: 2022-07-18)