

Industriellt styrsystem

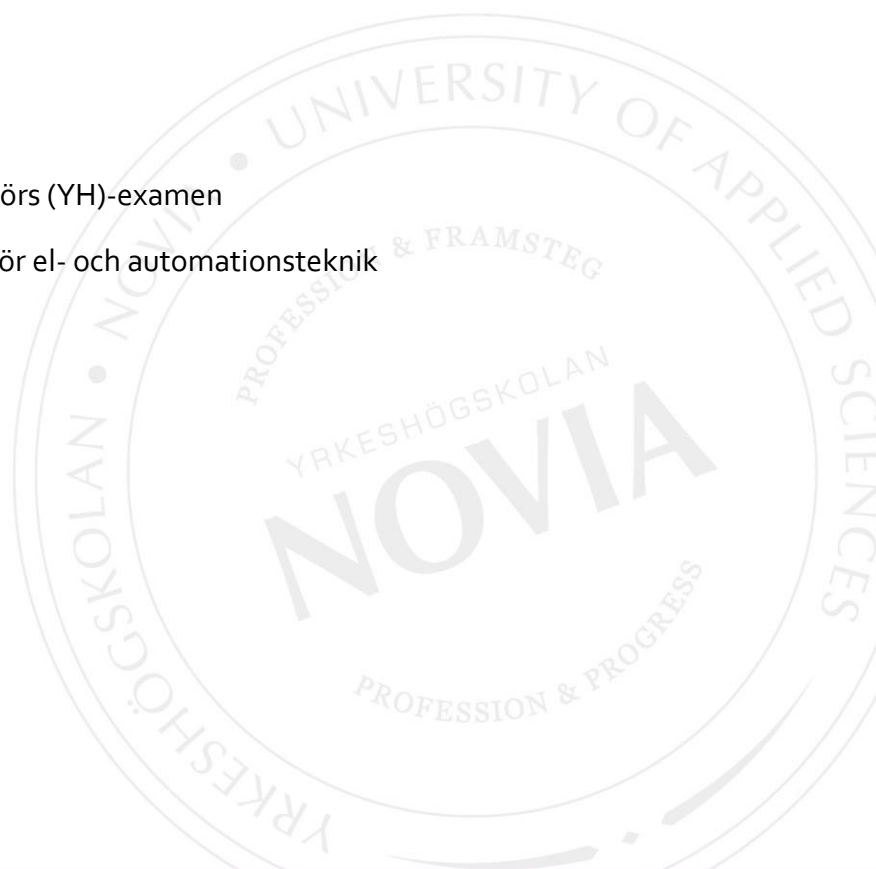
Styrsystem för industriella kompressorer

Mathias Lindström

Examensarbete för ingenjör (YH)-examen

Utbildningsprogrammet för el- och automationsteknik

Vasa 2021



EXAMENSARBETE

Författare: Mathias Lindström

Utbildning och ort: El- och automationsteknik, Vasa

Inriktning: Automationsteknik

Handledare: Matts Nickull

Titel: Industriellt styrsystem

Datum: 19.01.2021

Sidantal: 25

Bilagor: 2

Abstrakt

Detta examensarbete behandlar uppbyggnaden av ett styrsystem för styrning av kompressorer för industriellt bruk. Styrsystemet använder sig av Siemens hårdvara samt mjukvara.

Arbetets syfte var att ersätta ett gammalt föråldrat styrsystem till ett nytt modernare och skapa ett lätt hanterbart program för styrning av det. Utöver det skulle det befintliga styrsystemets uppkopplingar redas ut för att få en överblick av kablaget.

I examensarbetets teoridel behandlas vilken grundkunskap som krävs för att utföra arbetet och även hur en PLC opererar och vad den innehåller. Även hur man skall gå tillväga för att planera samt använda Siemens mjukvara tas upp. I arbetets praktiska del behandlas hur programmet byggs upp och vad som krävs för att få ett väl fungerande program.

Resultatet av arbetet är ett nytt fungerande styrsystem som simulerats och testats på labbrigg. I framtiden kan styrsystemet användas för en bättre funktion och kommunikation med det överliggande systemet.

Språk: svenska

Nyckelord: styrsystem, PLC, kommunikation

BACHELOR'S THESIS

Author: Mathias Lindström

Degree Programme: Electrical Engineering and Automation

Specialization: Automation

Supervisor: Matts Nickull

Title: Industrial Control System

Date: 19.01.2021

Number of pages: 25

Appendices: 2

Abstract

This bachelor's thesis deals with the construction of a control system for controlling compressors for industrial use. The control system uses hardware and software from Siemens.

The purpose of the thesis is to replace an old obsolete control system with a new modern one and create an easily manageable program for controlling it. In addition, the connections of the existing control system would be sorted out to get an overview of the cabling.

The theoretical part of the thesis examines the basic knowledge required to carry out the work and how a PLC operates and what it contains. It also deals with how to proceed to plan and use Siemens software. The practical part of the thesis deals with how the program is built and what is required to have a well-functioning program.

The result of the thesis is a new functional control system that has been simulated and tested on lab rigs. In the future, the control system can be used for a better function and communication with the overlying system.

Language: Swedish

Key words: control system, PLC, communication

Innehållsförteckning

Bilagor	1
Förkortningar	1
1 Inledning.....	1
1.1 Uppdragsgivare.....	1
1.2 Projekt	2
2 Helhetsbeskrivning.....	3
2.1 Jämförelse av PLC och programmerbart relä	4
3 Projektplanering.....	5
3.1 Funktionsspecifikation.....	5
3.2 Tidsplan.....	6
4 Styrsystem.....	7
4.1 PLC	7
4.1.1 Uppgifter hos en CPU	7
4.1.2 Hur opererar en PLC.....	8
4.2 Simatic Manager	8
4.2.1 OB.....	9
4.2.2 FB.....	9
4.2.3 FC	9
4.2.4 DB.....	10
4.3 Hårdvara.....	10
4.3.1 Simatic S7-ET200S (SIMATIC DP, IM151-8 PN/DP CPU f. ET200S).....	11
4.3.2 Digitala in- och utgångskort	12
5 Programmets uppbyggnad	13
5.1 Skapa projekt och hårdvara	13
5.2 Skapandet av programmets delar	15
5.3 Timerfunktion.....	17
5.4 Slutliga funktionen.....	18
6 Testning och simulering	20
7 Inkoppling av styrsystem.....	21
7.1 Inkoppling av styrcentral.....	21
7.2 Inkoppling av kompressorenheter	22
8 Resultat	23
9 Diskussion.....	24
10 Litteraturförteckning	25

Bilagor

Bilaga 1 SIMATIC DP, IM151-8 PN/DP CPU f. ET200S

Bilaga 2 Digital electronic module 8DO DC24V/0.5

Förkortningar

PLC	Programmable Logic Controller
I/O	In- och utgångar
OB	Organization Block
FB	Function Block
FC	Function
DB	Data Block
FBD	Function Block Diagram
LAD	Ladder
STL	Statement List
CPU	Central Processing Unit

1 Inledning

Den industriella digitaliseringens framfart har blivit en stor faktor för en stor del av företag och inte mindre för befolkningen världen runt. Den ökade automatiseringen har höjt kraven för industrier och maskiner inom den specifika industrin. Redan vid projektering måste det bestämmas styrsätt, graden av automation och på vilket sätt maskinerna skall integreras till styrsystemet. (Haag, 2011)

Detta examensarbete är gjort åt Kingspan Group. Arbetet handlar om att skapa ett styrsystem för deras kompressorer i fabriken. Arbetet består av en teoretisk del som behandlar planering, programvara och komponenter som används i arbetet. Den praktiska delen behandlar programmets uppbyggnad och funktionalitet. För arbetet användes Siemens Simatic Manager.

1.1 Uppdragsgivare

Uppdragsgivare för detta examensarbete är Paroc Panel System Oy Ab som ägs av Kingspan Group. Det är ett företag som befinner sig i södra Finland, Pargas. Paroc Panel System producerar högklassiga så kallade sandwichelement med ytskikt av stålplåt och innehåller Paroc stenull. Sandwichelementen används i huvudsak till fasader, innerväggar och innertak i kommersiella, industriella, kontors- och offentliga byggnader.

Paroc Panel System tidigare ägt av Paroc Oy Ab, såldes till Kingspan Group i december 2016. Kingspan Group är världsledande inom isolerings- och byggnadsmaterial för högpresterande lösningar med låga koldioxidutsläpp. Med över 130 tillverkningsanläggningar i mer än 60 länder över hela världen strävar de efter hållbara, innovativa och högkvalitativa produkter, system och tjänster för att ge bättre byggresultat för sina kunder. (Kingspan, 2020) (ParocPanels, 2020)

1.2 Projekt

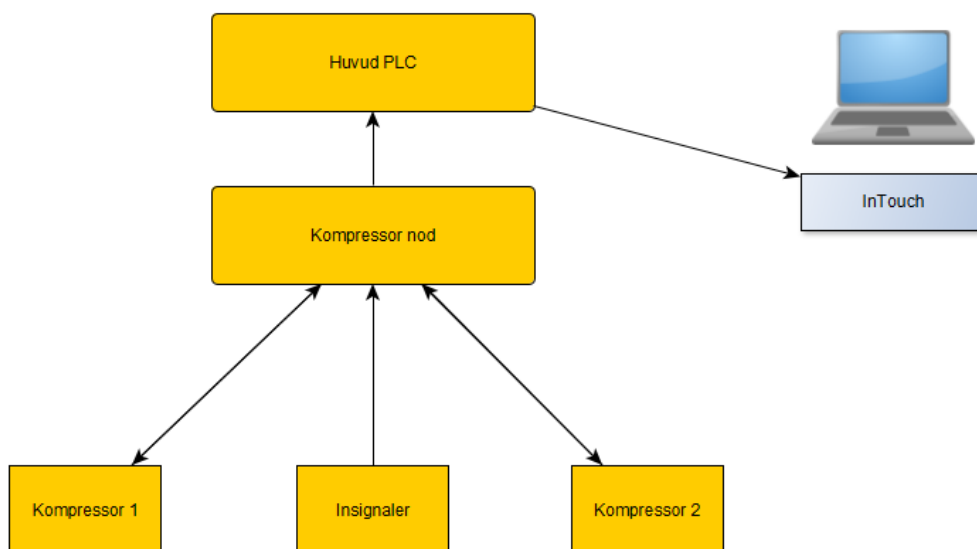
Detta projekt består av två delar. Den första delen baserar sig på teorin till PLC-styrssystem som används i projektet, planering av arbetet och utgångspunkterna vid arbetets startpunkt. Där kommer det gamla styrsystemet att behandlas och varför det behövdes bytas ut, funktionsspecifikationen för det nya styrsystemet samt PLC hårdvara och mjukvara.

Den andra delen består av programplanering, uppbyggnad av programkod samt testning och simulering. Där kommer den teoretiska delen behandlas och hur den användes för att skapa programmet till styrsystemet och hur slutresultatet av programmet blev. Slutligen behandlas det hur testning och simulering utförts samt resultat av funktionaliteten hos styrsystemet med en sammanfattning och diskussionsdel.

2 Helhetsbeskrivning

Fabrikens två kompressorer hade vid planering av examensarbetet inget fungerande styrsystem, utan de styrdes manuellt på plats via tryckknappar och en inbyggd operatörspanel. I källarutrymmet där kompressorerna befinner sig, finns ett gammalt styrsystem som använts tidigare och är baserat på reläteknik, ett programmerbart logikreläsystem. Eftersom kompressorerna bytts ut till nyare modeller och det befintliga styrsystemet inte längre är i bruk på grund av föråldrad programvara, finns det ingen kommunikation med det överliggande styrsystemet. Därför är det av stort behov att få ett nytt fungerande styrsystem som möjliggör kommunikation med det överliggande styrsystemet.

I dagsläge är operatörspersonen eller underhållspersonen tvungen att vara på plats i kompressorutrymmet för att operera kompressorerna. Ett fungerande styrsystem för kompressorerna resulterar i en förbättrad kommunikation från det överliggande styrsystemet. Det underlättar operatören med att identifiera och åtgärda eventuella problem som uppstått hos kompressorerna samt få en jämn belastning på dem.



Figur 1. Blockschema över styrsystemets kommunikation.

2.1 Jämförelse av PLC och programmerbart relä

Det finns fler stora fördelar med PLC-styrning jämfört med ett programmerbart relä. De största skillnaderna är att ett PLC-styrsystem är modulärt medan det programmerbara relät inte är det. Det betyder att det gamla styrsystemet hade begränsat med in- och utgångar medan det nya är mer flexibelt för modifieringar. En annan stor fördel som bör tas upp är att ett PLC har ett märkbart större minne, i form av ett minneskort som gör det mer användarvänligt för lagring av program. (Akytec.de, 2019)

Men detta gör också att ett PLC-styrsystem är dyrare att införskaffa jämfört med ett programmerbart relä. Detta leder till att användaren bör fundera över eventuella behov och kompatibilitet för styrsystemet under projekteringen för att uppnå förväntat resultat och budgetmål.

3 Projektplanering

Projekt har funnits i alla tider, och ett projekt strävar till att uppnå vissa mål och resultat inom en viss angiven tid. En bra planering av ett projekt betyder att projektet är lättare att genomföra. Planeringen för detta projekt utfördes i form av funktionsspecifikation för att klargöra funktioner mellan kund och utförare. Utöver det har en tidsplan utförts för att uppskatta hur projektet skall framskrida.

3.1 Funktionsspecifikation

För att bäst utföra ett projekt i liten eller stor omfattning är det behövligt att börja med att skapa en funktionsspecifikation. Det görs för att kunder och arbetsutföraren skall vara medvetna om hur slutresultatet skall se ut och undvika missförstånd. Eftersom detta projekt inte utfördes som ett officiellt kundarbete, gjordes det en enklare variant av funktionsspecifikationen för att fastställa slutresultatet och förenkla arbetets tillvägagångssätt.

Följande krav och behov har beaktats under planeringen av funktionsspecifikationen.

- Kompressorerna skall köras turvis för att minska på belastningen på ett längre tidsperspektiv. Tiden som en kompressor skall operera per gång skall kollas upp och optimeras till mellan 1 – 2 dygn och det skall vara möjligt att ändras enkelt i programmet.
- När körtiden för ena kompressorn tar slut och byte skall ske, skall den ha en fördröjning på 20 – 30 minuter för att den andra kompressorn skall ha tid att komma upp i varv och bli varm.
- Om lufttrycket i fabriken sjunker pga. stort förbruk av tryckluft, skall den vilande kompressorn startas för att försörja det stora lufttrycksbehovet.
- När en kompressor startar för att hjälpa till vid större lufttrycksbehov, skall den ha en minimum-körtid på 30 minuter. Detta för att kompressorn skall kunna köra upp skapligt och hinna köras varm för att minska på slitage.

- För varje enskild kompressor skall det finnas en två-lägesbrytare för automatläge och serviceläge. Om den körande kompressorn ändras från automat-läge till service-läge skall den stängas av direkt och den andra kompressorn startas. När den sedan ändras tillbaka till automat-läge skall det fortsätta normalt enligt programmet.
- Fjärrstyrningsstart och stopp skall beaktas i programmering av programmet för framtida uppbyggnad och implementering i det överliggande styrsystemet.

3.2 Tidsplan

För att ett projekt skall kunna utvecklas och slutföras enligt önskemål behöver man lägga upp en tidsplan. Det gjordes också för detta projekt. Uppdragsgivaren gav en tid för när projektet kunde påbörjas och de beställde komponenter som behövdes för projektet till den utsatta tiden.

Som tidsplan för projektet lades sommarmånaderna under 2019, börjande från juni och avslutande augusti månad.

4 Styrssystem

Ett styrsystem är en enhet eller en uppsättning av flera enheter som hanterar kommandon, leder eller reglerar andra enheter eller systems sätt att agera. Industriella styrsystem används inom den industriella produktionen för att styra utrustning eller maskiner. Detta kräver att operatören har en god uppfattning över teorin och grunderna till ett styrsystem. (Elektrologik, 2020)

För att kunna förverkliga styrsystemet i projektet behövs en CPU, in- och utgångsmoduler, strömkälla och kommunikationsmodul.

4.1 PLC

En programmerbar logisk styrenhet är ett industriellt datorkontrollsystem som kontinuerligt övervakar ingångsenheternas tillstånd och fattar beslut baserat på ett anpassat program för att styra tillståndet på utgångsdata till enheter. (Siemens, Siemens, 2020)

Nästan alla produktionslinjer, maskinfunktioner eller processer kan förbättras kraftigt med denna typ av styrsystem. Den största fördelen med att använda en PLC är dock förmågan att ändra och replikera operationen eller processen medan man samlar in och kommunicerar viktig information.

En annan fördel med ett PLC-styrsystem är att det är modulärt. Det vill säga, man kan mixa och matcha de typer av in- och utmatningsenheter som bäst passar användarens applikation.

4.1.1 Uppgifter hos en CPU

En PLC innehåller en CPU som innehåller ett internt program vilket berättar för PLC hur det skall utföra olika funktioner. De funktionerna kan vara t.ex.

- Utföra kontrollinstruktionerna i användarens program. Det programmet lagras i ett "icke-flyktigt" minne, vilket innebär att programmet inte går förlorat om strömmen kopplas bort.
- Kommunicera med andra enheter, som kan inkludera I/O enheter, programmeringsenheter, nätverk eller till och med andra PLC-enheter.

4.1.2 Hur opererar en PLC

Det finns fyra grundläggande steg i driften av alla PLC-enheter. Skanning av systemet och kopplade ingångar, programmet, utgångar och kommunikation. Dessa steg sker kontinuerligt i en upprepande loop. (Plcdev.com, 2020)

1. Start skanning av system.

Detta steg innehåller kommunikation med programmeringsterminalerna och intern diagnostik mm.

2. Kontrollerar tillståndet för alla ingångsenheter som är anslutna.
3. Utför den användarskapade programlogiken.
4. Aktiverar eller avaktiverar alla utgångsenheter som är anslutna enligt programlogiken.
5. Startar om loopen.

4.2 Simatic Manager

Simatic Manager, Step 7 är programvaran för programmering och konfigurering av Siemens utrustning som tillhör S7-300 och 400 serien. Det är alltså i denna programvara som all hårdvarukonfigurering och programkod kommer att slutföras inom. Det är ett av världens mest kända programmeringsverktyg inom industriell automation. I dagens läge är dock Siemens nya programvara vid namnet TIA Portal mer använt vid nya installationer eftersom det är ett nytt uppdaterat programmeringsverktyg. Dock används fortfarande Simatic Manager vid en stor del av industrier runtom i världen. (Siemens, Siemens, 2020)

För att utföra programmering av ett styrsystem i Siemens programvara, krävs kunskap inom olika typer av programblock. Det finns olika typer av programblock för att uppnå önskat resultat. Dessa programblock är färdigt inbyggda i mjukvaran och kan konfigureras och programmeras enligt behov. Programblocken används i både Simatic Manager och TIA Portal med samma funktioner.

4.2.1 OB

Organisationsblock representerar gränssnittet mellan operativsystemet och användarprogrammet. Det anropas av operativsystemet och kontrollerar cykliskt programmet samt avbryter programkörningen vid fel. Utöver det hanterar det också uppstart av automationssystemet och sköter om felhantering.

För att optimera styrsystemet enligt egna förväntade behov går det att använda sig av olika organisationsblock. För optimering finns det olika funktioner i organisationsblocken, t.ex. uppstart-, tidsfel-, hårdvarufel- eller synkroniseringsfel. Dessa olika organisationsblock har individuella prioriteringsgrader och beroende på prioriteringsgrad avgörs vilket alarm eller alarmgrad som kommer att påverkas. Detta betyder att vissa organisationsblock kan avbryta hela processen medan andra endast ger ett alarm. (Berger, 2012)

4.2.2 FB

Siemens använder termen Function Block för programrutiner som kan ha ett internt minne. Dessa block är delar av programmet vars anrop kan programmeras via blockparametrar. De har ett variabelminne som lagras i ett instansdatablock. Detta datablock är permanent tilldelat till det funktionsblocket som anropar det och inget annat kan påverka dess data. (Berger, 2012)

Fördelen med funktionsblock är att man kan programmera en programkod som kan komma att behöva återanvändas i flertal tillfällen, som t.ex. styrning för transportörmotorer. Detta möjliggör användning av samma programmerade funktionsblock på t.ex. alla transportörer i en produktionslinje men med anpassade givare och inställningar för varje transportör. Varje enskilt anrop av funktionsblocket skapar ett individuellt instansblock, vilket gör att transportörerna inte är sammankopplade. Denna metod sparar tid samt underlättar förståelse för programmet.

4.2.3 FC

En funktion används för att programmera återkommande eller komplexa automatiseringsfunktioner. De kan tilldelas parametrar och returnera ett värde som kallas för funktionsvärde till det anropade blocket. Funktioner lagrar inte information och har inget tilldelat datablock. Det betyder att de temporära variablerna är bortglömda efter att funktionen har utförts. (Berger, 2012)

4.2.4 DB

Datablock innehåller programmets variabeldata. För programmering av datablocken bestämmer användaren i vilken form data kommer att sparas, t.ex. boolean, integer, word osv.

Det finns två sätt att använda datablock på, som globala datablock eller som instans datablock. Globala datablock lagrar data som kan användas av alla block. Ett instans datablock är däremot bundet till ett funktionsblock och lagrar delar av det funktionsblockets lokala data. (Berger, 2012)

4.3 Hårdvara

Hårdvaror innebär i huvudsak ett kretskort, integrerade kretsar samt andra elektriska komponenter. Det innebär också hållbarhet och oföränderlighet i flertal fall, till skillnad från mjukvara som enkelt kan varieras. Användaren kan enkelt sätta ett helt nytt program i en hårdvara och göra så att den skapar en helt ny upplevelse för användaren. (Elsip, 2017)

Olika hårdvaror har olika egenskaper som t.ex. processor eller minneskrets. Detta leder till att val av hårdvara kan vara kritiskt för att uppfylla styrsystemets behov enligt funktionsspecifikationen. Utöver det behöver hårdvaran vara kompatibel med det överliggande styrsystemet.

4.3.1 Simatic S7-ET200S (SIMATIC DP, IM151-8 PN/DP CPU f. ET200S)

Siemens Simatic S7-ET200S är en styrenhet av modulär design, vilket betyder att användaren själv kan designa modulen med olika sorters signalmoduler eller signalkort för att uppfylla styrsystemets behov. Signalkorten kan vara av digitala eller analoga in-, och utgångstyper vilket ger en flexibilitet för användaren. (Siemens, Siemens, 2020)



Figur 2. Simatic S7-200S. (Siemens, Siemens, 2020)

4.3.2 Digitala in- och utgångskort

In- och utgångskort (I/O), är enheterna som tar emot och skickar ut signaler i styrsystemet. De sammankopplar enheter som t.ex. givaren eller maskiner med själva PLC-styrsystemet.

Ett ingångskort känner av status för ingångssignaler som givare, tryckknappar eller strömbrytaren. Utgångskort styr däremot styrenheter såsom reläer, motorstartare eller lampor. (Automation.com, 2018)



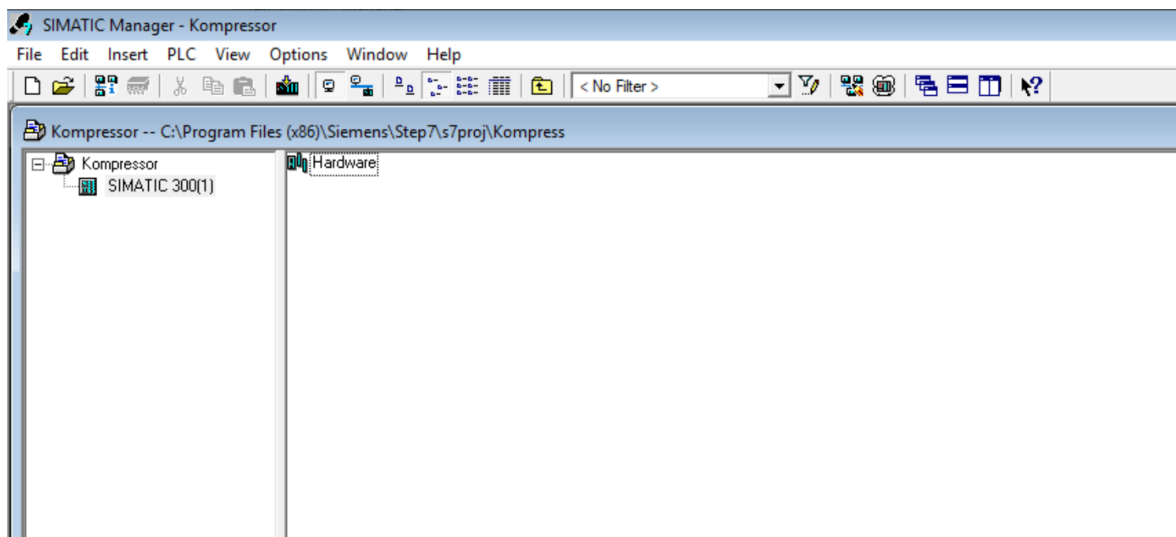
Figur 3. Siemens in- utgångs kort. (Siemens, Siemens, 2020)

5 Programmets uppbyggnad

Programmets uppbyggnad är arbetsuppgiften som är mest tidskrävande. Det beror på att allt som planerades i funktions-specifikationen skall implementeras och beaktas i programmet. Programmet skall också skapas på ett sätt så att andra användaren enkelt ska klara av att förstå och underhålla det. För att enklast komma igång med uppbyggnaden av programmet finns det manualer från Siemens som rekommenderas att följa för att uppnå förväntat resultat.

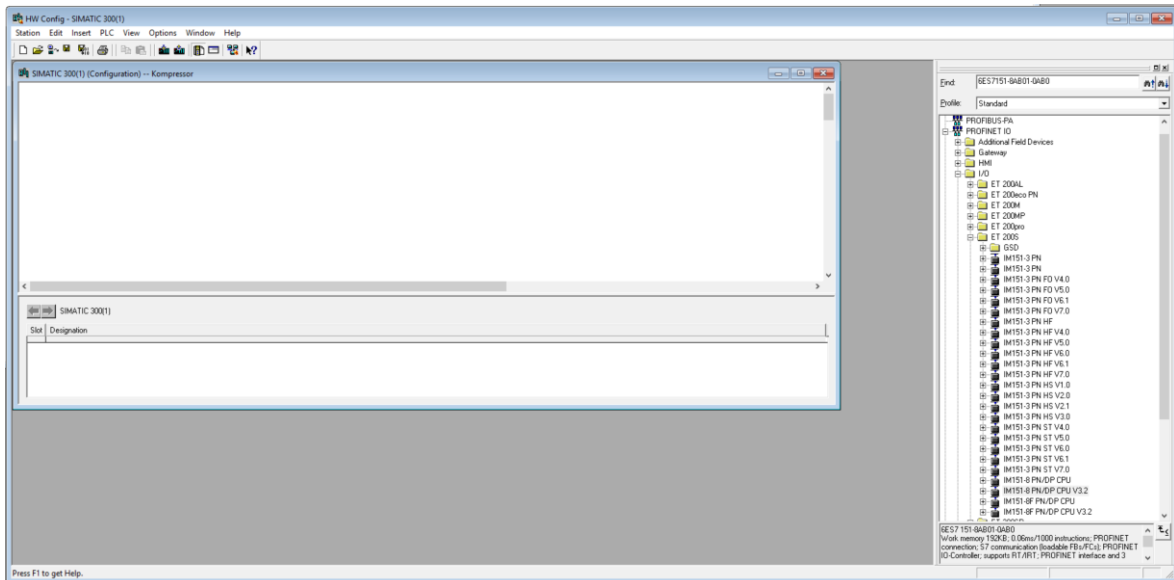
5.1 Skapa projekt och hårdvara

Att skapa ett nytt projekt är väldigt enkelt och användarvänligt. Det enda som krävs är att välja var projektet skall sparas och ge det ett namn. Efter det behöver man ha information på vad för PLC modell man kommer att använda för projektet. I figur 4. kan man se att ett projekt har skapats med namnet ”Kompressor” och PLC-modellen hör till Siemens 300-serien.



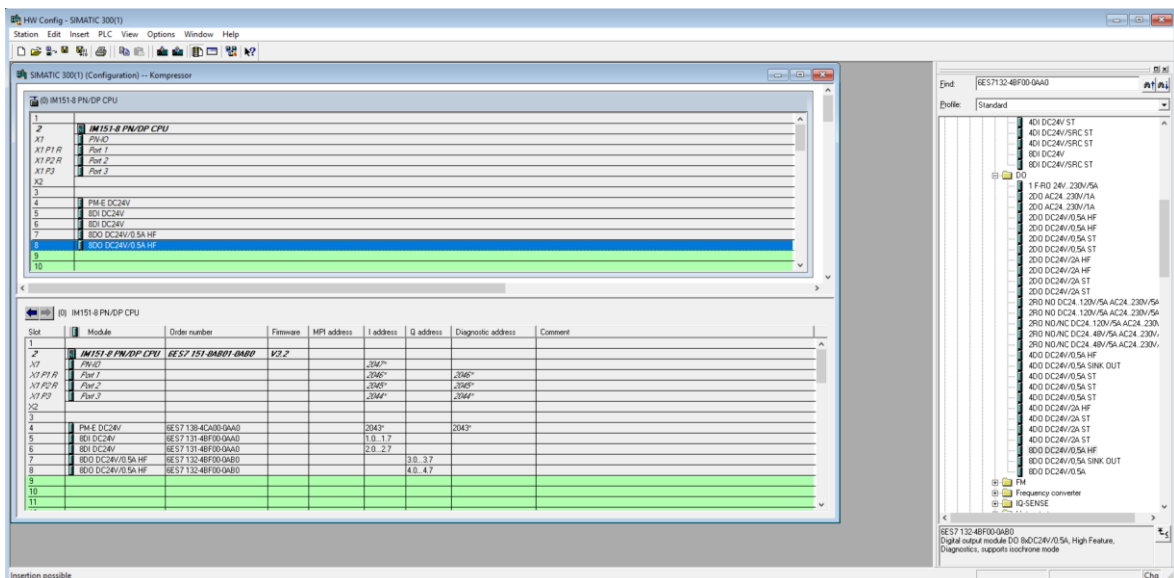
Figur 4. Skapandet av programmets hårdvara.

För att sedan lägga in hårdvara och konfigurera den, behöver man veta t.ex. vilka in- och utgångskort som kommer att användas. Det gör genom att gå in på ikonen ”Hardware” som syns i figur 4.



Figur 5. Välj CPU-modell för hårdvara.

Väl inne på hårdvarans konfiguration börjar man med att välja modellen för vilken CPU som kommer att användas under projektet. Det görs genom att leta upp den i högra katalogdelen som syns i figur 5. Ett annat sätt att utföra det på, är genom att skriva in produktnumret i sökfliken som sedan letar upp det per automatik.



Figur 6. Val av korttyper som används.

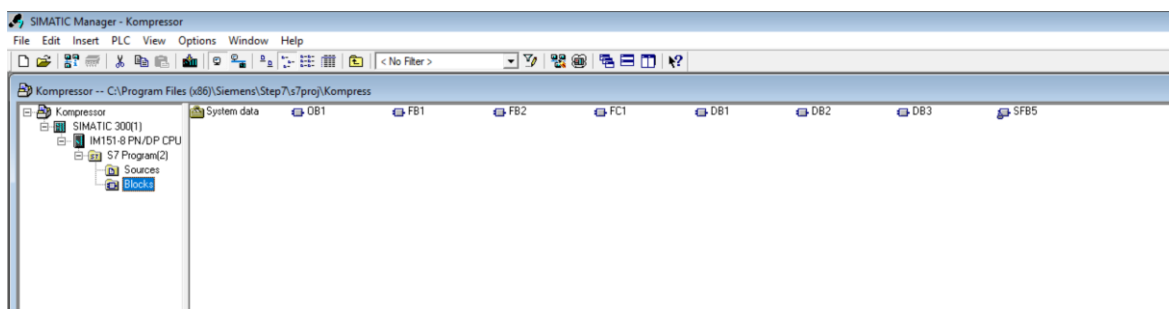
Det sista som krävs under hårdvarukonfigurationen är att välja och lägga in de korttyper som kommer att användas. Det görs på samma sätt som val av CPU modellen, genom att i högra katalogdelen leta upp det rätta kortet som sedan genom "drag and drop" metoden

placeras i hårdvaran. Exempel på det kan man se i figur 6. Det är viktigt att man placerar dem i den ordning som de i verkligheten är monterade i modulen.

I samband med konfiguration av in- och utgångskort behöver man också tilldela varje enskilt kort vilken in- och utgångsminnesposition de kommer att använda sig av i programmet. Det är viktigt att kolla upp så inget annat kort använder sig av samma minnespositioner.

5.2 Skapandet av programmets delar

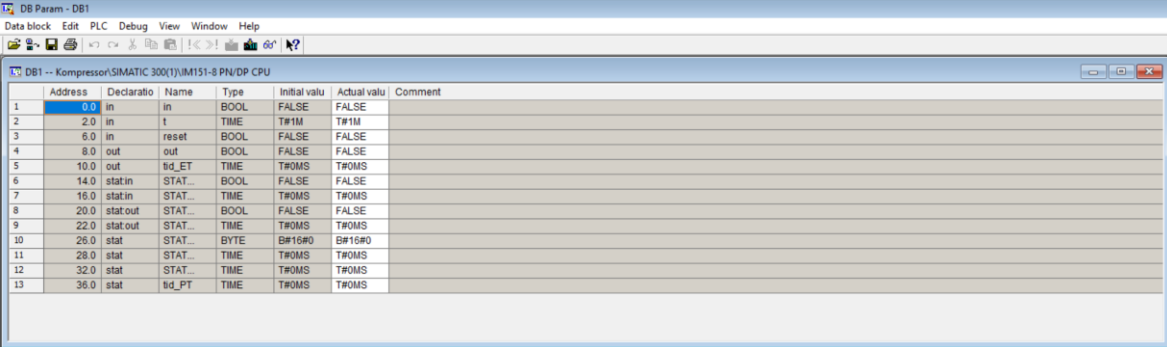
Vid skapandet av programmets delar kan det vara bra att göra en skiss över vad som skall styras och hur det ska uppnås. I detta fall så behövs ett organisationsblock, två funktionsblock, en funktion och tre datablock. Utmaningen med programmet var att skapa en timerfunktion som skulle klara av 24 – 48 timmar. Det finns ingen sådan färdig funktion vilket resulterade i skapandet av en egen funktion som uppfyller det kravet.



Figur 7. Skapandet av block för programmet.

Skapandet av blocken görs genom att högerklicka på mappen ”Blocks” och välja ”Insert new object”. Då kan man välja vilken sorts block man vill lägga till. Organisationsblocket läggs till per automatik vid skapandet av projektet, medan funktionsblocken och funktionen måste läggas till. Det som man behöver tänka på vid skapandet av dem är att lägga till kommentarer så det lätt går att få en uppfattning av vad blockets funktion är.

Eftersom timerfunktionerna för att klara av tider upp till flertal dygn programmerades i funktionsblock krävdes också datablock för att lagra data från dem. Datablocken måste då vara instansdatablock så de är sammankopplade med funktionsblocket.



The screenshot shows the 'DB Param - DB1' window in SIMATIC Manager. The title bar indicates the project is 'Kompressor/SIMATIC 300(1)/UM151-8 PN/DP CPU'. The main area displays a table of variables for data block DB1.

	Address	Declaration	Name	Type	Initial valu	Actual valu	Comment
1	0.0	in	in	BOOL	FALSE	FALSE	
2	2.0	in	t	TIME	T#1M	T#1M	
3	6.0	in	reset	BOOL	FALSE	FALSE	
4	8.0	out	out	BOOL	FALSE	FALSE	
5	10.0	out	tid_ET	TIME	T#0MS	T#0MS	
6	14.0	stat.in	STAT...	BOOL	FALSE	FALSE	
7	16.0	stat.in	STAT...	TIME	T#0MS	T#0MS	
8	20.0	stat.out	STAT...	BOOL	FALSE	FALSE	
9	22.0	stat.out	STAT...	TIME	T#0MS	T#0MS	
10	26.0	stat	STAT...	BYTE	B#16#0	B#16#0	
11	28.0	stat	STAT...	TIME	T#0MS	T#0MS	
12	32.0	stat	STAT...	TIME	T#0MS	T#0MS	
13	36.0	stat	tid_PT	TIME	T#0MS	T#0MS	

Figur 8. Variabler i ett datablock.

I figur 8 kan man se de lagrade variablerna som kommer att användas av det sammankopplade funktionsblocket.

5.3 Timerfunktion

Funktionen för dygnstimern är programmerad i Siemens STL programmeringsspråk, vilket är ett textbaserat programmeringsspråk. Det skapas genom att definiera variabler som in- och utgångar samt statiska variabler. Där statiska variabler är sådana som endast används inne i funktionsblocket.

The screenshot shows the Siemens SIMATIC Manager interface. The title bar indicates the project is 'Komp1 SFBS' for a SIMATIC 300 CPU. The main window displays the STL code for a function block named 'Komp1 SFBS (reset)'. The code is organized into sections for variable declarations, reset behavior, and normal behavior.

Name	Data Type	Address	Initial Value	Exclusion address	Termination address	Comment
out	Bool	8.0	FALSE			
tid_ET	Time	10.0	T#0MS			

```

FBI : Title:
Komp1 SFBS (reset)
Network1 : Title:
//Kopiera tidsvärdet
L   #t                #t
T   #tid_PT           #tid_PT

//Reset behavior
A   #reset            #reset
L   0
T   #tid_ET          #tid_ET
T   #tid_PT          #tid_PT
T   #STAT0.STIME    #STAT0.STIME
T   #STAT0.ATIME    #STAT0.ATIME
T   #STAT0.STATE    #STAT0.STATE
CALL #STAT0         #STAT0
IN:=FALSE
PT:=T#0MS
Q  :=#out            #out
ET:=#tid_ET         #tid_ET

CLR
=   #out             #out

BEU

// Normal Behavior
NOP 0

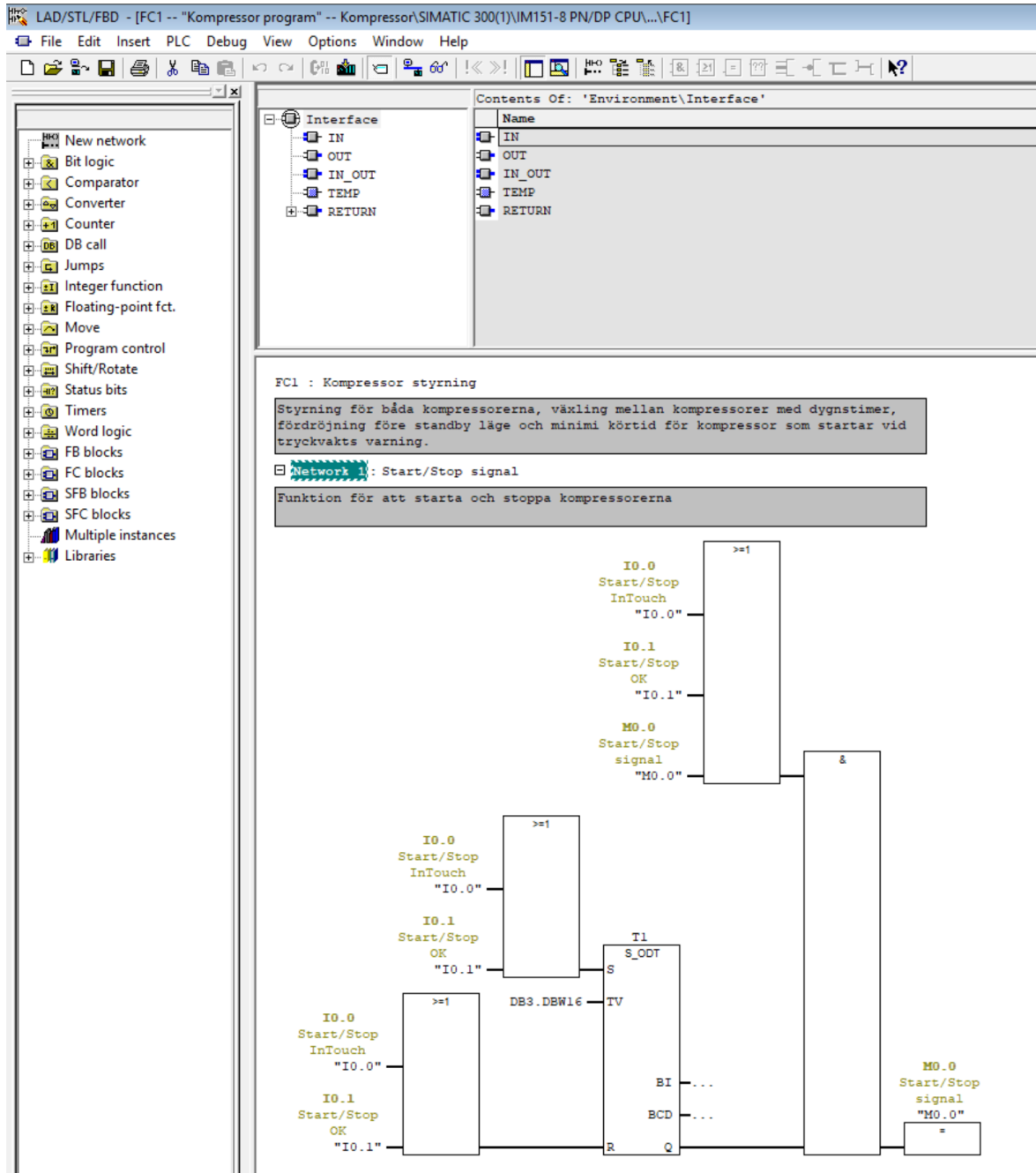
CALL #STAT0         #STAT0
IN:=#in             #in
PT:=#tid_PT        #tid_PT
Q  :=#out            #out
ET:=#tid_ET         #tid_ET

```

Figur 9. Funktion på dygnstimern.

5.4 Slutliga funktionen

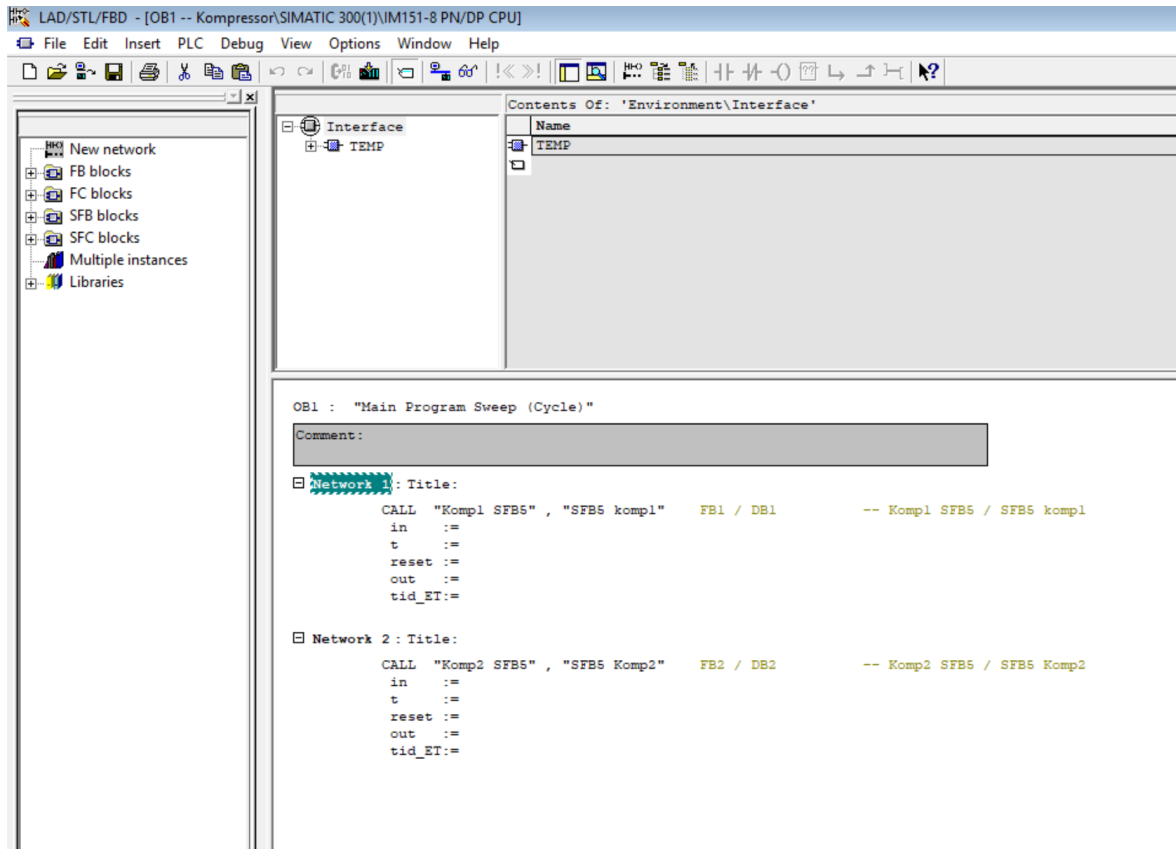
Slutligen skapades själva funktionen som styr utgångarna för att starta och stoppa kompressorerna. Det gjordes i FBD programspråk, vilket innebär att det byggs upp i blockdiagram där man definierar in- och utgångar eller minnesvariabler i blocken för att skapa en önskad logik.



Figur 10. Bild på första nätverket av funktionen.

Utöver funktionen behöver man till sist göra anropningar till blocken via organisationsblocket. Eftersom funktionsblocken är sammankopplade med ett eget

datablock så behöver man anropa dem i samma anropning. I figur 11 kan man se ett exempel på hur det kan utföras.



Figur 11. Anropningar i organisationsblocket.

6 Testning och simulering

Testning och simulering är en viktig del av processen för att undvika misstag eller förstöra hårdvara som används. Under skapandet av ett styrsystem eller program gör man ofta en hel del ändringar och modifikationer från projektets start tills det är slutfört. I större projekt är det också flera personer inblandade i samma projekt vilket gör dokumentationen och kommunikationen till en viktig del, så alla har koll på de ändringar och tester som gjorts.

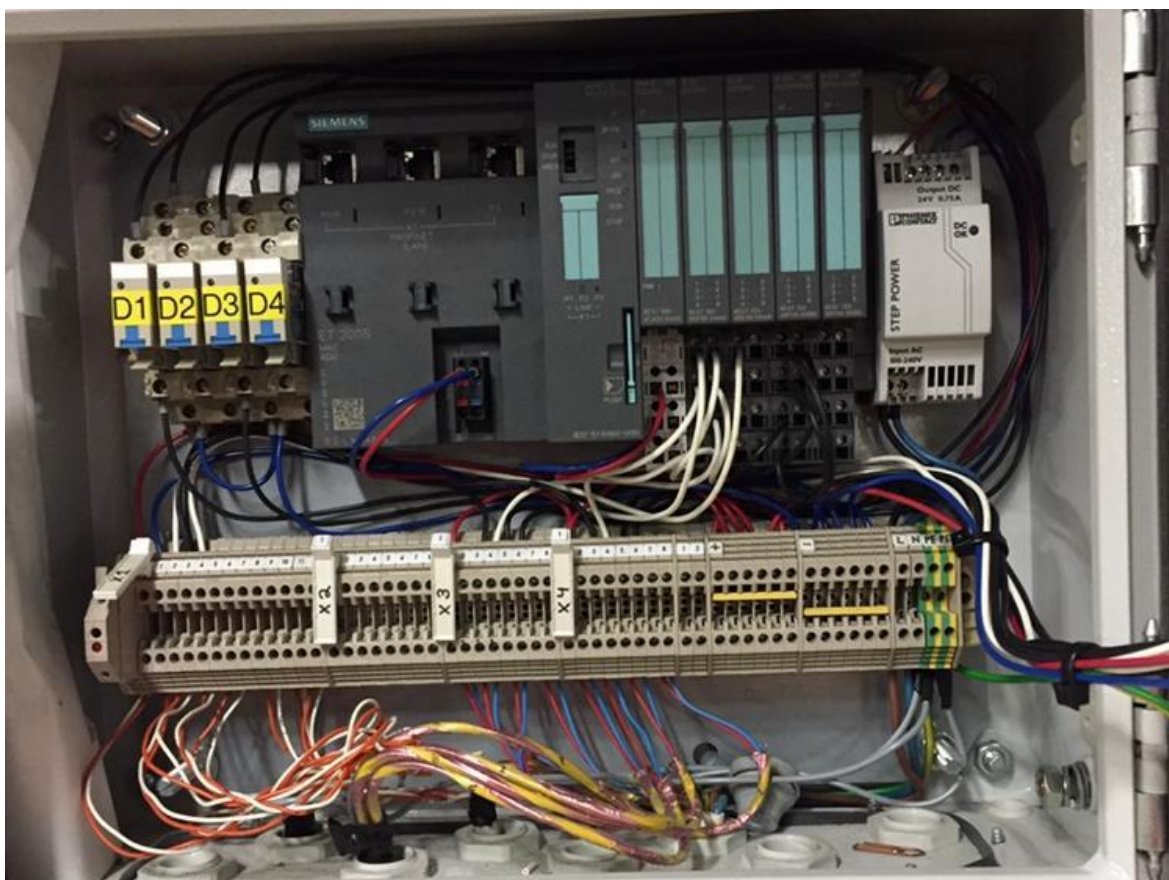
Testning och simulering av programmet utfördes regelbundet med hjälp av Siemens simuleringsfunktion i mjukvaran samt genom användning av en labbrigg. Det gjordes för att pröva att funktionerna som skapades uppfyllde förväntningarna och inte hade fel funktioner.

7 Inkoppling av styrsystem

Inkopplingen av det nya styrsystemet började med att montera bort det gamla. Den nya PLC-enheten monterades sedan på plats för att påbörja inkopplingen av styrsignaler enligt det planerade kretsschemat.

7.1 Inkoppling av styrcentral

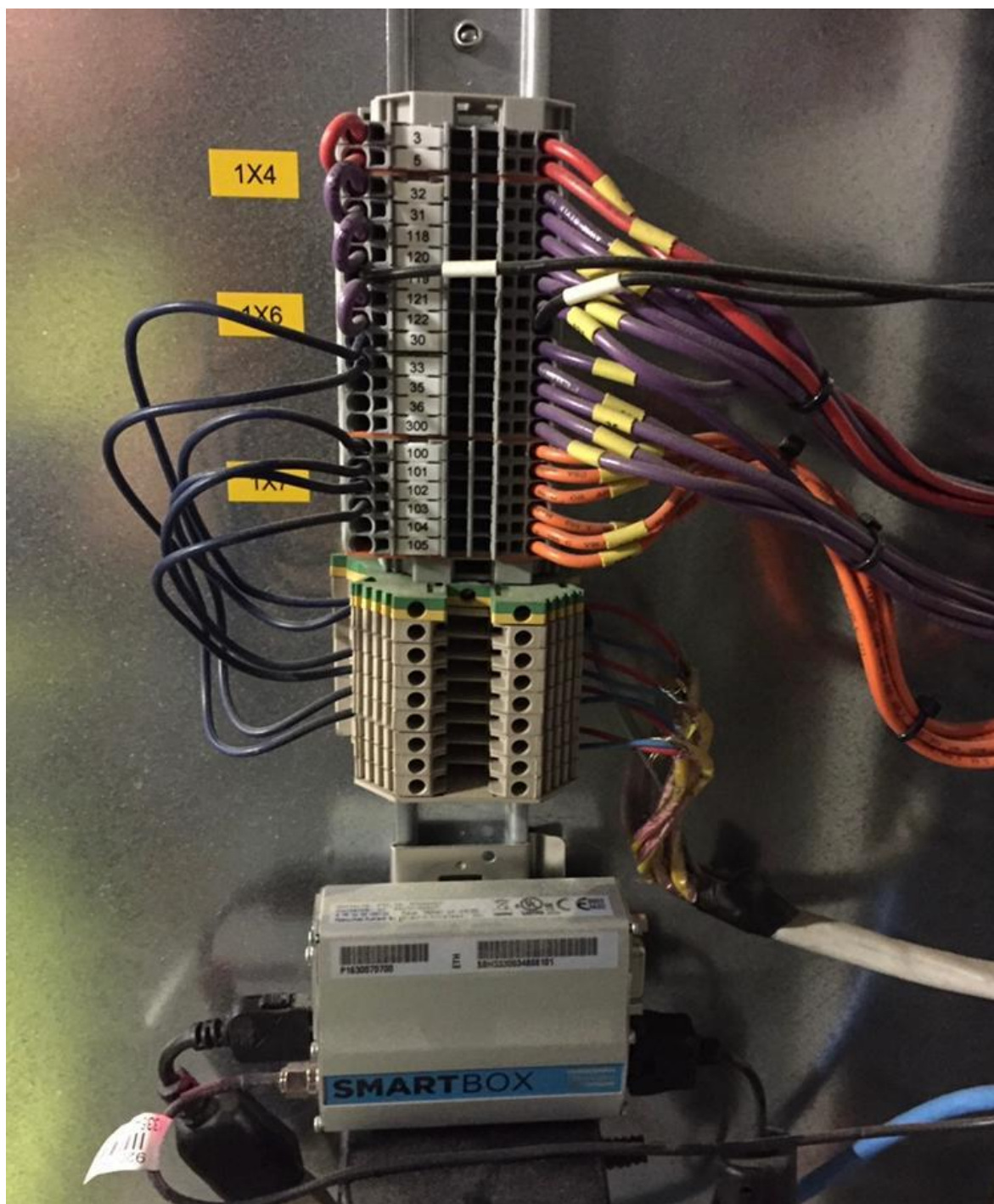
Inkoppling av in- och utgångar kopplades till PLC-enhetens signalmoduler. För att hålla kompressorernas och styrsystemets 24 VDC potentialer åtskilda användes det vanliga relämoduler för att styra utsignalerna till kompressorerna. Det utfördes genom att ansluta spänningsförsörjning från kompressorerna i reläets öppnande eller slutande kontakter och en signalanslutning tillbaka till kompressorn i fråga. PLC-enhetens utsignaler styrde däremot samtliga relä spolar för att kontrollera kompressorerna. Relämodulerna är markerade med beteckningarna D1, D2, D3 och D4 i figur 12.



Figur 12. Styrcentral.

7.2 Inkoppling av kompressorenheter

Kompressorenheterna var planerade med färdig plats för inkoppling av fjärrstyrningsmöjlighet. Från manualerna som tillhörde kompressorerna framkom det hur inkoppling av fjärrstyrningen skulle utföras.



Figur 13. Inkoppling av kompressor.

8 Resultat

Målet med detta arbete var att skapa ett styrsystem som skulle ersätta det gamla föråldrade styrsystemet för att styra fabriken kompressorer. Detta för att underlätta styrning och underhållning av kompressorerna i framtiden. Genom utförandet ska man i framtiden få information om kompressorerna samt lufttryck till det överliggande styrsystemet i övervakningscentralen.

Projektet resulterade i ett nytt styrsystem för kompressorerna och ett program där man enkelt kan ändra på fördröjningar och kötider för kompressorerna. Alla funktioner blev klara i programmet och alla komponenter monterades på plats i en styrcentral. Sluttestet kunde inte utföras, detta pga. att ena kompressorn inte hade funktionen inställd för styrning av ett utomstående styrsystem vilket företaget som levererade kompressorn behövde lägga till.

9 Diskussion

Examensarbetet var en intressant utmaning att utföra, det med tanke på att jag inte tidigare byggt upp ett styrsystem från noll och hade koll på vad allt som skulle implementeras i det. Därför gick en hel del tid till att läsa dokument och forum för att få en bättre överblick av tankesättet som krävs för att utföra arbeten som detta.

Arbetet var tidskrävande på grund av att det gamla styrsystemet inte var så bra dokumenterat och ingen hade så bra koll på hur det var uppkopplat. Det gick en del tid till att reda ut det för att ha koll på vilka kablar som var kopplade var. Utöver det så gick programmeringen långsamt framåt i början under forskningsperioden men började löpa på i bra fart ju längre projektet utvecklades.

Det största problemet under projektet var att skapa en timerfunktion som skulle verka under 24 – 48 timmar för kompressorerna. Det fanns ingen färdig timerfunktion som klarade av så långa tidsintervall vilket ledde till mer tid för forskning av lösningar på det. Det var väldigt lärorikt att skapa en sådan funktion och gjorde arbetet mer krävande. Detta var den mest intressanta delen av programmet och slutresultatet av den funktionen blev bra och väl användbar. Kunskapen från detta kommer att vara nyttigt i framtida problemlösningar för liknande funktioner vid behov.

10 Litteraturförteckning

- Akytec.de. (2019). https://akytec.de/en/blog/sps_vs_plr. Hämtat från Akytec.de.
- Automation.com. (2018). <https://www.automation.com/en-us/articles/2018/a-beginners-plc-overview-part-3-of-4-plc-inputs-an>. Hämtat från Automation.com.
- Berger, H. (2012). *Automating with STEP7 in LAD and FBD*.
- Elektrologik. (2020). *Elektrologik*. Hämtat från <http://www.elektrologik.se/styrssystem.html>
- Elsip. (2017). *Elsip*. Hämtat från <http://elsip.se/vad-ar-en-hardvara/>
- Haag, B. (2011). *Industriell systemteknik*. Studentlitteratur.
- Kingspan. (2020). <https://www.kingspan.com/group/>. Hämtat från Kingspan.com.
- ParocPanels. (2020). <https://www.parocpanels.com/sv-se>. Hämtat från ParocPanels.
- Plcdev.com. (2020). http://www.plcdev.com/how_plcs_work. Hämtat från Plcdev.com.
- Siemens. (2014). <https://support.industry.siemens.com/cs/document/87560033/et200s-technological-functions-%E2%80%94-1count?dti=0&lc=en-CO>. Hämtat från Siemens.
- Siemens. (2020). *Siemens*. Hämtat från Siemens: <https://support.industry.siemens.com/cs/start?lc=en-US>

SIEMENS

Data sheet

6ES7151-8AB01-0AB0

SIMATIC DP, IM151-8 PN/DP CPU f. ET200S, 192 KB work memory, int. PROFINET interface (with three RJ45 ports) as IO controller, without battery MMC required



General information	
HW functional status	01
Firmware version	V3.2
Product function	
• Isochronous mode	No
Engineering with	
• Programming package	STEP 7 V5.5 or higher
Supply voltage	
Rated value (DC)	24 V
permissible range, lower limit (DC)	20.4 V
permissible range, upper limit (DC)	28.8 V
Reverse polarity protection	Yes; against destruction
external protection for power supply lines (recommendation)	2 A min.
Mains buffering	
• Mains/voltage failure stored energy time	5 ms
Input current	
Inrush current, typ.	1.8 A

I ² t	0.13 A ² ·s
from supply voltage 1L+, max.	352 mA; 426 mA with DP master module
Output current	
for backplane bus (5 V DC), max.	700 mA
Power loss	
Power loss, typ.	5.5 W
Memory	
Work memory	
• integrated	192 kbyte
• expandable	No
• Size of retentive memory for retentive data blocks	64 kbyte
Load memory	
• Plug-in (MMC)	Yes
• Plug-in (MMC), max.	8 Mbyte
• Data management on MMC (after last programming), min.	10 y
Backup	
• present	Yes; Ensured by SIMATIC Micro Memory Card (maintenance-free)
CPU processing times	
for bit operations, typ.	0.06 μs
for word operations, typ.	0.12 μs
for fixed point arithmetic, typ.	0.16 μs
for floating point arithmetic, typ.	0.59 μs
CPU-blocks	
Number of blocks (total)	1 024; (DBs, FCs, FBs); the maximum number of loadable blocks can be reduced by the MMC used.
DB	
• Number, max.	1 024; Number range: 1 to 16000
• Size, max.	64 kbyte
FB	
• Number, max.	1 024; Number range: 0 to 7999
• Size, max.	64 kbyte
FC	
• Number, max.	1 024; Number range: 0 to 7999
• Size, max.	64 kbyte
OB	
• Description	See S7-300 operation list
• Size, max.	64 kbyte
• Number of free cycle OBs	1; OB 1

• Number of time alarm OBs	1; OB 10
• Number of delay alarm OBs	2; OB 20, 21
• Number of cyclic interrupt OBs	4; OB 32, 33, 34, 35
• Number of process alarm OBs	1; OB 40
• Number of DPV1 alarm OBs	3; OB 55, 56, 57
• Number of isochronous mode OBs	1; OB 61; only for PROFINET
• Number of startup OBs	1; OB 100
• Number of asynchronous error OBs	6; OB 80, 82, 83, 85, 86, 87 (OB83 only for centralized I/O and PROFINET IO)
• Number of synchronous error OBs	2; OB 121, 122
Nesting depth	
• per priority class	16
• additional within an error OB	4
Counters, timers and their retentivity	
S7 counter	
• Number	256
Retentivity	
— adjustable	Yes
— lower limit	0
— upper limit	255
— preset	Z 0 to Z 7
Counting range	
— adjustable	Yes
— lower limit	0
— upper limit	999
IEC counter	
• present	Yes
• Type	SFB
• Number	Unlimited (limited only by RAM capacity)
S7 times	
• Number	256
Retentivity	
— adjustable	Yes
— lower limit	0
— upper limit	255
— preset	No retentivity
Time range	
— lower limit	10 ms
— upper limit	9 990 s
IEC timer	
• present	Yes

• Type	SFB
• Number	Unlimited (limited only by RAM capacity)
Data areas and their retentivity	
Flag	
• Number, max.	256 byte
• Retentivity available	Yes
• Retentivity preset	MB 0 to MB 15
• Number of clock memories	8; 1 memory byte
Data blocks	
• Retentivity adjustable	Yes; via non-retain property on DB
• Retentivity preset	Yes
Local data	
• per priority class, max.	32 768 byte; Max. 2048 bytes per block
Address area	
I/O address area	
• Inputs	2 048 byte
• Outputs	2 048 byte
of which distributed	
— Inputs	2 048 byte
— Outputs	2 048 byte
Process image	
• Inputs, adjustable	2 048 byte
• Outputs, adjustable	2 048 byte
• Inputs, default	128 byte
• Outputs, default	128 byte
Subprocess images	
• Number of subprocess images, max.	1; With PROFINET IO, the length of the user data is limited to 1600 bytes
Digital channels	
• Inputs	16 336
— of which central	496
• Outputs	16 336
— of which central	496
Analog channels	
• Inputs	1 021
— of which central	124
• Outputs	1 021
— of which central	124
Hardware configuration	
Number of modules per system, max.	63; Centralized
Mounting rail	

<ul style="list-style-type: none"> • Number of mounting rails that can be used • Length of mounting rail, max. 	<p>1</p> <p>Station width: ≤ 1 m or < 2 m</p>
Time of day	
Clock	
<ul style="list-style-type: none"> • Hardware clock (real-time) • retentive and synchronizable • Backup time • Deviation per day, max. • Behavior of the clock following POWER-ON • Behavior of the clock following expiry of backup period 	<p>Yes</p> <p>Yes</p> <p>6 wk; At 40 °C ambient temperature, typically</p> <p>10 s; Typ.: 2 s</p> <p>Clock continues running after POWER OFF</p> <p>Clock continues to run with the time at which the power failure occurred</p>
Operating hours counter	
<ul style="list-style-type: none"> • Number • Number/Number range • Range of values • Granularity • retentive 	<p>1</p> <p>0</p> <p>0 to 2³¹ hours (when using SFC 101)</p> <p>1 h</p> <p>Yes; Must be restarted at each restart</p>
Clock synchronization	
<ul style="list-style-type: none"> • supported • to MPI, master • to MPI, slave • to DP, master • to DP, slave • in AS, master • in AS, slave • on Ethernet via NTP 	<p>Yes</p> <p>No</p> <p>No</p> <p>Yes; With DP master module</p> <p>Yes; With DP master module</p> <p>No</p> <p>No</p> <p>Yes; As client</p>
Interfaces	
Interfaces/bus type	1x PROFINET (3 RJ45 ports)
1. Interface	
Interface type	PROFINET
Physics	Ethernet
Isolated	Yes
automatic detection of transmission rate	Yes
Autonegotiation	Yes
Autocrossing	Yes
Change of IP address at runtime, supported	Yes
Interface types	
<ul style="list-style-type: none"> • Number of ports • integrated switch 	<p>3; RJ45</p> <p>Yes</p>
Protocols	
<ul style="list-style-type: none"> • MPI 	No

• PROFINET IO Controller	Yes; Also simultaneously with IO-Device functionality
• PROFINET IO Device	Yes; Also simultaneously with IO Controller functionality
• PROFINET CBA	Yes
• PROFIBUS DP master	No
• PROFIBUS DP slave	No
• Open IE communication	Yes; Via TCP/IP, ISO on TCP, and UDP
• Web server	Yes
• Point-to-point connection	No
PROFINET IO Controller	
• Transmission rate, max.	100 Mbit/s; full duplex
Services	
— PG/OP communication	Yes
— Routing	Yes; With DP master module
— S7 communication	Yes; with loadable FBs
— Isochronous mode	Yes; OB 61; only for PROFINET IO
— IRT	Yes
— MRP	Yes
— Shared device	Yes
— Prioritized startup	Yes
— Number of IO devices with prioritized startup, max.	32
— Number of connectable IO Devices, max.	128
— Of which IO devices with IRT, max.	64
— of which in line, max.	64
— Number of IO Devices with IRT and the option "high flexibility"	128
— of which in line, max.	61
— Number of connectable IO Devices for RT, max.	128
— of which in line, max.	128
— Activation/deactivation of IO Devices	Yes
— Number of IO Devices that can be simultaneously activated/deactivated, max.	8
— IO Devices changing during operation (partner ports), supported	Yes
— Number of IO Devices per tool, max.	8
— Device replacement without swap medium	Yes
— Send cycles	250 μ s, 500 μ s, 1 ms; 2 ms, 4 ms (not in the case of IRT with "high flexibility" option)
— Updating time	Minimum value depends on communication share set for PROFINET I/O, on the number of I/O devices, and on the number of configured user data items.


— Updating times	250 µs to 512 ms (depends on operating mode; for more details, refer to Operating Instructions, "Interface Module IM151-8 PN/DP CPU")
Address area	
— Inputs, max.	2 kbyte
— Outputs, max.	2 kbyte
— User data consistency, max.	1 024 byte; with PROFINET I/O
PROFINET IO Device	
Services	
— PG/OP communication	Yes
— Routing	Yes
— S7 communication	Yes; with loadable FBs
— Isochronous mode	No
— IRT	Yes
— MRP	Yes
— PROFINergy	Yes; With SFB 73 / 74 prepared for loadable PROFINergy standard FB for I-Device
— Shared device	Yes
— Number of IO Controllers with shared device, max.	2
Transfer memory	
— Inputs, max.	1 440 byte; Per IO Controller with shared device
— Outputs, max.	1 440 byte; Per IO Controller with shared device
Submodules	
— Number, max.	64
— User data per submodule, max.	1 024 byte
PROFINET CBA	
• acyclic transmission	Yes
• cyclic transmission	Yes
Open IE communication	
• Number of connections, max.	8
• Local port numbers used at the system end	0, 20, 21, 23, 25, 80, 102, 135, 161, 443, 8080, 34962, 34963, 34964, 65532, 65533, 65534, 65535
2. Interface	
Interface type	External interface via master module 6ES7138-4HA00-0AB0
Physics	RS 485
Isolated	Yes
Power supply to interface (15 to 30 V DC), max.	No
Protocols	
• MPI	No
• PROFINET IO Controller	No
• PROFINET IO Device	No

• PROFINET CBA	No
• PROFIBUS DP master	Yes
• PROFIBUS DP slave	No
• Open IE communication	No
• Web server	No
PROFIBUS DP master	
• Transmission rate, max.	12 Mbit/s
• Number of DP slaves, max.	32; Per station
Services	
— PG/OP communication	Yes
— Routing	Yes
— Global data communication	No
— S7 basic communication	Yes; 1 blocks only
— S7 communication	Yes
— S7 communication, as client	No
— S7 communication, as server	Yes
— Equidistance	Yes
— Isochronous mode	No
— SYNC/FREEZE	Yes
— Activation/deactivation of DP slaves	Yes
— Number of DP slaves that can be simultaneously activated/deactivated, max.	8
— Direct data exchange (slave-to-slave communication)	Yes
— DPV1	Yes
Address area	
— Inputs, max.	2 048 byte
— Outputs, max.	2 048 byte
User data per DP slave	
— Inputs, max.	244 byte
— Outputs, max.	244 byte
Protocols	
Redundancy mode	
Media redundancy	
— Switchover time on line break, typ.	200 ms; PROFINET MRP
— Number of stations in the ring, max.	50
Open IE communication	
• TCP/IP	Yes; via integrated PROFINET interface and loadable FBs
— Number of connections, max.	8
— Data length for connection type 01H, max.	1 460 byte
— Data length for connection type 11H, max.	32 768 byte

— several passive connections per port, supported	Yes
• ISO-on-TCP (RFC1006)	Yes; via integrated PROFINET interface and loadable FBs
— Number of connections, max.	8
— Data length, max.	32 768 byte
• UDP	Yes; via integrated PROFINET interface and loadable FBs
— Number of connections, max.	8
— Data length, max.	1 472 byte
Web server	
• supported	Yes
• User-defined websites	Yes
• Number of HTTP clients	5
Communication functions	
PG/OP communication	Yes
Data record routing	Yes; With DP master module
Global data communication	
• supported	No
S7 basic communication	
• supported	Yes; 1 blocks
• User data per job, max.	76 byte
• User data per job (of which consistent), max.	76 byte
S7 communication	
• supported	Yes
• as server	Yes
• as client	Yes; via integrated PROFINET interface and loadable FBs
• User data per job, max.	See online help of STEP 7 (shared parameters of the SFBs/FBs and of the SFCs/FCs of S7 Communication)
PROFINET CBA (at set setpoint communication load)	
• Setpoint for the CPU communication load	50 %
• Number of remote interconnection partners	32
• Number of functions, master/slave	30
• Total of all master/slave connections	1 000
• Data length of all incoming connections master/slave, max.	4 000 byte
• Data length of all outgoing connections master/slave, max.	4 000 byte
• Number of device-internal and PROFIBUS interconnections	500
• Data length of device-internal und PROFIBUS interconnections, max.	4 000 byte
• Data length per connection, max.	1 400 byte
Remote interconnections with acyclic transmission	

— Sampling interval, min.	500 ms
— Number of incoming interconnections	100
— Number of outgoing interconnections	100
— Data length of all incoming interconnections, max.	2 000 byte
— Data length of all outgoing interconnections, max.	2 000 byte
— Data length per connection, max.	1 400 byte
Remote interconnections with cyclic transmission	
— Transmission frequency: Transmission interval, min.	1 ms
— Number of incoming interconnections	200
— Number of outgoing interconnections	200
— Data length of all incoming interconnections, max.	2 000 byte
— Data length of all outgoing interconnections, max.	2 000 byte
— Data length per connection, max.	450 byte
HMI variables via PROFINET (acyclic)	
— Number of stations that can log on for HMI variables (PN OPC/iMap)	3; 2x PN OPC/1x iMap
— HMI variable updating	500 ms
— Number of HMI variables	200
— Data length of all HMI variables, max.	2 000 byte
PROFIBUS proxy functionality	
— supported	Yes
— Number of linked PROFIBUS devices	16
— Data length per connection, max.	240 byte; Slave-dependent
iPAR server	
• supported	Yes
Number of connections	
• overall	12
• usable for PG communication	11
— reserved for PG communication	1
— adjustable for PG communication, min.	1
— adjustable for PG communication, max.	11
• usable for OP communication	11
— reserved for OP communication	1
— adjustable for OP communication, min.	1
— adjustable for OP communication, max.	11
• usable for S7 basic communication	10
— reserved for S7 basic communication	0

— adjustable for S7 basic communication, min.	0
— adjustable for S7 basic communication, max.	10
• usable for S7 communication	10; with loadable FBs
— adjustable for S7 communication, max.	10
• total number of instances, max.	32
• usable for routing	4; With DP master module
S7 message functions	
Number of login stations for message functions, max.	12; Depending on the configured connections for PG/OP and S7 basic communication
Process diagnostic messages	Yes; ALARM_S, ALARM_SC, ALARM_SQ, ALARM_D, ALARM_DQ
simultaneously active Alarm-S blocks, max.	300
Test commissioning functions	
Status block	Yes; Up to 2 simultaneously
Single step	Yes
Number of breakpoints	4
Status/control	
• Status/control variable	Yes
• Variables	Inputs, outputs, memory bits, DB, times, counters
• Number of variables, max.	30
— of which status variables, max.	30
— of which control variables, max.	14
Forcing	
• Forcing	Yes
• Forcing, variables	I/O
• Number of variables, max.	10
Diagnostic buffer	
• present	Yes
• Number of entries, max.	500
— adjustable	No
— of which powerfail-proof	100; Only the last 100 entries are retained
Interrupts/diagnostics/status information	
Alarms	Yes
Diagnostics function	Yes
Diagnostics indication LED	
• for maintenance	Yes; MT
• Bus fault BF (red)	Yes; BF-PN
• Group error SF (red)	Yes
• Monitoring 24 V voltage supply ON (green)	Yes

• Bus activity PROFINET (green)	Yes; P1-/P2-/P3-Link
Potential separation	
between PROFIBUS DP and all other circuit components	Yes
Isolation	
Isolation tested with	500 V DC
Degree and class of protection	
IP degree of protection	IP20
Configuration	
Configuration software	
• STEP 7	Yes; V5.5 or higher
Programming	
• Command set	see instruction list
• Nesting levels	8
• System functions (SFC)	see instruction list
• System function blocks (SFB)	see instruction list
Programming language	
— LAD	Yes
— FBD	Yes
— STL	Yes
— SCL	Yes; Optional
— CFC	Yes; Optional
— GRAPH	Yes; Optional
— HiGraph®	Yes; Optional
Know-how protection	
• User program protection/password protection	Yes
• Block encryption	Yes; With S7 block Privacy
Cycle time monitoring	
• lower limit	1 ms
• upper limit	6 000 ms
• adjustable	Yes
• preset	150 ms
Dimensions	
Width	120 mm; DP master module: 35 mm
Height	119.5 mm
Depth	75 mm
Weights	
Weight, approx.	320 g; DP master module: Approx. 100 g
last modified:	11/12/2020 




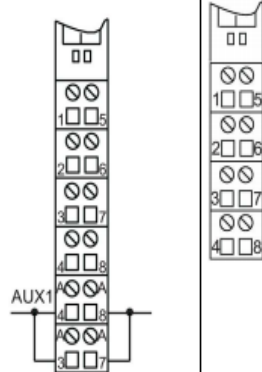
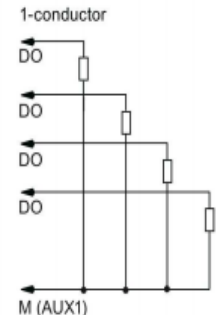
General terminal assignment

Note

Terminals A4, A8, A3 and A7 are only available at specified terminal modules.

Terminal assignment for 8DO DC24V/0.5A (6ES7132-4BF00-0AA0)				
Terminal	Allocation	Terminal	Allocation	Notes
1	DO ₀	5	DO ₁	<ul style="list-style-type: none"> DO_n: Output signal, channel n AUX1: Mass M (from power module) or voltage bus (freely usable up to 230 V AC)
2	DO ₂	6	DO ₃	
3	DO ₄	7	DO ₅	
4	DO ₆	8	DO ₇	
A4	AUX1	A8	AUX1	
A3	AUX1	A7	AUX1	

Usable terminal modules

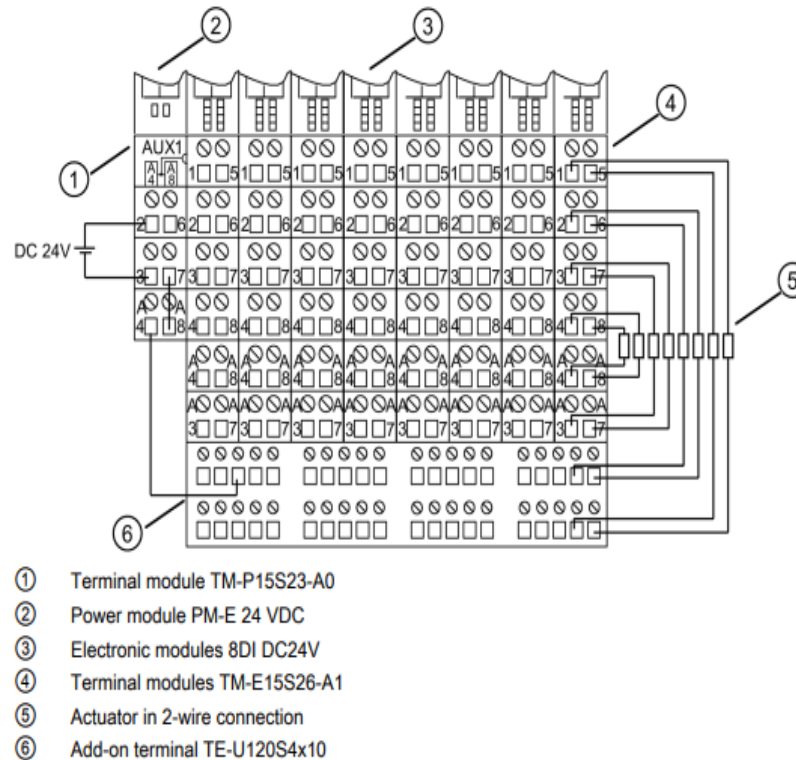
Usable terminal modules for 8DO DC24V/0.5A (6ES7132-4BF00-0AA0)		
TM-E15C26-A1 (6ES7193-4CA50-0AA0)	TM-E15C24-01 (6ES7193-4CB30-0AA0)	 Spring terminal
TM-E15S26-A1 (6ES7193-4CA40-0AA0)	TM-E15S24-01 (6ES7193-4CB20-0AA0)	 Screw-type terminal
TM-E15N26-A1 (6ES7193-4CA80-0AA0)	TM-E15N24-01 (6ES7193-4CB70-0AA0)	 Fast Connect
		Connection examples 1-conductor 

Properties

1.1 Digital electronic module 8DO DC24V/0.5 A (6ES7132-4BF00-0AA0)

2-wire connection

The following configuration example shows a two-wire connection with the electronic modules 8DO DC24V. You require further terminals so that sufficient terminals are available for the mass M connection when the TM-E15S26-A1 terminal modules are used. In the example this is implemented by the add-on terminal TE-U120S4x10, which you can mount as of a width of 120 mm (8 EMs). You can naturally also use other terminals for this configuration (for example, ET 200S potential distribution module 4POTDIS).



Block diagram

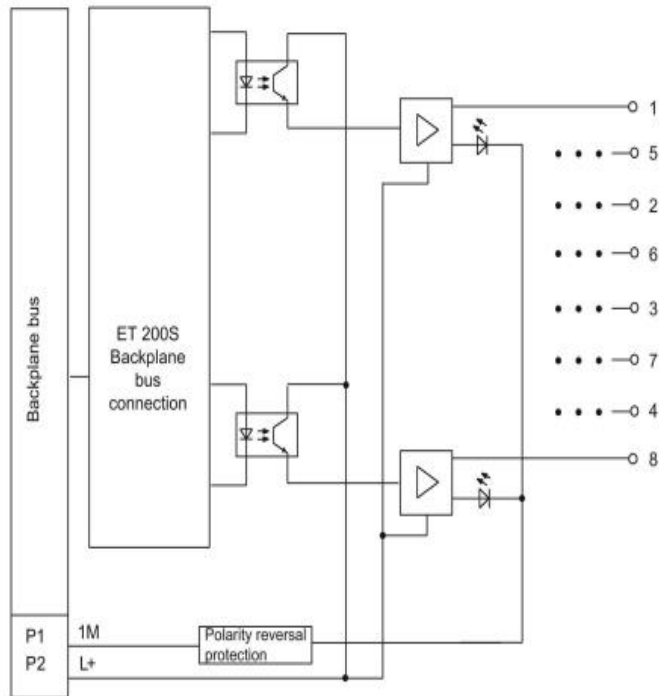


Figure 1-1 Block diagram of the 8DO DC24V 0.5A