

Kompatibilitet och konvertering mellan olika PLC-tillverkare inom industriell automation



Djuro Petrovic

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

© Copyright Djuro Petrović

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

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

Tryckt i Sverige
Lunds universitet
Helsingborg 2024

Sammanfattning

Automationslösningar i form av PLC-programmering ökar i rasande fart och PLC-tillverkare som Siemens och Rockwell Automation är bland de mest använda och erkända i Europa. PLC-enheterna från dessa tillverkare tillsammans med motsvarande utrustning som exempelvis HMI-paneler är standardiserade och löser komplexa problem inom industriell automation. Utöver skillnader i prestanda och kommunikation är även tillämpningen av den internationella standarden IEC 61131-3 olika, vilket granskas i denna studie.

Syftet med detta examensarbete är att undersöka hur noga Siemens och Rockwell Automations egenutvecklade programmeringsmjukvaror för konfigurering av PLC-enheter, TIA portal respektive Studio 5000, tillämpar IEC 61131-3. Utöver den generella tillämpningen av IEC 61131-3 riktas fokus på att identifiera vilka delar PLC-tillverkarna har gemensamt inom IEC 61131-3 med syfte att identifiera likheter och därmed kompatibiliteten. För att lyfta fram skillnader och förbättringsmöjligheter av kompatibiliteten ska de delar av IEC 61131-3 som PLC-tillverkarna inte har gemensamt hittas. Slutligen identifieras de mest tillämpade programmeringsspråken m.h.a. dokumentfiler av Siemens och Rockwell Automation för att kunna konvertera de delar där kompatibiliteten mellan PLC-tillverkarna är störst.

Efter genomfört examensarbete framkommer att Siemens tillämpar 61.0% av IEC 61131-3 medan Rockwell Automation hamnar på 27.8%. Utifrån analysen av IEC 61131-3 och dokument om överensstämmelse (eng. *compliance*) hittas 20 och 10 dominanta tabeller av TIA portal respektive Studio 5000 varav 9 är gemensamma. Vad gäller de svaga tabellerna hittas 5 gemensamma därav 3 inom programmeringsspråket IL. Därefter visas ST och LD vara de mest tillämpade programmeringsspråken i båda utvecklingsmiljöerna med 85.1% och 82.8% för TIA portal respektive 80.9% och 75.3% för Studio 5000 inom kategorin med uppfyllda funktioner. Avslutningsvis konverteras funktionerna inom de fullständiga, dominanta tabellerna med samtliga funktioner uppfyllda i första hand i ST och när funktionerna inte kan skrivas i ST används LD istället. Syftet med denna del är att belysa likheter och skillnader mellan PLC-tillverkarna när samma funktioner implementeras.

Nyckelord: PLC, TIA portal, Studio 5000, IEC 61131-3, ST, LD

Abstract

Automation solutions in the form of PLC programming are increasing rapidly and PLC manufacturers such as Siemens and Rockwell Automation are among the most used and recognized in Europe. The PLC units from these manufacturers, along with corresponding equipment such as HMI panels, are standardized and solve complex problems in industrial automation. Beyond differences in performance and communication, the application of the international standard IEC 61131-3 also varies, which is being examined in this study.

The purpose of this thesis is to investigate how closely Siemens and Rockwell Automation's proprietary programming software for configuring PLC units, TIA Portal and Studio 5000 respectively, apply IEC 61131-3. Beyond the general application of IEC 61131-3, the focus is directed at identifying which parts the PLC manufacturers have in common within IEC 61131-3 to identify similarities and thus compatibility. To highlight differences and potential improvements in compatibility, the parts of IEC 61131-3 that the PLC manufacturers do not share are identified. Finally, the most applied programming languages are identified using documentation files from Siemens and Rockwell Automation to convert the parts where compatibility between PLC manufacturers is greatest.

After completing the thesis, it is emerged that Siemens applies 61.0% of IEC 61131-3, while Rockwell Automation applies 27.8%. Based on the analysis of IEC 61131-3 and compliance documents, 20 and 10 dominant tables are found in TIA Portal and Studio 5000 respectively, of which 9 are common. Regarding the weak tables, 5 common ones are found, 3 of which are within the programming language IL. Thereafter, ST and LD are shown to be the most commonly used programming languages in both development environments with 85.1% and 82.8% for TIA Portal and 80.9% and 75.3% for Studio 5000 in the category of fulfilled functions. Finally, the functions within the complete, dominant tables are converted, with all functions fulfilled, primarily in ST, and when the functions cannot be written in ST, LD is used instead. The purpose of this section is to highlight the similarities and differences between the PLC manufacturers when the same functions are implemented.

Keywords: PLC, TIA Portal, Studio 5000, IEC 61131-3, ST, LD

Innehåll

1	Inledning	1
1.1	Bakgrund	1
1.2	Syfte	1
1.3	Målformulering	2
1.4	Tidigare målformulering	2
1.5	Problemformulering	2
1.6	Motivering av examensarbetet	3
1.7	Etiska aspekter	3
1.8	Avgränsningar	3
1.9	Resurser	4
2	Teknisk bakgrund	5
2.1	Programmerbart styrsystem - PLC	5
2.2	IEC 61131-3	6
2.2.1	Ladder diagram (LD)	7
2.2.2	Function block diagram (FBD)	8
2.2.3	Structured text (ST)	8
2.2.4	Instruction list (IL)	10
2.2.5	Sequential function chart (SFC)	10
2.3	TIA portal	11
2.4	Studio 5000	12
3	Metod	13
3.1	Val och specificering av frågeställningar	13
3.2	Litteraturstudie	14
3.3	Källkritik	14
3.4	Analys av litteraturstudien	15
3.5	Programskrivning	15
4	Analys	16
4.1	IEC 61131-3 tillsammans med TIA portal och Studio 5000	16
4.1.1	TIA portal i enlighet med IEC 61131-3	16
4.1.2	Studio 5000 i enlighet med IEC 61131-3	16
4.2	Kompatibilitet mellan TIA portal och Studio 5000	16
4.3	Kompatibilitetsbrist mellan TIA portal och Studio 5000	17
4.4	De mest kompatibla programmeringsspråken	17
4.5	Konvertering	17
5	Resultat	18
5.1	Den generella tillämpningen av IEC 61131-3	18
5.2	Kompatibilitet mellan TIA portal och Studio 5000	20
5.2.1	De dominanta tabellerna i TIA portal	20

5.2.2	De dominanta tabellerna i Studio 5000	21
5.2.3	Filtrering av de dominanta tabellerna mellan TIA portal och Studio 5000	21
5.3	Kompatibilitetsbrist mellan TIA portal och Studio 5000	22
5.3.1	De svaga tabellerna i TIA portal	22
5.3.2	De svaga tabellerna i Studio 5000	22
5.3.3	Filtrering av de svaga tabellerna mellan TIA portal och Studio 5000	23
5.4	De mest kompatibla programmeringsspråken	24
5.4.1	De mest kompatibla programmeringsspråken i TIA portal	24
5.4.2	De mest kompatibla programmeringsspråken i Studio 5000	24
5.4.3	Bestämmelser och kriterier	25
5.5	Konvertering	26
5.5.1	Funktioner inom de dominanta tabellerna	26
5.5.2	Funktioner inom de svaga tabellerna	33
6	Diskussion och slutsats	35
6.1	Svar på frågeställningarna	35
6.2	Tillämpning	36
6.3	Framtida utvecklingsmöjligheter	36
7	Terminologi	38
8	Referenser	39

Förord

Jag vill tacka AFRY som har gjort det möjligt att genomföra examensarbetet inom en av de mest spännande branscherna som finns ute i näringslivet: industriell automation.

Jag vill även tacka min handledare Anders Jakobsson och Ahmet Barac på AFRY som har stöttat mig under arbetets gång, visat intresse och bidragit med konstruktiv kritik.

1 Inledning

I detta avsnitt ges en övergripande presentation av examensarbetets bakgrund, syfte och frågeställningar. I underrubrikerna 1.1 och 1.2 läggs vikten på företaget examensarbetet genomförs hos och syftet med examensarbetet. I 1.3 beskrivs målformuleringen innehållande de problem som arbetet avser att lösa. Underrubriken 1.4 beskriver tidigare frågeställningar och underrubriken 1.5 fokuserar på specificering av de problem beskrivna under 1.3 och resulterar i specifika, raka frågeställningar examensarbetet dedikeras till att lösa. Varför just de specifika frågeställningarna valts motiveras under 1.6 samt vad åstadkommet resultat av det hela kan innebära för företaget och näringslivet. Slutligen diskuteras etiska aspekter i 1.7, avgränsningar som gäller under 1.8 samt resurser som används under 1.9.

1.1 Bakgrund

AFRY AB är ett internationellt företag inom teknik, design, arkitektur, automation och rådgivning. Genom relevant kompetens inom tidigare nämnda områden omvandlas digitaliserade lösningar till verklighet åt kunder, både lokalt i Sverige men även globalt världen över. Företaget är grundat 1895 under namnet Ångpanneföreningen (ÅF) men efter sammanslagningen med den finländska Pöyry 2019 heter det numera AFRY.

Eftersom industriell automation är en viktig gren inom AFRY genomförs detta examensarbete inom detta område. Arbetet fokuserar på de två mest frekvent använda PLC-tillverkare inom företaget - Siemens och Rockwell Automation. De problem som examensarbetet avser att undersöka är granskning av kompatibiliteten mellan PLC-tillverkarna i en teoretisk studie samt praktisk konvertering mellan dessa. Fokus på kompatibiliteten kommer att ligga på detaljnivå, vilket resulterar i att arbetet blir ett unikt sådant innehållande nya perspektiv.

1.2 Syfte

Syftet med examensarbetet är att, genom en statistisk undersökning, visa hur anpassningen till den internationella standarden IEC 61131-3 kan påverka kompatibiliteten mellan PLC-tillverkarna och om det är av vikt att följa standarden.

Vidare är syftet att informera automationsingenjörer om de viktiga aspekterna de ska tänka på vad gäller konvertering av programkod mellan Siemens och Rockwell Automation. Genom att veta vilka delar som är mest kompatibla mellan PLC-tillverkarna kan automationsingenjörerna lättare avgöra bästa tillvägagångssättet vid konvertering av längre programkoder.

Efter genomfört examensarbete förväntas Siemens ha kommit något längre än Rockwell Automation vad gäller tillämpningen av IEC 61131-3. Detta baseras på faktumet att Siemens är mer internationellt erkänd och mer tillämpad inom industrin [1]. Trots olika tillämpningar av IEC 61131-3 förväntas konverteringen vara ungefär densamma.

1.3 Målformulering

Målformuleringen delas in i två delar: den statistiska undersökningen och konverteringen av programkod skrivna för Siemens och Rockwell Automation.

Målet med den statistiska undersökningen är att jämföra dokumentfiler utgivna av PLC-tillverkarna samtidigt som dokumentet för den internationella standarden IEC 61131-3 följs. Genom att jämföra dokumentfilerna tillsammans med IEC 61131-3 kan ett resultat erhållas i form av uppfyllda kriterier för respektive PLC-tillverkare. Dessa ska sammanställas i tabeller och diagram för att visa den slutgiltiga tillämpningen av IEC 61131-3. Resultatet på denna del av examensarbetet förväntas ge en tydlig bild av hur Siemens och Rockwell Automation förhåller sig till IEC 61131-3.

Målet med konverteringen är att visa praktiskt hur programskrivningen skiljer sig mellan PLC-tillverkarnas egna utvecklingsmiljöer: TIA portal av Siemens och Studio 5000 av Rockwell Automation. Denna del kommer även att stärka resultatet på den statistiska undersökningen och kompatibiliteten mellan PLC-tillverkarna.

1.4 Tidigare målformulering

I det tidiga skedet av examensarbetet introducerades ett antal mål och frågeställningar, vilka skiljer sig från de ovannämnda i avsnitt 1.3. Efter en detaljerad analys om vad som kändes mest relevant för AFRY bestämdes ett antal nya mål och frågeställningar. Frågeställningarna omfattade konvertering av funktionsblock mellan Siemens och Mitsubishi Electric, tillsammans med den statistiska undersökningen av IEC 61131-3. Utöver dessa skulle det mest multikompatibla programmeringsspråket hittas utifrån konverteringen av funktionsblock. Tidsbrist och otillräcklig tillgång till dokumentfiler påverkade möjligheten att besvara frågeställningarna, vilket ledde till formuleringen av nya frågeställningar som beskrivs i 1.5.

1.5 Problemformulering

Här presenteras frågeställningar som ska undersökas under examensarbetets gång:

1. Hur väl förhåller sig TIA portal och Studio 5000 till IEC 61131-3?
2. Inom vilka tabeller i IEC 61131-3 uppfylls samtliga funktioner i respektive utvecklingsmiljö?
3. Inom vilka tabeller i IEC 61131-3 uppfylls inga funktioner i respektive utvecklingsmiljö?
4. Vilket/vilka programmeringsspråk i IEC 61131-3 är mest tillämpat/tillämpade i respektive utvecklingsmiljö?
5. Utifrån 4.: Hur skiljer sig programkod mellan TIA portal och Studio 5000 vid användning av tabellerna från 2?

1.6 Motivering av examensarbetet

Allt fler föråldrade system behöver uppgraderas eller bytas ut helt mot automatiserade lösningar. Detta gör att automationskunskaper alltid kommer att vara aktuella och relevanta på arbetsmarknaden, vilket bidrog till valet av examensarbetet [2].

Efter genomfört examensarbete kommer AFRY kunna använda de viktiga teoretiska aspekter som krävs vid översättning och konvertering av programkod mellan Siemens och Rockwell Automation. Den teoretiska studien står för ca halva examensarbetet och kommer att resultera i vilken vikt man bör lägga på tillämpningen av IEC 61131-3 när exempelvis en ny PLC-tillverkare skapas och skall marknadsföras. En PLC-tillverkare som uppfyller kriterier och normer av en internationellt erkänd standard som IEC kommer med högre sannolikhet lyckas att vinna förtroende inom automationssektorn. [3].

Vidare genomförs automationsprojekt för kunder baserat på deras önskemål om vilken PLC-tillverkare som skall användas. Detta gör att automationsingenjörerna måste kunna programmera i de olika PLC-tillverkarnas utvecklingsmiljöer såsom TIA portal och Studio 5000. Resultatet av examensarbetet kommer att belysa likheter och skillnader vid konvertering av en programkod som exempelvis är skriven i Studio 5000 och skall konverteras till TIA portal. Genom att veta vilka delar inom Studio 5000 som är kompatibla med TIA portal kan man anpassa konverteringen. Delar som kan ingå i konverteringen kan vara allt från deklaration av variabler och datatyper till användning av *timers* och andra tidsrelaterade funktioner.

Dessa motiv faller in för samtliga parter, författaren själv, AFRY och näringslivet/omgivande samhälle.

1.7 Etiska aspekter

Denna jämförande studie mellan Siemens och Rockwell Automation kan öka produktivitet och säkerhet i industrier. Genom att öka implementeringsstyrkor inom IEC 61131-3 främjar detta examensarbete standardisering och effektivitet. Detta underlättar och förbättrar samarbetsförmågan mellan PLC-tillverkarna.

1.8 Avgränsningar

På grund av bredden på den internationella standarden IEC 61131-3 och begränsad tid på examensarbetet riktas fokus endast på de enskilda, fem programmeringsspråken inom PLC. Delen om mjukvaru- och kommunikationsmodellen utesluts därför helt.

Vad gäller PLC-tillverkare är det Siemens och Rockwell Automation som är studiens utgångspunkt, vilket innebär att andra PLC-tillverkare såsom Mitsubishi Electric, ABB och Schneider Electric inte behandlas i detta examensarbete. Som beskrivet tidigare i 1.4 sattes Mitsubishi Electric som studiens mål tillsammans med Siemens men p.g.a. att dokumentfilen om överensstämelsen med IEC 61131-3 inte kan hittas ersätts PLC-tillverkaren mot Rockwell Automation.

1.9 Resurser

För att genomföra konverteringen mellan TIA portal och Studio 5000 används följande:

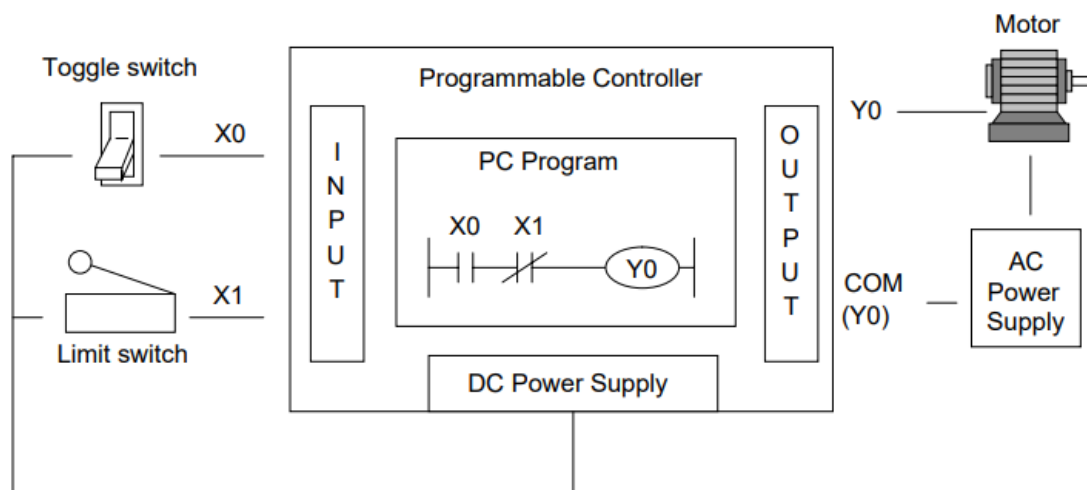
- TIA portal V15.1
- Studio 5000

2 Teknisk bakgrund

I följande avsnitt beskrivs de delar som ingår i genomförandet av detta examensarbete. Delarna är själva PLC-n, standarden den grundar sig på och standardens fem programmeringsspråk. Därefter följer introduktionen till TIA portal och Studio 5000.

2.1 Programmerbart styrsystem - PLC

Ett programmerbart styrsystem eller en PLC, eng. *Programmable Logic Controller*, är en form av dator som tar emot insignaler, bearbetar dem enligt ett installerat program och slutligen ger ut utsignaler. Dessa utsignaler ansluts i sin tur vidare till en process, vanligtvis av mekanisk typ. En PLC betraktas som en krets bestående av logiska grindar som bland annat *AND*, *OR* och *NOR*. Dessutom innehåller en PLC ett minne som är programmerat efter programmerarens önskemål, se Figur 2.1. PLC-n utgör hjärnan i tillverkningsprocessen och ansvarar för logiken bakom hela systemet [4].



Figur 2.1: PLC i blockdiagram med ingångar, CPU och utgångar [5].

En typisk tillämpning av PLC-maskiner är att samla information och få kännedom om ett tillstånd, utvärdera tillståndet och slutligen vidta en åtgärd baserat på tillståndet. I Figur 2.1 hittas ingångar till PLC-enheten som är i form av tryckknappar (*X0* och *X1*) som ser till att leda elektrisk ström vid tryck. Genom att få in en signal genom input-sektionen kommer CPU i PLC-n att bearbeta detta genom den inbyggda och programmerade logiken inuti kretsen, vilket innebär utvärdering av tillståndet från tryckknapparna. Slutligen aktiveras *AC Power Supply* som är ansluten till en motor enligt blockdiagrammet.

2.2 IEC 61131-3

Inom PLC-programmeringen finns normer och standarder för hur PLC-enheter programmeras. En sådan är IEC 61131-3 som innefattar ett direktiv för samtliga PLC-fabrikat sedan 1993. Standarden publicerades av SC65B WG7, den internationella standardiseringsorganisationen *International Electrotechnical Commission*, därav förkortningen IEC [6].

Tillkomsten av med IEC 61131-3 baserades på den ökade komplexiteten inom den industriella automationen. Tack vare standardiseringen tillverkades PLC-enheter i enlighet med IEC 61131-3. Detta gav programmerare möjlighet att jobba med flera olika PLC-tillverkare samtidigt utan att behöva specifik kompetens för varje enskild PLC-tillverkare. Standarden IEC 61131 består i själva verket av en kombination av 10 olika standarder (IEC 60050, IEC 60559:1989, IEC 60617-12:1997, IEC 60617-13:1993, IEC 60848:2002, ISO/AFNOR: 1989, ISO/IEC 10646-1:1993, ISO 8601, ISO 7185, ISO 7498) och är den första som trädde i kraft vad gäller acceptans inom industriell automation [7].

IEC 61131-3 beskriver och ger form till fem olika programmeringsspråk som får lov att användas inom industriell automation och inga skillnader bör förekomma mellan olika PLC-fabrikat. Dock skiljer sig implementeringen av IEC 61131-3 mellan olika tillverkare i praktiken. De fem programmeringsspråken som ingår i standarden är ladderdiagram, funktionsblockdiagram, strukturerad text, instruktionslista och sekventiellt funktionsdiagram.

Utöver de fem programmeringsspråken inkluderas även andra delspecifikationer [7]:

Part 1: General information. Omfattar generella definitioner och typiska funktioner som kännetecknar just det PLC-fabrikatet.

Part 2: Equipment requirements & tests. Definierar de elektriska, mekaniska och funktionella kraven på PLC-enheterna men även kvalifikationstesten. Miljöförhållandena (temperatur, luftfuktighet) och arbetsgränser (tålighet) av PLC-enheterna ingår också.

Part 3: Programming languages. Användning av de fem programmeringsspråken FBD, IL, LD, SFC och ST som är samorienterade till harmoniserad och fraktidsorienterad version. Här definieras även mjukvaru- och kommunikationsmodellen.

Part 4: User guidelines. Avsedd som guide genom de olika faserna i ett automationsprojekt och är övningsorienterad.

Part 5: Communications. Ansvarar för kommunikation mellan olika PLC-tillverkare och med andra enheter. Funktioner som val av enhet, dataöverföring, alarm och åtkomstkontroll ingår i kommunikationsdelen.

Part 6: Safety-related PLC. Målet är anpassning till den internationella standarden IEC 61508 ("Functional safety of electrical/electronic/programmable electronic safety-related systems") och maskindirektivet IEC 62061 ("Safety of machinery

- Functional safety of safety-related electrical, electronic and programmable electronic control systems”) till PLC-enheter.

Part 7: Fuzzy Control Programming. Målet är att förse användarna och PLC-fabrikaten med förståelse och integrering av *fuzzy*-kontrollapplikationer baserade på IEC 61131-3 och främja portabiliteten av dessa mellan olika PLC-tillverkare.

Part 8: Guidelines for the application and implementation of programming languages for PLC. Dokumentet ger svar på frågor som standarden inte gör. Den inkluderar en guide till implementering och användningsmanual för *end users*.

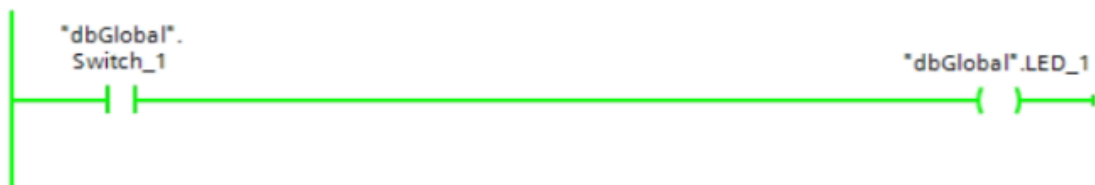
2.2.1 Ladder diagram (LD)

Innan programmerbara styrsystem fanns användes reläbaserade kretsar för att styra olika maskiner och utrustning genom att koppla reläerna på specifika sätt. Kostnaden för relärer samt deras konstanta underhåll resulterade i skapandet av ladderdiagram med syfte att efterlikna de gamla reläbaserade kretsarna [8].

Logiken bakom LD-programmering är precis som det låter, eng. *ladder*, en stege som har en början, en riktning och ett slut. Ett ladderdiagram presenteras i Figur 2.2 där variabeln Switch_1 är en kontakt (eng. *normally open contact*) och LED_1 är en tilldelningsvariabel (eng. *assignment*). Det är viktigt att nämna att datatypen för samtliga variabler klassas som *normally open/closed contact* och *assignment* är av typen Boolean och antar värdena sant eller falskt. I Figur 2.2 går ingen elektrisk ström förbi Switch_1 eftersom Switch_1 är falskt och därmed blir LED_1 också falskt. I Figur 2.3 har Switch_1 modifierats till sant och LED_1 blir därmed sant. Färgen grön illustrerar de delar där elektrisk ström kan passera medan färgen blå begränsar densamma.



Figur 2.2: Switch_1 falskt medför att LED_1 blir falskt.

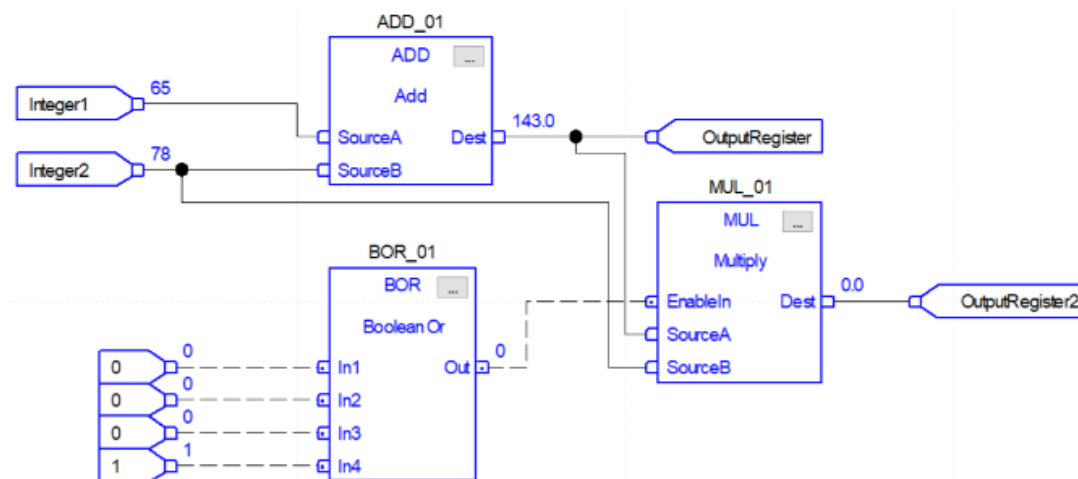


Figur 2.3: Modifiering av Switch_1 till sant sätter LED_1 till sant.

”dbGlobal” i Figur 2.2 och 2.3 är ett datablock där variabler läggs till och sparas. Variablerna Switch_1 och LED_1 finns i detta datablock och kan anropas med punknotation, precis som i de flesta högnivåprogrammeringsspråken. Figur 2.2 och 2.3 är skapade i TIA portal.

2.2.2 Function block diagram (FBD)

Funktionsblockdiagram är ett grafiskt programmeringsspråk som består av block med ingångar och utgångar liknande logiska grindar. FBD är block med en viss inbyggd logik som exekveras och meningen med FBD är att undvika onödig repetition av kod. Detta grafiska programmeringsspråk anses vara lämpligt och användbart när man vill programmera något snabbt samtidigt som felsökning (eng. *debug/troubleshoot*) av program är betydligt enklare än i andra programmeringsspråk i IEC 61131-3. Programmering i FBD kräver inga ingenjörsmässiga skickligheter som strukturerad text kräver och kan skrivas av vanliga användare. Något enklare exempel på FBD-programmering illustreras i Figur 2.4 nedan där blocken ADD, BOR och MUL används för att bland annat addera talen Integer1 och Integer2 och multiplicera summan OutputRegister med Integer2 [9].



Figur 2.4: Funktionsblocken ADD, BOR och MUL inom FBD [10].

2.2.3 Structured text (ST)

Till stor skillnad från de grafiska och visuella programmeringsspråken LD och FBD är ST lik de traditionella programmeringsspråken och är därmed ett högnivåprogrammeringsspråk precis som C++, Java och Python. De mest grundläggande operationer som används vid ST-programmering är IF, ELSE (IF), WHILE, FOR och CASE [11]. Ett exempel på några av nämnda operationer finns i Figur 2.5 nedan där programmet hittar det högsta talet i arraylistan #arrayList. För varje position i denna arraylista, från 0 till 10, hittas det högsta talet och sätts in i variabeln

#tempMax, som sedan läggs till i #max. Programkoden är anpassad efter TIA portal där variabelreferensen ”#” används.

```
#tempMax := 0;

FOR #tempCount := 0 TO 10 DO
    IF #arrayList[#tempCount] > #tempMax THEN
        #tempMax := #arrayList[#tempCount];
    END_IF;
END_FOR;

#max := #tempMax;
```

Figur 2.5: FOR-loop och IF-statement i ST skriven i en vanlig textredigerare.

Fördelen med ST är att den är textbaserad vilket innebär att det är möjligt att skriva kod i en textredigerare (ex. Notepad++), utan att ha tillgång till en utvecklingsmiljö som exempelvis TIA portal. Utöver detta sparas lagringsutrymme vid större projekt vid användning av ST än exempelvis LD eller FBD. En extra finess med ST är möjligheten att lägga till kommentarer i koden för att underlätta för andra som läser och/eller använder koden [11].

2.2.4 Instruction list (IL)

Instruktionslista är ett programmeringsspråk som påminner om Assembly och är ett sätt att kommunicera direkt med datorns hårdvara. IL består av instruktioner som följs i ordning och därav namnet instruktionslista [12].

Fördelar med att använda IL är effektiviteten i både exekveringshastighet och minskad användning av lagringsutrymme jämfört med LD och FBD. IL är inget populärt programmeringsspråk bland PLC-programmerare p.g.a. bristen på grafiska och visuella effekter och det faktum att det är ett lågnivåprogrammeringsspråk som Assembly. I Figur 2.6 visas bland annat: O (hitta tillståndet), S ("SET") och R ("RESET") [4] [12].

Network 1:				
Comment				
1	O	"Start"		%I0.0
2	S	"Auto_mode"		%M0.0

Network 2:				
Comment				
1	O	"Stop"		%I0.1
2	R	"Auto_mode"		%M0.0

Network 3:				
Comment				
1	O	"PE_Low"		%I0.2
2	A	"PE_High"		%I0.3
3	=	#Large_box		

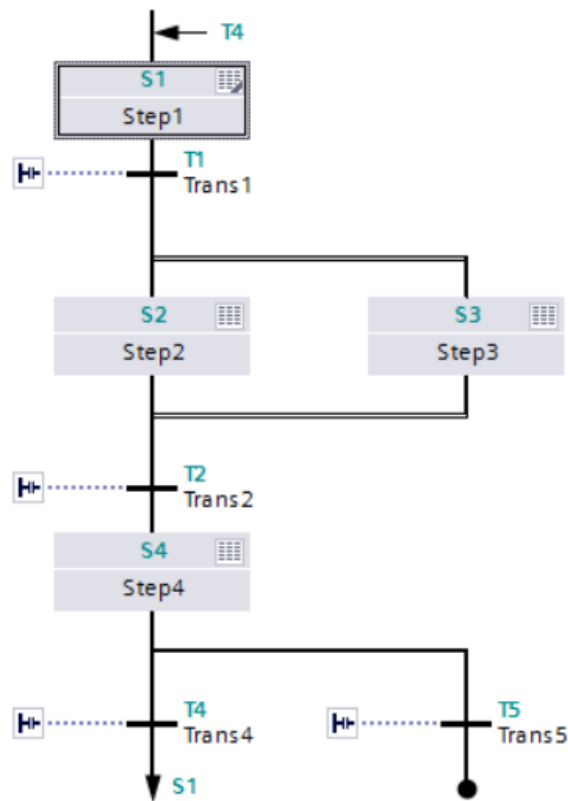
Network 4:				
Comment				
1	O	"PE_Low"		%I0.2
2	AN	"PE_High"		%I0.3
3	=	#Small_box		

Figur 2.6: Kod skriven som en instruktionslista i IL [13].

2.2.5 Sequential function chart (SFC)

Sekventiella funktionsblock är en steg-för-steg-metodik att bygga programkod som utför en rad uppgifter i sekvens. Programmeringsspråket är att föredra när flera scenarion/utfall är inblandade och när det handlar om större projekt som behöver delas upp i mindre delar [14]. En enkel struktur för ett SFC-program illustreras i Figur 2.7 med tillstånden/stegen S1-S4 och övergångsvillkoren T1-T2 och T4-T5.

Det speciella med SFC är kombinationen av de andra programmeringsspråken. Baserat på önskemål och typ av projekt kan övergångsvillkoren implementeras i exempelvis ST och LD. Övergångsvillkoren är villkor innan varje tillstånd som måste uppfyllas för att kunna exekvera det nästkommande tillståndet. I Figur 2.7 börjar programmet i det initiala tillståndet S1. För att programmet ska kunna exekvera nästa tillstånd, i detta fall S2 och S3, behöver övergångsvillkoret T1 uppfyllas.



Figur 2.7: Programstruktur för SFC med tillstånd och övergångsvillkor.

2.3 TIA portal

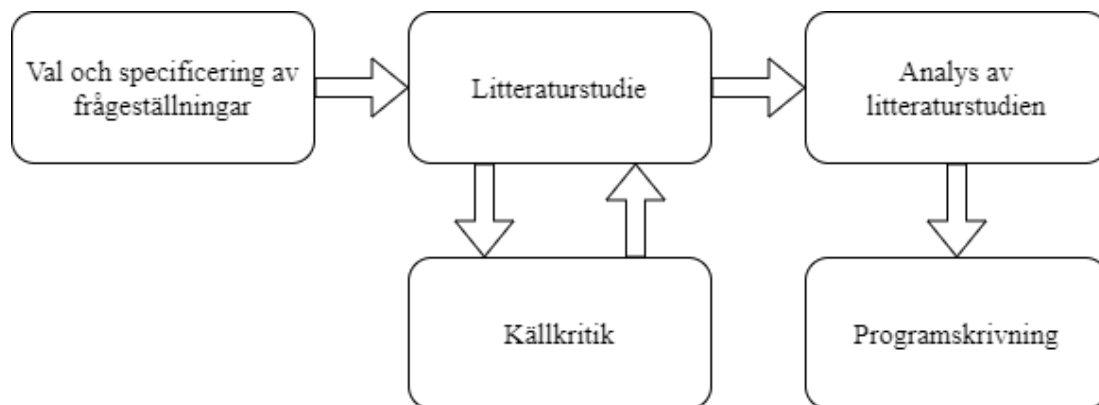
Totally Integrated Automation portal eller TIA portal är Siemens egenutvecklade utvecklingsmiljö med syfte att programmera och konfigurera Siemens PLC-enheter, HMI och PC-system. Genom integrering av olika funktioner och verktyg kan automationsprojekt utvecklas tillsammans med användning av programmeringsspråk som omfattas av IEC 61131-3. TIA portal ger möjlighet att simulera avancerade automationsprojekt innan de tillämpas i befintliga och verkliga system. Fjärranslutning och molntjänster för att övervaka och styra PLC-enheter, HMI och PC-system ingår också. Under examensarbetets praktiska del används TIA portal V15.1 som släpptes av Siemens i december 2018 [15].

2.4 Studio 5000

Studio 5000 är en integrerad utvecklingsmiljö utvecklad av Rockwell Automation där programmering och konfiguration av Allen-Bradley:s PLC-enheter sker. De fem programmeringsspråken som stöds av IEC 61131-3, tillsammans med HMI och nätverkskonfiguration, används för att skapa både komplexa och enklare automationsprojekt. Online-övervakning och redigering möjliggör realtidsuppdateringar och felsökning av Rockwell-system. Studio 5000 används vid konvertering i detta examensarbete och släpptes i november 2012 [16].

3 Metod

I denna sektion görs en generell och översiktlig beskrivning av tillvägagångssätt och metoder som används under examensarbetet. Sektionen är uppdelad i fem faser: val och specificering av frågeställningar 3.1, litteraturstudie 3.2, källkritik 3.3, analys av litteraturstudien 3.4 och slutligen programskrivning 3.5. Faserna illustreras i Figur 3.1 nedan.



Figur 3.1: Examensarbetet delades in i fem faser.

3.1 Val och specificering av frågeställningar

Fas I: Med hjälp av AFRYs samarbete genomfördes en kartläggning på problem och hinder som uppkommer inom industriell automation på företaget. Kartläggningen resulterade i att examensarbetet dedikerades till att redogöra för de mest förekommande PLC-fabrikaten inom företaget, Siemens och Mitsubishi Electric. Vad gäller specificering av frågeställningar bestämdes följande: teoretisk jämförelse av tillämpningen av IEC 61131-3 mellan PLC-fabrikaten. Den teoretiska studien skulle omfatta den generella tillämpningen av IEC 61131-3 och de mest tillämpade programmeringsspråken. Därefter skulle det mest multikompatibla programmeringsspråket hittas utifrån konvertering/översättning av funktionsblock skrivna i de olika programmeringsspråken.

P.g.a. otillräcklig tillgång till dokumentfiler om överensstämmelsen med IEC 61131-3 för Mitsubishi Electric kunde inte den teoretiska delen genomföras. Vad gäller konverteringen av funktionsblock med syfte att påvisa det mest multikompatibla programmeringsspråket krävdes programskrivning i ett programmeringsspråk och därefter översättning till de resterande. Detta steg skulle genomföras för samtliga programmeringsspråk.

Eftersom tiden för examensarbetet är begränsad och dokumentfiler för Mitsubishi Electric inte kunde hittas omformulerades frågeställningarna och en ny PLC-tillverkare introducerades - Rockwell Automation. Denna gång kunde den teoretiska delen genomföras för Siemens och Rockwell Automation där tillämpningen av IEC 61131-3 granskades tillsammans med de olika programmeringsspråken. Vad gäller att hitta

det mest multikompatibla programmeringsspråket vid konvertering ändrades denna frågeställning mot att konvertera olika programkoder i de programmeringsspråken som visar sig vara de mest tillämpade i den teoretiska delen. Programkoderna omfattade funktioner som var mest tillämpade hos respektive PLC-tillverkare och dessa programmerades i PLC-tillverkarnas utvecklingsmiljöer TIA portal och Studio 5000.

3.2 Litteraturstudie

Fas II: I det tidiga skedet av examensarbetet, men även under genomförandet, genomförs en litteraturstudie i form av inhämtning av begrepp och terminologi såsom PLC, den internationella standarden IEC 61131-3 och dess programmeringsspråk. Begreppen och terminologin från källorna utgör en god förståelse och stark grund för genomförandet av examensarbetet. Källornas trovärdighet och pålitlighet diskuteras i nästföljande avsnitt 3.3.

3.3 Källkritik

Fas III: Efter bearbetning av litteraturstudien i Fas II granskas källornas trovärdighet och pålitlighet.

Källorna [1], [11], [14] är från RealPars. RealPars är ett utbildningscenter beläget i Nederländerna med specialisering inom industriell automation, kontrollsystem och PLC-programmering. Företaget erbjuder kurser inom dessa områden med särskild vikt på PLC-programmering. Källan anses vara trovärdig och pålitlig utifrån följande kategorier: expertis och erfarenhet, utbildningslärande, kvalitetsinnehåll, samarbeten och referenser samt användarfeedback.

Källorna [8], [10], [13] är från SolisPLC. Liknande RealPars är SolisPLC en online-baserad utbildningsplattform som erbjuder kurser inom industriell automation och PLC-programmering. Målgruppen är allt från nybörjare till erfarna automationsingenjörer. SolisPLC tillhandahåller utbildningsmaterial och videolektioner som täcker PLC-programmering, HMI-design och SCADA-system. Källan anses vara trovärdig och pålitlig utifrån följande kategorier: expertis och erfarenhet, omfattande utbildningsmaterial, praktiskt fokus och positiv användarfeedback.

Källorna [2], [9], [12] är från Schneider Electric Blogg, Control respektive Motion Control Tips. Schneider Electric Blogg är Schneiders officiella blogg och erbjuder artiklar och nyheter inom energi, automation och hållbarhet. Som en av världens ledande leverantörer av energihantering och automatisering erbjuder Schneider Electric pålitliga och aktuella branschinsikter baserade på expertis och erfarenhet.

Control är en respekterad och erkänd tidskrift och webbplats inom processkontroll och industriell automation och erbjuder artiklar, nyheter, analyser och rapporter om de senaste teknologierna. Control är välansedd för sina insiktsfulla och tekniskt djupgående artiklar och skrivs av experter och yrkesverksamma inom det tekniska fältet.

Motion Control är en online-plattform som fokuserar på automation. Den erbjuder

nyheter, tekniska rapporter och studier om servomotorer, drivsystem och andra relaterade teknologier. Motion Control är en betrodd källa för ingenjörer och tekniker som arbetar inom detta område.

Källan [3] är en vetenskaplig artikel skriven av Bhakhar Rikin och är publicerad i ResearchGate. ResearchGate är en välkänd plattform för akademiker och forskningsartiklar. Den används av forskare och experter världen över och har dessutom ett gott rykte inom forskarsamhället, vilket bidrar till källans trovärdighet.

Källorna [4], [7] är böcker om PLC-enheter respektive industriell automation och är skrivna av etablerade experter inom dessa områden. Böckerna används ofta som läroböcker i tekniska utbildningar, bland annat på Lunds universitet. Vidare är utgivarna Elsevier och Springer de största och mest respekterade förlagen inom vetenskaplig och teknisk litteratur. Givet författarnas auktoritet och internationellt erkända och respekterade förlag anses källorna vara trovärdiga och därmed pålitliga.

Källorna [5], [15], [16], [17], [18] är från PLC-tillverkarna Mitsubishi Electric, Siemens respektive Rockwell Automatic. Dessa PLC-tillverkare anses i de flesta rapporter vara de populäraste och mest erkända i hela världen. Med lång erfarenhet, bred produktfortfölj, innovation och kvalitet samt tillförlitlighet anses dessa PLC-tillverkare vara trovärdiga.

Källorna [6], [19] är från IEC som är en internationell standardiseringsorganisation för elektrisk, elektronisk och relaterad teknik. IEC är globalt erkänd och tillämpad världen över, har bred täckning så som energiproduktion och industriell automation och dessutom samarbetar med andra standardiseringsorgan som ISO och ITU. Dessa faktorer bidrar till källans trovärdighet.

3.4 Analys av litteraturstudien

Fas IV: Denna del innebär analys av källorna och resulterar i vilka figurer och exempel som används i avsnittet 2. Vidare jämförs dokumentfilerna av Siemens och Rockwell Automation med syfte att granska tillämpningen av IEC 61131-3.

3.5 Programskrivning

Fas V: Den allra sista fasen omfattar programskrivning i TIA portal och Studio 5000. Programskrivning eller konvertering omfattar funktioner som både Siemens och Rockwell Automation tillämpar och illustrerar likheter och skillnader mellan PLC-tillverkarna. Syftet är att skriva program innehållande dessa funktioner i TIA portal, konvertera dem till Studio 5000 och slutligen jämföra programmen.

4 Analys

Denna del av rapporten innefattar en mer detaljerad bild av föregående avsnitt och de strategier som tillämpas för att besvara frågeställningarna beskrivna under 1.5.

Analysdelen delas in i underrubriker för att strukturera upp de olika arbetsmoment som ingår. Underrubrikerna är IEC 61131-3 tillsammans med TIA portal och Studio 5000, kompatibilitet och kompatibilitetsbrist mellan TIA portal och Studio 5000, de mest kompatibla programmeringsspråken och konvertering mellan utvecklingsmiljöerna.

4.1 IEC 61131-3 tillsammans med TIA portal och Studio 5000

Undersökningen av den internationella standarden IEC 61131-3 i form av egenskaper och funktioner genomförs där de olika 62 tabellerna från *Part 3: Programming languages* studeras [19].

4.1.1 TIA portal i enlighet med IEC 61131-3

Nästa steg fokuserar på analysen av dokumentet om överensstämmelse med IEC 61131-3 där samtliga tabeller från *Part 3: Programming languages* ingår [17]. Genom att studera tabellerna noggrant kan både antal möjliga instruktioner/funktioner samt antal uppfyllda/inte uppfyllda instruktioner för varje enskilt programmeringsspråk dokumenteras i en sammanställningstabell. Dessa värden presenteras slutligen i ett diagram och illustrerar en tydlig bild av hur Siemens TIA portal förhåller sig till IEC 61131-3. Liknande tabell görs för de enskilda programmeringsspråken LD, FBD, ST, IL och SFC där antalet uppfyllda funktioner sammanställs.

4.1.2 Studio 5000 i enlighet med IEC 61131-3

Samma procedur som för Siemens genomförs för PLC-tillverkaren Rockwell Automation där kompatibilitetstabellerna från *Part 3: Programming languages* i IEC 61131-3 studeras och dokumenteras i samma tabell som för TIA portal. Resultatet består av ett antal instruktioner som Studio 5000 tillämpar respektive inte tillämpar enligt dokumentet om överensstämmelsen med IEC 61131-3 [18]. Även detta resultat dokumenteras och presenteras i ett andelstalsdiagram. Precis som för TIA portal studeras de enskilda programmeringsspråken i Studio 5000 och jämförs mellan varandra. Dock förblir denna del av resultatet ofullständig p.g.a. otillräcklig dokumentering kring de enskilda programmeringsspråkens tillämpning i enlighet med IEC 61131-3.

4.2 Kompatibilitet mellan TIA portal och Studio 5000

Granskning av dokumentfilerna av Siemens och Rockwell Automation resulterar i ett antal uppfyllda tabeller för respektive PLC-tillverkare. För att kunna besvara var TIA portal och Studio 5000 är mest kompatibla undersöks först vilka tabeller respektive PLC-tillverkare uppfyller. Efter identifiering av de fullständiga tabellerna görs en

filtrering där de gemensamma tabellerna hittas. Kriteriet för att en tabell ska räknas som fullständig eller dominant är att samtliga funktioner inom denna måste vara uppfyllda. Detta kriterium beskrivs mer under avsnitt 5.1.

4.3 Kompatibilitetsbrist mellan TIA portal och Studio 5000

Genom att använda samma struktur som i föregående avsnitt undersöks de tabeller där inga funktioner uppfylls. Först hittas de tabeller med inga funktioner uppfyllda för TIA portal och därefter för Studio 5000. Efter filtrering observeras de gemensamma tabellerna och diskuteras längre fram i arbetet.

4.4 De mest kompatibla programmeringsspråken

Eftersom grunden för detta examensarbete är standarden IEC 61131-3 undersöks även de fem programmeringsspråken för båda PLC-tillverkarna. Genom analys av dokumenten om överensstämmelse med IEC 61131-3 noteras de funktioner som uppfylls av respektive programmeringsspråk [17], [18]. Dokumentationen av utvecklingsmiljöerna visar vilka funktioner som kan uppfyllas av vilket programmeringsspråk. För Studio 5000 visas endast vilken funktion som uppfylls utan att ange specifikt vilket programmeringsspråk det gäller.

4.5 Konvertering

När de fullständigt uppfyllda tabellerna hittas och de mest kompatibla programmeringsspråken identifieras genomförs konvertering av de enskilda funktionerna ur dessa tabeller i de olika programmeringsspråken.

Konverteringen innebär skrivning av program innehållande funktioner hittade inom 5.2. Programskrivning genomfördes först i TIA portal och därefter konvertering av programmen i Studio 5000.

5 Resultat

Detta avsnitt presenterar svaren till samtliga frågeställningar och är uppdelat och strukturerat efter ordningen på frågeställningarna beskrivna under 1.5. Strukturen följer även det som beskrevs tidigare i analysdelen under 4.

5.1 Den generella tillämpningen av IEC 61131-3

För att få grepp om hur noga Siemens och Rockwell Automation implementerar *Part 3: Programming Languages* inom IEC 61131-3 studeras de 62 tabellerna som ser till att aspekter såsom PLC, definition av data, de fem programmeringsspråken mm stöds.

I Tabell 5.1 nedan sammanställs tre olika antal: antal funktioner som uppfylls och inte uppfylls av PLC-tillverkarna samt det sammanlagda antalet funktioner som finns i IEC 61131-3. Målet med denna del av den statistiska undersökningen är att visa den generella tillämpningen av IEC 61131-3 utan kategorier.

Tabell 5.1: Antal funktioner som är implementerade resp. inte implementerade.

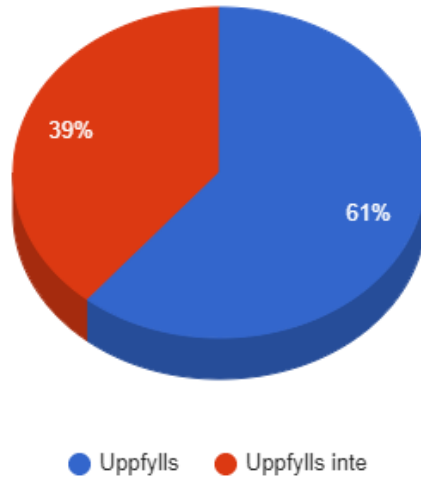
Funktioner	Siemens	Rockwell Automation
Stöds	551	251
Stöds inte	352	652
Summa	903	903

Kriteriet: För dokumentet utgivet av Siemens [17] anges vilka funktioner som uppfylls för *varje* programmeringsspråk. Till följd av detta bestäms kriteriet att en funktion skall räknas som uppfylld när minst ett programmeringsspråk uppfyller densamma. Alltså, enligt tabellen *Table 30 - Functions of time data types* i dokumentet uppfylls funktionen *1a: ADD* endast för programmeringsspråket ST i TIA portal. Följaktligen räknas denna funktion som uppfylld. För Rockwell Automations dokument om överensstämmelse med IEC 61131-3 [18] anges endast om en funktion är uppfylld, utan att ange vilket programmeringsspråk det gäller. För vissa funktioner finns det dock undantag där programmeringsspråk nämns. Det bestämda kriteriet under examensarbetet visar sig vara detsamma enligt Rockwell Automation där följande står i dokumentet:

"The IEC61131-3 specification defines five programming languages and a set of common elements. All languages are defined as optional, but at least one must be supported to claim compliance with the specification. [...]" [18]

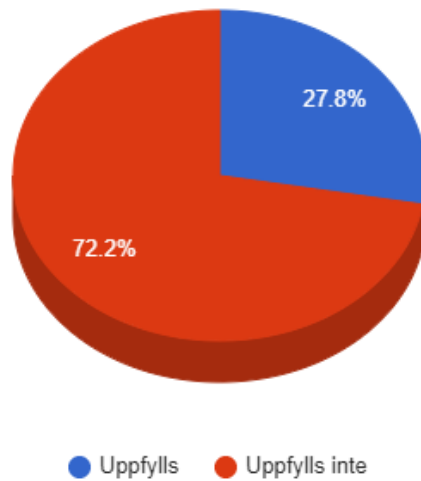
Någon mer grafisk sammanställning av Tabell 5.1 visas i Diagram 5.1 och 5.2 nedan där TIA portal visar en siffra på 61.0% implementerade funktioner av IEC 61131-3 just inom *Part 3: Programming Languages*. För Studio 5000 landar procentsiffran på 27.8%.

TIA portal



Figur 5.1: Grafisk presentation av uppfyllda funktioner i TIA portal.

Studio 5000



Figur 5.2: Grafisk presentation av uppfyllda funktioner i Studio 5000.

5.2 Kompatibilitet mellan TIA portal och Studio 5000

För att kunna besvara frågeställningen gällande kompatibiliteten mellan Siemens och Rockwell Automation krävs en detaljerad bild av den tillämpade standarden IEC 61131-3. Därför undersöks samtliga 62 tabeller inom *Part 3: Programming Languages* för att avgöra vilka områden/delar som PLC-fabrikaten har gemensamt.

Under avsnitt 5.2.1 och 5.2.2 presenteras och sammanställs de tabeller med samtliga funktioner uppfyllda i TIA portal respektive Studio 5000. Det avslutande avsnittet 5.2.3 för denna frågeställning filtrerar ut den gemensamma nämnaren mellan Siemens och Rockwell Automation. Denna del fortsätter i avsnitt 5.5 med praktisk konvertering av de funktioner som finns i dessa tabeller samtidigt som de diskuteras och jämförs mellan PLC-fabrikaten.

5.2.1 De dominanta tabellerna i TIA portal

Med hjälp av Siemens-dokumentet [17] hittas 20 tabeller med samtliga funktioner uppfyllda enligt kriteriet beskrivet tidigare i 5.1. Dessa tabeller anses vara den dominanta sidan av TIA portal och tabellerna listas ut nedan. Tabellerna omfattar bland annat standardfunktioner inom aritmetik och bit-shift, elementära datatyper, ladder-element och kantdetektering.

1. Table 1: Character set features.
2. Table 2: Identifier features.
3. Table 4: Numeric literals.
4. Table 6: Two-character combinations in character strings.
5. Table 8: Date and time of day literals.
6. Table 10: Elementary data types.
7. Table 23: Standard functions of one numeric variable.
8. Table 24: Standard arithmetic functions.
9. Table 25: Standard bit shift functions.
10. Table 26: Standard bitwise Boolean functions.
11. Table 27: Standard selection functions.
12. Table 28: Standard comparison functions.
13. Table 29: Standard character string functions.
14. Table 35: Standard edge detection function blocks.
15. Table 36: Standard counter function blocks.
16. Table 58: Graphic execution control elements.

17. Table 59: Power rails.
18. Table 60: Link elements.
19. Table 61: Contacts.
20. Table 62: Coils.

Antal tabeller med samtliga funktioner uppfyllda är 20 av 62.

5.2.2 De dominanta tabellerna i Studio 5000

Med samma kriterium hittas 10 dominanta tabeller för Studio 5000 med samtliga funktioner uppfyllda utifrån analysen av Rockwell-dokumentet [18]. Dessa tabeller presenteras i listan nedan.

1. Table 2: Identifier features.
2. Table 20: Use of *EN* input and *ENO* output.
3. Table 23: Standard functions of one numeric variable.
4. Table 24: Standard arithmetic functions.
5. Table 25: Standard bit shift functions.
6. Table 26: Standard bitwise Boolean functions.
7. Table 35: Standard edge detection function blocks.
8. Table 58: Graphic execution control elements.
9. Table 59: Power rails.
10. Table 60: Link elements.

Antal tabeller med samtliga funktioner uppfyllda är 10 av 62.

5.2.3 Filtrering av de dominanta tabellerna mellan TIA portal och Studio 5000

Utifrån resultaten från de föregående avsnitten 5.2.1 och 5.2.2 kan följande tabeller filtreras ut och de sammanställs nedan. Funktioner som omfattas inom dessa tabeller kommer sedan att konverteras i respektive utvecklingsmiljö och slutligen diskuteras där likheter och skillnader belyses.

1. Table 2: Identifier features.
2. Table 23: Standard functions of one numeric variable.
3. Table 24: Standard arithmetic functions.
4. Table 25: Standard bit shift functions.

5. Table 26: Standard bitwise Boolean functions.
6. Table 35: Standard edge detection function blocks.
7. Table 58: Graphic execution control elements.
8. Table 59: Power rails.
9. Table 60: Link elements.

Antal gemensamma tabeller är 9 av 62.

5.3 Kompatibilitetsbrist mellan TIA portal och Studio 5000

Denna del fokuserar på kompatibilitetsbristen mellan TIA portal och Studio 5000. De tabeller som sorteras under kompatibilitetsbrist är de tabeller med inga funktioner uppfyllda, oavsett programmeringsspråk.

Tabellerna med inga uppfyllda funktioner sammanställs för respektive utvecklingsmiljö i de följande avsnitten 5.3.1 och 5.3.2 nedan.

5.3.1 De svaga tabellerna i TIA portal

I de listade tabellerna nedan har inga krav uppfyllts i TIA portal oavsett programmeringsspråk. Funktionerna inom dessa tabeller kan inte konverteras eftersom de inte stöds och kommer därför endast att diskuteras kort i nästa avsnitt.

1. Table 31: Functions of enumerated data types.
2. Table 39: Program declaration features.
3. Table 44: Action block features.
4. Table 52: Instruction list operators.
5. Table 53: Function block invocation and Function invocation features for IL language.
6. Table 54: Standard function block input operators for IL language.

Antal tabeller med inga funktioner uppfyllda är 6 av 62.

5.3.2 De svaga tabellerna i Studio 5000

Som för TIA portal hittas tabeller med inga funktioner uppfyllda i Studio 5000. Tabellerna listas ut nedan och, precis som för TIA portal, tas dem upp i avsnittet för slutsatser under 6.

1. Table 3 a): Pragma feature.
2. Table 6: Two-character combinations in character strings.
3. Table 7: Duration literal features.
4. Table 8: Data and time of day literals.
5. Table 15: Location and size prefix features for directly represented variables.
6. Table 22: Type conversion function features.
7. Table 30: Functions of time data types.
8. Table 31: Functions of enumerated data types.
9. Table 39: Program declaration features.
10. Table 45: Action qualifiers.
11. Table 50: Task features.
12. Table 52: Instruction list operators.
13. Table 53: Function block invocation and Function invocation features for IL language.
14. Table 54: Standard function block input operators for IL language.

Antal tabeller med inga funktioner uppfyllda är 14 av 62.

5.3.3 Filtrering av de svaga tabellerna mellan TIA portal och Studio 5000

Tabellerna inom avsnitten 5.3.1 och 5.3.2 filtreras ut och följande tabeller kan erhållas:

1. Table 30: Functions of time data types.
2. Table 39: Program declaration features.
3. Table 52: Instruction list operators.
4. Table 53: Function block invocation and Function invocation features for IL language.
5. Table 54: Standard function block input operators for IL language.

Antal gemensamma tabeller är 5 av 62.

5.4 De mest kompatibla programmeringsspråken

Följande frågeställning syftar till att undersöka och hitta de mest tillämpade programmeringsspråken i enlighet med IEC 61131-3 och programskrivning i desamma. Programskrivningen genomförs i båda utvecklingsmiljöerna.

5.4.1 De mest kompatibla programmeringsspråken i TIA portal

Innan den praktiska konverteringen av funktioner som ingår i de dominanta, gemensamma tabellerna från 5.2.3 bestäms först det mest tillämpade programmeringsspråket. Utifrån analysen av dokumentet för TIA portal [17] erhålls följande resultat när det kommer till programmeringsspråkskompatibilitet till IEC 61131-3: ST stödjer 51.9% av funktioner, LD 50.5%, FBD 48.7%, IL 23.9% och slutligen SFC 2.9%. Tabell 5.2 nedan visar motsvarande resultat i antal uppfyllda instruktioner för respektive programmeringsspråk.

Tabell 5.2: Antal funktioner som uppfylls av programmeringsspråken i TIA portal.

Programmeringsspråk	Siemens
FBD	440 av 903
IL	216 av 903
LD	456 av 903
SFC	27 av 903
ST	469 av 903

Baserat på resultatet från tabellen ovan påvisas ST vara det mest tillämpade programmeringsspråket i TIA portal.

Om endast de uppfyllda funktionerna tas i beaktning (551 funktioner från avsnitt 5.1) erhålls följande procent: ST 85.1%, LD 82.8%, FBD 79.9%, IL 39.2% och SFC 4.9%.

5.4.2 De mest kompatibla programmeringsspråken i Studio 5000

Dokumentet för Studio 5000 [18] innefattar endast kommentarer för funktionerna där det inte nämns uttryckligen vilket programmeringsspråk det gäller förutom LD, ST och några enstaka gånger FBD. Tabell 5.3 nedan illustrerar därför endast resultat för programmeringsspråken LD och ST där ST låg på 22.4% medan LD följer strax efter med 20.9%.

Om endast de uppfyllda funktionerna tas i beaktning (251 funktioner från avsnitt 5.1) erhålls följande procent: ST 80.9% och LD 75.3%.

Tabell 5.3: Antal funktioner som uppfylls av programmeringsspråken i Studio 5000.

Programmeringsspråk	Rockwell Automation
FBD	otillräcklig information
IL	otillräcklig information
LD	189 av 903
SFC	otillräcklig information
ST	203 av 903

5.4.3 Bestämmelser och kriterier

Baserat på resultaten i föregående avsnitten 5.4.1 och 5.4.2 visas ST och LD vara de mest tillämpade programmeringsspråken i respektive utvecklingsmiljö, dock med ett frågetecken vad gäller Studio 5000 p.g.a. brist av specifika kommentarer i dokumentet. Detta innebär att konvertering av funktionerna inom de listade tabellerna från 5.2.3 genomförs i första hand i ST medan LD används som alternativt programmeringsspråk. Resultatet av procentalen för ST och LD för TIA portal och Studio 5000 sammanställs under Tabell 5.4 nedan som följer.

Tabell 5.4: Tillämpning av LD och ST i procent med samtliga funktioner i beaktning.

Programmeringsspråk	Siemens	Rockwell Automation
LD	50.5%	20.9%
ST	51.9%	22.4%

I Tabell 5.5 nedan presenteras tillämpningsprocenten för ST och LD utifrån de uppfyllda funktionerna enligt IEC 61131-3.

Tabell 5.5: Tillämpning av LD och ST i procent med uppfyllda funktioner i beaktning.

Programmeringsspråk	Siemens	Rockwell Automation
LD	85.1%	82.8%
ST	80.9%	75.3%

5.5 Konvertering

Härnäst presenteras tabellernas specifika funktioner och deras praktiska konvertering i båda utvecklingsmiljöerna. Vid avvikelser läggs kommentarer till som förklarar eventuella skillnader och andra anmärkningsvärda observationer. Programkod börjar alltid i TIA portal och därefter följer Studio 5000.

5.5.1 Funktioner inom de dominanta tabellerna

Tabeller som listades under avsnitt 5.2.3 presenteras här med konvertering av samtliga funktioner som ingår i desamma.

Table 2: Identifier features

Programkod 1: TIA portal.

```
1 IF #BUTTON715 THEN
2     #MOTOR715ON := 600; // Varvtal
3 ELSIF #BUTTON815 THEN
4     #MOTOR715ON := 1200;
5 ELSIF #CONVEYOR_Sw_5 THEN
6     #MotorSw5 := 300;
7     #track_Ax := 7;
8 ELSIF #_buttonX7 THEN
9     #_240VMOTOR7 := 2200;
10 ELSE
11     #MOTOR715ON := #DEFAULT_VALUE_MOTOR;
12     #MotorSw5 := #DEFAULT_VALUE_MOTOR;
13     #_240VMOTOR7 := #DEFAULT_VALUE_MOTOR;
14 END_IF;
```

Programkod 2: Studio 5000.

```
1 if BUTTON715 then
2     MOTOR715ON := 600; // Varvtal
3     elsif BUTTON815 then
4         MOTOR715ON := 1200;
5         elsif CONVEYOR_Sw_5 then
6             MotorSw5 := 300;
7             track_Ax := 7;
8         elsif _buttonX7 then
9             _240VMOTOR7 := 2200;
10            else MOTOR715ON := DEFAULT_VALUE_MOTOR;
11                MotorSw5 := DEFAULT_VALUE_MOTOR;
12                _240VMOTOR7 := DEFAULT_VALUE_MOTOR;
13            end_if;
```

Funktioner: Användning av stora bokstäver och tal (MOTOR715ON, BUTTON815), stora och små bokstäver med tal och understreck (CONVEYOR_Sw_5) samt stora och små bokstäver med tal som börjar med understreck (_buttonX7, _240VMOTOR7).

Avvikelser: TIA portal använder och anropar variabler med # och praktiserar stora bokstäver vid användning av samtliga satser: IF, WHILE, CASE o.s.v. I Studio 5000 är det valfritt med stora bokstäver där inga regler om variabelreferens tillämpas.

Table 23: Standard functions of one numeric variable

Programkod 3: TIA portal.

```
1 #overtravel_POS := ABS(#overtravel);
2
3 #randomNumberAdjusted := SQRT(#randomNumber);
4
5 #angle_0 := SIN(#actual_angle);
6 #angle_1 := COS(#actual_angle);
7 #angle_2 := TAN(#actual_angle);
8 #angle_3 := ASIN(#actual_angle);
9 #angle_4 := ACOS(#actual_angle);
10 #angle_5 := ATAN(#actual_angle);
```

Programkod 4: Studio 5000.

```
1 overtravel_POS := ABS(overtravel);
2
3 randomNumberAdjusted := SQRT(randomNumber);
4
5 angle_0 := SIN(actual_angle);
6 angle_1 := COS(actual_angle);
7 angle_2 := TAN(actual_angle);
8 angle_3 := ASIN(actual_angle);
9 angle_4 := ACOS(actual_angle);
10 angle_5 := ATAN(actual_angle);
```

Funktioner: Användning av generella (ABS, SQRT) och trigonometriska funktioner (SIN, COS, TAN, ASIN, ACOS, ATAN).

Avvikelser: Både TIA portal och Studio 5000 tillämpar funktionerna på samma sätt.

Table 24: Standard arithmetic functions

Programkod 5: TIA portal.

```
1 #gain_4_adj := (#gain_4 + 15) * 3;  
2 #gain_5_adj := ((273 - #gain_5) - 117) ** 2;  
3 #gain_6_adj := #gain_5_adj / 2;  
4 #gain_7_adj := #gain_6_adj MOD 3;  
5 #temp := #gain_7_adj;
```

Programkod 6: Studio 5000.

```
1 gain_4_adj := (gain_4 + 15) * 3;  
2 gain_5_adj := ((273 - gain_5) - 117) ** 2;  
3 gain_6_adj := gain_5_adj / 2;  
4 gain_7_adj := gain_6_adj mod 3;  
5 temp := gain_7_adj;
```

Funktioner: Användning av standardoperationer (ADD, MUL, SUB, DIV, MOD, EXPT, MOVE). I ST är dessa tillämpade med specialtecken (+, *, -, /, MOD, **, :=).

Avvikelser: Både TIA portal och Studio 5000 tillämpar funktionerna på samma sätt.

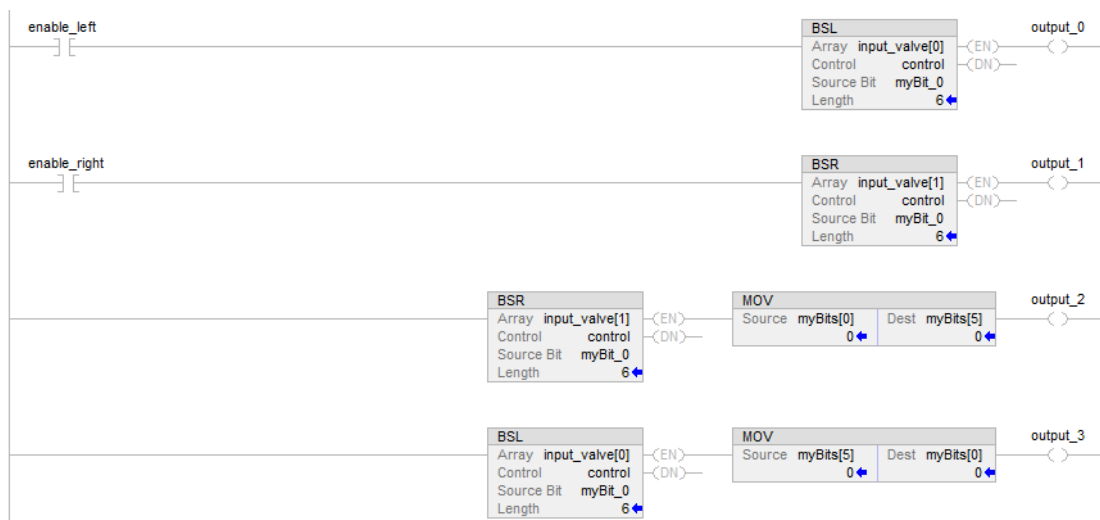
Table 25: Standard bit shift functions

Programkod 7: TIA portal.

```

1 // Left-shifted with SHL by N bits
2 #output_0 := #enable_left AND
3           SHL(IN := #input_valve, N := #N);
4 // Right-shifted with SHR by N bits
5 #output_1 := #enable_right AND
6           SHR(IN := #input_valve, N := #N);
7 // Right-rotated with ROR by N bits
8 #output_2 := #enable_right_rotation AND
9           ROR(IN := #input_valve, N := #N);
10 // Left-rotated with ROL by N bits
11 #output_3 := #enable_left_rotation AND
12           ROL(IN := #input_valve, N := #N);

```



Figur 5.3: Programkod för Studio 5000 i LD med shift-funktioner.

Funktioner: Användning av funktionen bit-shift med kommandon SHL, SHR, ROR, ROL. SHL och SHR flyttar N stycken bitar till vänster respektive höger. De bitar som inte får plats nollställs och sätts till värdet 0. ROR och ROL uppfyller samma funktion: flyttar N stycken bitar till vänster respektive höger med undandag att de bitar som inte får plats roteras till andra sidan (*circular*).

Avvikelser: Bit-shift-funktionerna SHL, SHR, ROR, ROL finns ej tillgängliga i ST för Studio 5000. Dessa funktioner finns i LD och används för att uppnå samma resultat.

Table 26: Standard bitwise Boolean functions

Programkod 8: TIA portal.

```
1 #_240VMOTOR7_ON := #sensor_1 AND #BUTTON715;
2 #sensor_1 := #safety_infrastructure OR #undetected_12;
3 #_240VMOTOR7_ON := #undetected_12 XOR #_buttonX7;
4 #undetected_12 := NOT #failure_detected;
```

Programkod 9: Studio 5000.

```
1 _240VMOTOR7_ON := sensor_1 AND BUTTON715;
2 sensor_1 := safety_infrastructure OR undetected_12;
3 _240VMOTOR7_ON := undetected_12 XOR _buttonX7;
4 undetected_12 := NOT failure_detected;
```

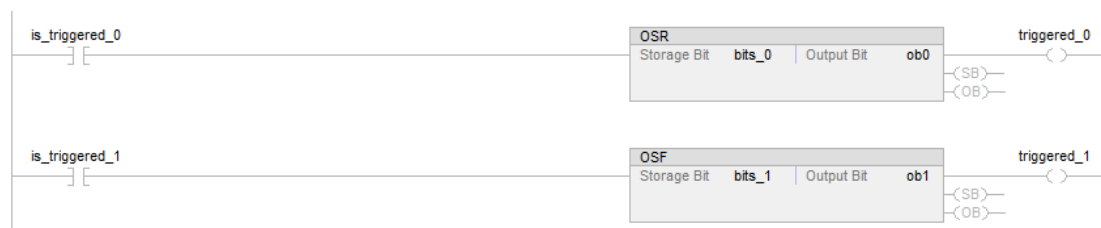
Funktioner: Användning av logiska funktioner/grindar (AND, OR, XOR, NOT).

Avvikelser: Både TIA portal och Studio 5000 tillämpar funktionerna på samma sätt.

Table 35: Standard edge detection function blocks

Programkod 10: TIA portal.

```
1 // Rising edge detector
2 "R_TRIG_DB"(CLK := #is_triggered,
3           Q => #triggered);
4 // Falling edge detector
5 "F_TRIG_DB"(CLK := #is_triggered_1,
6           Q => #triggered_1);
```

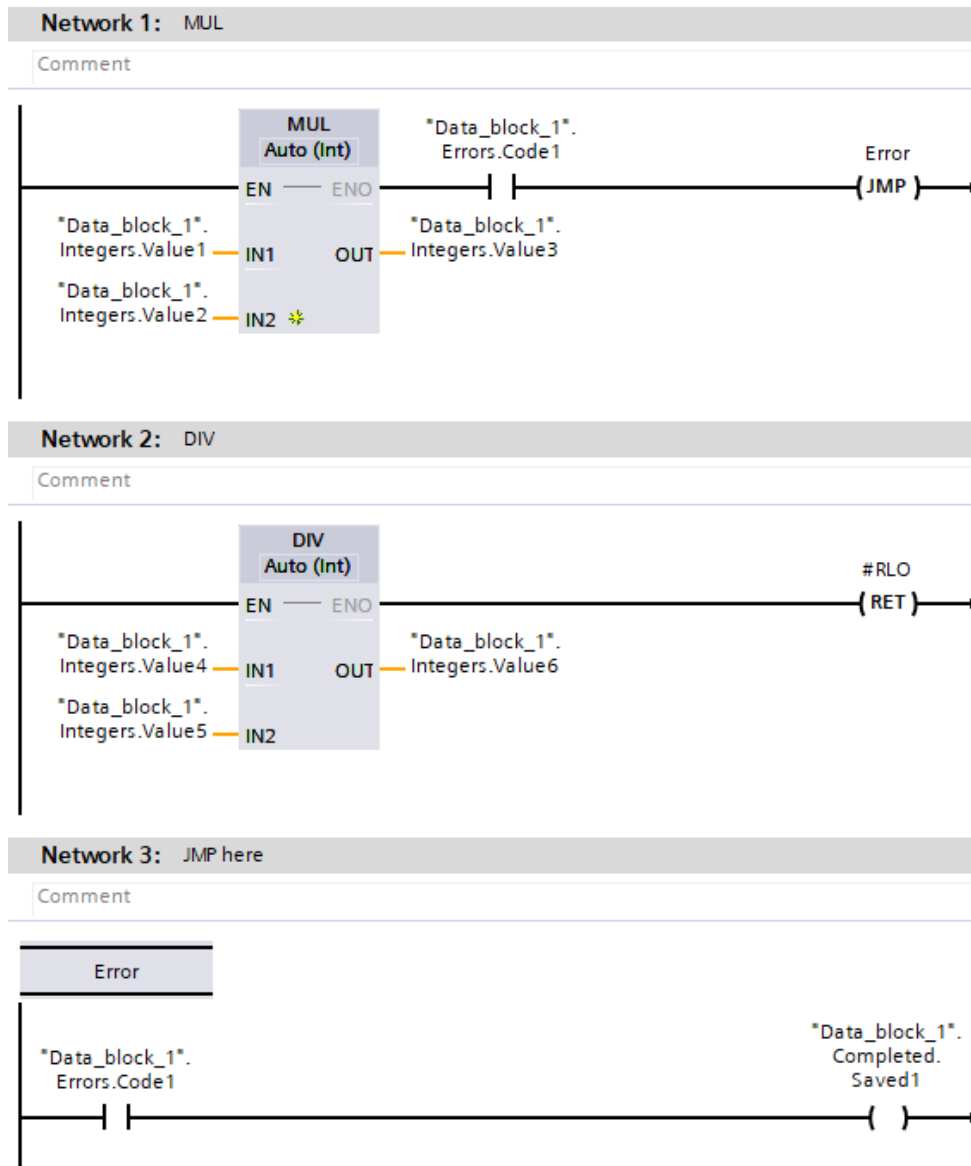


Figur 5.4: Programkod för Studio 5000 i LD med stigande och fallande funktioner.

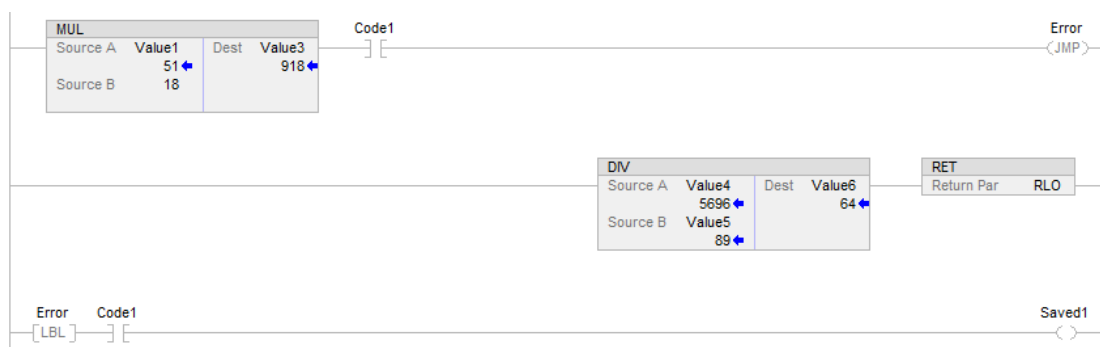
Funktioner: Användning av funktioner *Rising edge detector* och *Falling edge detector* där positiv och negativ kant detekteras.

Avvikelser: *Rising* "R_TRIG_DB" för TIA portal i ST och OSR-block för Studio 5000 i LD. *Falling* "F_TRIG_DB" för TIA portal i ST och OSF-block för Studio 5000 i LD. OSR och OSF står för *One Shot Rising* respektive *One Shot Falling*.

Table 58: Graphic execution control elements



Figur 5.5: Programkod för TIA portal i LD med funktionerna JMP och RET.



Figur 5.6: Programkod för Studio 5000 i LD med funktionerna JMP och RET.

Funktioner: Användning av *(un)conditional jump* och *(un)conditional return* med kommandon JMP och RET.

Avvikelser: Varken för TIA portal eller Studio 5000 finns dessa funktioner i ST. Tillämpning i LD resulterar i samma ladderdiagram.

Table 59: Power rails

Funktioner: Användning av *Left power rail* och *Right power rail*. Dessa hittas i Figur 5.5 och 5.6: vertikal linje till vänster om blocket MUL och linjen till höger om *assignment/output energize* JMP.

Avvikelser: Både TIA portal och Studio 5000 tillämpar funktionerna på samma sätt.

Table 60: Link elements

Funktioner: Användning av funktioner *Horizontal link* och *Vertical link*. Dessa hittas i Figur 5.5 och 5.6.

Avvikelser: Både TIA portal och Studio 5000 tillämpar funktionerna på samma sätt.

5.5.2 Funktioner inom de svaga tabellerna

De funktioner som ingår i tabellerna från avsnitt 5.3.3 presenteras nedan.

Table 30: Functions of time data types

Funktioner: Användning av funktioner relaterade till tid saknas i båda utvecklingsmiljöerna. Funktionerna är *Numeric and concatenation functions*: ADD (TIME), ADD_TIME, ADD (TIME_OF_DAY), ADD_TOD_TIME, ADD (DATE_AND_TIME), ADD_DT_TIME, SUB (TIME), SUB_TIME, SUB (DATE), SUB_DATE_DATE, SUB (TIME_OF_DAY), SUB_TOD_TIME, SUB (TIME_OF_DAY), SUB_TOD_TOD, SUB (DATE_AND_TIME), SUB_DT_TIME, SUB (DATE_AND_TIME), SUB_DT_DT, MUL (TIME), MULTIME, DIV (TIME), DIVTIME, CONCAT_DATE_TOD.

Type conversion functions: DT_DATE_TOD, DT_TO_DATE.

Table 39: Program declaration features

Funktioner: Användning av funktioner relaterade till deklaration av funktionsblock. Funktionerna är: RETAIN, NON_RETAIN, VAR_IN_OUT, VAR_GLOBAL...END_VAR, VAR_ACCESS...END_VAR, VAR_EXTERNAL inom PROGRAM, VAR_EXTERNAL CONSTANT inom PROGRAM, VAR_VAR_TEMP.

Table 52: Instruction list operators

Funktioner: Användning av funktioner kopplade till IL. Kommandon är: LD, ST, S, R, AND, &, OR, XOR, NOT, ADD, SUB, MUL, DIV, MOD, GT, GE, EQ, NE, LE, LT, JMP, CAL, RET,).

Table 53: Function block invocation and Function invocation features for IL language

Funktioner: Användning av funktioner inom IL. Exempel på några av funktionerna är:

Programkod 11: Exempel på tillämpning inom IEC 61131-3.

```
1 // CAL of function block with non-formal argument list
2 CAL      C10(%IX10, FALSE, A, OUT, B)
3 CAL      CMD_TMR(%IX5, T#300ms, OUT, ELAPSED)
```

Programkod 12: Exempel på tillämpning inom IEC 61131-3.

```
1 // CAL of function block with formal argument list
2 CAL C10 (
3     CU := %IX10,
4     Q => OUT)
5
6 CAL CMD_TMR (
7     IN := %IX5,
8     PT := T#300ms,
9     Q => OUT,
10    ET => ELAPSED,
11    ENO => ERR)
```

Programkod 13: Exempel på tillämpning inom IEC 61131-3.

```
1 // Function invocation with formal argument list
2 LIMIT (
3     EN := COND,
4     IN := B,
5     MN := 1,
6     MX := 5,
7     ENO => TEMPL
8 )
9 ST      A
```

Table 54: Standard function block input operators for IL language

Funktioner: Användning av funktioner inom IL gällande funktionsblock. Funktionerna är: S1, R, S, R1, CLK, CU, R, PV, CD, PV, CU, CD, R, PV, IN, PT (TP), IN, PT (TON), IN, PT (TOF).

6 Diskussion och slutsats

I detta avsnitt sammanställs och diskuteras de resultat och svar på frågeställningarna under rubrik 1.5. Underrubriken 6.1 sammanfattar svaren på frågeställningarna medan underrubrikerna 6.2 och 6.3 omfattar tillämpning av arbetet samt utvecklingsmöjligheter.

6.1 Svar på frågeställningarna

Tillämpningen av den internationella standarden IEC 61131-3 undersöktes för TIA portal och Studio 5000 där andelen av de tillämpade funktionerna enligt standarden visades ligga på 61.0% respektive 27.8%. Siffrorna visade en påfallande skillnad mellan PLC-tillverkarna där Siemens utvecklingsmiljö, TIA portal, tillämpade IEC 61131-3 med en markant marginal. Det visades dock att konverteringen av de fullständiga tabellerna, som refereras till de dominanta med samtliga funktioner uppfyllda, kunde genomföras utan några större skillnader mellan utvecklingsmiljöerna.

Vidare studerades tillämpningen av de 62 tabellerna i enlighet med IEC 61131-3 med syfte att undersöka kompatibiliteten mellan TIA portal och Studio 5000. Resultatet ledde till ett antal gemensamma tabeller där både TIA portal och Studio 5000 tillämpade samtliga funktioner inom dessa gemensamma tabeller. Av 20 fullständigt tillämpade tabeller för TIA portal och 10 för Studio 5000 hittades 9 gemensamma sådana. De gemensamma tabellerna gav insikt om vilka områden TIA portal och Studio 5000 är mest kompatibla. I andra termer visade tabellerna inom vilka områden en programkod kunde skapas i den ena utvecklingsmiljön och därefter skapa precis samma programkod i den andra utvecklingsmiljön.

En liknande studie genomfördes för de tabeller där varken TIA portal eller Studio 5000 hade tillämpat någon funktion överhuvudtaget. För TIA portal hittades 6 sådana och 14 för Studio 5000. Filtrering av dessa resulterade i 5 tabeller som delades mellan TIA portal och Studio 5000. Dessa tabeller illustrerade var kompatibiliteten mellan TIA portal och Studio 5000 brast och att dessa bör tas i beaktning vid framtida utveckling inom utvecklingsmiljöerna. Det är viktigt att notera att 3 av dessa 5 tabeller omfattar det något ålderdomliga programmeringsspråket IL, vilket nödvändigtvis inte kommer att resultera i någon bättre kompatibilitet mellan TIA portal och Studio 5000. Anledningen till det är att andra programmeringsspråk är mer användarvänliga och standardiserade, vilket diskuteras härnäst.

Utöver kompatibilitetsfrågorna är det även viktigt att nämna programmeringsspråken inkluderade inom IEC 61131-3. För TIA portal visades ST vara det mest tillämpade programmeringsspråket med hela 51.9%. Därefter följer LD med 50.5%, FBD 48.7%, IL 23.9% och slutligen SFC med endast 2.9%. För Studio 5000 visades ST och LD följa samma fotspår som hos TIA portal, men p.g.a. otillräcklig och ofullständig dokumentering i Rockwell-dokumentet [18] för Studio 5000 kunde resultat dokumenteras endast för dessa programmeringsspråk. ST ledde vägen med 22.4% med LD precis bakom med 20.9%. Baserat på resultatet för TIA portal gjordes antagandet att inget annat

programmeringsspråk för Studio 5000 toppade ST.

En närmare bild på de enskilda programmeringsspråken inom kategorin med uppfyllda funktioner kunde följande observeras:

För TIA portal: ST 85.1%, LD 82.7%, FBD 79.9%, IL 39.2%, SFC 4.9%.

För Studio 5000: ST 80.9% och LD 75.3%.

Med kännedom om vilka tabeller som gjorde TIA portal och Studio 5000 kompatibla genomfördes en konvertering där samtliga funktioner användes i kortare programkoder för att visa eventuella skillnader och avvikelser. Denna del hade i syfte att illustrera skillnader trots att både TIA portal och Studio 5000 tillämpade dessa i enlighet med IEC 61131-3. Programkoderna skrevs i det mest kompatibla programmeringsspråket ST och när dessa inte kunde genomföra en specifik funktion fortsatte programkodningen i det nästmest kompatibla programmeringsspråket LD o.s.v. Konverteringen blev ungefär densamma med några enstaka ändringar såsom praktisering av stora bokstäver för *statements* och *loops* i TIA portal men inte i Studio 5000, variabelreferens med # i TIA portal o.s.v. De största skillnaderna var Bit-shift- och Edge-detection-funktioner som kunde skapas i ST för TIA portal men inte för Studio 5000. För Studio 5000 skapades dessa funktioner i LD istället. Trots användning av olika programmeringsspråk uppfyllde funktionerna samma syfte som i ST för TIA portal.

Avslutningsvis undersöktes närmare de 5 tabeller med inga uppfyllda funktioner där 3 av dessa hörde ihop med IL. Eftersom IL endast tillämpade 23.9% av hela IEC 61131-3 är inte detta förvånande, men p.g.a. det alternativa textbaserade programmeringsspråket ST lär inga större ändringar äga rum för IL.

6.2 Tillämpning

Resultaten i detta examensarbete kan användas till grund för vidare undersökning och forskning mellan TIA portal och Studio 5000 men även mellan andra PLC-tillverkare. Ett företag som AFRY som arbetar med flera PLC-tillverkare kan dra nytta av den generella tillämpningen av IEC 61131-3 som genomfördes i detta examensarbete och baserat på det välja vilken PLC-tillverkare som kan passa bra beroende på situation. Om en automationslinje behöver bytas ut helt mot en ny PLC-tillverkare som från grunden var Rockwell Automation kan detta examensarbete vara ett tillräckligt svar till varför utbytet kan ske med Siemens PLC-enheter.

6.3 Framtida utvecklingsmöjligheter

Möjligheter till förbättring är breda och omfattar teoretiska och praktiska aspekter. De teoretiska skulle innebära mer undersökning av dokumentation för båda PLC-tillverkarna som skulle stärka den nuvarande undersökningen. De praktiska aspekterna innebär konvertering av alla funktioner i Studio 5000 för samtliga programmeringsspråk eftersom dessa inte nämns explicit i Rockwell-dokumentet [18].

Vidare är det viktigt att nämna att programmeringsspråken inte är de enda som ingår enligt IEC 61131-3 utan även mjukvaru- och kommunikationsmodellen. Genom att analysera dessa delar kan en klarare bild erhållas i form av skillnader i tillämpning av TIA portal och Studio 5000.

7 Terminologi

AFNOR - Association Française de Normalisation.

Dominant tabell - En standardtabell där samtliga funktioner är tillämpade av en PLC-tillverkare [19].

FBD - Function block diagram.

HMI - Human-Machine Interface.

IEC - International Electrotechnical Commission.

IL - Instruction list.

ISO - International Standards Organization.

ITU - International Telecommunication Union.

LD - Ladder diagram.

PLC - Programmable Logic Controller.

SFC - Sequential function chart.

ST - Structured text.

Svag tabell - En standardtabell där inga funktioner är tillämpade av en PLC-tillverkare [19].

TIA portal - Totally Integrated Automation portal.

8 Referenser

- [1] Kevin Cope. *What are the Major PLC Manufacturers*. RealPars, Rotterdam, October 2018.
- [2] Jan Kaihøj. *Öka anläggningens produktivitet och lönsamhet genom att modernisera automationssystemet*. Schneider Electric Blogg, May 2020.
- [3] Bhakhar Rikin. *IEC STANDARDS FOR PLC*. ResearchGate, February 2015.
- [4] William Bolton. *Programmable Logic Controllers*. Elsevier. 6:e upplaga. ISBN: 978-0-128-02929-9, December 2015.
- [5] Mitsubishi Electric. *FX Programming Manual, revision J*. ResearchGate, September 2015.
- [6] International Electrotechnical Commission. *IEC 61131-3:2013*. Webstore - IEC, February 2013.
- [7] Karl-Heinz John och Michael Tiegelkamp. *IEC 61131-3: Programming Industrial Automation Systems*. Springer. 2:a upplaga. ISBN: 978-3-642-12014-5, May 2010.
- [8] Vladimir Romanov. *PLC Programming / How to Read Ladder Logic Ladder Diagrams*. SolisPLC, April 2020.
- [9] Shawn Dietrich. *PLC Programming With Function Block Diagrams*. Control, August 2023.
- [10] Vladimir Romanov. *Introduction to Function Block Programming in RSLogix 5000*. SolisPLC, June 2020.
- [11] Gregory Duranso. *Basics of Structured Text (ST) Programming / Examples Applications*. RealPars, April 2021.
- [12] Miles Budimir. *What are Instruction Lists (ILs) for PLC programming?* Motion Control Tips, June 2017.
- [13] Redouane Ait Ali Yahia. *An Introduction to Statement List (STL) Programming in Siemens TIA Portal*. SolisPLC, July 2022.
- [14] Tom Mejer Antonsen. *Sequential Function Chart (SFC) Programming for Beginners*. RealPars, February 2024.
- [15] Siemens. *Totally Integrated Automation Portal*. February 2024.
- [16] Rockwell Automation. *Automation System Design Software*. September 2020.
- [17] Siemens. *Standards compliance according to IEC 61131-3 (3rd Edition)*. SIMATIC, April 2015.
- [18] Rockwell Automation. *Logix 5000 Controllers IEC 61131-3 Compliance*. Rockwell Automation-Litterature Library, March 2022.

- [19] International Electrotechnical Commission. *Programmable controllers - Part 3: Programming languages*. IEC, December 2013.