CODEN:LUTEDX/(TEIE-3083/1-75/(2019)

Industrial Electrical Engineering and Automation

Undersökning av PLC till PLC-kommunikation

Oliver Persson

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

Undersökning av PLC till PLC-kommunikation

LUNDS UNIVERSITET
Campus Helsingborg

LTH Ingenjörshögskolan vid Campus Helsingborg

Examensarbete:
Oliver Persson

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

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

Tryckt i Sverige
Lunds universitet
Lund 2019

Sammanfattning

Examensarbetet gick ut på att undersöka lösningar på PLC till PLC-kommunikation mellan olika PLC-system. Ändamålet är att lägga grund till en produkt som kan ta hand om all kommunikation mellan flera maskiner med olika PLC-system på en fabrikslinje.
Det utreddes vilka kommunikationsprotokoll som var lämpliga om PLC:er från Omron och Siemens användes. Efteråt undersöktes tredjeparts leverantörers lösningar på hur de kan hantera dessa kommunikationsprotokoll. När en lämplig lösning hittades så implementerades ett proof-of-concept-test för att se att lösningen fungerar som önskat och för att samla prestandastatistik.

De kommunikationssätt som var viktigast blev acyklisk kommunikation över Ethernet/IP med CIP- protokollet för Omron. För Siemens användes OPC UA.

Tre stycken leverantörer lämnade förslag. Representanter från Siemens, Beckhoff och B&R visade sina lösningar. Efter övervägning valdes Beckhoff lösningen att utredas vidare. Slutsatserna som kunde dras från implementeringen presenterades. De viktigaste slutsatserna var att Beckhoffs lösning är kraftfull, expanderbar och modulär med familjär programmering för många utvecklare.

Nyckelord: Industriell kommunikation, OPC UA, Beckhoff.

Abstract

This bachelor thesis consisted of examining solutions for PLC to PLC communication between different PLC systems. The end goal is to give a basis for a product that will handle all communications between multiple machines in a factory production line.
It was examined which communication protocols that was best suited to if PLCs from Siemens and Omron was used.
Afterwards third-party solutions were evaluated based on how they can handle the selected communication protocols.
When a suitable solution was found it was implemented as a proof-of-concept test that will prove that it can handle the communication and to collect performance statistics.

The communication protocols that was found to be most important was acyclic communication over Ethernet/IP using the CIP protocol on Omron. For Siemens OPC UA was found to be the best alternative.

Three companies presented their proposals. Represents from Siemens, Beckhoff and B&R presented solutions. After consideration Bekchoff was chosen for further development. Conclusions that were drawn from this implementation were presented. The foremost conclusions from Beckhoffs solution were that it is a powerful, expandable and modular with programming familiar to most developers.

Keywords: Industrial communication, OPC UA, Beckhoff.

Förord

Detta examensarbetet gav en mycket bra inblick i hur arbetslivet ser ut för en ingenjör. Jag har fått lära känna många olika automationssytem och fått erfarenhet med dessa. Det har varit ett omfattande arbete med många ingående sektioner.

Ecolean, som utfärdar examensarbetet, har varit väldigt tillmötesgående och har försett alla resurser som var nödvändiga inklusive arbetsplats, dator och mjukvara.

Det har varit mycket intressant att få lära sig så många nya PLC system och fått träna PLC programmering. Beckhoff, Siemens och Omron är vanliga system som jag har stor chans att stöta på senare och en bara fotnot att ha på mitt CV.

Innehållsförteckning

1 Inledning

Inledning som beskriver varför examensarbetet utfördes och vilka mål och motiveringar som fanns bakom.

1.1 Bakgrund

1.1.1 Vilka problem löser examensarbetet?

I en produktionslinje kan det finnas en mängd olika automationssystem bland de ingående maskinerna. Problemet är att dessa modeller av styrsystem och deras kommunikationsprotokoll inte alltid är samma typ av system som övriga maskiner i produktionslinjerna använder. Det betyder att två maskiner med olika kommunikationsprotokoll inte kan samarbeta direkt med varandra.

Examensarbetet ska fokusera på att hitta en prototyp som kan göra att maskiner med olika automationssystem kan kommunicera.

1.1.2 Vad är nytt med examensarbetet?

När PLC av olika fabrikat används kan de inte direkt kommunicera med varandra. Företaget har tidigare löst det problemet med direkt koppla in båda system mot hårdvara och reläer om de båda behöver ta del av samma signaler. Detta examensarbete ska undersöka om det finns en smidigare lösning genom att använda direkt PLC till PLC- kommunikation som endast behöver en koppling för att kunna skicka mycket data över ett nätverk. Dessutom kan kommunikation över nätverk möjliggöra att data centraliseras för enklare dataloggning, trendanalyser, rapportering osv.

1.2 Syfte

Examensarbetet ska ta fram en lösning som gör det möjligt för olika PLC:er att kommunicera med varandra även om de inte använder samma typ av protokoll för dataöverföring. Detta ska innebära att maskiner som är baserade på dessa PLC kan kommunicera med andra maskiner i samma fabrikslinje för att kunna få en överordnad styrning och övervakning av hela fabrikslinjen istället för att hantera enskilda maskiner.

1.3 Målformulering

Examensarbetet ska resultera i en prototyp som kan låta olika PLC direkt kommunicera med varandra. Kommunikation över Profinet, Ethernet/IP samt OPC UA (Open Platform Communications Unified Architecture) behöver kunna hanteras. Ett lyckat examensarbete ska resultera i att flera PLC av minst två fabrikat kan utbyta variabler med varandra (både läsa och skriva) via den framtagna prototypen. Dessutom ska prototypens prestanda mätas i form av ett test som verifierar kommunikationshastigheten för ett helt kommunikationsförlopp där alla ingående PLC har fått uppdaterade värden från de andra PLC. Prototypen bör kunna utföra en uppdateringscykel några gånger per sekund. Ingen exakt tid har specificerats.

1.4 Problemformulering

Följande frågor besvaras i examensarbetet:

1. Vilka kommunikationsprotokoll som används av utvalda PLC kan hanteras?
2. Finns det lösningar från leverantörer som kan anpassas?
3. Vilka automationssystem kan hanteras?
4. Vilka kriterier behöver lösningen bedömas efter?
5. Vilken är den bästa lösningen utifrån uppfyllnad av kriterierna i punkt 4?
6. Vilka begränsningar har lösningen?

1.5 Motivering av examensarbete

Examensarbetet ger en direkt inblick i hur det kan vara att arbeta som automationsingenjör. Examensarbetet är lämplig för programmet då det främst handlar om industriell automation men även digital kommunikation.
Det är en bra arbetslivserfarenhet och ger kunskaper och erfarenhet i en mängd aspekter av arbete på större industriföretag. Under examensarbetets gång kommer kontakt med andra företag och leverantörer ytterligare expandera inlärningsmöjligheter. Både allmänna och specificerad kunskap kan införskaffas.

1.6 Avgränsningar

Examensarbetet behöver endast stödja ett urval av PLC-modeller och deras kommunikationsprotokoll. Protokoll som behövs stödjas är Profinet, Ethernet/IP samt OPC UA. Andra typer av kommunikation behövs inte stödjas.
Lösningen ska testas på en testrigg som använder följande PLC-modeller: Siemens S7-1500 och Omron NJ5. Övriga modeller behövs inte testas.
Examensarbetet ska endast resultera i ett lyckat test att en lösning fungerar. Eventuell implementering ingår inte i detta examensarbete.

2 Teknisk bakgrund

Detta kapitel beskriver termer och teknik som används i examensarbetet.

2.1 Virtuell maskin

En virtuell maskin är ett sätt att köra flera operativsystem utan att behöva flera installationer. Virtuella maskiner möjliggör att köra ett helt separat system inuti ett annat system. Den virtuella maskinen har sina egna program och inställningar. På alla sätt kan den betraktas som sitt eget system. Maskinen är indirekt kopplad mot omvärlden via I/O(Input/Output) kopplingar till de fysiska gränssnitt värdsystemet har.

Virtuella maskiner passar väldigt bra till att köra program som har kompatibilitetsproblem med vissa operativsystem. Kompatibilitetsproblem elimineras helt eftersom en virtuell maskin med ett operativsystem som stöds fullt ut kan användas.
[1]

2.2 WireShark

WireShark är ett program med öppen källkod som kan avlyssna och visa datorkommunikation. WireShark kan hantera en enorm mängd kommunikationsprotokoll och visa användbar information om dessa.

I detta examensarbete har WireShark använts för att avlyssna paket som skickas över Ethernet på en utvecklingsdator. Mottagna paket visas i ett grafiskt gränssnitt som innehåller all data och information om paketen. Dessutom visas metadata såsom paketnummer och ankomsttider.
[2]

2.3 Kommunikationsgränssnitt

2.3.1 Profinet

Industriellt kommunikationsgränssnitt. Används av Siemens PLC:er för en mängd kommunikationsmöjligheter. Baserat på Ethernet kablar och kontakter. Siemens styrutrustning använder ofta exklusivt Profinet eller en likande variant baserad på seriell kommunikation kallad Profibus. Detta examensarbete har endast berört Profinet varianten.

2.3.2 Ethernet/IP

Industriellt kommunikationsprotokoll använt av Omron PLC:er. Används för att sköta PLC till PLC kommunikation. Stödjer både cykliska och acykliska kommunikationsmöjligheter. Använder Ethernetkontakter och Ethernetkablar. [4]

2.3.3 OPC UA

OPC UA står för OPC Unified Architecture som är en standard utvecklad av OPC Foundation. De erbjuder en standard för datalagring och kommunikation som är oberoende av vilket fabrikat som kommunikationspartnerna är. OPC UA erbjuder inbyggd säkerhet i form av kryptering och signering. [3]

2.4 Hårdvara

2.4.1 Siemens

Siemens är ett av de fabrikat av PLC:er som används i detta examensarbete. Intressanta modeller är PLC:er av typen SIMATIC S7-300 samt SIMATIC S7-1500. 1500 varianten är modernare och mer kraftfull. Den har en display och tryckknappar för enkel tillgång till viktig information. 1500 serien har även inbyggt stöd för OPC UA vilket kommer till nytta i detta examensarbete. Kommunikation är baserat på ProfiBUS och ProfiNET. Endast ProfiNET är av intresse i detta examensarbete.

2.4.2 Omron

Omron är det andra fabrikatet av PLC:er som används i examensarbetet. Modeller som används är av NJ501 serien. Dessa PLC:er är mer traditionellt utformade med en kontrollenhet utan knappar och paneler. Omron använder EtherCAT för I/O kommunikation men Ethernet/IP port för PLC-kommunikation. [4]

2.5 Kommunikationsprotokoll

2.5.1 Cyklisk kommunikation Omron

Det finns främst två sätt att kommunicera över Ethernet/IP med Omron. Den första metoden är att konfigurera cyklisk kommunikation mellan PLC:er. Med cyklisk kommunikation menas att kommunikationen sker konstant i cykler med en viss cykeltid oberoende av vad PLC:en är programmerad att göra.

För att upprätta kommunikationen krävs en konfiguration består av att rita upp uppkopplingen som den ser ut i verkligheten och sedan tilldela taggar till de variabler som skall överföras. Taggar läggs ihop och överförs cykliskt med en konstant cykeltid. [4]

2.5.2 Acyklisk kommunikation Omron

Det andra sättet att kommunicera över Ethernet/IP med Omron PLC:er är att använda acyklisk kommunikation över ett protokoll kallat CIP (Common Industrial Protocol), Common Industrial Protocol. Denna kommunikation styrs med funktionsblock i PLC-kod, detta gör kommunikationen acyklisk då den är oberoende av cykeltider utan kan utföras som önskat enligt koden.

Endast en av parterna i kommunikationen behöver ha kod. Den andra parten kräver endast publika variabler som möjliggör åtkomst över nätverk. [4]

2.5.3 Siemens TCP/IP

Siemens PLC:er stödjer kommunikation med TCP/IP (Transmission control protocol/Internet Protocol) paket. Dessa paket skickas med ett funktionsblock i PLC-kod. På samma sätt behövs paketet tas emot av ett annat funktionsblock hos PLC:en som är mottagare. Paketen är av vanligt TCP/IP utförande med header och data.

2.5.4 Siemens ISO-on-TCP

Alternativ variant av TCP/IP som används och konfigureras på exakt samma sätt. Använder en annan header för att snabba upp kommunikation. Det ger en direkt uppkoppling mellan två punkter eftersom mycket av adresseringen i TCP är nerskalad eller borttagen. ISO-on-TCP är mycket mindre vida spridd än vanlig TCP/IP den är baserad på. [11]

2.5.5 Siemens I-device

Siemens PLC:er har ett sätt att låta en PLC agera som en extra I/O modul mot en annan PLC. Detta betyder att flera Siemens PLC:er kan kopplas samman där en agerar master och direkt läser eller skriver till en annan PLC som agerar passiv slav. PLC:en som agerar slav kan fortfarande utföra sina vanliga uppgifter som PLC, kommunikationen sker cykliskt precis på samma sätt som en PLC kommunicerar mot sina I/O-moduler.

För att upprätta kommunikationen så konfigureras den ena PLC:en till vad Siemens kallar en I-device. Denna I-device sammanlänkas sedan med en annan PLC som agerar master och konfigurerar vilka variabler den vill läsa och skriva till.

2.6 Utvecklingsmiljöer

2.6.1 Omron Sysmac studio

För att programmera Omron PLC:er används deras utvecklingsmiljö kallad Sysmac Studio. Sysmac stödjer konfigurationer och programmering i Ladder blandat med Strukturerad Text.

2.6.2 Siemens TIA Portal

Siemens använder TIA Portal (Totally Integrated Automation Portal) som har en samling av programvara som krävs för att konfigurera och programmera deras PLC:er. PLC-program kan skrivas i Ladder, funktionsblock eller strukturerad text.

2.6.3 Beckhoff TwinCAT

Bekhoffs PLC:er skiljer sig från de andra i och med att deras PLC är baserad på PC. En PLC-runtime körs som bakgrundsprocess. Tillhörande utvecklingsmiljö är baserad på Microsoft Visual Studio och kallas TwinCAT. TwinCAT använder strukturerad text för PLC-programmering men har även stöd för C++.

3 Metod

Här beskrivs hur arbetsgången för examensarbetet gick till.

3.1 Utforska arkitekturen

Första praktiska momentet är att lära känna de system som skall behandlas i resten av examensarbetet. Följande kapitel behandlar de PLC:er och deras kommunikationsprotokoll som studerades.

3.1.1 Omron

Två stycken PLC:er införskaffades. Dessa var av modellerna Omron NJ501-1400 samt Omron NJ501-1420. De har identiska kommunikationsprotokoll och gränssnitt. 1st Ethernet/IP port och 1st EtherCAT port. EtherCAT är inte av större intresse för examensarbetet då det är ett protokoll för att snabbt kommunicera med I/O moduler och drivsteg.

Examensarbetet inriktade sig på kommunikation från styrsystem till styrsystem. Ethernet/IP är det gränssnitt som är relevant för det ändamål.

Tillgängligt är främst två typer av kommunikation som är cyklisk eller acyklisk.

3.1.2 Cyklisk kommunikation Omron

Cyklisk kommunikation upprättas mellan två Omron PLC:er via endast konfiguration. Några variabler skapades som tilldelas taggar som konfigurationen använder för att cykliskt överföra dessa variabler. Kommunikationen sker över Ethernet/IP med direkt Ethernet anslutning mellan PLC:er.

3.1.3 Acyklisk kommunikation Omron

Omron har stöd för acyklisk kommunikation där data skickas vid anrop i PLC-program. Protokollet som används för den här typen av kommunikation kallas CIP.

Ett program skapades på en NJ501-1400 PLC för att prova på de olika funktioner som finns tillgängliga. I programmet anropas block som startar en uppkoppling, utför kommunikationen och avslutar uppkopplingen.

Programmet skrevs från ett exempel i Omron manual. [4]

3.1.4 Siemens

Först införskaffades en Siemens S7-1500 kontroller och en S7-300. Sedan efter implementeringsproblem så ersattes S7-300 kontrollen med en andra S7-1500. Dessa Siemens kontroller använder PROFINET gränssnittet för att kommunicera utåt.

PROFINET stödjer en mängd kommunikationsprotokoll varav de som var av intresse för examensarbetet är: TCP/IP, ISO-on-TCP, OPC UA samt Profinet I/O med ProfinetRT.

3.1.5 TCP/IP med Siemens

Första kommunikationsprotokollet av intresse var TCP/IP.
TCP/IP överföringen behöver PLC-kod i både avsändaren och mottagaren.
En Siemens PLC programmerades till att skicka en variabel med ett färdigt funktionsblock.
En annan Siemens PLC programmerades till att ta emot denna variabel, även här med ett färdigt funktionsblock.

3.1.6 ISO-on-TCP med Siemens

ISO-on-TCP implementerades genom att använda samma block som används för TCI/IP kommunikationen. Ändringar från TCP/IP som behövde göras var att ändra en parameter så att programmet använder ISO istället och ändra konfigurationen i uppkopplingen så att den anger ISO även här.

3.1.7 I-device med Siemens

Siemens variant på cyklisk kommunikation görs genom att konfigurera en PLC som en I-device vilket gör den till en direkt slav till en annan PLC.
Med den konfigurationen skapades ett register med variabler som överfördes cykliskt med fast cykeltid mellan två Siemens PLC:er.

3.1.8 OPC UA med Siemens

Nyare Siemenskontroller i S7-1500 serien kan användas som server i den ganska nya standarden OPC UA. Det finns en specifik manual för denna funktion. [5]

För att prova serverfunktionaliteten i Siemens PLC:en så konfigurerades den till att agera som server. Sedan genererades ett certifikat. Servern konfigurerades så att den tillät en mängd olika krypteringssätt, inklusive ingen kryptering. Dessutom så konfigurerades den till att tillåta alla användare och alla deras certifikat.

En testklient som kallas UA Expert laddades ner på PC för att testa om servern går att nå och om den kan utbyta data både krypterat och okrypterat.

3.2 Leverantörer

Leverantörer bjöds in för att visa deras lösningar.

3.2.1 Inbjudan av leverantörer

Tre stycken leverantörer blev inbjudna att visa deras lösningar. Siemens, Beckhoff och B&R. Säljare och tekniker åkte ut på plats med hårdvara och förberedd mjukvara för att visa hur och om deras produkter var lämpliga för att hantera de kommunikationsmetoder som examensarbetet har inriktat sig mot. De presenterade sina förslag vid flertalet tillfällen.

3.2.2 Kriterier för jämförelse

En lista med kriterier togs fram som ligger till grund när leverantörernas produkter och lösningar skall utvärderas. Listan presenteras i analysdelen, kap 4.1.

3.3 Fördjupad undersökning av arkitektur

Under tidsperioden som leverantörer visade sina lösningar så utfördes fördjupade test av kommunikationsprotokollen och hur dessa fungerar. Mer komplicerade program skrevs för att prova några aspekter så som överföringshastighet, datastrukturer och datamängd. Testerna utfördes inte symmetriskt mellan de olika protokollen då de har olika förutsättningar och olika aspekter som är av intresse.

En Ethernet-switch införskaffades och konfigurerades till att spegla paket som skulle in till en PLC och skickade även dessa paket till PC så att WireShark kunde övervaka kommunikationen.

3.3.1 Övervaka Omron-kommunikation

Med hjälp av WireShark så övervakades den cykliska varianten av kommunikation med Omron. Programmet som skrevs för testet i Kap. 3.2.4 återanvändes i detta test.
Testet gav en inledande uppskattning av överföringshastigheten och hur kommunikationen går till. Tider uppskattas genom att jämföra och beräkna skillnader mellan paket över nätverket.

3.3.2 Övervaka Siemens-kommunikation

Även Siemens-testerna återskapades fast denna gång under övervakning med WireShark. Överföringsinformation för TCP, ISO-on-TCP och Profinet RT analyserades. Likadant som för Omron testerna gav dessa mätningar övergripande förståelse på hur kommunikationen ser ut och dess prestanda.

3.3.3 Strukturer

För att underlätta datasortering och överföring användes Strukturer för variabler. Möjligheter och användning av strukturer testades i båda systemen. Strukturer skapades som innehåller olika datatyper.

3.4 Utvärdering av lösningar

Lösningarna som leverantörerna kan leverera behövs utvärderas och jämföras.

3.4.1 Sammanställning

Resultatet från inbjudan av leverantörerna visade deras lösningars egenskaper, begränsningar och utförande.

Lösningarna sammanställdes och redovisades så att de kunde jämföras med varandra i mån om att välja en av dem att gå vidare med.

3.4.2 Val av vidareutveckling

Efter övervägande så valdes alternativet från Beckhoff. Detta alternativ valdes då det var väldigt lätt att komma igång med samt att det hade direkt stöd för de kommunikationssätt som lämpligast för Omron och Siemens. Vilka kommunikationsprotokoll som valdes diskuteras i analysdelen kap 4.2.1

3.5 Utveckling av slutgiltigt test

När lösningen var vald så kunde utvecklingen av proof-of-concept test påbörjas. Totalt tre stycken test genomfördes i Beckhoffs utvecklingsmiljö. Ett test som påvisar att Beckhoff kan agera mellanhand mellan Siemens PLC:er. Ett test som visar att Beckhoff kan agera mellanhand mellan Omron PLC:er samt slutligen ett fullständigt test där Siemens PLC:er och Omron PLC:er kunde kommunicera via Beckhoff. Resultaten diskuteras i kap. 5.4

3.5.1 TwinCAT

För att bli bekant med utvecklingsmiljön konsulterades Beckhoff Information Portal. [6] Projekten med koden som användes i testerna med Beckhoff lånades. Dessa projekt blev basen och inspirationen för egen kod.

3.5.2 OPC UA mot Siemens test

Först påbörjades implementationen av en OPC UA klient i TwinCAT. Detta är den mest omfattande delen av det slutgiltiga testet.

De nödvändiga blocken för OPC UA behövde testas och implementeras. Funktionsblock byggdes upp som har OPC UA funktioner så de kan individuellt testas och användas.

Sammanslagna funktionsblock skapades av de funktionerna som blev implementerade.
När alla funktionsblock var testade och fungerade så skapades ett test som utförde den kompletta dataöverföringen mellan de två Siemens PLC:er via OPC UA.
Detta test använde programkod för tidmätning som noterade och beräknade medelvärdet på överföringshastigheten vid varje cykel för att få en uppskattning av prestandan för dataöverföringen. Koden implementerades i en av de Siemens PLC:er som deltog i kommunikationen.

3.5.3 CIP mot Omron test

Samma test som för OPC UA utfördes med CIP mot Omron istället.

Ett test med samma metod för testet av OPC UA skapades. I Omron PLC:en översattes tidsmätningskoden från Siemens PLC:en som användes i testet innan. Minimum, maximum och medelvärde noteras precis som de andra testen.

3.5.4 Kombinerat test Omron och Siemens

När testerna för Omron och Siemens var klara startades det slutgiltiga kombinerade testet vilket är målet för examensarbetet.

Koden baserades på OPC UA testet men använde kod och konfigurationer från CIP testet.
Först så testades det att båda kommunikationssätten kunde köras samtidigt(parallellt). En mätning gjordes i detta skede med samma metod och kod som i Siemens testet.
Kommunikation till alla fyra PLC:er, två Siemens, två Omron, utfördes i samma cykel. Inga variabler utbyttes mellan de systemen.
Följande test lade till extra variabler i alla PLC:er så de läste vars minst en variabel från de andra PLC:na. Koden i TwinCAT uppdaterades och nu kör det kompletta testet med datautbyte från alla fyra PLC:er. Denna implementering låter Omron och Siemens PLC:er att utbyta data via TwinCAT vilket uppfyller examensarbetets mål.

Resultat noterades från alla tester och presenteras i kap 5.4

3.6 Källkritik

Här motiveras varför källorna som anges i kap 8 anses pålitliga.

3.6.1 Publikationer

[1] och [2] är publikationer tillgängliga från Lunds Tekniska högskolas bibliotek. De är publicerade av vetenskapliga förlag som har ansvar för att innehållet är korrekt.

3.6.2 Officiella manualer

Refererade manualer är skapade av de företag som även har skapat produkten manualen gäller för. De själva vet bäst hur deras egna produkter kan användas. Manualer är endast en guide och beskrivning på hur en produkt kan användas och risken är därmed låg för att informationen skulle vara medvetet felaktig. Eventuella fel kan korrigeras i nya utgåvor av manualen när fel hittas.

4 Analys

4.1 Framtagna kriterier

Dessa kriterier är de som togs fram tidigt i examensarbetet i syfte att ge en grund som förslag kan jämföras mot.

- Arbetsåtgång – Installation och utveckling

Den viktigaste punkten. Bedömning om hur mycket arbete kommer att behövas för att utveckla och implementera en lösning.

- Komplexitet– Kompetens och kunskap

Bedömer om lösningen behöver specialiserad kompetens för att kunna implementeras.

- Footprint– Storlek och Yta

Bedömer om lösningen tar mycket plats och om det kan bli problem att implementera i befintliga fabrikslinjer.

- Expanderingsmöjligheter– Nya överföringsmetoder eller nya maskiner

Bedömer lösningens möjligheter till att använda andra PLC-system än de som har behandlats i examensarbetet. Bedömer även om andra funktioner kan implementeras såsom dataloggning.

- Begränsningar

Bedömer vilka begränsningar som kan vara relevant för det ändamål som examensarbetet behandlar.

- Säkerhet

Bedömer lösningens hantering av IT-säkerhet där saker som kryptering och lösenord kan ingå.

4.2 Särskilda val

Presenterar de viktigare valen under examensarbetets gång.

4.2.1 Val av kommunikationsprotokoll

De flesta kommunikationsprotokoll utreddes av begäran från handledare. Under examensarbetets gång så sållades en del av protokollen bort om det fanns bättre alternativ.

Vid examensarbetets start valdes OPC UA, acyklisk kommunikation med CIP, TCP/IP, ISO-on-TCP, Profinet RT.
Flertalet protokoll föll bort på grund av brister eller att de var överflödiga.

Omron var begränsad till Ethernet/IP och därmed begränsad till de få tillgängliga kommunikationssätten. En variant på I/O kommunikation fanns tillgänglig men användes inte då en annan metod var av högre intresse. Acyklisk kommunikation med CIP visade sig vara mer lämpad för önskemålen.

Siemens kommunikation baserad på Profinet hade ett stort urval av tillgängliga kommunikationsprotokoll. De som undersöktes var OPC UA, TCIP/IP, ISO-on-TCP samt Profinet RT. (En del av Profinet I/O device) Den första metoden som sållades bort var ISO-on-TCP då den var en överflödig kopia av TCP/IP. Samma kodblock användes till både TCP/IP och ISO-on-TCP. Endast konfigurering skilde mellan implementeringen. TCP/IP är vanligare än ISO-on-TCP och därför lades all fokus på TCP/IP.
ProfinetRT verkade lovande då det endast krävde konfigurering. Dock är protokollet begränsat till cyklisk data och därmed täcker det inte alla behov. Därmed behövs ett annat kommunikationsprotokoll som hanterar acyklisk data. ProfinetRT lades på is eftersom det bara skulle vara en del av en lösning. Det kan komma att användas i framtiden men är inte nödvändigt för detta examensarbete.

Två av leverantörerna hade problem att använda TCP/IP mot Siemens. Det visade sig även att TCP/IP implementationen i Siemens endast kan skicka data. För att läsa data behöver den andra partnern skicka. Båda parter måste kunna agera som både master och slav.

Kvarvarande protokoll som fick utförliga implementeringar var CIP messaging för Omron och OPC UA för Siemens.

4.2.2 Val av lösning

Det val av leverantör som vidareutvecklades är nödvändigtvis inte den lösning som kommer att användas, men examensarbetet kommer ge en utförlig redovisning av lösningen och hur mycket arbete som krävs för en fullständig implementering.

Vid tidpunkten då examensarbetet behövde gå vidare med ett av förslagen så hade B&R inte kunnat lösa några av de kommunikationssätten som önskades. De behölls som en potentiell kandidat men för detta examesarbetes del behövs B&R inte tittas vidare på. De två andra förslagen presenterades.

På handledares begäran skedde vidareutveckling med Beckhoff-förslaget. Beckhoff-lösningen var att deras vanliga PLC-runtime system redan kan klara av de kommunikationsprotokoll som behövs.

4.3 Problem som uppstod

Här beskrivs de problem som uppstod under examensarbetet och visar hur dessa blev lösta.

4.3.1 Siemens S7-300

En av de första Siemes PLC:er som införskaffades var en äldre modell i S7-300 serien. Denna PLC hade problem att ansluta med TCP mot andra PLC:er. Den hade en äldre variant av överföring som hade andra överföringsblock. Även men hjälp av automationsutvecklare med erfarenhet i Siemens kunde inte TCP köras korrekt.

Lösningen är ett av de enklaste sätten att lösa problem med hårdvara. PLC:en byttes ut mot en modernare och kraftfullare variant. En andra PLC i S7-1500 serien användes som ersättning. Om det hade varit tid över så hade tid tilldelats till S7-300 PLC så att även denna kan användas med lösningen, dock så prioriterades detta bort.

4.3.2 Allmän felsökning och bugkorrigering

Projektet hade stort fokus på mjukvara med fysisk dataöverföring. Många buggar och fel uppstod under examensarbetet. Eventuella fel löstes efterhand som de uppstod. Flera metoder och resurser användes vid felsökning. Många fel kunde avhjälpas genom manualer och supportsidor av de relevanta tillverkarna och utvecklarna. Beckhoff, Omron och Siemens har alla var sin supportsida. [6] [7] [8]
Några problem löstes via diskussion med automationsingenjörer med erfarenhet av de specifika systemen som hade problem. En del exempelprojekt fanns att tillgå, både projekt från de tester som utfördes men även allmänt publicerade kodexempel. Som sista utväg kontaktades leverantörernas support direkt via Email.
Alla problem behövs inte beskrivas. De är en naturlig del av mjukvaruutveckling.

Nedan följer några exempel på de allvarligaste felen som uppstod.

- Strukturer och deras namn.

  Omron CIP read och write block klarar av att skicka en hel struktur i en exekvering. Vid de första testerna med skrivning och läsning av strukturer gavs error flagga och felkod. Felkoden motsvarade fel med datatyper. Det visade sig att typnamnen på strukturer och dess medlemmar behövde vara exakt samma i båda ändar. Strukturerna hade samma utformning men vissa namn på ingående variabler skiljdes åt. För att all data ska tolkas och hamna rätt måste strukturerna se exakt likadana ut i både sändare och mottagare.

  Felet avhjälptes genom att se över namnen på samtliga strukturer, understrukturer och medlemmar så de stämmer överens i alla kommunikationsmedlemmar.

- Virtuella maskiners koppling till fysiska portar

  När OPC UA och Ethernet/IP överföringsblock först testades hade de båda generiska fel i blocken. Främst gavs ”timeout, no response” eller motsvarande. Efter dubbelkontroll av adresser och variabler antogs att systemen antagligen inte kunde nå alls. Ett ping-kommando via Windows terminalfält visade att PLC:en kunde nås från PC men inte från den virtuella maskinen som kördes inuti den. Vid diskussion med handledare ändrades nätverksinställningarna för att direkt koppla den virtuella maskinen till den fysiska Ethernet-portens adapter och förbjuda uppkoppling via andra medel så som Wi-Fi. Detta avhjälpte omedelbart problem som hade uppstått med OPC-UA men CIP blocken hade fortfarande problem. Slutsatsen blev att kommunikation inte hade skett korrekt över Ethernet där partnern var kopplad utan troligtvis över ett av de andra nätverken.

- Överföringsblocken för CIP i TwinCAT saknade riktig feedback på när blocket var klart. Problem uppstod där antigen blocken repeterade konstant efter att bli klara eller så kunde de endast köra en gång. Lösningen blev att ha en extra variabel som styrde när blocket skulle startas om. Denna variabel styrs i huvudprogrammet där överföringsblocket ligger i ett funktionsblock med denna variabel som inparameter.

4.3.3 Olösta problem på grund av tidsbrist

Examensarbetet drog över den tid som ursprungligen hade planerats. En del av de problem som uppstod i sista minuten prioriterades bort och en kompromiss hittades istället för att lösa orsaken till problemen. Det var främst två problem som det inte allokerades tid till för att lösas. CIP överföringsblocken i TwinCAT kunde inte hantera boolean variabler. Detta verkade bero på att Omron system använder 16 bitars minnessekvenser till att hantera booleans medan TwinCAT använder 8 bitar. [9] [10]

För att testerna skulle blir klara i god tid så valdes det att utesluta boolean variabler från testerna.

Det andra problemet som inte togs omhand var läsning av OPC UA variabler i bulk. Så kallar ReadList där en hel lista variabler läses på en gång. Ideellt så skulle alla variabler i en OPC UA Server läsas med samma block men detta mål hanns inte med att fullföljas. Det visade sig att OPC UA ReadList visade felaktiga data om flera olika datatyper lästes samtidigt. Ett funktionsblock togs fram för att kunna läsa en lista med variabler men den gick oanvänd i de slutgiltiga testerna. Endast enskilda Read-block användes för varje variabel. Tyvärr hade examensarbetet i detta skede redan överskridit den ursprungliga planeringen och ett fullständigt komplett test uppnåddes inte.

5 Resultat

Här presenteras resultaten som togs fram under examensarbetets gång.

5.1 Kommunikationsprotokoll

Beskriver kommunikationsprotokollen enligt framtagna observationer och tester.

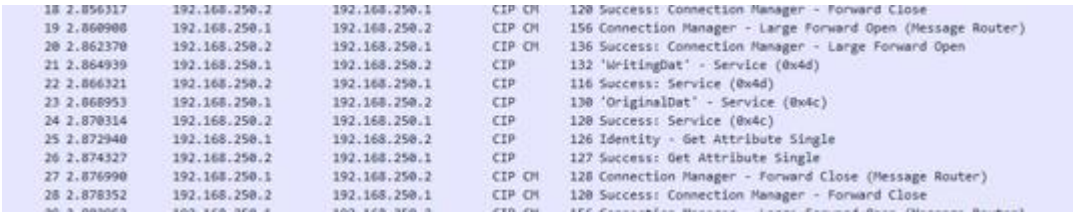
5.1.1 Cyklisk kommunikation Omron

Den cykliska varianten av kommunikation med Omrons PLC:er visade sig inte uppfylla de önskemål som fanns. Tidigt i examensarbetet lades denna typ av kommunikation till sidan och ersattes helt av den acykliska varianten.

5.1.2 Omron CIP acyklisk kommunikation

Till skillnad från den cykliska varianten av kommunikation så krävs minimal konfigurering men mycket programmering istället. Denna typ av kommunikation visade sig vara den lämpligaste metoden för kommunikation mot Omron PLC:er då det var den enda varianten i Omron som var anpassad för att kunna kommunicera mot andra enheter än endast Omron PLC:er.

Övervakning av Omron CIP visade konsekvent samma överföringsbeteende på alla olika funktioner. Connect, Disconnect, Read, Write och Request hade samma utseende och tider på nätverket. Varje sekventiell request kunde utföras varje 4 ms.
Figur 1 visar resultatet av detta test.



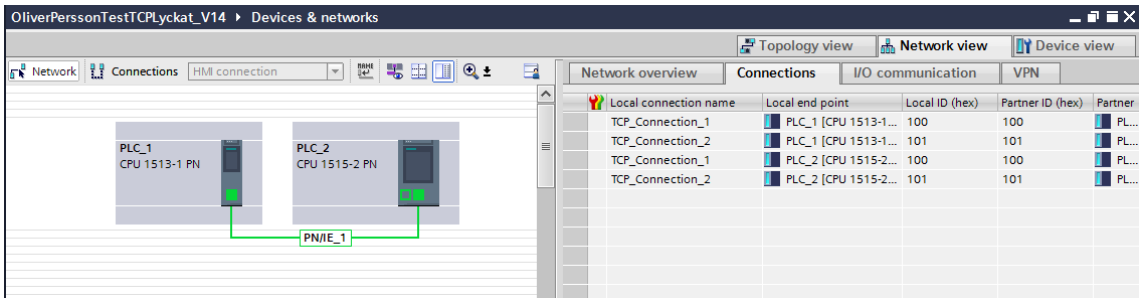
Figur 1. Omron CIP, Wireshark avlyssning

Övervakningen gav intrycket att CIP messaging är pålitligt och förutsägbart då avlyssningen presenterad i figur 1 visar att denna typ av kommunikation har väldigt små eller inga variationer i varken tid eller sekvens. Paketen som skickas över nätverket kommer i jämna tidsmellanrum och i samma ordning.

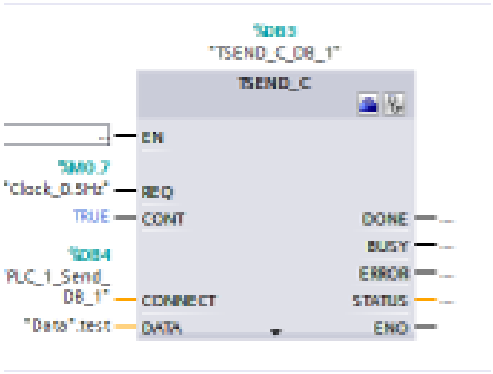
Koden för kommunikationen visas i appendix A.

5.1.3 Siemens TCP

TCP/IP kommunikation med Siemens PLC:er visade sig vara smidigt att hantera mot andra Siemens PLC:er men uppvisade problem mot andra fabrikat. Detta var den metod som leverantörer hade mest problem att hantera.
Fördelar med denna metod är att konfigurering på Siemens-sidan upprättas direkt i Siemens utvecklingsmiljö med förberedda val. Det mesta kan direkt konfigureras grafiskt som i figur 2.



Figur 2. Grafisk konfiguration TCP

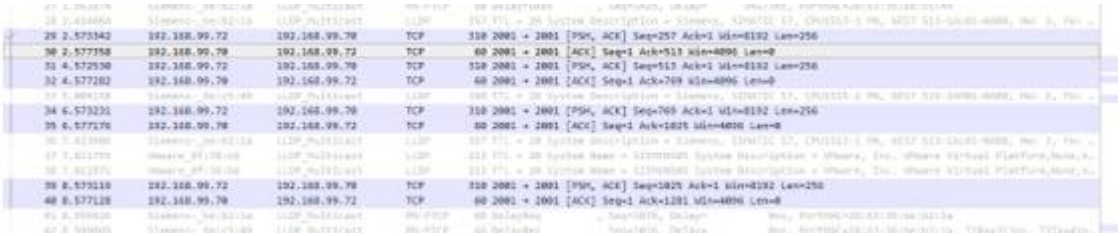


Figur 3. Send-block för TCP

Figur 3 visar det programblock som används för att programmera dataöverföringen mellan Siemens PLC:er.

Övervakning av kommunikationen med Wireshark visade att kommunikationen kunde variera kraftigt i hastighet och timing. Tiden innan ett ACK-paket returnerades skiljde sig åt varje gång kommunikationen stoppades och startades om igen. Reslutaten från endast övervakning kan dessutom inte anses fullständig då det är tiden tills att data blir tillgänglig hos mottagaren som är av intresse, inte tiden tills att ACK-paket returneras.

Figur 4 visar en av de mätningar som utfördes.



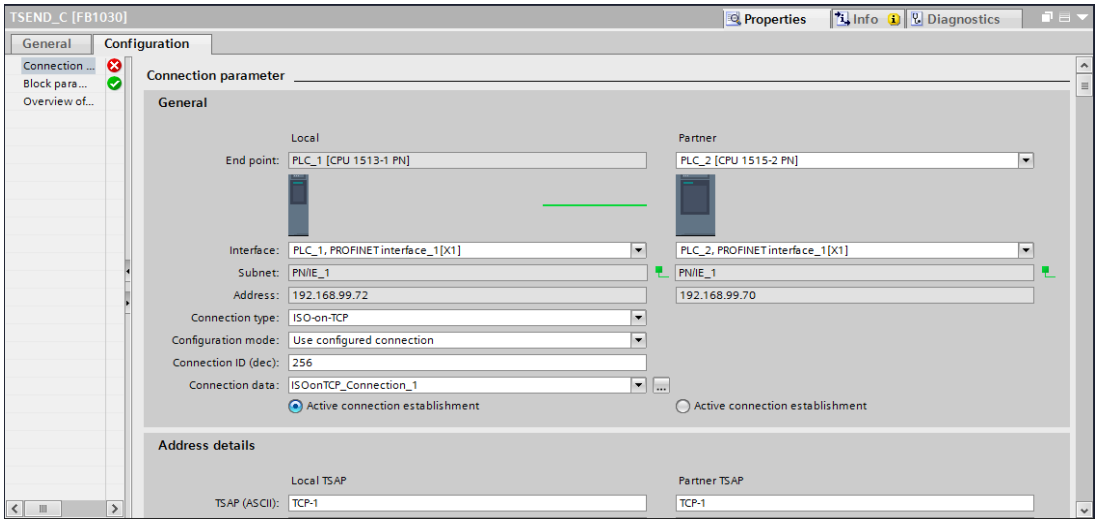
Figur 4. TCP, Wireshark-avlyssning

Kommunikation med stora variabelmängder visade att ACK-paket inte var synkroniserade. Vid stora dataöverföringar skickades ett ACK för varje två datapaket.

Programmet som utförde kommunikationen körde inte kommunikationen i maximal cykeltid då kommunikationen kördes med trigger från PLC:ens interna klocka. Detta var en brist i koden för testet. Ingen lösning på problemet hittades då det prioriterades bort. Om problemet hade lösts så hade överföringsprestanda bättre kunnat noteras.

5.1.4 Siemens ISO-on-TCP

ISO-on-TCP är en specialiserad variant av TCP. Detaljerna skiljer sig vilket är beskrivet i en artikel. [11]
ISO-on-TCP kommunikationen upprättas identiskt som den vanliga TCP/IP kommunikationen. Enda skillnaden i implementeringen är att ISO-on-TCP väljs vid konfigureringen av uppkopplingen och programblocket visat i figur 5. Skillnaden på nätverksnivå mellan de två protokollen är hur paketet som skickas ser ut och hanteras. De exakta skillnaderna studerades inte i detta examensarbetet.

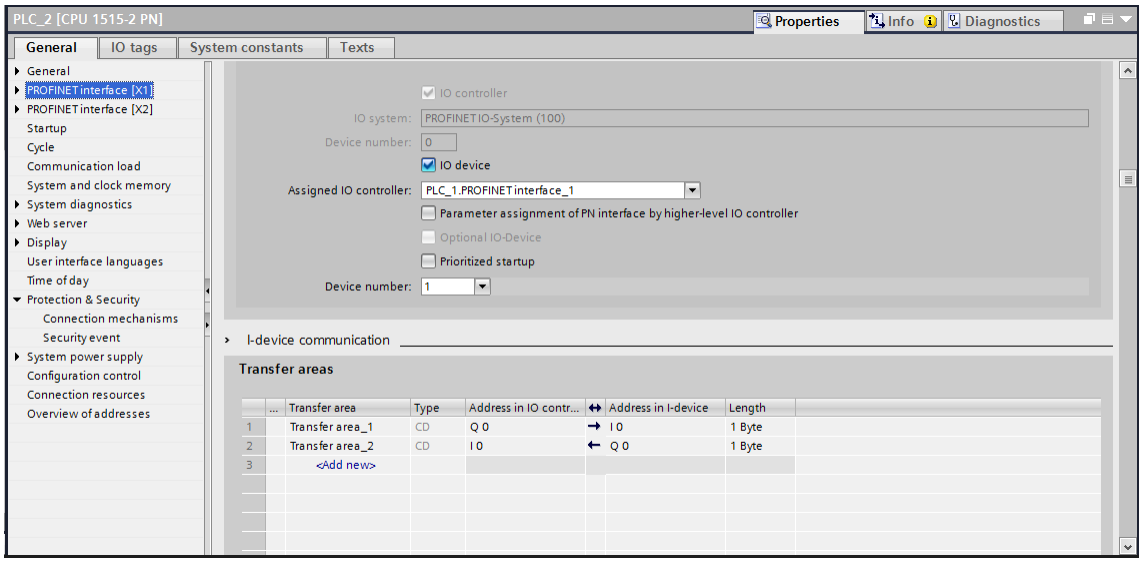


Figur 5. ISO-on-TCP configuration för TSEND_C block

Av de fabrikat som examensarbetet har berört var det endast Siemens system som kunde använda ISO-on-TCP och examensarbetet går ut på att hitta sätt att kommunicera mot andra system så denna kommunikationsmetod valdes bort.

5.1.5 Siemens I-device

Konfigurationen visas i figur 6. Med denna metod kan stora processer förenklas genom att alla I/O samlas i en PLC som kan skicka alla parametrar vidare till en överliggande PLC. På detta sätt kan flera PLC:er kopplas samman till ett nätverk i flera nivåer. Denna kommunikationsmetod visade sig effektiv men hade samma problem som andra metoder i att den endast hanterar cykliska data. Prototypen behöver kunna hantera acyklisk data för att kunna köra synkroniserade överförings cykler mellan alla PLC:er. Detta för att överföringen inte blir överbelastad och att datan i PLC:erna är synkroniserad så att det inte uppstår problem med variationer i olika PLC:er.



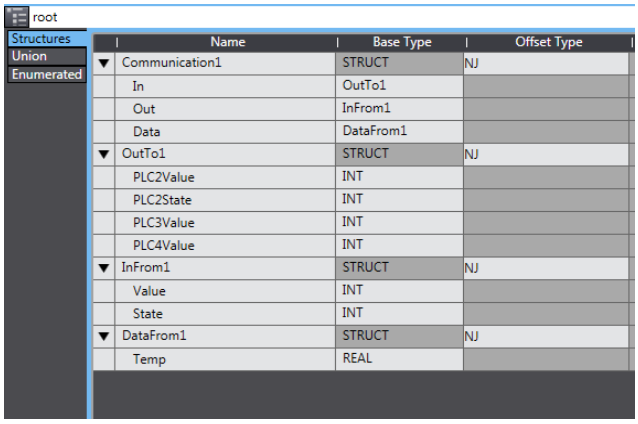
Figur 6. Siemens IO Device konfiguration

5.1.6 OPC UA

OPC UA visade sig vara den mest användbara och mångsidiga metoden för kommunikation med Siemens. Den är designad för att kunna ha standardiserad kommunikation oavsett fabrikat på PLC:er.
Det var dock en komplicerad metod som krävde mycket förberedelser med certifikat, valbar kryptering och variabeladressering. Trots dessa nackdelar valdes OPC UA som den lämpligaste metoden för kommunikation mot Siemens.

5.2 Användning av strukturer

Strukturer förenklar datasamlingar och i vissa fall förenklas kommunikationen då hela strukturer kan skickas med endast en begäran. Understrukturer kan användas till noggrannare datasamlingar. Figur 7 visar en implementering av strukturer med Omron.



Figur 7. Strukturer med Omron

5.3 Utvärdering av leverantörer

Leverantörerna jämförs enligt kriterierna framtagna i kap 4.3.1.
Den fullständiga beskrivningen innehåller intressanta egenskaper, uppfyllnad av kriterier samt för- och nackdelar.

5.3.1 Beckhoff

Egenskaper
• PC-baserad med Windows
• PLC-runtime körs på PC
• Flera PLC:er kan köras på samma enhet
• Satsar på EtherCAT
• Kör TwinCAT integrerat i Microsoft Visual Studio
• Hanterar CIP över Ethernet/IP, OPC UA, Siemens I-device
• TCP/IP mot Siemens är inte bevisat att fungera
• Mycket programmering i PLC
• Använder importerade bibliotek
• Konfigurerar portar och drivrutiner för OPC UA, Ethernet/IP, ProfiNet
• Erbjuder expandering med Gigabit switch, 8portar med 100mb vars.
• Mycket initial programmering
• Återanvändbar kod och konfigurering

Jämförelsekriterier
Arbetsåtgång – Mycket kodning. Finns färdiga block för kommunikationen men de måste implementeras i program.
Komplexitet – Nödvändig kunskap i Strukturerad text eventuellt C++. TwinCAT 3 är själva systemet som används och behöver speciell kunskap
Footprint – Låg, endast en PLC/PC enhet
Expandering – Windows PC, kan köra massa andra program och runtimes.
Begränsningar – OPC UA och ProfinetRT/ Profinet I device är de enda kommunikationssätten bevisade mot Siemens men det finns några fler av intresse. Prestanda är hårdvarabundet.
Säkerhet – Ger ingen extra säkerhet. Säkerhet sköts på nätverket med exempelvis brandväggar. OPC UA har inbyggda möjligheter till säkerhet, både autentisering och kryptering erbjuds.

Fördelar
• Tidigare programmeringserfarenheter kommer till nytta
• Klassiska PLC språk men även stöd för C++
• Kort utbildning
• Oändligt förnybar gratis licens vid programutveckling
• Färdiga metoder och bibliotek

Nackdelar
• Mycket kod
• TCP/IP mot Siemens inte bevisat
• Ingen integrerad säkerhet

5.3.2 WinCC OA

Egenskaper
• Mjukvara som kan köras på en mängd system, främst linux
• Säljes som licenser
• SCADA (Supervisory Control And Data Acquisition)
• Skalbara lösningar och licenser
• Använder Managers, är moduler med var sin uppgift
• Event manager tar hand om de andra managers
• Database manager sköter datalagring
• Drivers sköter uppkoppling mot andra enheter
• OPC UA driver, S7 driver m.m.
• Datautbyte sker med datapoints och funktioner för dessa
• Kan använda script
• Eventbaserad, hämtar endast data vid förändring
• Licens är hårdvarabunden

Jämförelsekriterier
Arbetsåtgång - Skalbart beroende på önskad nivå av system. Mycket arbete kan läggas på utveckling av färdiga moduler, konfigurerade program och script. Detta minskar framtida arbete men behövs betydligt med initialt arbete.
Komplexitet – Specifik kunskap om WinCC OA (WinCC Open Architecture). Använder inte programmering utan kräver kunskap om att hantera programmiljön.
Footprint – Hårdvaraoberoende, beror på val av hårdvara.
Expandering – Flera licenser kan läggas till. Andra WinCC OA system kan kopplas samman. De individuella systemen kan inte expanderas.
Begränsningar – Förslaget är begränsat följande drivers, S7, S7+, OPC UA, Ethernet/IP
Leveranstid - 5 dagar enligt hemsida [12]
Säkerhet – SSL kryptering

Fördelar
• Direkt användare gränssnitt. Inte mycket kod.
• Kan installeras på en mängd olika hårdvara
• Skalbart – enkelt att lägga till befintliga moduler
• Sammankopplingsbart med andra WinCC OA system
• Hämtar data endast vid förändring genom jämförelser
• Databashantering ingår
• 1st HMI (Human Machine Interface) ingår
• Strukturerad data som är kopplad till maskinen denna data kom från
• Beräkningsfunktioner direkt på inkommande data
• Färdiga matematiska formler
• Enkelt att återanvända tidigare arbete
• Native S7 driver mot Siemens PLC

• WinCC OA agerar alltid master. Inget arbete krävs i PLC kontrollerna

Nackdelar
• Unik miljö, kräver specialiserad kunskap
• Mycket inledande arbete
• Licens är hårdvarabunden
• Designat för att göra mycket mer än vad behövs
• Behöver bli partner för full tillgång och support
• Relativ lång utbildning

5.3.3 B&R

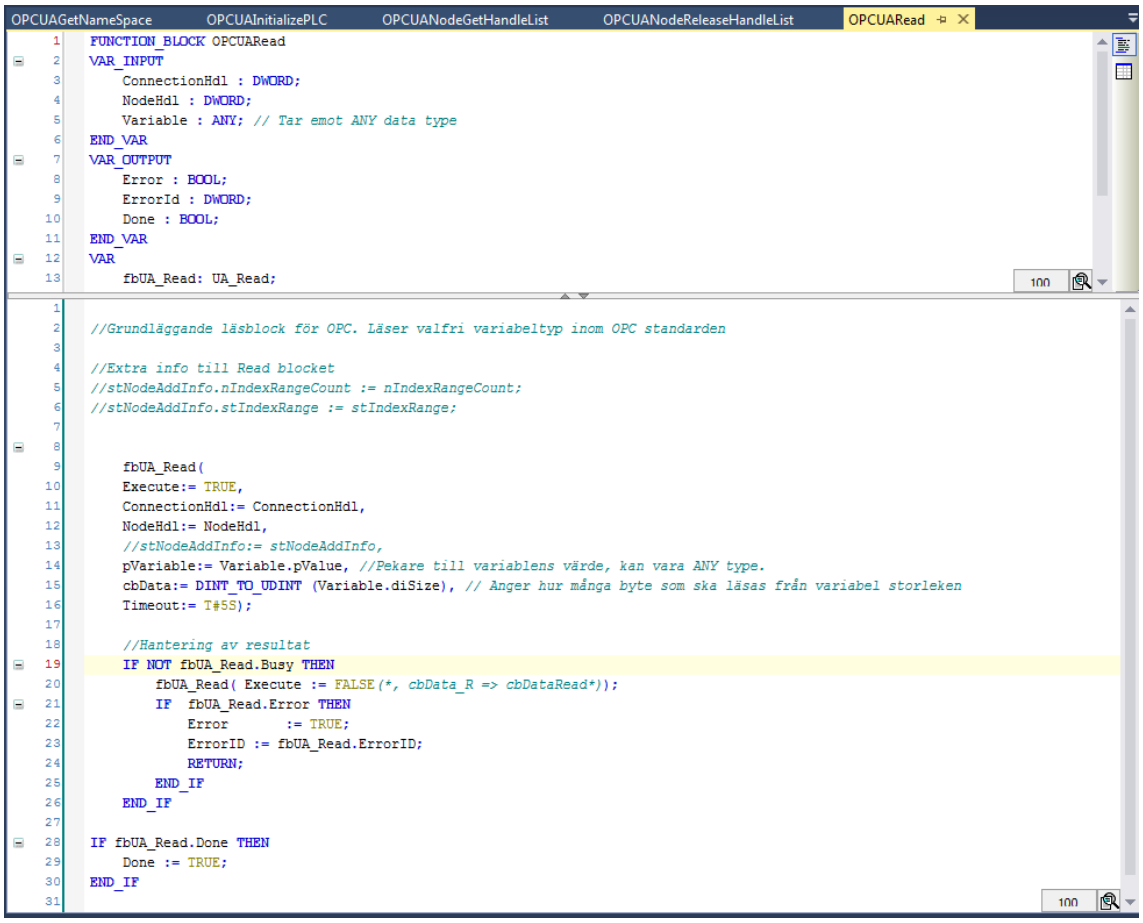
B&R ingick inte i bedömningen då de inte hade löst kommunikation mot Omron vid det skede som bedömningen behövde göras.

5.4 Slutgiltigt test Beckhoff med TwinCAT 3

Slutgiltiga resultaten för examensarbetet som ger fullständig utvärdering av Beckhofflösningen.

5.4.1 OPC UA med TwinCAT

OPC UA testet gick smidigt då varje enskild del kunde utvecklas och testas för sig själv. Alla OPC UA-block som fanns tillgängliga i TwinCAT implementerades som funktioner. Figur 8 visar implementeringen av OPC UA Read som ett funktionsblock som ett exempel. Lista för samtliga implementeringar hittas i appendix G.

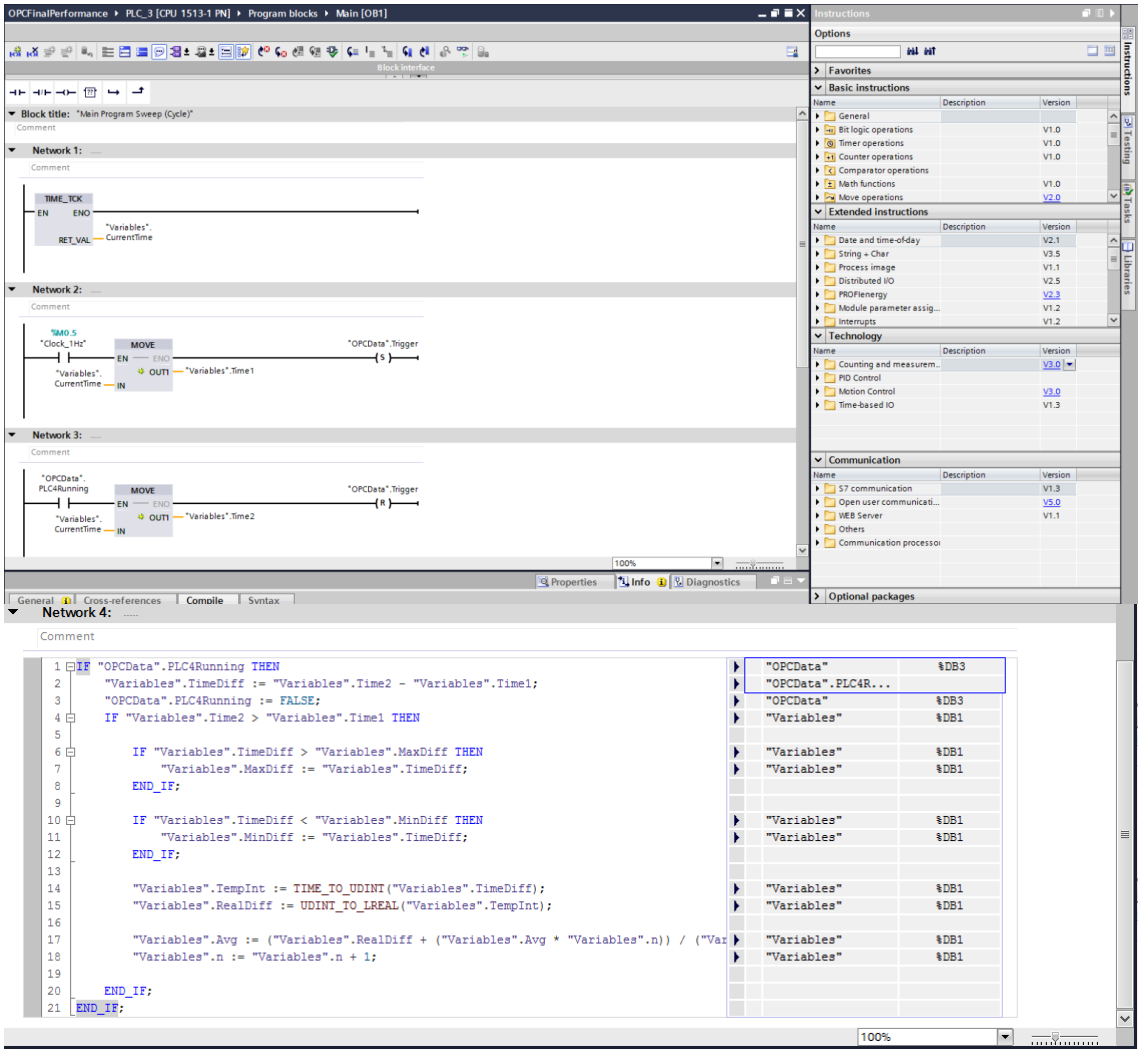


Figur 8. OPC UA Read Funktionsblock

Dessa funktioner lades samman i sammanslagna block så uppkoppling och nerkoppling mot en server underlättades.

OPCUAInitializePLC och OPCUATerminatePLC använder alla block som behövs för att varje PLC ska starta och avsluta OPC UA kommunikation, inklusive variabeladressering. Dessa blocken minskar repetitionen som behövdes i koden.

Siemens PLC mätte tiderna. Figur 9 visar hur tidmätningen gick till i Siemens.

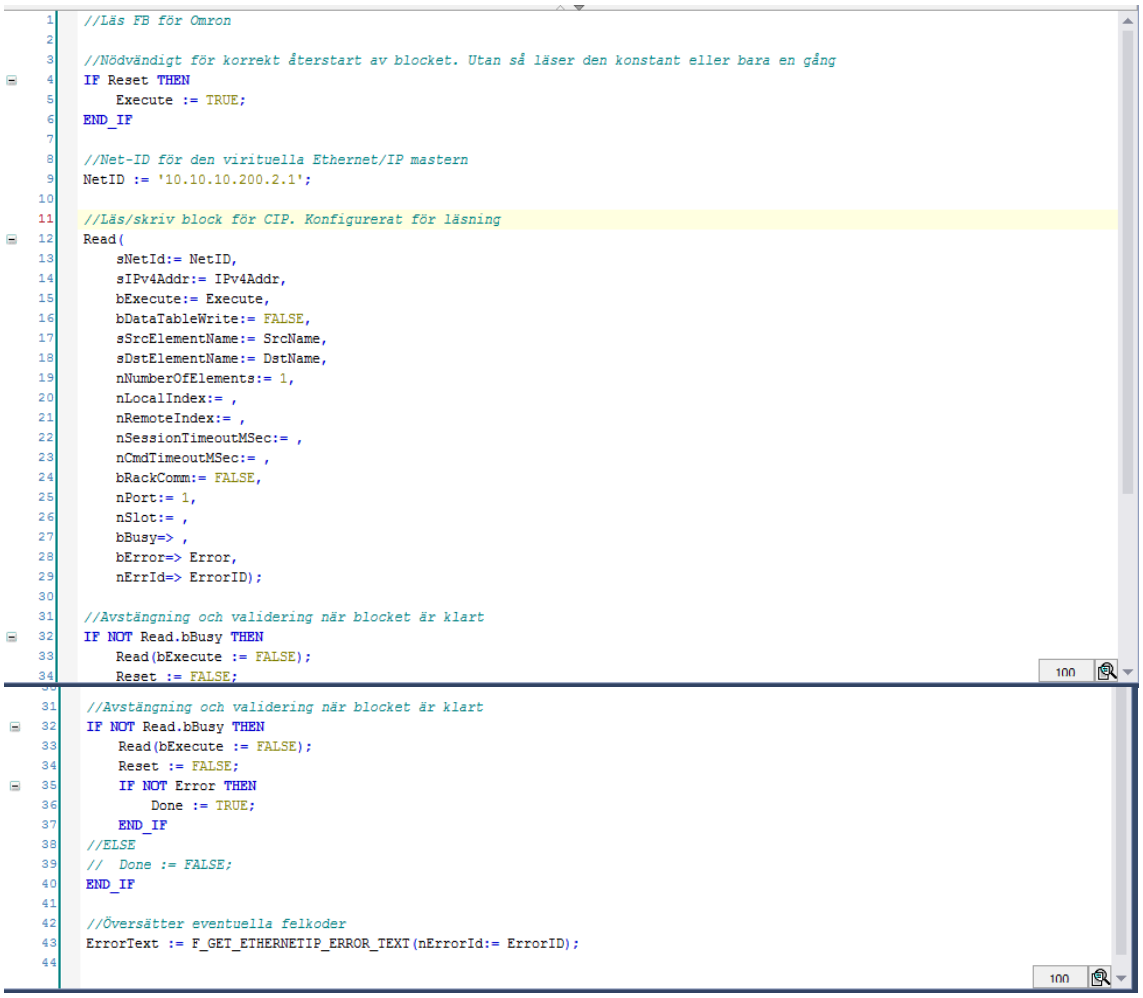


Figur 9. Tidmätning med Siemens

Mätmetoden går ut på att en variabel i Siemens PLC läses konstant av TwinCAT. När denna variabel ändras på en timer i Siemens så startar TwinCAT det fullständiga läsförloppet och skrivförloppet. Mätresultatet presenteras i tabell 3.

5.4.2 Ethernet/IP med TwinCAT

Endast ett funktionsblock används för Ethernet/IP kommunikationen. Två block skapades som specialisering, ett som läser och ett som skriver. Läsblocket visas i figur 10.

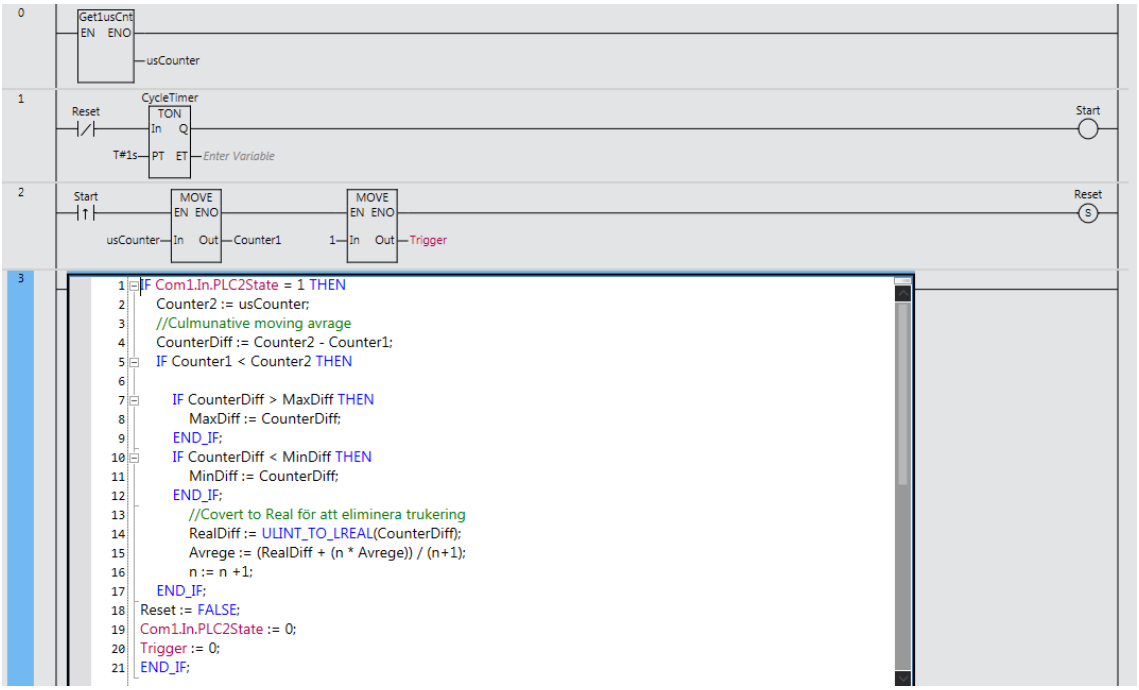


```
//Läs FB för Omron

//Nödvändigt för korrekt återstart av blocket. Utan så läser den konstant eller bara en gång
IF Reset THEN
    Execute := TRUE;
END_IF

//Net-ID för den virituella Ethernet/IP mastern
NetID := '10.10.10.200.2.1';

//Läs/skriv block för CIP. Konfigurerat för läsning
Read(
    sNetId:= NetID,
    sIPv4Addr:= IPv4Addr,
    bExecute:= Execute,
    bDataTableWrite:= FALSE,
    sSrcElementName:= SrcName,
    sDstElementName:= DstName,
    nNumberOfElements:= 1,
    nLocalIndex:= ,
    nRemoteIndex:= ,
    nSessionTimeoutMSec:= ,
    nCmdTimeoutMSec:= ,
    bRackComm:= FALSE,
    nPort:= 1,
    nSlot:= ,
    bBusy=> ,
    bError=> Error,
    nErrId=> ErrorID);

//Avstängning och validering när blocket är klart
IF NOT Read.bBusy THEN
    Read(bExecute := FALSE);
    Reset := FALSE;
```

```
//Avstängning och validering när blocket är klart
IF NOT Read.bBusy THEN
    Read(bExecute := FALSE);
    Reset := FALSE;
    IF NOT Error THEN
        Done := TRUE;
    END_IF
//ELSE
//  Done := FALSE;
END_IF

//Översätter eventuella felkoder
ErrorText := F_GET_ETHERNETIP_ERROR_TEXT(nErrorId:= ErrorID);
```

Figur 10. CIP Read Funktionsblock

Denna gången mätes tider i Omron PLC med samma metod som i Siemens. Denna mätmetod visas i figur 11.



Figur 11. Tidsmätning med Omron

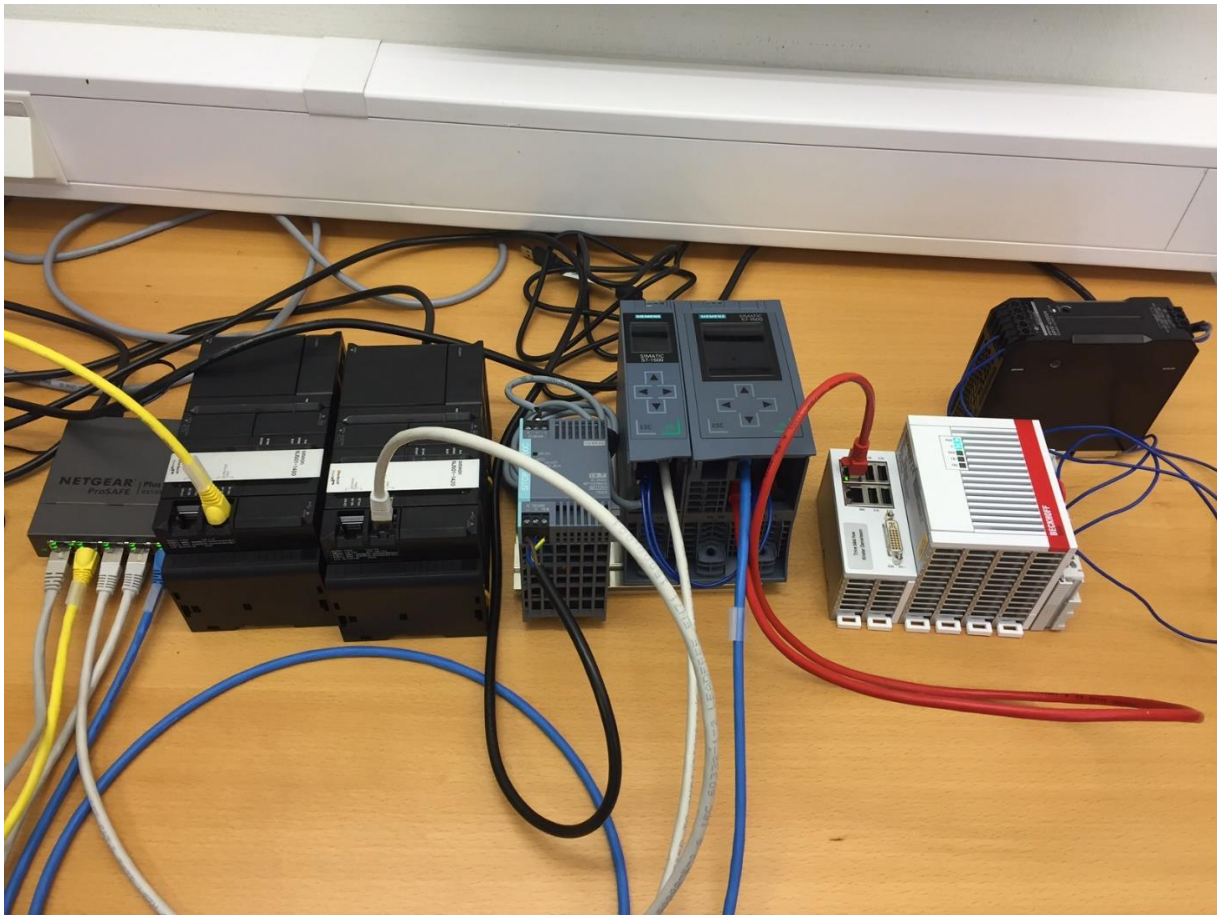
Fullständig kod för testet finns i appendix E

5.4.3 Kombinerat test

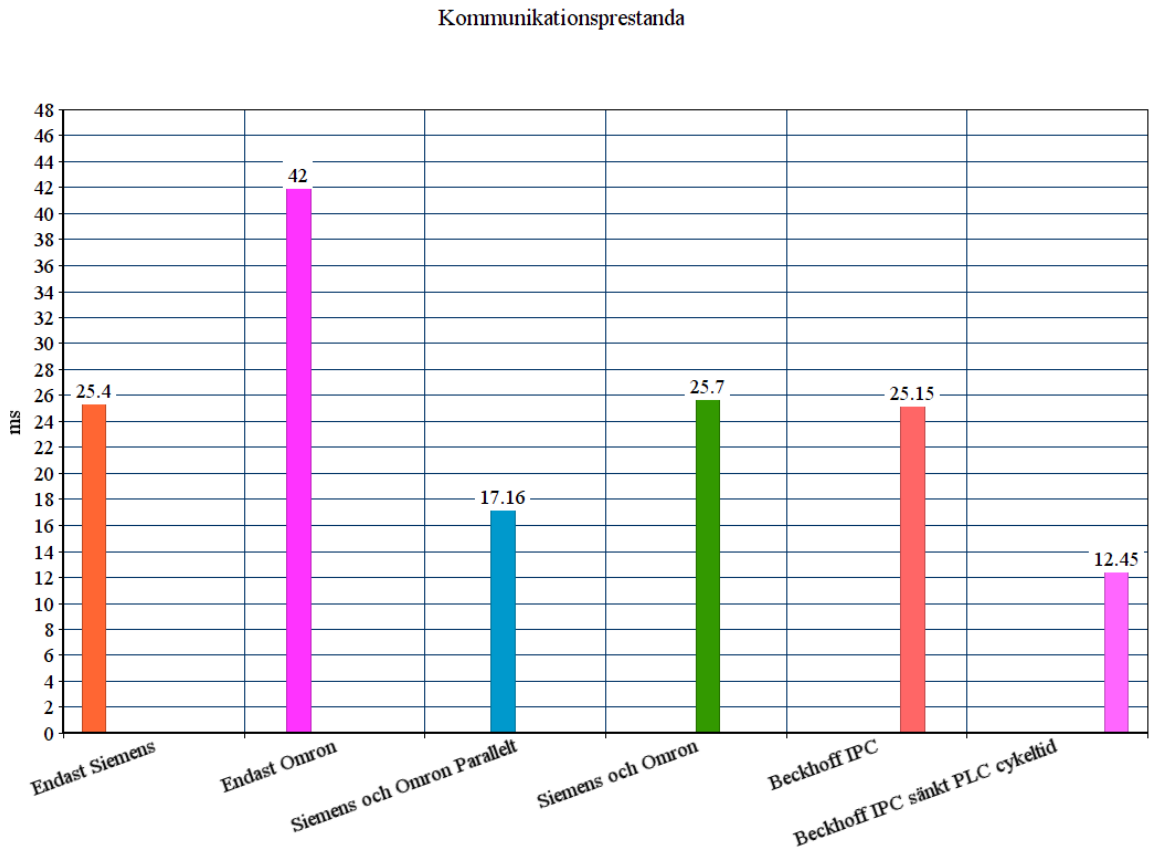
Det kombinerade testet använde koden från båda föregående tester presenterat i appendix F. Mätningen utfördes i Siemens med samma kod som i OPC UA-testet, visat i figur 9.

Detta testet var det slutgiltiga testet som bevisar examensarbetets mål är uppfyllt. Med detta test kan Siemens PLC:er och Omron PLC:er utbyta data mellan varandra. Detta via att kommunicera med Beckhoff PLC som en gemensam nod. Från Siemens till Beckhoff användes OPC UA och från Omron till Beckhoff användes CIP via Ethernet/IP.

Först användes utvecklingsdatorn som PLC med TwinCAT runtime. Sedan kördes precis samma test fast på Beckhoff IPC (Industrial PC) som hårdvara istället för på PC. Figur 12 visar uppkopplingen med all hårdvara.



Figur 12. Slutgiltigt test med Beckhoff IPC som master. Från vänster: Ethernet switch, Omron NJ501-1400, Omron NJ501-1420, Siemens S7-1500, Siemens S7-1500 (större modell), Beckhoff IPC.



Tabell 1: Medelvärden av uppmätt tid per fullständig överföringscykel

Tabell 1 visar genomsnittliga tiden för en fullständig överföringscykel från att kommunikationen startar en cykel tills att nästkommande cykel startar.
Första två kolumnerna från vänster visar den uppmätta tiden för TwinCAT-programmet när Omron och Siemens kommunikation kördes var för sig. Tredje kolumnen från vänster visar när programmet körde båda kommunikationsprogrammen samtidigt.

Fjärde kolumnen från vänster visar resultatet från det riktigt testet som är målet för examensarbetet. Dessa värdena representerar prestandan på TwinCAT när samtliga PLC:er utbyter data mot alla de andra. De två sista kolumnerna visar resultatet från när samma program användes på lånad Beckhoff IPC som är ett den hårdvara som kommer användas om detta examensarbete skulle leda till vidareutveckling från detta proof-of-concept test. Sista kolumnen visar resultat när denna IPC körde sin snabbast möjliga programcykeltid.

Samtliga mätningar visar god prestanda relativt mot önskad prestanda. Ingen exakt siffra gavs men en uppmaning att lösningen bör klara några hundratal millisekunder per cykel.
Resultaten i tabell 1 visar att cykeltiden är betydligt lägre och därmed anses detta proof-of-concept test lyckat.

5.5 Slutgiltig bedömning

Dessa bedömningar är baserade på upplevelser och bedömningar av TwinCAT.

TwinCAT är en grundläggande men kraftfull utvecklingsmiljö. Det mesta behövs göras från grunden och konfigurering är ej intuitivt. Fördelen är att kodningen är öppen och lätthanterlig. Baserad på MS Visual Studio gör det bekvämt och igenkännligt för utvecklare som har erfarenhet av Visual Studio. Påminner mycket om andra högnivå utvecklingsmiljöer. Strukturerad text är standard programspråk som lämpar sig väldigt bra till hanteringen av funktionsblock och bibliotek.

TwinCAT valdes för att alla intressanta kommunikationsprotokoll finns som färdiga block som kräver en nerladdning av den relevanta modulen. Det var snabbt och enkelt att installera modulen då den automatiskt integrerar sig mot TwinCAT.

De flesta problemen som uppstod var på grund av svag dokumentation på Beckhoffs hjälphemsida. [6] OPC UA dokumentationen saknade några detaljer och det som fanns var kort, men alltför specifikt. Dokumentation för Ethernet/IP blocket som användes saknades nästan helt. Däremot var supportkontakten utmärkt, de var både kunniga och snabba att svara.

6 Slutsats

Här presenteras kortfattat slutgiltiga resultatet och svar på tidigare frågeställning.

6.1 Sammanfattning

Sammanfattar examensarbetet.

6.1.1 Val av kommunikationsprotokoll

Flertalet kommunikationsprotokoll testades. Omron var begränsat till en cyklisk och en acyklisk variant. Av dessa var endast den acykliska varianten av CIP över Ethernet/IP av intresse.

Siemens har ett stort urval av kommunikationsprotokoll. Enda sambandet mellan dessa är att de utförs över ProfiNet. OPC UA var den mest intressanta överföringsmetoden. OPC UA är ett modernt protokoll utvecklat i samarbete mellan några av de största tillverkarna av PLC-styrsystem. [12] Detta förenklar kompatibilitet mellan olika PLC-system. TCP/IP med Siemens var även intressant då den var väldigt enkel att använda och konfigurera mellan Siemens system. Vidare tester mot andra system visade att TCP/IP endast var enkelt att använda mot andra Siemenskontroller och skapade en hel del problem mot övriga system. Även om TCP/IP kunde läsas och skrivas var ofta data ej korrekt då Siemens använder annorlunda datahantering än de systemen som testades emot.

TCP/IP valdes bort från sluttestet då det inte var av prioritet. Mot Omron användes Ethernet/IP med acyklisk kommunikation och mot Siemens användes OPC UA.

6.1.2 Val av leverantör

Tre stycken leverantörer svarade med ett erbjudande av lösning. Beckhoff, B&R samt WinCC OA systemet som är en del av Siemens. Beckhofflösningen är baserad på deras normala PLC-runtimesystem. De har färdiga moduler med funktioner för dataöverföringen. B&Rlösningen var lik Beckhofflösningen där deras PLC-system med tilläggsfunktioner används för att lösa kommunikationen.

WinCC OA är ett SCADA-system som har och driver moduler för de överföringsprotokoll som används.

WinCC OA var av lågt intresse då det är en unik miljö som kräver specialiserad kunskap. Användare hade blivit låsta i deras system. B&R- och Beckhoff-systemen är mer öppna och kunskaper från liknande system kan användas.

Av dessa två system var det endast Beckhoff som var testat att klara av de kommunikationssätten som behövdes. B&R är fortfarande en kandidat men Beckhofflösningen var den som undersöktes vidare.

6.1.3 Beckhofflösningen

Ett komplett test med PLC:er från både Siemens och Omron. Testet skrevs med TwinCATsystemet från Beckhoff. Först testades kommunikation mot Omron och Siemens var för sig, när dessa tester visade att de fungerade gjordes ett kombinerat test. När allting fungerade lånades en Beckhoff IPC och testet kördes på den riktiga hårdvaran. Allt, utom några i sista minuten olösta problem, fungerade som önskat.

6.1.4 Framtida utvecklingsmöjligheter

Under examensarbetets gång noterades några möjligheter till att vidareutveckla Beckhofflösningen efter examensarbetets avslut. Främst bland förslagen var möjligheten till datainsamling och loggning. All data skickas till TwinCAT innan data skickas vidare till övriga PLC:er. Detta gör det till en naturlig knytpunkt att samla in data till exempelvis en databas. Insamlade data kan då presenteras och analyseras från databasen.

6.2 Svar på frågeställning

Svarar på frågeställningen från Inledningen (kap. 1.4.1)

- Vilka kommunikationsprotokoll som används av utvalda PLC:er kan hanteras?

OPC UA samt acyklisk kommunikation med CIP är de som har bevisats. Om ett projekt drivs vidare kan fler kommunikationsprotokoll implementeras vidare för ett bredare användningsområde där fler fabrikat av PLC:er ingår.

- Finns det lösningar från leverantörer som kan anpassas?

Ja, flera lösningar har presenterats. Beckhoff, B&R samt WinCC OA lösningar kan användas. Beckhoff var den leverantör vars lösning testades vidare.

- Vilka automationssystem kan hanteras?

Siemens, Omron och Beckhoff har testats och kan hanteras. Inga begränsningar mot andra automationssystem stöttes på men testades inte i detta examensarbete. Vidareutveckling krävs för att testa fler automationssystem.

- Vilka kriterier behöver lösningen bedömas efter?

Kriterierna lösningarna har bedömts efter är arbetsåtgång, komplexitet, footprint, expanderingsmöjligheter, begränsningar och säkerhet. För hela bedömningen framställdes även viktiga egenskaper samt för- och nack-delar.

- Vilken är den bästa lösningen utifrån uppfyllnad av kriterierna?

Lösningen som detta examensarbete tittade på är att använda Beckhoff PLC för att ta hand om kommunikation men det är inte nödvändigtvis den bästa lösningen som finns utan den som bedömdes lämpligast från de lösningar som presenterades.

- Vilka begränsningar har lösningen?

Inga direkta begränsningar stöttes på under utvecklingen utöver de begränsningar som sattes för examensarbetet i helhet.

6.3 Reflektion över resultatet

Besvarar på hur syftet från kap 1.2 har uppfyllts och hur resultatet kan komma att användas.

6.3.1 Uppfyllnad av syftet

Examensarbetet har fullständigt uppfyllt det syfte som angetts i början. En lösning har undersökts och ett test har visat att den gör vad som önskat.

6.3.2 Användning av resultat

Resultatet ger en bra uppfattning om hur Beckhofflösningen är att använda och implementera. Om vidareutveckling önskas kan kod från detta examensarbete ligga till grund eller inspiration för en fullständig implementering.

6.4 Etiska aspekter

Diskuterar några etiska aspekter som berörts under examensarbetet.

6.4.1 Sekretess

Under examensarbetet har vissa detaljer delats som ligger under sekretess.
Dessa detaljer är främst offerter och priser från leverantörerna. Det är viktigt att hålla dessa detaljer hemliga då konkurrenter kan orättvist ge sämre erbjudande när de ser vad de behöver konkurrera mot. Att hålla offerter hemliga bidrar till att offertutgivare behöver ge bättre erbjudande då de inte vet om deras erbjudande är bättre än andra.

6.4.2 Datalagring och datainsamling

Examensarbetet har inte arbetat med datalagring, endast tillfällig förflyttning av data.
Däremot kan lösningen som undersöktes expandera med metoder för datainsamling och lagring. Detta kan leda till att överflödig datainsamling hos kunder som använder sig av en sådan lösning.
Detta kan ge fördelar i felsökning och effektivisering men de skulle kunna se övriga användardata såsom produktionsmängder. Nackdelarna som följer är att produktionsdata som inte behövs visas. Slutsatser kan dras från denna data vilket kunder kan önska hålls hemligt.

Detta examensarbete kan inte beröra någon senare implementering av datalagring utan det faller på företaget att hantera datainsamling på ett korrekt sätt.

7 Terminologi

ACK
: Acknowledgement paket som berättar att kommunikation gick korrekt till

CIP – Common Industrial Protocol
: Samlat kommunikationsprotokoll som är grunden för Ethernet/IP

EtherCAT
: Kommunikationsprotokoll med fokus på snabb I/O-kommunikation

Ethernet/IP
: Kommunikationsprotokoll över Ethernet

I/O – Input/Output
: Ett sätt att benämna ingångar och utgångar

IPC – Industrial PC
: Dator anpassad för industriella ändamål och förhållanden

ISO-on-TCP
: Alternativ till TCP/IP

HMI – Human Machine Interface
: Gränssnitt mellan människa och maskin, typiskt en pek-display som visar info eller ger kommando

OPC UA – Open Platform Communications Unified Architecture
: Öppet kommunikationsprotokoll designat för att vara oberoende av specifika system

PLC – Programmable Logic Controller
: Styrenhet för automationssystem

ProfiNet
: Kommunikationsprotokoll som använder Ethernet som bas.

ProfiNetRT – Profinet Real Time
: Realtidsdel av ProfiNet

SCADA – Supervisory Control And Data Acquisition
: System som kan utföra datainsamling och rapportering.

TCP/IP – Transmission control protocol/Internet Protocol
: Modell för överföringsprotokoll

TIA Portal – Totally Integrated Automation Portal
Samlad utvecklingsmiljö för Siemens.

WinCC OA – WinCC Open Architecture
SCADA-system från Siemens

8 Källförteckning

[1]
Nair, R, & Smith, J. 2005. *Virtual Machines: Versatile Platforms For Systems And Processes.* Elsevier Science. E-bok.

[2]
Ramirez, G, Burke, J, & Orebaugh, A. 2006, *Wireshark & Ethereal Network Protocol Analyzer Toolkit (Jay Beale's Open Source Security Series).* Syngress. E-bok.

[3]
OPC Foundation, *Unified Architecture*
https://opcfoundation.org/about/opc-technologies/opc-ua/
(Hämtad 2018-06-04)

[4]
Omron, *NJ/NX-series CPU Unit Built-in EtherNet/IP Port User's Manual*
https://assets.omron.eu/downloads/manual/en/v2/w506_nx_nj-series_cpu_unit_built-in_ethernet_ip_port_users_manual_en.pdf (Hämtad 2018-03-05)

[5]
Siemens, *SIMATIC OPC UA S7-1500*
http://w3.siemens.com/mcms/automation-software/en/tia-portal-software/step7-tia-portal/simatic-step7-options/opc-ua-s7-1500/pages/default.aspx (Hämtad 2018-03-05)

[6]
Beckhoff, *Beckhoff Information System*
https://infosys.beckhoff.com/index_en.htm (Hämtad 2018-03-05)

[7]
Omron, *Tjänster och support*
https://industrial.omron.se/sv/services-support/technical-tools/downloads#nj5
(Hämtad 2018-03-05)

[8]
Siemens, *Industry Online Support*
https://support.industry.siemens.com/cs/start?lc=en-SE
(Hämtad 2018-03-05)

[9]
Omron, *NJ/NX-series CPU Unit Software User's Manual*
https://assets.omron.eu/downloads/manual/en/v6/w501_nx_nj-series_cpu_unit_software_users_manual_en.pdf (Hämtad 2018-03-05)

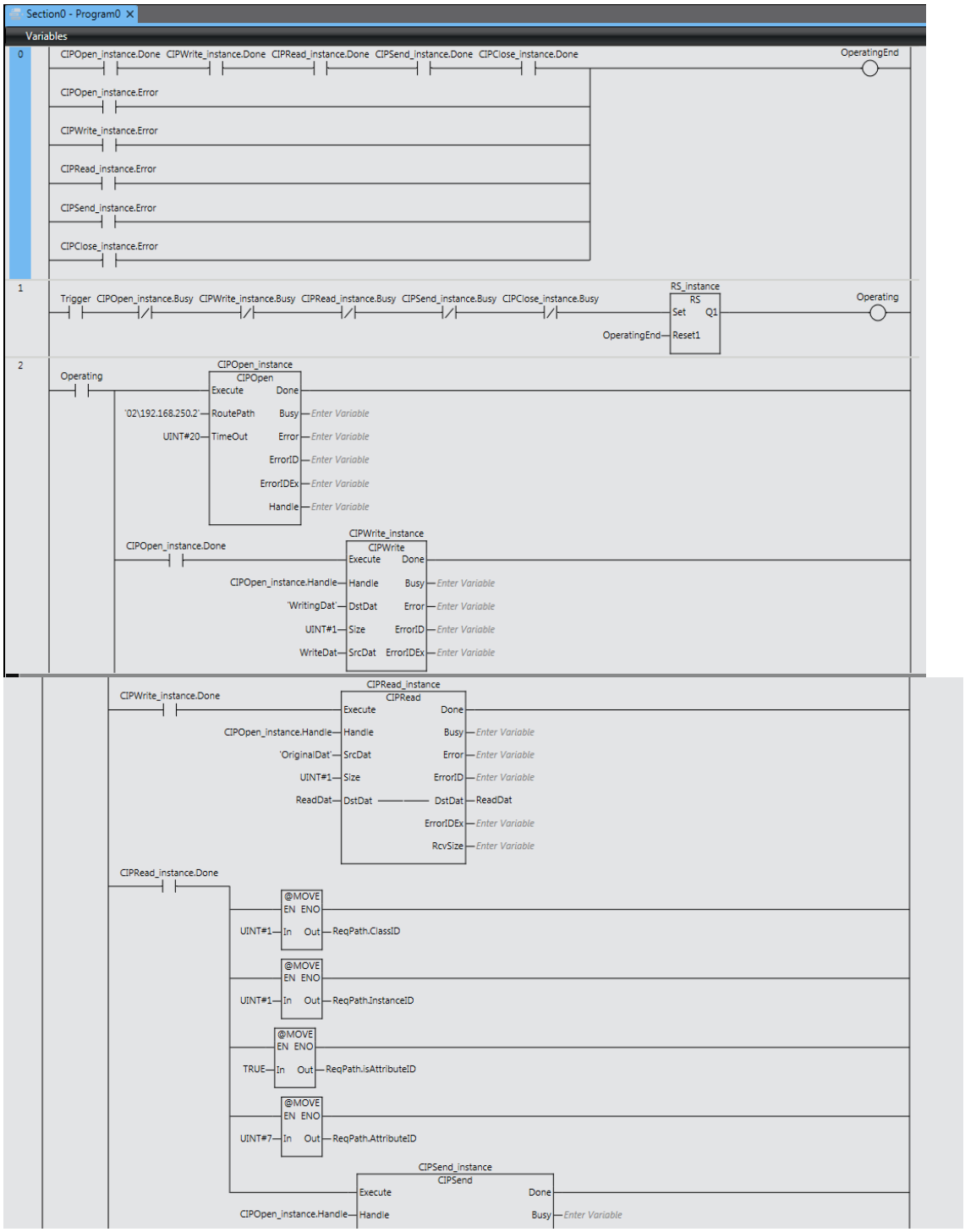
[10]
Beckhoff, *TwinCAT PLC Control: Standard Data Types BOOL*
https://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/html/tcplcctrl_plc_data_types_overview.htm&id= (Hämtad 2018-03-05)

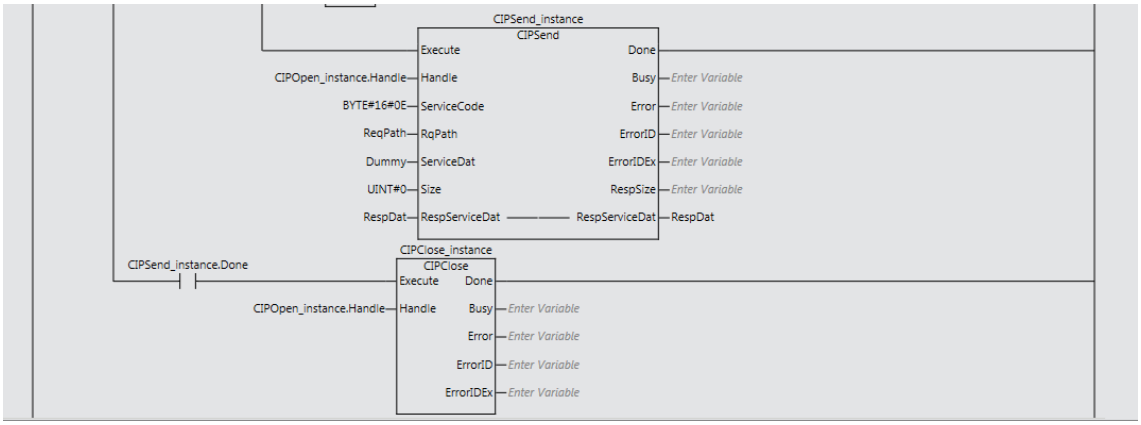
[11]
Siemens, *What properties, advantages and special features does the ISO-on-TCP protocol offer?*
https://support.industry.siemens.com/cs/document/26484227/what-properties-advantages-and-special-features-does-the-iso-on-tcp-protocol-offer-?dti=0&lc=en-WW
(Hämtad 2018-03-07)

[12]
OPC Fundation, 2018, *Markets and Collaborations*
https://opcfoundation.org/markets-collaboration/
(Hämtad 2018-03-07)

9 Appendix

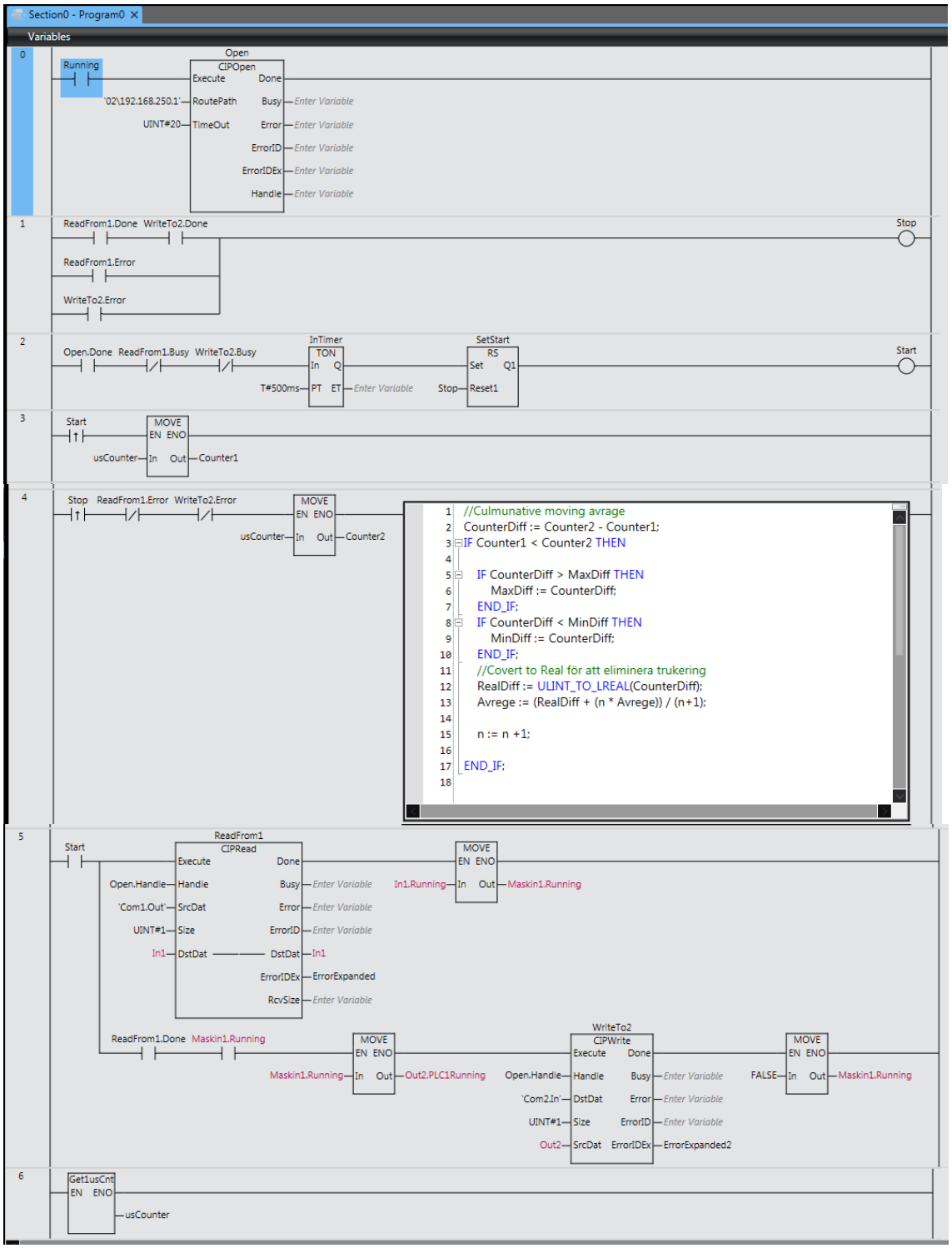
Appendix A: CIP test med Omron





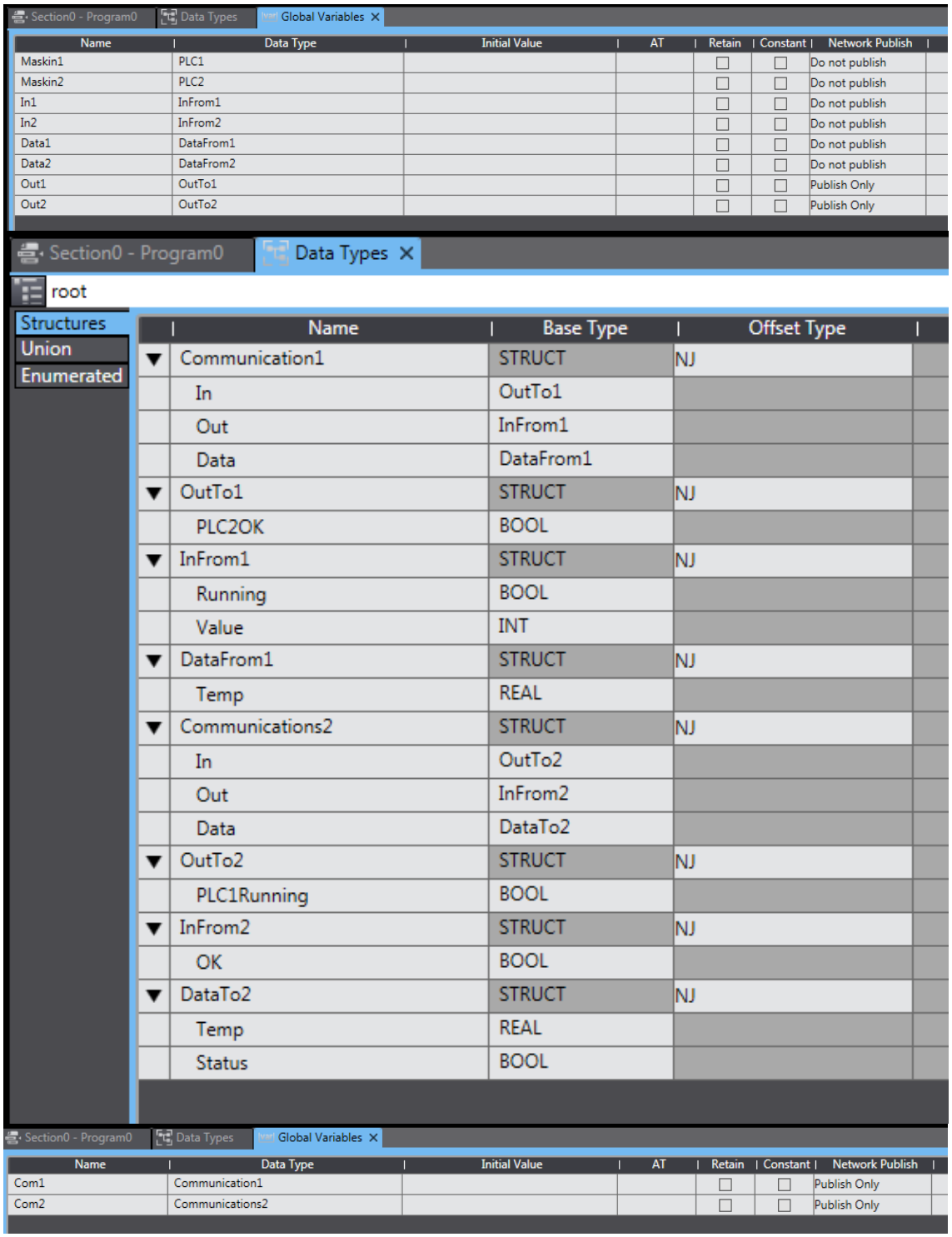
Figur 13: Ladder program för kommunikation mellan Omron PLC.

Appendix B: Simulerat test med CIP Omron, PLC del



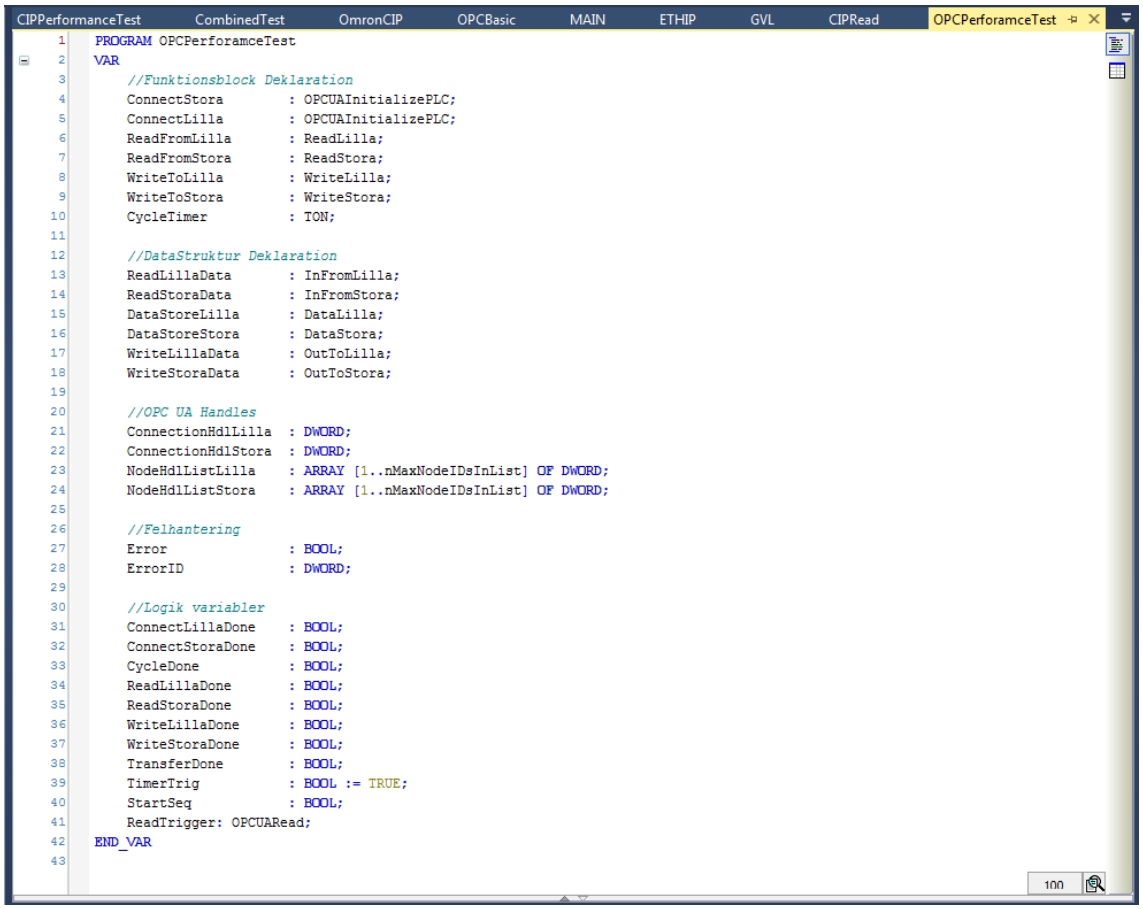
Figur 14: Cykeltidsmätning skapad i Omron program.

Appendix C: Simulerat test med CIP Omron, Kontroller del

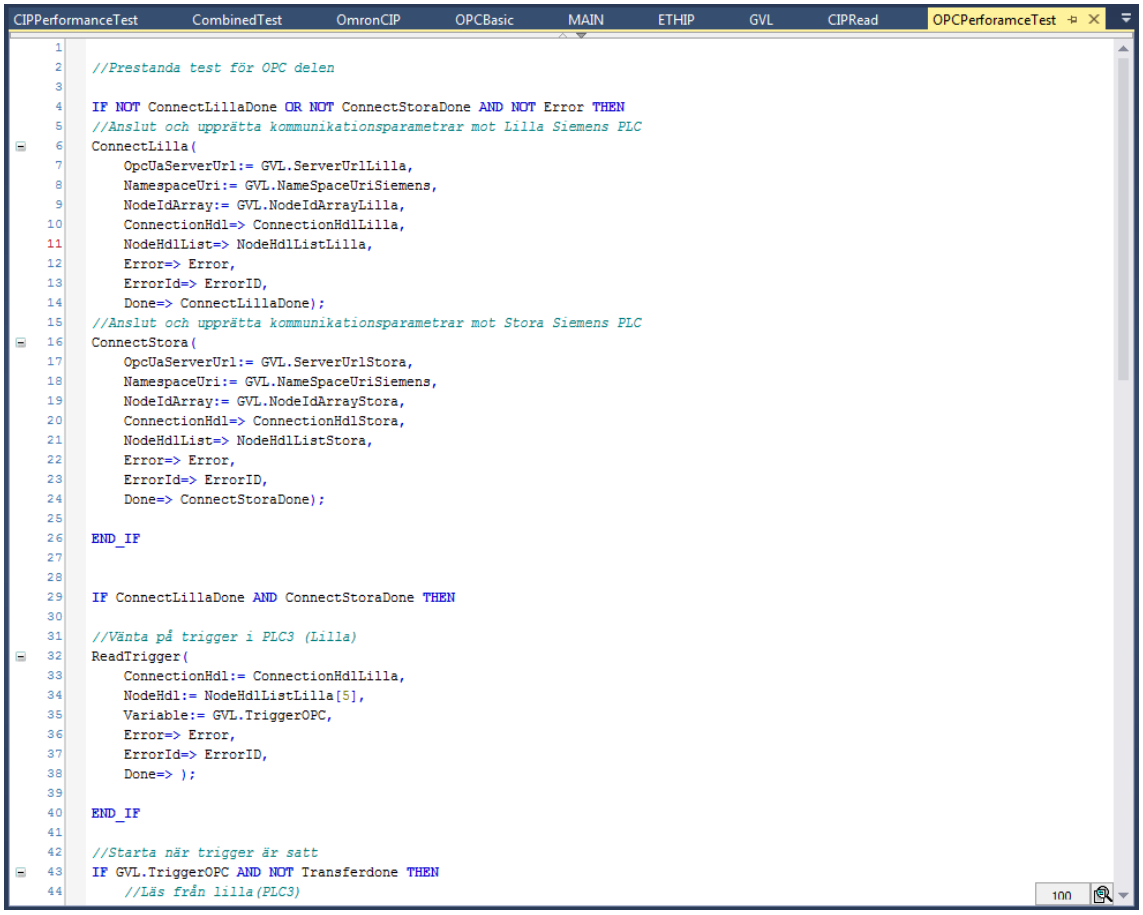


Figur 15: Strukturer att skriva till samt läsa från i Omron PLC.

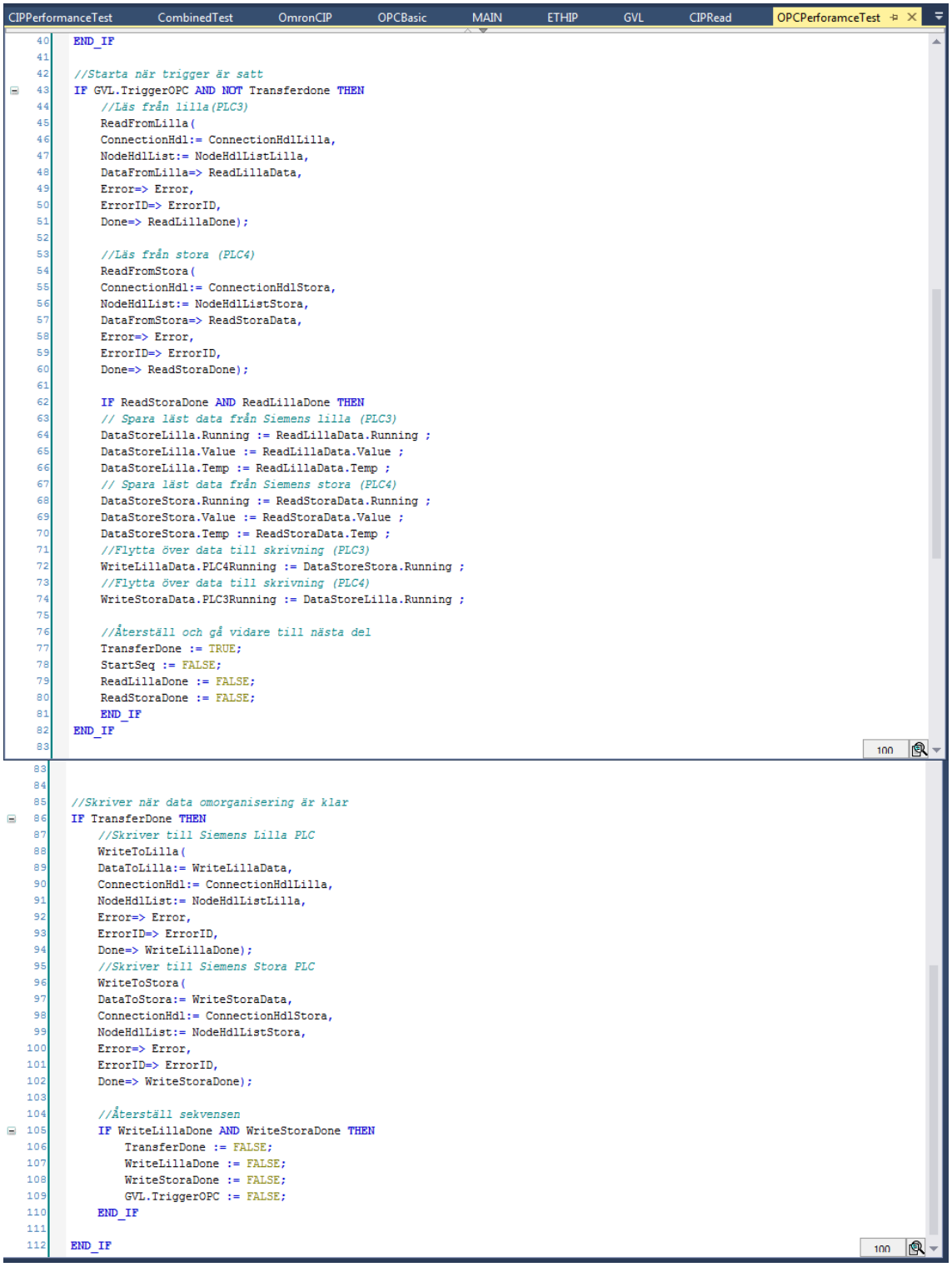
Appendix D: TwinCAT prestandatest av OPC UA på Siemens



Figur 16: Initieringsdel av TwinCAT program. Kommunicerar med Siemens PLC.




```
1
2   //Prestanda test för OPC delen
3
4   IF NOT ConnectLillaDone OR NOT ConnectStoraDone AND NOT Error THEN
5   //Anslut och upprätta kommunikationsparametrar mot Lilla Siemens PLC
6   ConnectLilla(
7       OpcUaServerUrl:= GVL.ServerUrlLilla,
8       NamespaceUri:= GVL.NameSpaceUriSiemens,
9       NodeIdArray:= GVL.NodeIdArrayLilla,
10      ConnectionHdl=> ConnectionHdlLilla,
11      NodeHdlList=> NodeHdlListLilla,
12      Error=> Error,
13      ErrorId=> ErrorID,
14      Done=> ConnectLillaDone);
15  //Anslut och upprätta kommunikationsparametrar mot Stora Siemens PLC
16  ConnectStora(
17      OpcUaServerUrl:= GVL.ServerUrlStora,
18      NamespaceUri:= GVL.NameSpaceUriSiemens,
19      NodeIdArray:= GVL.NodeIdArrayStora,
20      ConnectionHdl=> ConnectionHdlStora,
21      NodeHdlList=> NodeHdlListStora,
22      Error=> Error,
23      ErrorId=> ErrorID,
24      Done=> ConnectStoraDone);
25
26  END_IF
27
28
29  IF ConnectLillaDone AND ConnectStoraDone THEN
30
31  //Vänta på trigger i PLC3 (Lilla)
32  ReadTrigger(
33      ConnectionHdl:= ConnectionHdlLilla,
34      NodeHdl:= NodeHdlListLilla[5],
35      Variable:= GVL.TriggerOPC,
36      Error=> Error,
37      ErrorId=> ErrorID,
38      Done=> );
39
40  END_IF
41
42  //Starta när trigger är satt
43  IF GVL.TriggerOPC AND NOT Transferdone THEN
44      //Läs från lilla(PLC3)
```



Figur 17: TwinCAT program. Kommunicerar med Siemens PLC.

Appendix E: TwinCAT prestandatest av CIP med Ormon

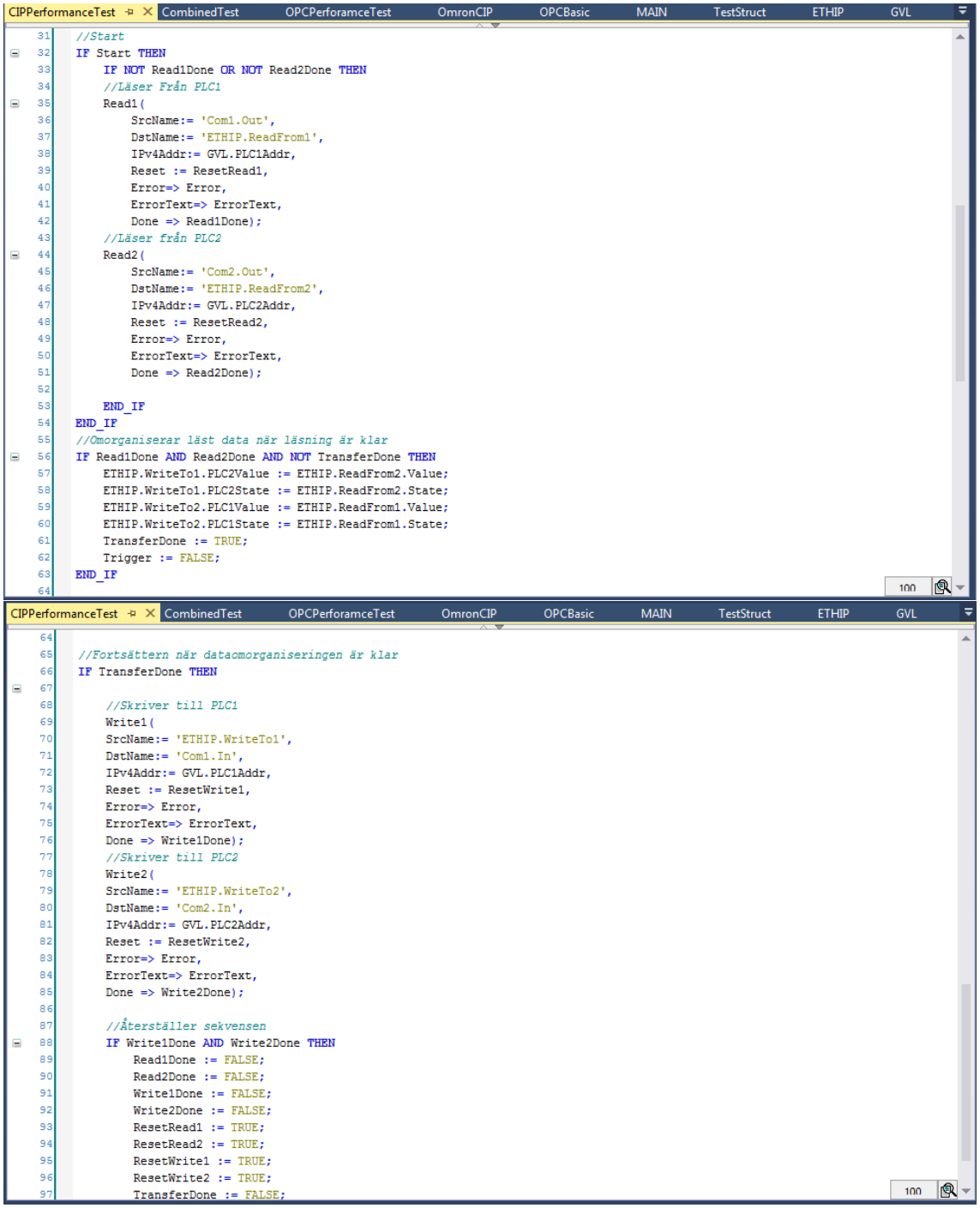
CIPPerformanceTest | CombinedTest | OPCPerforamceTest | OmronCIP | OPCBasic | MAIN | TestStruct | ETHIP | GVL

```
PROGRAM CIPPerformanceTest
VAR
    Read1: CIPRead;
    Read2: CIPRead;
    Error : BOOL;
    ErrorText : STRING;
    Read1Done: BOOL;
    Read2Done: BOOL;
    Write1Done : BOOL;
    Write2Done : BOOL;
    TransferDone: BOOL;
    Write1: CIPWrite;
    Write2: CIPWrite;
    Test: BOOL;
    StartTimerTrig: TON;
    Trigger: BOOL := TRUE;
    Start: BOOL;
    ReadToggle: CIPRead;
    ResetToggle : BOOL := TRUE;
    ResetRead1 : BOOL := TRUE;
    ResetRead2: BOOL := TRUE;
    ResetWrite1: BOOL := TRUE;
    ResetWrite2 : BOOL := TRUE;
END_VAR
```

CIPPerformanceTest | CombinedTest | OPCPerforamceTest | OmronCIP | OPCBasic | MAIN | TestStruct | ETHIP | GVL

```
//Slutgiltigt test för Omron

//Alternativ startsekvens med timer
(*
StartTimerTrig(IN:= Trigger, PT:= T#5S, Q=> Start);
IF NOT Trigger THEN
    Trigger := TRUE;
END_IF
*)

//Läser konstant efter en variabel. Vid förändring startar förloppet
ReadToggle(
    SrcName:= 'Trigger',
    DstName:= 'ETHIP.Toggle',
    IPv4Addr:= GVL.PLC1Addr,
    Reset := ResetToggle,
    Error=> Error,
    ErrorText=> ErrorText,
    Done=> );

//När den pollade variablen ändras till 1 startar förloppet
IF ETHIP.Toggle = 1 THEN
        Start := TRUE;
    ELSE
        Start := FALSE;
        ResetToggle := TRUE;
END_IF


//Start
IF Start THEN
    IF NOT Read1Done OR NOT Read2Done THEN
    //Läser Från PLC1
```

100



Figur 18: TwinCAT program. Kommunicerar med Omron PLC.

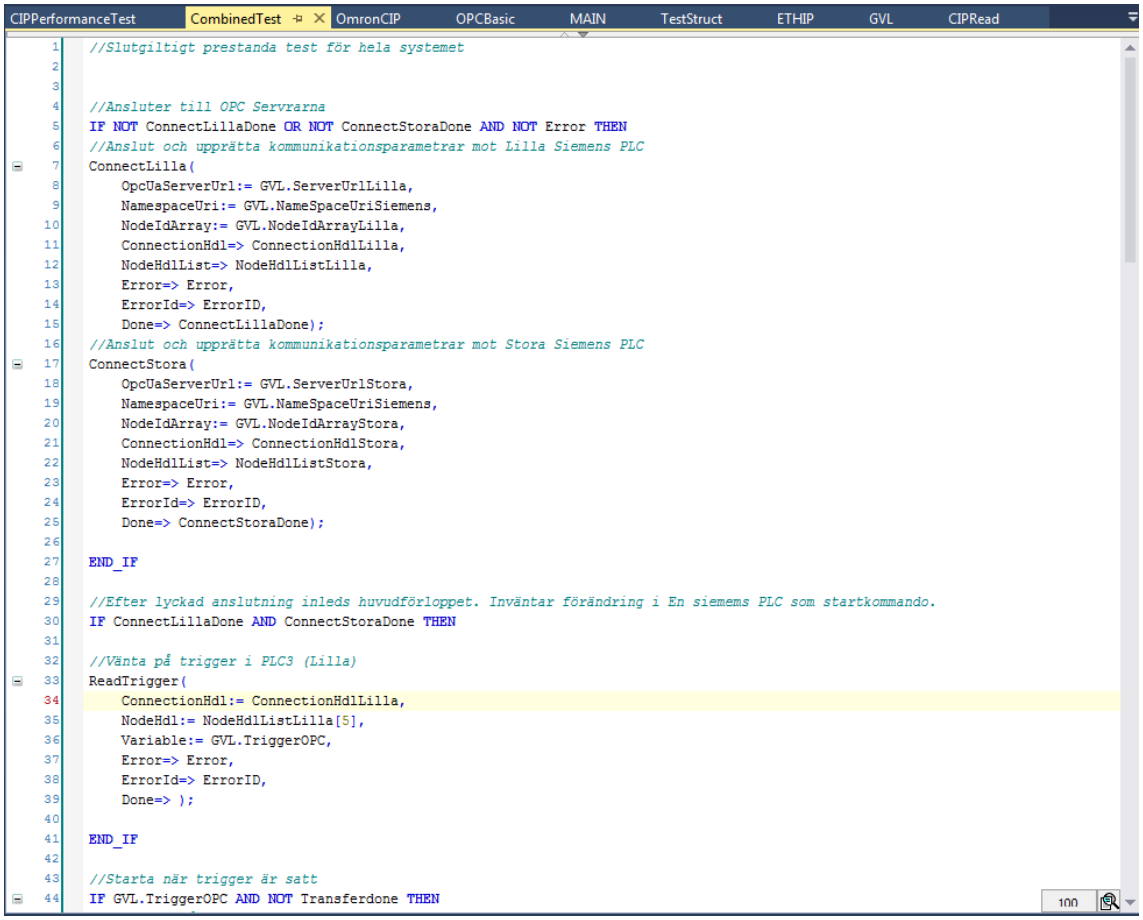
Appendix E: Kombinerat test med TwinCAT för både Omron och Siemens

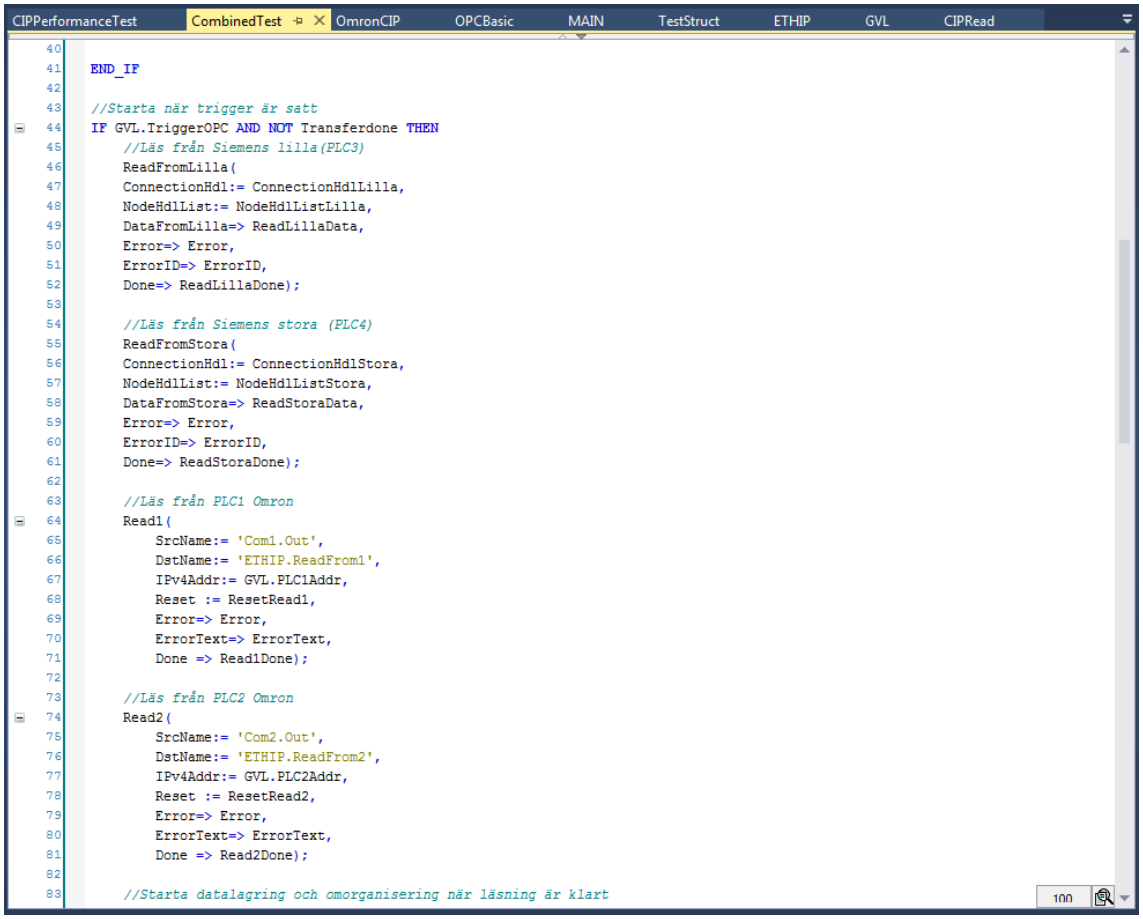
CIPPerformanceTest | CombinedTest | OmronCIP | OPCBasic | MAIN | TestStruct | ETHIP | GVL | CIPRead

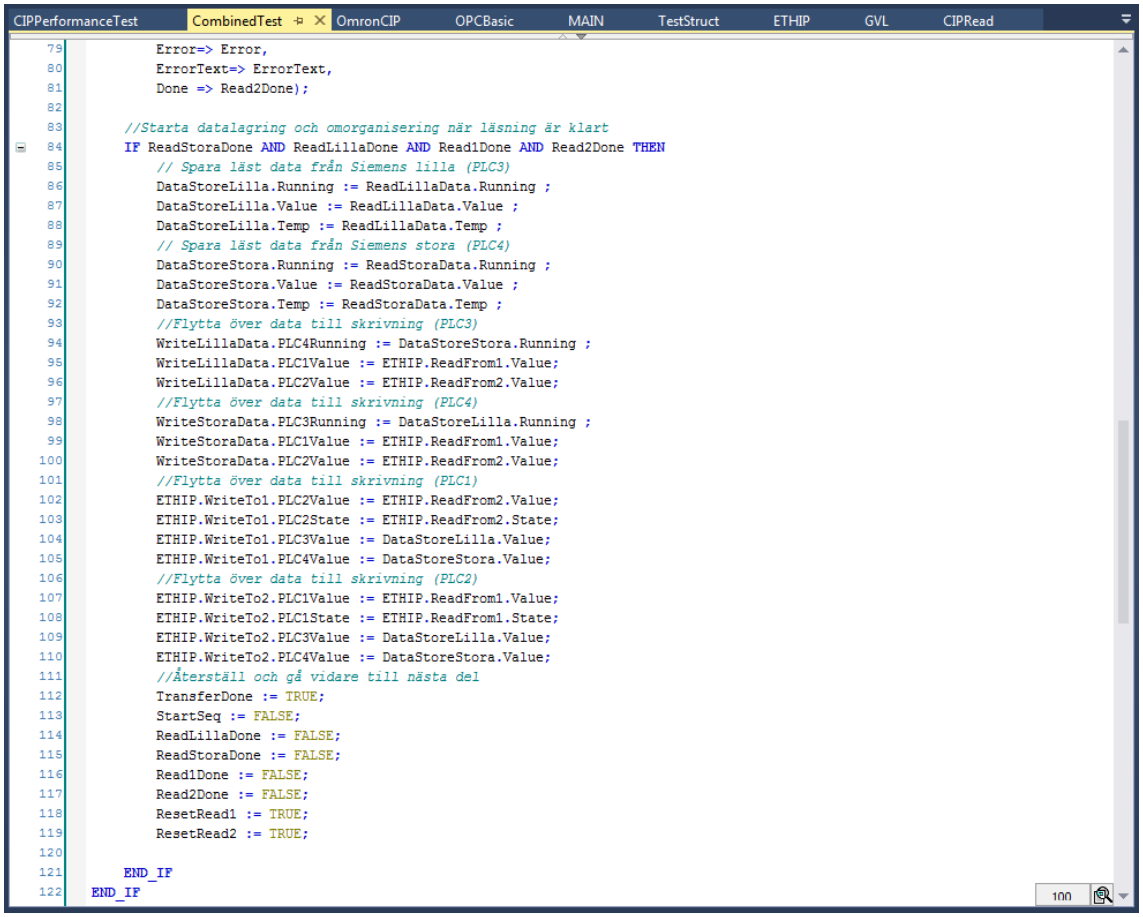
```
PROGRAM CombinedTest
VAR
        //Funktionsblock Deklarationer
    ConnectStora        : OPCUAInitializePLC;
    ConnectLilla        : OPCUAInitializePLC;
    ReadFromLilla       : ReadLilla;
    ReadFromStora       : ReadStora;
    WriteToLilla        : WriteLilla;
    WriteToStora        : WriteStora;
    CycleTimer          : TON;
    Read1: CIPRead;
    Read2: CIPRead;
    Write1: CIPWrite;
    Write2: CIPWrite;
    ReadTrigger: OPCUARead;
    ReadToggle: CIPRead;

    //DataStruktur Deklaration
    ReadLillaData       : InFromLilla;
    ReadStoraData       : InFromStora;
    DataStoreLilla      : DataLilla;
    DataStoreStora      : DataStora;
    WriteLillaData      : OutToLilla;
    WriteStoraData      : OutToStora;

    //OPC UA Handles
    ConnectionHdlLilla  : DWORD;
    ConnectionHdlStora  : DWORD;
    NodeHdlListLilla    : ARRAY [1..nMaxNodeIDsInList] OF DWORD;
    NodeHdlListStora    : ARRAY [1..nMaxNodeIDsInList] OF DWORD;

    //Felhantering
    Error               : BOOL;
    ErrorID             : DWORD;
```

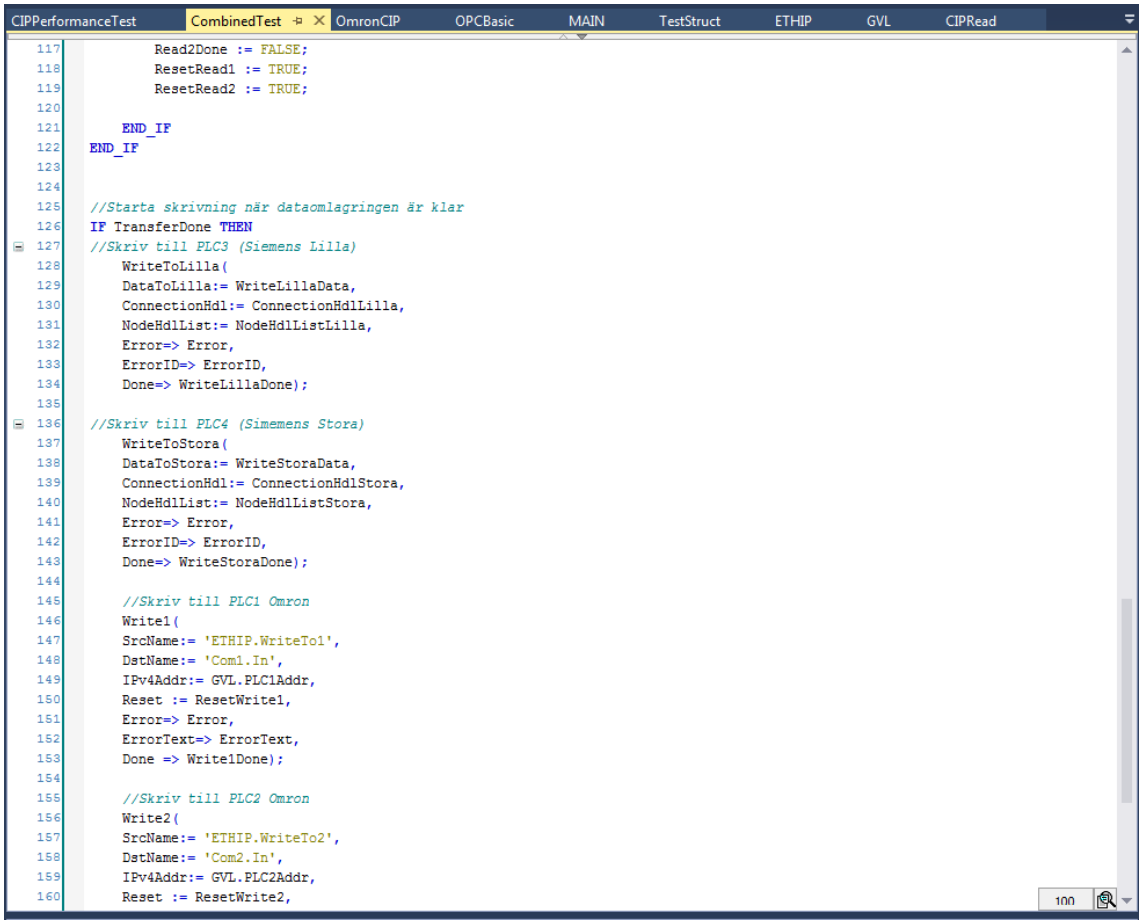
CIPPerformanceTest | CombinedTest | OmronCIP | OPCBasic | MAIN | TestStruct | ETHIP | GVL | CIPRead

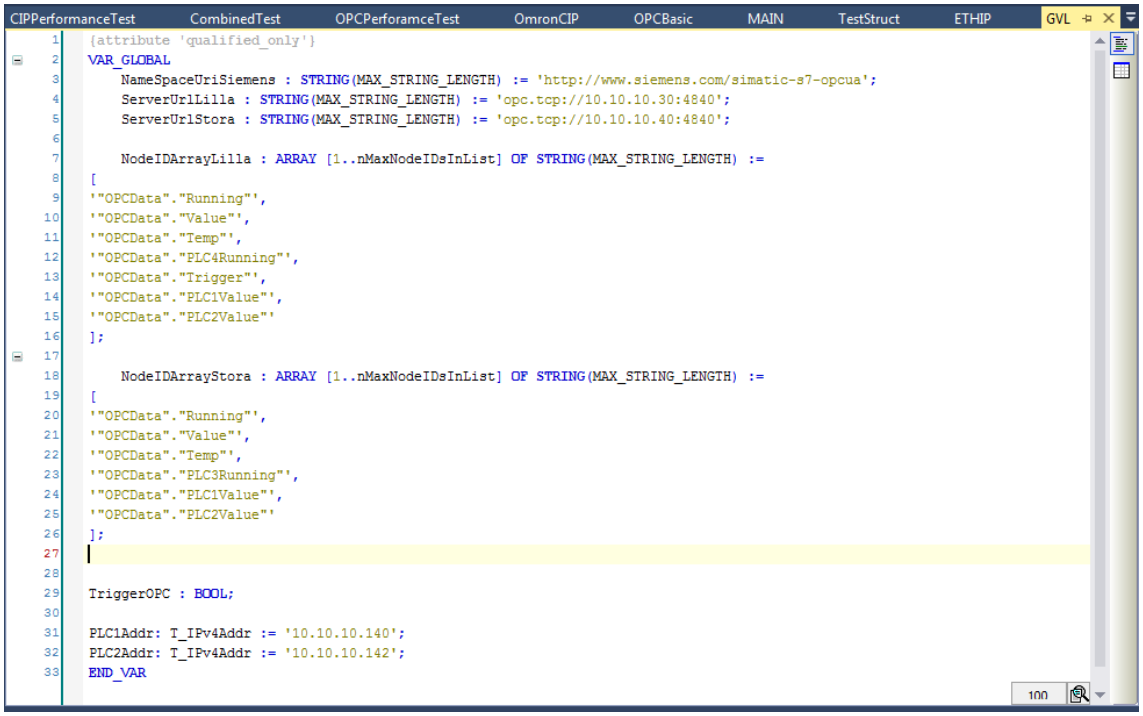
```
    NodeHdlListStora    : ARRAY [1..nMaxNodeIDsInList] OF DWORD;

    //Felhantering
    Error               : BOOL;
    ErrorID             : DWORD;
    ErrorCIP            : BOOL;
    ErrorText           : STRING;

    //Logik variabler
    ConnectLillaDone    : BOOL;
    ConnectStoraDone    : BOOL;
    CycleDone           : BOOL;
    ReadLillaDone       : BOOL;
    ReadStoraDone       : BOOL;
    WriteLillaDone      : BOOL;
    WriteStoraDone      : BOOL;
    TransferDone        : BOOL;
    TimerTrig           : BOOL := TRUE;
    StartSeq            : BOOL;
    Read1Done: BOOL;
    Read2Done: BOOL;
    Write1Done : BOOL;
    Write2Done : BOOL;
    Trigger: BOOL := TRUE;
    Start: BOOL;
    ResetToggle : BOOL := TRUE;
    ResetRead1 : BOOL := TRUE;
    ResetRead2: BOOL := TRUE;
    ResetWrite1: BOOL := TRUE;
    ResetWrite2 : BOOL := TRUE;

END_VAR
```


CIPPerformanceTest
CombinedTest
OmronCIP
OPCBasic
MAIN
TestStruct
ETHIP
GVL
CIPRead
//Slutgiltigt prestanda test för hela systemet
//Ansluter till OPC Servrarna
IF NOT ConnectLillaDone OR NOT ConnectStoraDone AND NOT Error THEN
//Anslut och upprätta kommunikationsparametrar mot Lilla Siemens PLC
ConnectLilla(
OpcUaServerUrl:= GVL.ServerUrlLilla,
NamespaceUri:= GVL.NameSpaceUriSiemens,
NodeIdArray:= GVL.NodeIdArrayLilla,
ConnectionHdl=> ConnectionHdlLilla,
NodeHdlList=> NodeHdlListLilla,
Error=> Error,
ErrorId=> ErrorID,
Done=> ConnectLillaDone);
//Anslut och upprätta kommunikationsparametrar mot Stora Siemens PLC
ConnectStora(
OpcUaServerUrl:= GVL.ServerUrlStora,
NamespaceUri:= GVL.NameSpaceUriSiemens,
NodeIdArray:= GVL.NodeIdArrayStora,
ConnectionHdl=> ConnectionHdlStora,
NodeHdlList=> NodeHdlListStora,
Error=> Error,
ErrorId=> ErrorID,
Done=> ConnectStoraDone);
END_IF
//Efter lyckad anslutning inleds huvudförloppet. Inväntar förändring i En siemems PLC som startkommando.
IF ConnectLillaDone AND ConnectStoraDone THEN
//Vänta på trigger i PLC3 (Lilla)
ReadTrigger(
ConnectionHdl:= ConnectionHdlLilla,
NodeHdl:= NodeHdlListLilla[5],
Variable:= GVL.TriggerOPC,
Error=> Error,
ErrorId=> ErrorID,
Done=>);
END_IF
//Starta när trigger är satt
IF GVL.TriggerOPC AND NOT Transferdone THEN
100


CIPPerformanceTest
CombinedTest
OmronCIP
OPCBasic
MAIN
TestStruct
ETHIP
GVL
CIPRead
END_IF
//Starta när trigger är satt
IF GVL.TriggerOPC AND NOT Transferdone THEN
//Läs från Siemens lilla(PLC3)
ReadFromLilla(
ConnectionHdl:= ConnectionHdlLilla,
NodeHdlList:= NodeHdlListLilla,
DataFromLilla=> ReadLillaData,
Error=> Error,
ErrorID=> ErrorID,
Done=> ReadLillaDone);
//Läs från Siemens stora (PLC4)
ReadFromStora(
ConnectionHdl:= ConnectionHdlStora,
NodeHdlList:= NodeHdlListStora,
DataFromStora=> ReadStoraData,
Error=> Error,
ErrorID=> ErrorID,
Done=> ReadStoraDone);
//Läs från PLC1 Omron
Read1(
SrcName:= 'Com1.Out',
DstName:= 'ETHIP.ReadFrom1',
IPv4Addr:= GVL.PLC1Addr,
Reset := ResetRead1,
Error=> Error,
ErrorText=> ErrorText,
Done => Read1Done);
//Läs från PLC2 Omron
Read2(
SrcName:= 'Com2.Out',
DstName:= 'ETHIP.ReadFrom2',
IPv4Addr:= GVL.PLC2Addr,
Reset := ResetRead2,
Error=> Error,
ErrorText=> ErrorText,
Done => Read2Done);
//Starta datalagring och omorganisering när läsning är klart
100

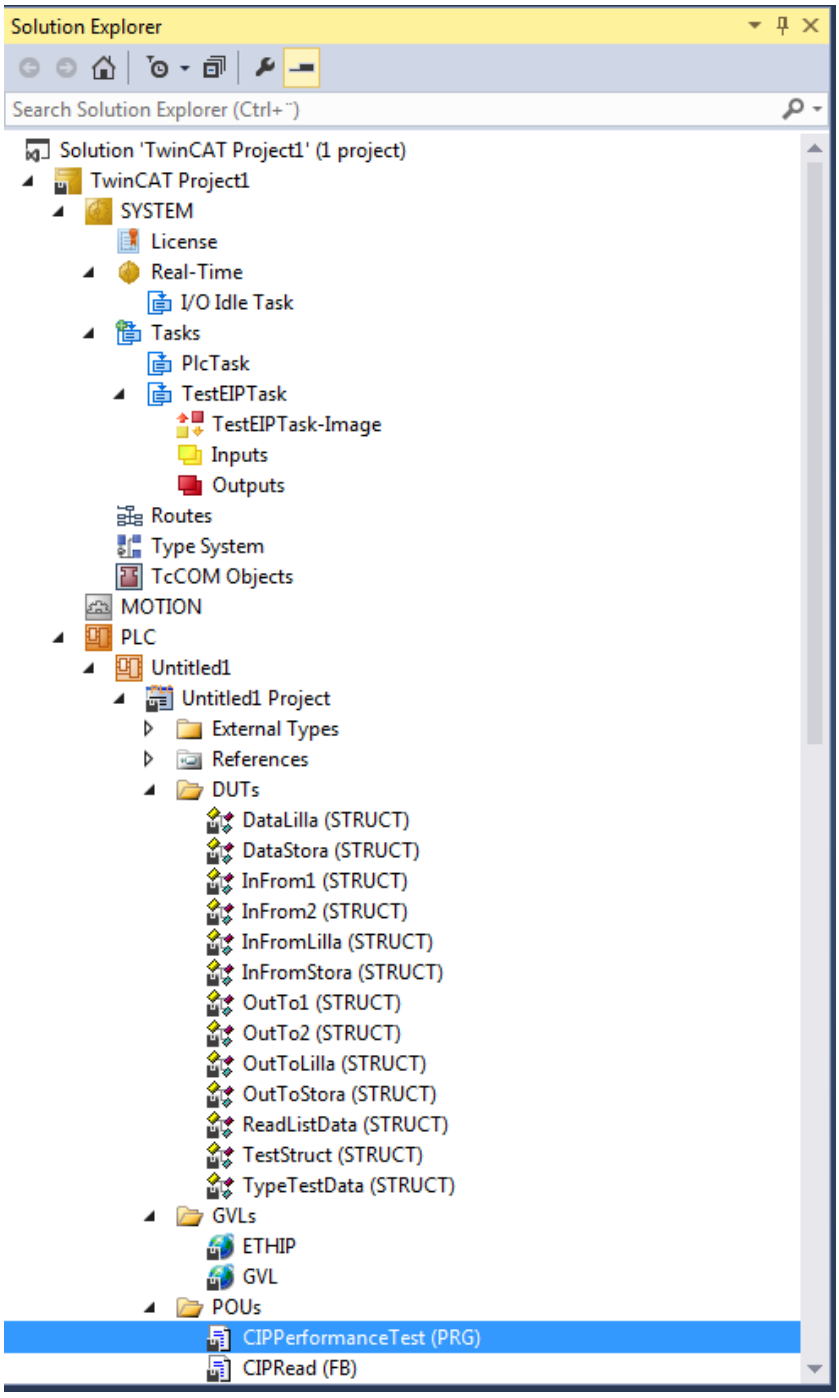

CIPPerformanceTest
CombinedTest
OmronCIP
OPCBasic
MAIN
TestStruct
ETHIP
GVL
CIPRead
79 Error=> Error,
80 ErrorText=> ErrorText,
81 Done => Read2Done);
82
83 //Starta datalagring och omorganisering när läsning är klart
84 IF ReadStoraDone AND ReadLillaDone AND Read1Done AND Read2Done THEN
85 // Spara läst data från Siemens lilla (PLC3)
86 DataStoreLilla.Running := ReadLillaData.Running ;
87 DataStoreLilla.Value := ReadLillaData.Value ;
88 DataStoreLilla.Temp := ReadLillaData.Temp ;
89 // Spara läst data från Siemens stora (PLC4)
90 DataStoreStora.Running := ReadStoraData.Running ;
91 DataStoreStora.Value := ReadStoraData.Value ;
92 DataStoreStora.Temp := ReadStoraData.Temp ;
93 //Flytta över data till skrivning (PLC3)
94 WriteLillaData.PLC4Running := DataStoreStora.Running ;
95 WriteLillaData.PLC1Value := ETHIP.ReadFrom1.Value;
96 WriteLillaData.PLC2Value := ETHIP.ReadFrom2.Value;
97 //Flytta över data till skrivning (PLC4)
98 WriteStoraData.PLC3Running := DataStoreLilla.Running ;
99 WriteStoraData.PLC1Value := ETHIP.ReadFrom1.Value;
100 WriteStoraData.PLC2Value := ETHIP.ReadFrom2.Value;
101 //Flytta över data till skrivning (PLC1)
102 ETHIP.WriteTo1.PLC2Value := ETHIP.ReadFrom2.Value;
103 ETHIP.WriteTo1.PLC2State := ETHIP.ReadFrom2.State;
104 ETHIP.WriteTo1.PLC3Value := DataStoreLilla.Value;
105 ETHIP.WriteTo1.PLC4Value := DataStoreStora.Value;
106 //Flytta över data till skrivning (PLC2)
107 ETHIP.WriteTo2.PLC1Value := ETHIP.ReadFrom1.Value;
108 ETHIP.WriteTo2.PLC1State := ETHIP.ReadFrom1.State;
109 ETHIP.WriteTo2.PLC3Value := DataStoreLilla.Value;
110 ETHIP.WriteTo2.PLC4Value := DataStoreStora.Value;
111 //Återställ och gå vidare till nästa del
112 TransferDone := TRUE;
113 StartSeq := FALSE;
114 ReadLillaDone := FALSE;
115 ReadStoraDone := FALSE;
116 Read1Done := FALSE;
117 Read2Done := FALSE;
118 ResetRead1 := TRUE;
119 ResetRead2 := TRUE;
120
121 END_IF
122 END_IF
100

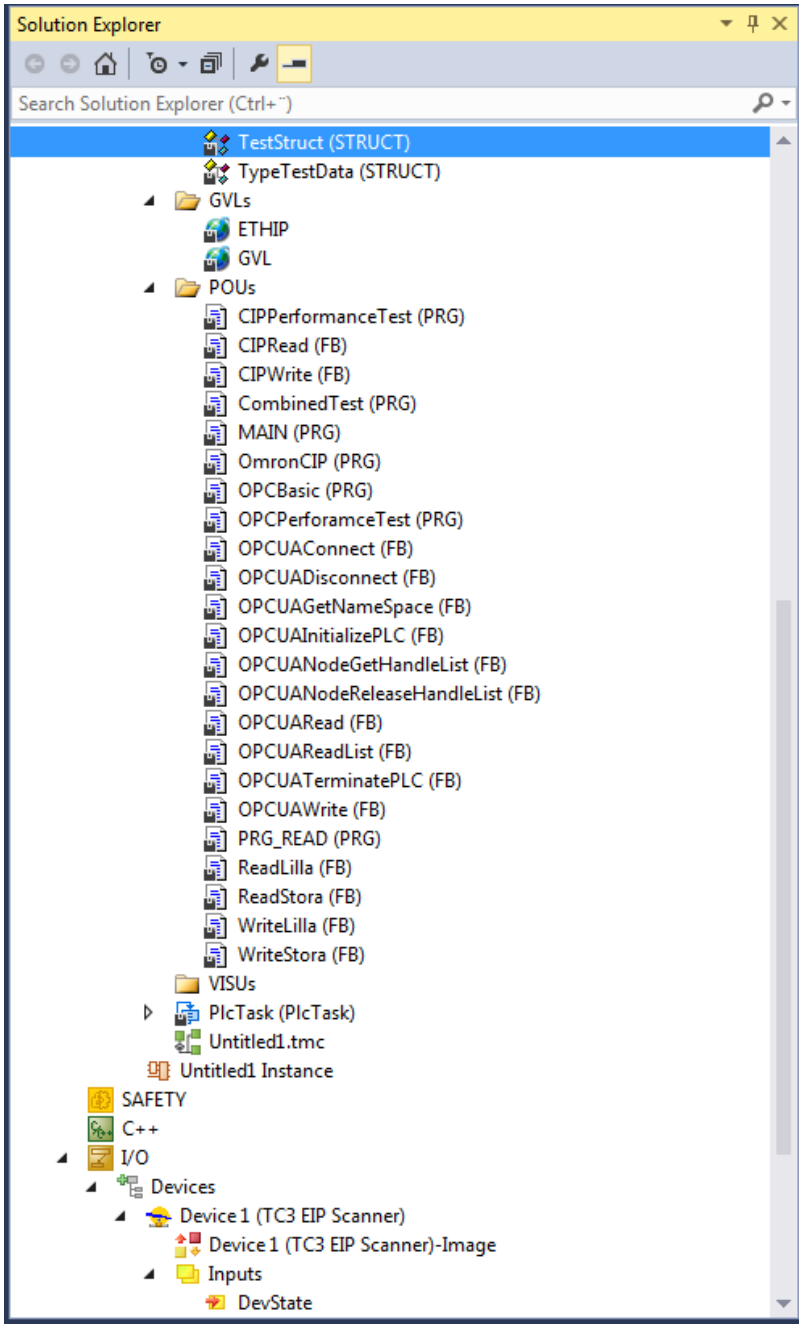

CIPPerformanceTest
CombinedTest
OmronCIP
OPCBasic
MAIN
TestStruct
ETHIP
GVL
CIPRead
117 Read2Done := FALSE;
118 ResetRead1 := TRUE;
119 ResetRead2 := TRUE;
120
121 END_IF
122 END_IF
123
124
125 //Starta skrivning när dataomlagringen är klar
126 IF TransferDone THEN
127 //Skriv till PLC3 (Siemens Lilla)
128 WriteToLilla(
129 DataToLilla:= WriteLillaData,
130 ConnectionHdl:= ConnectionHdlLilla,
131 NodeHdlList:= NodeHdlListLilla,
132 Error=> Error,
133 ErrorID=> ErrorID,
134 Done=> WriteLillaDone);
135
136 //Skriv till PLC4 (Simemens Stora)
137 WriteToStora(
138 DataToStora:= WriteStoraData,
139 ConnectionHdl:= ConnectionHdlStora,
140 NodeHdlList:= NodeHdlListStora,
141 Error=> Error,
142 ErrorID=> ErrorID,
143 Done=> WriteStoraDone);
144
145 //Skriv till PLC1 Omron
146 Write1(
147 SrcName:= 'ETHIP.WriteTo1',
148 DstName:= 'Com1.In',
149 IPv4Addr:= GVL.PLC1Addr,
150 Reset := ResetWrite1,
151 Error=> Error,
152 ErrorText=> ErrorText,
153 Done => Write1Done);
154
155 //Skriv till PLC2 Omron
156 Write2(
157 SrcName:= 'ETHIP.WriteTo2',
158 DstName:= 'Com2.In',
159 IPv4Addr:= GVL.PLC2Addr,
160 Reset := ResetWrite2,
100



Figur 19: TwinCAT program. Kommunikation med Omron PLC och Siemens PLC.

Appendix F: Komplett lista för innehåll i TwinCATprojekt





Figur 20: Navigerinsträd för hela TwinCAT projektet.