

Styrsystemutbyte från Uniview till ABB:s 800xA

Design, implementering och optimering



Raneem Abboud
Raidor Muhammad

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



LUNDS UNIVERSITET
Lunds Tekniska Högskola

Styrsystemutbyte från Uniview till ABB:s 800xA

Design, implementering och optimering

LTH vid Campus Helsingborg
Division of Industrial Electrical Engineering and Automation

av
Raneem Abboud & Raidor Muhammad
Juni, 2025

Copyright Raneem Abboud, Raidor Muhammad

LTH vid Campus Helsingborg

Lunds universitet
Box 882
251 08 Helsingborg

LTH at Campus Helsingborg
Lund University
Box 882
SE-251 08 Helsingborg
Sweden

Tryckt i Sverige
Lunds universitet
Lund 2025

Innehållsförteckning

Sammanfattning	7
Abstract	8
1 Inledning	10
1.1 Bakgrund	11
1.2 Syfte	12
1.3 Målformulering	12
1.4 Problemformulering	13
1.5 Motivering av examensarbete	13
1.6 Avgränsningar	13
1.7 Resurser	14
1.8 Arbetsfördelning	14
2 Teknisk bakgrund	15
2.1 Styrsystem historik	15
2.1.1 Lokal PLC med lokalt HMI	15
2.1.2 Flera PLC'er och ett övergripande SCADA-system	16
2.1.3 Distribuerade styrsystem (DCS)	17
2.2. ABB:s DCS-system 800xA	18
2.2.1 Soft Controller och simuleringsmiljö	18
2.2.2 Control Builder M	19
2.2.3 Plant Explorer Workplace	20
2.2.4 Function Designer	22
2.2.5 Process Graphics 2 (PG2)	22
2.2.6 OPC	23
2.3. Hårdvarukomponenter och signalöverföring	24
2.3.1 AC800M (CPU PM 866)	24
2.3.2 Modbus	25
2.3.3 Communications Link (COMLI)	26
2.4 Metoder att programmera PLC:er	27
2.4.1 IEC 61131-3 standard	27
2.4.2 ABB:s programmeringsspråk	28
2.4.3 ABB-specifika verktyg	28
2.5 Andra aspekter nödvändiga vid programmering	29
2.5.1 Beteckningssystem	29
2.5.2 Processförståelse	30

2.5.3 Förmåga att förstå elritningar	31
2.6 ECA 60	32
2.7 SDM2	33
2.8 Panna 3 i Uniview	34
3 Metod	35
3.1 Förberedelse och utbildningar	35
3.2 Fysisk förändring av kommunikationsvägar	36
3.3 Programmeringsmetodik och verktyg	37
3.3.1 Signalstruktur och I/O-lista	37
3.3.2 Programmering i Function Designer (FBD)	38
3.3.3 Exempel på programmering i Function Designer	39
3.3.4 Programmering i Control Builder (ST)	44
3.4 Operatörens gränssnitt och Faceplate	45
3.5 IFAT och FAT	47
3.6 Utcheckning och driftsättning	48
3.7 Förbättringar och ändringar	50
4 Analys	52
4.1 Olika val av programmeringsspråk	52
4.2 Utmaningar och lösningar	53
4.2.1 Felaktig dokumentation	53
4.2.2 Synkroniseringsproblem vid Modbus-läsning	54
4.2.3 Saknad historik	55
4.3 Begränsningar och stabilitetsproblem i utvecklingsverktygen	55
4.4 Källkritik	56
5 Reflektion och kritik	57
6 Resultat	58
7 Slutsats	60
7.1 Vidare arbete	61
8 Terminologi	62
9 Källförteckning	63
10 Appendix	65

Sammanfattning

Detta examensarbete har fokuserat på att modernisera styrsystemet för reservpanna 3 vid Hetvattencentralen i Landskrona, genom att ersätta ett äldre SCADA-baserat system Uniview med ett modernt distribuerat styrsystem (DCS) ABB 800xA. Arbetet har genomförts i samarbete med Deterministic Control AB, och med Landskrona Energi AB som slutkund.

Syftet med examensarbetet har varit att uppnå en fullständig systemmigrering där både hårdvara och mjukvara byts ut, samt att integrera äldre utrustning med moderna gränssnitt. Särskilt fokus har lagts på användningen av ABB:s verktyg inom 800xA-plattformen samt de programmeringsspråk som valts för att utveckla styrsystemet. Programmeringsarbetet har huvudsakligen utförts i det grafiska verktyget Function Designer med hjälp av Funktionsblocksdiagram FBD i Project Explorer, som möjliggjort en tydlig och strukturerad implementation av den grundläggande styrlogiken. För mer avancerade funktioner skapades en kontrollmodul (Control Module) i Control Builder med Structured Text (ST). Denna kombination av verktyg och programmeringsspråk har bidragit till att uppfylla slutkundens krav på funktionalitet och tydlighet.

Vidare utformades ett operatörsgränssnitt i PG2 med syftet att efterlikna och harmonisera med övriga processbilder inom anläggningen. I arbetet genomfördes även testmoment såsom Factory Acceptance Test (FAT), utcheckning och driftsättning, där systemets funktionalitet verifierades tillsammans med slutkunden.

Resultatet demonstrerar hur det nya styrsystemet uppfyller samtliga krav från slutkunden. Driftsättningen visade att brännaren kunde startas som den ska och att temperatur- och effektreulatorernas funktionalitet var korrekt i verkligheten. En betydligt förbättrade struktur, överskådlig och användarvänlig processbild i 800xA presenterades, vilket saknades i det gamla systemet. En provkörning upp till 5 MW kördes för att kontrollera systemets prestanda, vilket visade en stabil och precis reaktion av regulatorerna enligt inställt börvärde.

Arbetet har erbjudit en konkret tillämpning av både teoretisk kunskap och praktiska färdigheter inom automation, styrteknik och programmering i ett verkligt industriprojekt. Det har även stärkt kunskaperna inom systemintegration, felsökning och utveckling i ABB:s 800xA-miljö, områden som är viktiga inom industriell automation och relevanta för en framtida yrkesroll.

Nyckelord: P&ID diagram, 800xA-system, Control Builder, Function Designer, Process Graphics 2 (PG2), DCS, PLC, KKS.

Abstract

This thesis focuses on the modernization of the control system for Backup Boiler 3 at District heating plant in Landskrona. The objective was to replace an outdated SCADA-based system, Uniview, with the modern Distributed Control System (DCS) ABB 800xA. The project was carried out in collaboration with Deterministic Control AB, and with Landskrona Energi AB as the final customer.

The main goal was to achieve a complete system migration, involving upgrades of both hardware and software, as well as the integration of older equipment by modernizing the interfaces.

The thesis describes the methods, tools and communication solutions used to accomplish the system changes. Special focus has been placed on the use of ABB's tools within the 800xA platform and the programming languages chosen to develop the control system. The programming work has mainly been done in the graphical tool Function Designer using Function Block Diagram FBD on Project Explorer, which has enabled a clear and structured implementation of the basic control logic. For more advanced functions, a Control module was created in Control Builder with Structured Text (ST). This combination of tools and programming languages has greatly contributed to meeting the final customer's requirements for functionality and clarity.

Furthermore, a new operator interface was developed using PG2, designed to match and harmonize with other process displays within the facility. The thesis also included key test phases such as the Factory Acceptance Test (FAT), checkout, and commissioning, all conducted in close cooperation with the client to verify system functionality.

The results demonstrate how the new control system meets all the requirements of the end customer. The commissioning showed that the burner could be started as it should and that the temperature and power regulators were functioning correctly in reality. A significantly improved structure, clear and user-friendly process image in 800xA was presented, which was missing in the old system. A test run up to 5 MW was run to check the system performance, which showed a stable and precise reaction of the regulators according to the defined setpoint.

The thesis also addresses the technical and practical challenges encountered during the transition from the old to the new control system. Overall, the project provided a valuable opportunity to apply both theoretical knowledge and practical skills in automation, control engineering, and industrial programming. It also strengthened competence in system integration, troubleshooting, and development within ABB's 800xA environment skills that are essential in the field of industrial automation and relevant for future careers.

Keywords: P&ID diagram, 800xA-system, Control Builder, Function Designer, Process Graphics 2 (PG2), Uniview, DCS, PLC, KKS.

Förord

Detta examensarbete, som är avslutande del av programmet Elektroteknik och automation, har gjorts vid Lunds universitet LTH i samarbete med Deterministic Control AB och Landskrona Energi AB.

Vi vill passa på och tacka vår mentor Claes Olsson som har stöttat oss under arbetets gång, visat intresse och bidragit med betydande insatser. Stort tack till verkställande direktör Conny Franzon för värdefulla insikter och konstruktiv feedback under examensarbetets gång.

Vi tackar likaså vår examinator Ulf Jeppsson och vår handledare Erik Ottosson för vägledning och värdefulla råd under examensarbetets gång.

Ett speciellt tack till anställda på Deterministic AB som gav oss möjligheten att genomföra detta examensarbete. Vi tackar även för vänlig hjälpsamhet, mottagande, och att ni har varit lyhörda och ställt upp för oss när vi behövt vägledning och besvarat våra frågor.

Vi tar tillfället i akt att uttrycka tacksamhet till våra kärleksfulla familjer, partner och vänner för uppmuntran, stöd och uppmärksamhet genom denna satsning.

1 Inledning

I avsnittet introduceras bakgrunden till examensarbetet tillsammans med en beskrivning av dess syfte, målsättning, de problem som ligger till grund för studien samt avgränsningar.

1.1 Bakgrund

Examensarbetet genomförs i samarbete med Deterministic Control AB och har Landskrona Energi AB som slutkund. Arbetet har fokuserat på att uppdatera och förbättra styrsystemet för reservpanna 3 i Hetvattencentralen (HVC) belägen vid kraftvärmeverket Energiknuten i Landskrona, genom att ersätta det gamla SCADA-systemet med ABB:s moderna DCS-system 800xA. Examensarbetet är en del av ett större arbete som syftar till att modernisera och framtidssäkra styrningen av hela värmeproduktionen.

Landskrona Energi AB är ett kommunalägt energiföretag och ansvarar för flera viktiga delar av stadens energiförsörjning, bland annat fjärrvärme, elnät, stadsnät och elhandel.

En av företagets viktigaste anläggningar är Hetvattencentralen (HVC), där hetvatten produceras och distribueras till stadens fjärrvärmenät. **HVC** producerar bara värme och ingen el, vilket innebär att det fungerar som ett fjärrvärmeverk. I anläggningen finns 6 pannor där Panna 1, 2, 5 och 6 eldas med miljövänliga bränslen till exempel brännbart avfall, träflis och biogas. Däremot Panna 3 och 4 fungerar som reservkapacitet och startas enbart vid behov när det är hög belastning eller driftstörning. Båda pannor använder naturgasbränsle. [1]

Fjärrvärme är ett sätt att producera värme centralt på ett fjärrvärmeverk och sedan transporteras som varmt vatten via isolerade rör till olika bostäder. Det är ett energieffektivt och miljövänligt alternativ i motsats till att varje fastighet skulle ha sin egen värmekälla, exempelvis en panna eller värmepump. Fjärrvärme är vanligt i Sverige och används i över hälften av alla bostäder. Det är ett system där värmen kommer från förnybara energikällor till exempel flis, pellets och avfallsförbränningar. Det varma vatten leds tillbaka till anläggningen efter att det har använts för att värmas upp på nytt, vilket skapar ett effektivt värmekretslopp. [1]

Tidigare styrdes pannorna i HVC med hjälp av det gamla SCADA-systemet Uniview. Systemet betraktas nuförtiden som tekniskt föråldrat och det uppfyller inte moderna krav på säkerhet, funktionalitet och integration. Ett av de problem SCADA-systemet hade var att en del av nätverket sträckte sig fysiskt utanför fjärrvärmeverket vilket i sin tur ökade säkerhetsrisken. Efter modernisering av styrsystemet för några pannor, till exempel Panna 6, medför det äldre systemet begränsningar för operatörerna eftersom de styr och övervakar flera pannor från olika plattformar. Därför arbetar Landskrona Energi med ett moderniseringsprojekt där alla pannor stegvis uppgraderas till ABB:s 800xA-system, som erbjuder ett enhetligt DCS-system. Två av sex pannor har redan uppdaterats och detta examenarbete fokuserar på att flytta en av reservpannorna (Panna 3) till en och samma plattform ABB:s 800xA.

Deterministic Control AB (DCAB) har varit teknisk partner i examensarbete. Företaget är ett projektbaserat konsultbolag inom industriell automation, med lång erfarenhet av att leverera styrsystemlösningar. DCAB arbetar med PLC-programmering DCS-, HMI-, SCADA-lösningar samt elkonstruktion och driftsättning. De arbetar mot flera branscher såsom energi, VA, industri och marin och har stort erfarenhet av flera system som ABB:s 800xA, Siemens TIA Portal och andra etablerade plattformar. Deterministics breda erfarenhet och kompetens av ABB:s 800xA-systemet har varit en värdefull tillgång under genomförandet av examensarbete. [2]

1.2 Syfte

Syftet med examenarbetet är att modernisera och effektivisera övervakningen och styrningen av reservpanna 3 genom att ersätta det befintliga Uniview med ABB:s 800xA. Det förväntade resultatet är ett uppdaterat, mer tillförlitligt och användarvänligt system som underlättar operatörernas arbete och ökar driftfunktionaliteten.

1.3 Målformulering

I examensarbete ska följande mål nås:

- Utbyte av reservpanna 3 från Uniview till ABB:s 800xA-system.
- Utveckla ett nytt program för att hämta och hantera data från det befintliga systemet Uniview till ABB:s 800xA systemet, med bibehållen driftsäkerhet och anpassad funktionalitet.
- Designa och utveckla ett nytt användargränssnitt som är anpassat efter operatörernas behov och är identiskt med det befintliga systemet för de andra pannorna.

1.4 Problemformulering

Examensarbete ska besvara följande frågor:

1. Vilka tekniska utmaningar uppstår vid utbyte av reservpanna 3 från Uniview till ABB:s 800xA plattform?
2. Vilka förbättringar behöver göras i styrlogiken i 800xA (om det är omöjligt) för att öka systemets tillgänglighet och effektivitet?
3. Hur ska det nya gränssnittet designas och vilka funktioner måste vara med för att tillgodose operatörernas krav?

1.5 Motivering av examensarbete

Examensarbetet är kopplat till utbildningsinriktning mot automation och speglar uppgifterna som kan förekomma i framtida yrkesroller.

Detta examensarbete valdes eftersom det är ett praktiskt och konkret uppdrag som omfattar både utredning av vad som behöver göras och hur det kan genomföras. Det innebär också programmering och konfigureringsarbete av den nödvändiga förändringen. Examensarbetet ger möjlighet att kombinera teoretisk kunskap med praktiskt arbete genom att migrera ett föråldrat styrsystem till en modern DCS-plattform. Examensarbetet kommer att förbättra förståelse för framtida arbetsuppgifter inom industriområdet. Processen stärker problemlösning och problemlösning, vilka är centrala kompetenser i ingenjörskapet.

Slutkunden ser flera fördelar med examensarbetet; genom att konvertera reservpanna 3 till ABB:s 800xA-plattform blir styrsystemet enhetligt med resten av anläggningen, vilket förbättrar driften. Operatörerna får en effektivare och bättre arbetsmiljö där pannorna kan styras från ett gemensamt system.

1.6 Avgränsningar

Examensarbetet kommer att begränsas till att byta ut styrsystemet för reservpanna 3 från Uniview till ABB:s 800xA-plattform inkluderande förbättringar av styrlogiken samt utveckling av grafiska processbilder.

Fokus ligger på att använda programmet **Function Designer** för att skapa en fungerande programkod. Arbetet kommer även att innefatta simulerad demonstration för driften med grafiska processbilder under ett **FAT** (Factory Acceptance Test) samt signalutcheckning och driftsättning av det skarpa systemet tillsammans med slutkunden, Landskrona Energi AB.

Omkonstruktion av pannan, elritningar eller mätningar av dess parametrar ingår inte i examensarbetet.

1.7 Resurser

- ABB 800xA manualer.
- Utbildning inom programmet ABB system 800xA och Process Graphics (PG2).
- Deterministic Control AB utvecklingsmiljö, Denna miljö bestod av ABB:s 800xA Engineering Workplace inklusive tillhörande PLC-simulering (Soft Controller), där simulerades och testades styrsystemet innan det togs i drift.
- Elritningar och PI schema för Panna 3.
- Mentorerna med goda erfarenheter.
- Kontakter och material från Deterministic Control AB och Landskrona Energi AB.

1.8 Arbetsfördelning

	Raneem	Raidor
Teknisk bakgrund	60	40
Metod	40	60
Analys	50	50
Resultat	40	60
Slutsats	60	40
Poster	50	50

2 Teknisk bakgrund

Detta kapitel syftar till att ge en övergripande teknisk bakgrund om de verktyg, system och kommunikationsprotokoll som använts i examensarbetet. Det krävdes en förståelse av både hårdvara och mjukvara som ingår i automationssystemet för att genomföra en lyckad integration och styrning av processen.

Här presenteras styrsystemet AC800M tillsammans med de olika kommunikationsmoduler och de metoder som använts för signalöverföring mellan olika enheter. Vidare beskrivs även ABB 800xA-systemet, bland annat programmeringsverktyget Control Builder, simulering med SoftController och andra verktyg som används vid utveckling och felsökning.

En viktig del av bakgrunden handlar om de kommunikationsprotokoll som använts, till exempel Modbus RTU och Modbus TCP/IP. Dessutom beskrivs även standarder för programmering och beteckningssystemet för märkning av komponenter.

2.1 Styrssystem historik

Detta avsnitt beskriver olika varianter av styrsystem och hur de är uppbyggda. Ett styrsystem kan vara PLC'er, servrar och lokala operatörspaneler. Instrument- och styrsystem kan klassificeras på flera sätt beroende på applikation, bransch och syfte. En vanlig klassificeringsuppdelning på automationssystem är följande:

- Lokal PLC med lokalt HMI
- Flera PLC'er och ett övergripande SCADA-system
- Distribuerade styrsystem (DCS)

2.1.1 Lokal PLC med lokalt HMI

I början var automationssystem enkelt uppbyggda såsom lokal PLC med tillhörande lokalt placerad panel HMI. HMI står för Human Machine Interface, alltså en mellanliggande gränssnittsenhet som fungerar som lokalt maskingränssnitt i fält. Gränssnittet är vanligtvis monterade på PLC-panelen och i speciella fall ansluts de på distans genom att förlänga Ethernet-kabeln från PLC:n inuti panelen till önskad plats. HMI har ingen hårddisk, vilket betyder att den har ingen långtidshistorik.

Styrsystem startades med en PLC med ansvar för att hantera ingångar och utgångar och att utföra kontrollogiken. Ytterligare kommer HMI som möjliggör för operatörerna ett användarvänligt gränssnitt för att integrera med systemet. Med tiden utvecklades resterande delar av styrsystemet. [3]

2.1.2 Flera PLC'er och ett övergripande SCADA-system

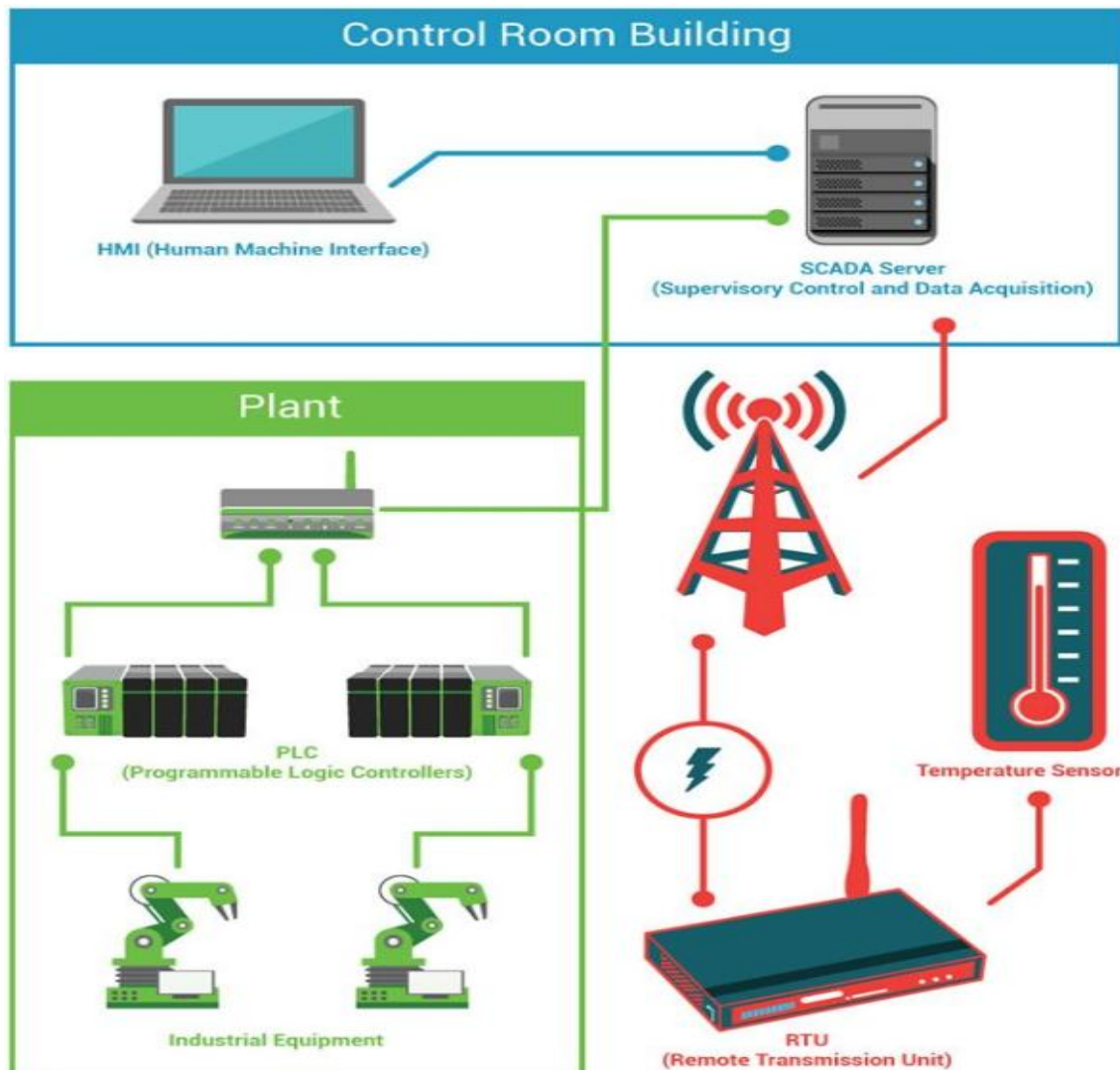
I början av 1990-talet utvecklades styrsystemen till en mer avancerad nivå av styrning, där en mjukvara i datorn introducerades som ett program för övervakning. SCADA-systemet har en hårddisk, vilket innebär att det finns möjlighet till historisk datalagring från systemet. Lagrade data kan också loggas i en SQL-databas (Structured Query Language), vilket ger enkel och effektiv dataanalys. Styrsystemet innehåller grafik som i slutändan kommunicerar med alla andra PLC'er, systemet kallas för SCADA (Supervisory Control and Data Acquisition), alltså övervakande kontroll- och datainsamlingssystem.

Systemet är vanligtvis uppbyggt av flera delsystem som ihop skapar arkitekturen:

- Remote Terminal Unit (RTU) för insamling av data
- Programmable Logic Controller (PLC) för styrning
- Human Machine Interface (HMI)
- Kommunikationsinfrastruktur
- Servrar för datainsamling, samordning och historik
- Processinstrument såsom regulatorer, ställdon och sensorer, se figur 1.

Systemet möjliggör några viktiga funktioner för operatörerna som: processvisualisering som möjliggör styrning av utrustning och övervakning, datainsamling och analys, visning av realtidstrender, historiska trender, rapportering, larmhantering och uppdatering av parametrar eller program online.

Systemet viktigaste egenskap är att den tillåter kommunikation mellan olika PLC'er även om de tillhör olika fabrikat, vilket underlättar integration i anläggningar mellan olika styrsystem under en gemensam plattform. Systemet kommer också att förbättra industriella anläggningar genom att standardisera systemarkitekturen, tillhandahålla realtidsdata, ökar produktiviteten, ökar drifttiden, effektiviserar gränssnitt och säkerställer regelefterlevnad. Systemet har också sina begränsningar, som till exempel det har en hög initial kostnad, den kan vara kommunikationsberoende som kan leda till fördröjd respons p.g.a. ett avbrott i nätverk eller störning samt att systemet har visat begränsad kontroll i realtid över snabba processer. [4]



Figur 1: Figuren visar typisk distribuerad SCADA-arkitektur [15]

2.1.3 Distribuerade styrsystem (DCS)

DCS-system står för Distributed Control system, det innebär utspridd reglering det vill säga ingen central regulator. Funktionaliteten hos DCS- och SCADA-system är mycket lika, men DCS tenderar att användas på stora kontinuerliga processanläggningar där hög tillförlitlighet och säkerhet är viktigt och kontrollrummet inte är avlägset. DCS-systems största skillnad från SCADA-system är att de är processororienterade och ständigt är i förbindelse med datakällorna medan SCADA-system är datainsamlingsorienterat och kan ha temporära uppkopplingar för datainsamling av data som lagras i databasen SQL (läs avsnitt 2.1.2).

DCS tar automatiseringen till nästa nivå genom att integrera flera PLC'er och andra komponenter. Systemen är designade för storskaliga komplexa processer som kräver en hög grad av koordination. Vanligtvis har systemet inte bara ett stort antal styrslingor, där autonoma styrenheter är fördelade i hela systemet, utan också ett centralt kontrollrum med operatörsarbetsplatser. Dessa arbetsstationer är anslutna till flera PLC'er och andra intelligenta enheter fördelade över hela anläggningen. Denna distribuerade arkitektur möjliggör större flexibilitet och redundans. [5]

Redundans används för att verksamhetskritiska system ställer höga krav på datakommunikation och tillförlitlig. Redundans undviker de allvarliga konsekvenser och garanterar maximal tillgänglighet till kommunikationstjänster, vilket innebär att vid problem med huvudförbindelsen eller utrustningen så ansluts förbindelserna via en annan väg. [6]

DCS passar bäst för storskaliga kontinuerliga processindustrier på grund av deras förmåga att hantera ett stort antal ingångar och utgångar tillsammans med deras robusta kommunikationskapacitet. Många anläggningar såsom kraftverk och fjärrvärmeverk är också beroende av DCS för exakt kontroll av temperatur, flödes hastigheter, tryck och andra kritiska parametrar. Systemen är idealiska för industrier som värme eller kraftproduktion. Några av DCS fördelarna är att den har hög tillförlitlighet och redundans, avancerade kontrollmöjligheter samt flexibilitet och skalbarhet, medan DCS vanligaste nackdelar att den är komplex, och har en hög initial kostnad.

Ett exempel på ett modernt och avancerat DCS-system är ABB:s 800xA-plattform. Därför kommer kommande avsnitt att förklara ABB:s 800xA verktyg och struktur och hur de användes i detta examensarbete.

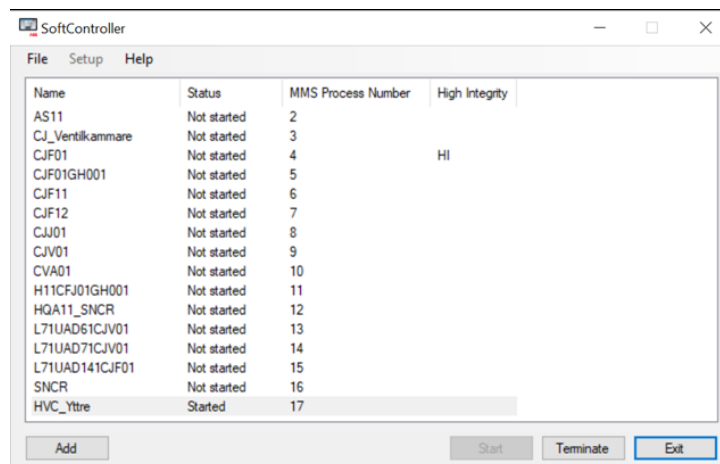
2.2. ABB:s DCS-system 800xA

ABB:s 800xA är ett Windows-baserat datasystem som används för att övervaka och styra industriella processer. Det är ett komplett system som gör arbetet enklare för både ingenjörer och operatörer där det samlar allt på ett och samma ställe - programmera styrsystem, köra simuleringar, felsökningar och kunna se realtidsdata.

I systemet används flera olika verktyg som tillsammans bildar en komplett miljö, där vissa av dessa verktyg behöver licens för kunna använda dem. Nedan beskrivs hur dessa verktyg är kopplade till varandra och vilken funktion varje del har i systemet.

2.2.1 Soft Controller och simuleringsmiljö

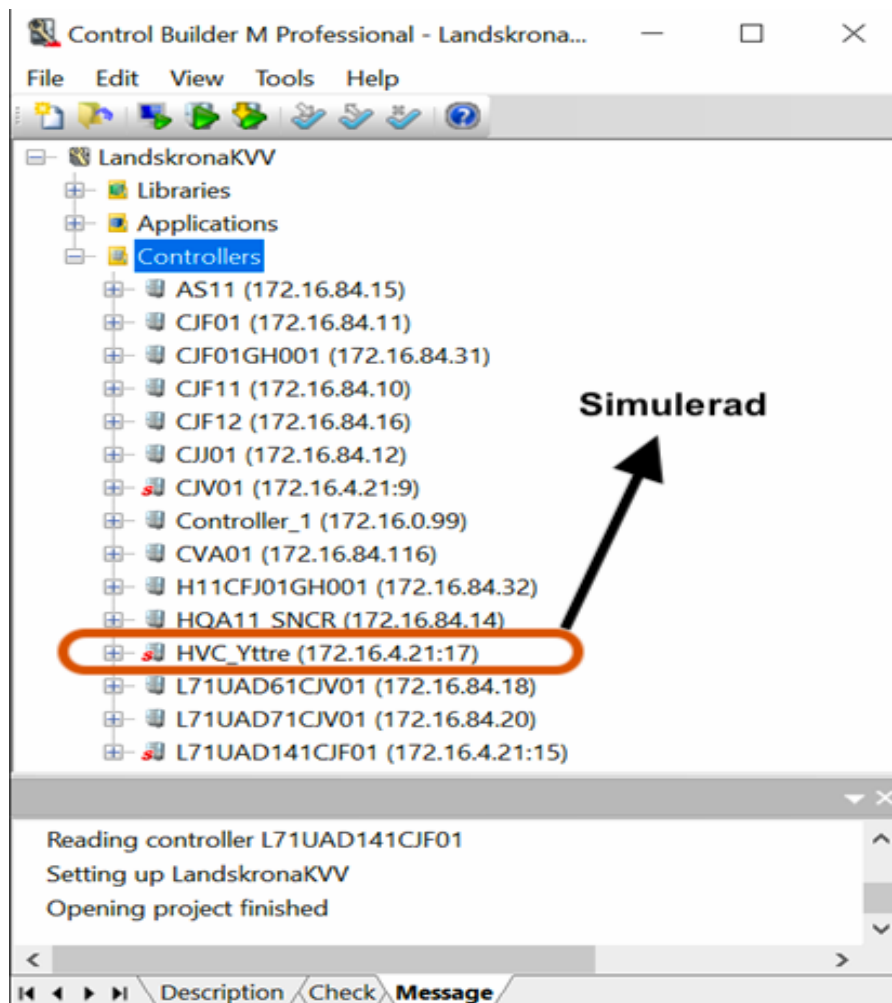
SoftController är en mjukvarubaserad kontroller som används i ABB:s 800xA-system för testning, simulering och utveckling av applikationer utan att den fysiska hårdvaran (till exempel: PLC) behöver vara inkopplad.



Figur 2: Mjukvaru-PLC, “HVC Ytre, startad” (DCAB, utvecklingsmiljö)

Det används i icke-säkerhetsklassade applikationer (**non-SIL Safety Integrity Level**) och är inte avsedd för system som kräver hög säkerhet, eftersom där används SoftController HI (High Integrity). Det som SoftController gör är att den beter sig som en riktig PLC, såsom att utföra styrlogiken och behandla in- och utsignaler. Dock har SoftController vissa skillnader jämfört med en fysisk kontroll, till exempel noggrannheten i beräkningar skiljer sig (64-bit för SoftController mot 32-bit för hårdvara) och har begränsade kommunikationsmöjligheter. Dessutom kräver SoftController särskild licens och används tillsammans med Control Builder i 800xA-systemet.

Simuleringen genomförs genom att programkoden nerladdas till SoftController, för att det ska fungera krävs att SoftController är aktiverad (se figur 2) och kopplad till rätt styrsystem via angiven IP-adress och i slutet (MMS Process Number) till exempel: (172.16.4.21:17) se figur 3. När enheten körs i simulerat läge, som visas nedan i figur 3, markeras detta med ett “**S**” i Control Builder för det ska skilja sig från den fysiska kontrollen. När det är nerladdat, testas koden antingen direkt i Control Builder, eller i processbilden för att visualisera och kontrollera funktionaliteten. [7]



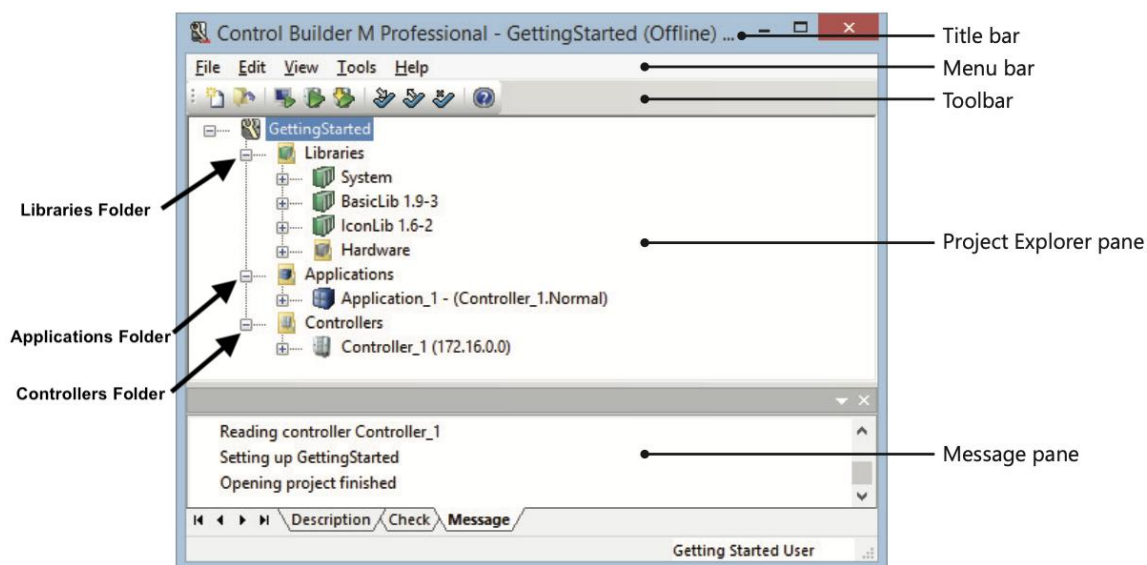
Figur 3: Simulerad Controller “HVC Yttre” markerad i Control Builder M.
(DCAB, Utvecklingsmiljö)

2.2.2 Control Builder M

Control Builder M är ett verktyg som används för att skapa och konfigurera PLC:ns källkod. Det är ett centralt verktyg i ABB:s 800xA-systemet och används för att konfigurera hårdvarukomponenter och programmera styrlogiken. Verktöget följer standarden IEC 61131–3 och stödjer flera programmeringsspråk som passar olika användare och behov (se avsnitt 2.4.2 för programmeringsspråk som följer standarden).

Det innehåller även kompletterande verktyg som Function Designer, Control Module Diagram Editor (CMD Editor) och Control Diagrams Editor som hjälper att bygga upp styrsystemets logik på ett visuellt och överskådligt sätt. Dessa verktyg gör det möjligt att se aktuella värden, ändra parametrar och felsöka logiken utan att stoppa systemet. [8]

I Control Builder används Projekt Explorer Pane för att navigera och organisera projektets olika delar, figur 4. Den är uppdelad till tre huvudmappar: **Libraries**, **Applications** och **Controllers** som ger tydlig översikt över vad som finns i projektet.



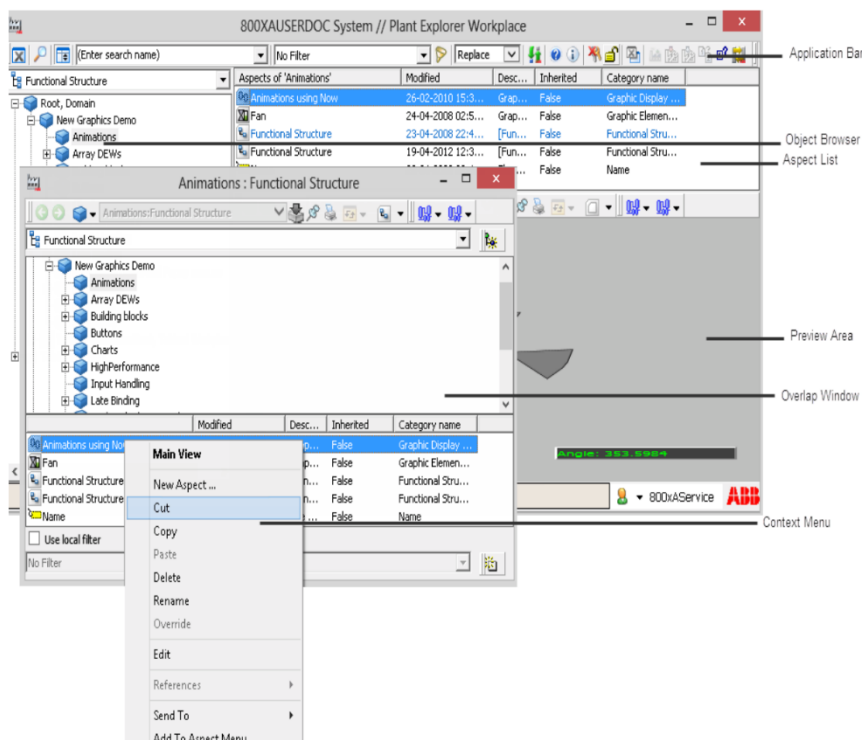
Figur 4: Projekt Explorer Pane i Control Builder. [7]

Libraries Folder används för att hantera både användardefinierade och inbyggda bibliotek som behövs för att skapa funktionalitet inom projektet. Applications Folder, här skapas applikationer som kopplas till rätt styrsystem. Och i Controllers Folder finns information om alla styrsystem, både de som är kopplade fysiskt och simulerade. [7]

2.2.3 Plant Explorer Workplace

Detta är en central del av ABB:s 800xA-system och används för att navigera, hantera och konfigurera de objekt som finns i systemet. Det används av ingenjörer och tekniker för att skapa och organisera en tydlig struktur i anläggningen, där alla delar av systemet placeras i en hierarkisk modell.

I Plant Explorer organiseras objekten i olika strukturer beroende på funktion, placering och användningsperspektiv. Verktøjets användargränssnitt efterliknar Windows Explorer, där objekt presenteras i ett trädformat på vänstra sidan (Object Browser). Skillnaden är dock att i Plant Explorer visas underobjekt (s.k. child objects) enbart i trädet, medan på högre sidan (Aspect List) visas en aspektlista som innehåller information och funktioner som är kopplad till det valda objektet, se figur 5. [9]



Figur 5: Användargränssnittet Plant Explorer Workplace. [9]

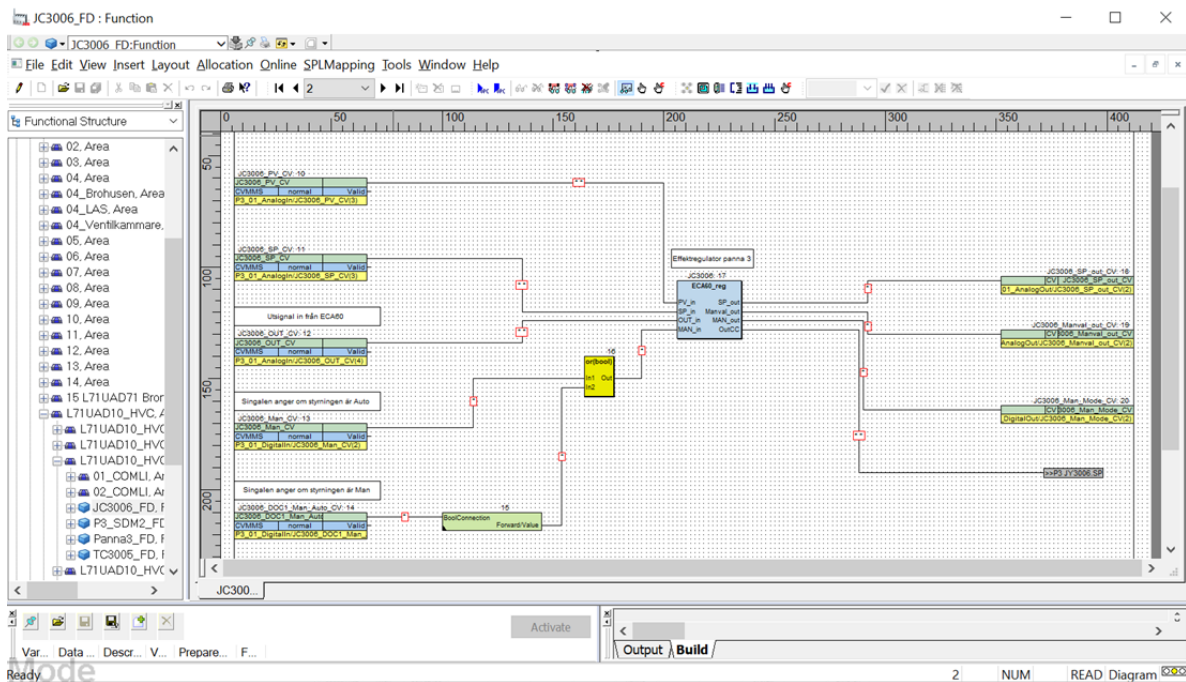
Aspect List används för att visa, redigera och lägga till aspekter såsom processgrafik, larmlistor, OPC-konfiguration och Trend Display. Plant Explorer ger tillgång till flera parallella strukturer som visar olika sätt att betrakta systemet: [9]

- **Control Structure:** Visar nätverket, fältbussar och kontrollstationer.
- **Functional Structure:** Återspeglar fabriken logiska processer och funktioner.
- **Object Type Structure:** Visar objekt baserat på deras typ.

2.2.4 Function Designer

Function Designer är ett grafiskt verktyg som gör det möjligt för ingenjörer att designa styrsystem baserat på processens funktioner, utan att behöva djupa kunskaper i programmering. I verktyget ritas så kallad Function Block Diagram (FBD), se figur 6. Det är blockschema där det kopplas ihop olika delar som motorer, ventiler, sensorer och andra styrkomponenter. Varje komponent representerar en aspekt av ett objekt i systemet (Aspect Object) och dessa kan kopplas med signaler, sekvenser och kontrollmoduler.

Fördelen med Function Designer är att den är lätt att använda och har många funktioner så som att se processen i realtid genom att starta OPC server och visa signaler. Dessutom behövs det inte att se på procesbilder för att öppna en Faceplate utan bara höger klicka på objekten och öppna det. [10]



Figur 6: Function Designer i Plant Explorer.
(DCAB, utvecklingsmiljön)

2.2.5 Process Graphics 2 (PG2)

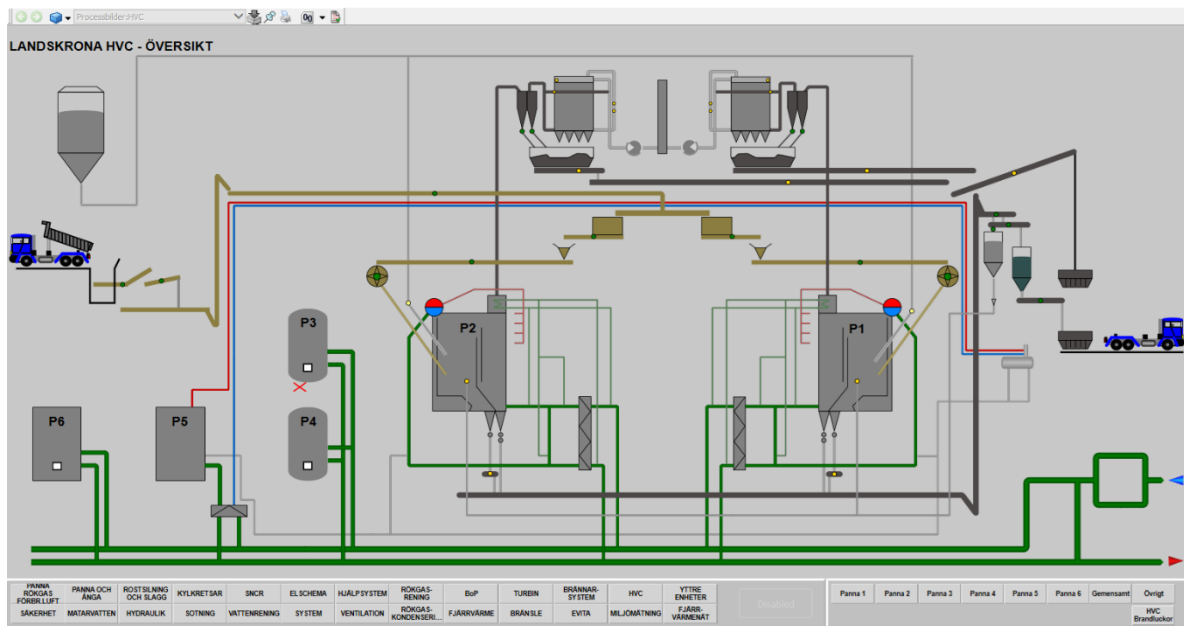
Process Graphics 2 är den nya generationens grafiks-system i ABB:s 800xA-system och används för att skapa processbilder – visuella skärmbilder där operatörerna kan övervaka och styra till exempel en anläggning.

PG2 gör det möjligt att bygga moderna, flexibla och användarvänliga grafiska gränssnitt. Genom att använda verktyg som Graphics Builder, ges möjlighet att dra in symboler som motorer, rör och tank mm. Därefter kopplas dessa symboler till olika signaler som i sin tur visas på processbilden som i figur 7.

I PG2 skapas tre olika typer av grafiska objekt:

- **Graphics Display:** hela skärmbilden med fast grafik och live-data.
- **Graphics Element:** mindre fönster som kan användas i olika processbilder.
- **Generic Element:** generella grafiska komponenter med flexibla funktioner.

Det är en viktig del i operatörens miljö i 800xA-systemet, och är byggt för att vara snabbt, tydligt och lätt att använda för både operatörer och ingenjörer. [11]



Figur 7: Grafisk processbild på HVC. Grafiken skapades som extra och frivilligt arbete. (DCAB, utvecklingsmiljö)

2.2.6 OPC

OPC står för (Object Linking and Embedding (OLE) for Process Control), det är en etablerad mjukvaruteknik som beskriver kommunikation av realtidsdata mellan hårdvara och mjukvarukomponenter inom industrin med andra ord koppling mellan Windows och automationshårdvara.

OPC är ett standardgränssnitt för data-, händelse- och historikåtkomst baserat på COM (Component Object Model). COM är en teknik från Microsoft för att koppla olika mjukvarumoduler även om de har olika programmeringsspråk. OPC är en standardiserad kommunikations plattform för SCADA- och DCS-system. Det är det största och vanligaste protokollet mellan olika tillverkare och har olika varianter:

- OPC DA (Data Access): överföra realtidsdata mellan server och klient
- OPC AE (Alarm and Event): överföring av larm och händelser
- OPC UA (Unified Architecture): är den nyaste, mest avancerade versionen och säkraste. Det som gör OPC UA den säkraste är att den kräver certifikat och kryptering mellan OPC UA, alltså till exempel att bestämma vem har rätt att skriva, läsa eller övervaka data. Denna version är oberoende av Windows/COM och använder flexibelt kommunikationsprotokoll. [3]

2.3. Hårdvarukomponenter och signalöverföring

I detta avsnitt presenteras de hårdvarukomponenter som möjliggör styrning av systemet, samt vilka metoder för signalöverföring som tillämpats i examenarbetets genomförande. I signalöverföring belyses både seriella och nätverksbaserade överföringsprotokoll bland annat Modbus RTU, Modbus TCP och COMLI. Målet är att beskriva dessa protokolls funktion och hur de används i praktiken vid övergång från Uniview till 800xA.

2.3.1 AC800M (CPU PM 866)

AC800M är en hårdvara som ingår i 800xA-paketet. Hårdvaran består av PLC PM866 och kommunikationskort CI853, CI853, och CI867 för olika protokoll. AC800M-styrenheten kan stödja både nätverks- och seriell kommunikation, som syns i figur 8.

CI853-modulen är kommunikationsgränssnittet från AC800M kontroller till externa enheter som använder Modbus RTU eller COMLI-protokoll, modulen kan monteras i AC800M. Kommunikationstypen för modulen är seriellt med RS-232 eller RS-485 och det underlättar koppling för äldre industriella enheter, till exempel de gamla fältutrustningar som ska integreras med 800xA-system. Detta leder till att modulen spelar en stor roll vid modernisering av äldre system.

CI867-modulen är kommunikationsgränssnitt från AC800M kontroller till Modbus TCP via Ethernet, kortet kan monteras i AC800M. Den använder Ethernet 100 Mbit/s, vilket ger hög hastighet och stor kapacitet som underlättar hantering av flera enheter samtidigt. Modulen är utrustad med dubbla kanaler CH1 och CH2, och används när det är många enheter eller behövs hög hastighet. CI867 är lämpligt för att integrera gamla enheter med ett modernt system som 800xA. [3]



Figur 8: Från vänster till höger syns kommunikationsmodulerna CI853, CI853, CI867, CI854A, följt av CPU modulen PM866 för styrsystemet AC800M. (Hämtad från Landskrona fältinstrument)

2.3.2 Modbus

Modbus är en kommunikationsstandard som används för att överföra data mellan enheter inom processkontroll och industriell automation. Det finns flera olika protokoll som skiljer sig åt beroende på i vilka enheter de är utformade för att användas med och hur de överför data, men de vanligaste varianterna är Modbus RTU och Modbus TCP.

Modbus RTU står för (Remote Terminal Unit). Enheter som använder detta protokoll kommunicerar vanligtvis via något av tre elektriska gränssnitt RS-232, RS-485 eller RS-422. Dessa gränssnitt används för **seriell kommunikation**. I Modbus RTU överförs data med hjälp av binärt format samt informationen kodas så att ett värde överförs som ett 16-bitars heltal, vilket innebär att de värde som kan representeras är mellan 0–65 535, se figur 18.

Både RS-232 och RS-485 har gemensamma principer som bygger på Master/Slav, där en master-enhet styr kommunikationen. Skillnaden mellan dessa är att RS-232 används för punkt-till-punkt-kommunikation mellan två enheter, medan RS-485 tillåter kommunikation mellan flera enheter på samma buss – upp till 32 noder (en master och upp till 31 slavar).

Modbus RTU som överföringsmetod har sina för- och nackdelar. Fördelarna är att den är den snabbaste seriella varianten av Modbus-protokollet med en överföringshastighet mellan 300 bps och 115.2 kbps. Nackdelar kan vara att det maximala avståndet mellan enheterna kan varieras beroende på överföringshastigheten där längre kablar kan påverka kommunikationskvaliten, och max avståndet kan vara cirka 1200 meter. Dessutom Modbus RTU har svårigheter vid felsökning av kommunikationsproblem. [6] [3]

Modbus TCP/IP står för (Transmission Control Protocol) och (Internet Protocol). Den är en modern version av Modbus-protokollet som är baserad på **nätverkss kommunikation**, vilket innebär att den använder Ethernet-kablar och switchar för kommunikation mellan enheter. Modbus TCP/IP kombinerar ett fysiskt nätverk (Ethernet), med en nätverksstandard (TCP/IP) och en standardmetod för att representera data (Modbus).

Datakommunikationshastigheterna har förbättrats från seriell busskommunikation på 31,25 kbit/sek till normala Ethernethastigheter på 10/100 Mbit/sek eller högre, och systemet kan hantera flera enheter jämfört med seriella protokoll. Ett Modbus TCP/IP-nätverk består av en klient och flera servrar, där klienten är kopplad till en eller flera switchar, och varje switch därefter är ansluten till flera servrar.

Modbus TCP har många fördelar såsom att den har förmågan att stödja obegränsat antal enheter med hjälp av IP-adressering i stället för seriella kommunikationsgränser. Lika viktigt är dess kompatibilitet till olika typer av nätverk samt att den är lätt att implementera.

Om gamla enheter som bara stöder seriell kommunikation, alltså Modbus RTU, ska anslutas till ett modernt system som använder Modbus TCP, så behövs en omvandlare (gateway), figur 9. Denna omvandlare tar emot seriella signaler och konverterar dem till Modbus TCP. [3]



Figur 9: En kommunikationsgateway (EDW-100) med stöd för Modbus RTU och Modbus TCP/IP. [12]

2.3.3 Communications Link (COMLI)

Kommunikationslänk (COMLI) är ett kommunikationsprotokoll som ABB utvecklat. Det är inte lika vanligt idag men kan fortfarande förekomma i äldre system. Det är en sort av **seriell kommunikation** såsom **Modbus RTU**, men den används först och främst för dataöverföring mellan äldre PLC'er. COMLI använder RS-232 eller RS-485 som fysiskt gränssnitt. [3]

2.4 Metoder att programmera PLC:er

I detta avsnitt förklaras IEC 61131-3 standarden och de olika typer av programmeringsspråk som används i allmänhet för att programmera alla sorter PLC:er samt vilka verktyg som speciellt erbjuds av ABB.

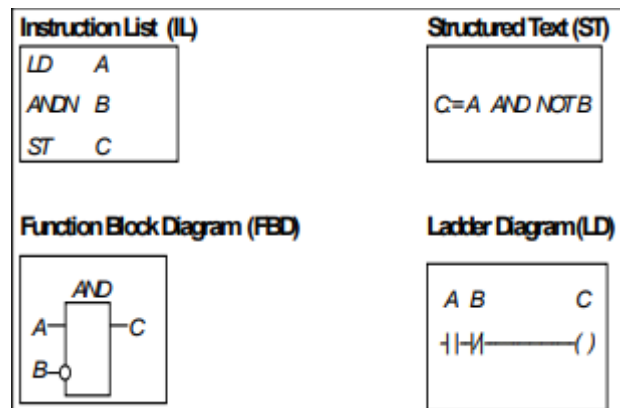
2.4.1 IEC 61131-3 standard

Denna standard förtydligar syntax och semantik för en enhetlig uppsättning programmeringsspråk för PLC:er. Dessutom underlättar den för användaren att byta programmiljö oavsett vilken programvara som används.

I IEC 61131-3 Standard finns det bara fyra programmeringsspråk definierade för att programmera alla typer av PLC:er. Två av dem är textbaserad, vilka är InstruktionsLista (IL), Strukturerad Text (ST), och de andra är grafiska, vilka är LadderDiagram (LD), och FunktionsBlockDiagram (FBD), se figur10.

- **Instruktionslista:** Är lågnivåspråk som liknar Assembler.
- **Strukturerad text:** Det är ett högnivåspråk som kan kommunicera med andra språk baserade på gemensamma strukturella element och liknar Pascal.
- **Ladder diagram:** Är det ursprungliga PLC-programmeringsspråket som ger en grafisk representation av kretselement.
- **Funktionsblockdiagram:** I detta språk beskrivs funktioner som block med in- och utgångsanslutningar. Det är det vanligaste språket inom processindustrin. Det

uttrycker beteendet hos funktioner, funktionsblock och program som en uppsättning sammankopplade grafiska block, som i elektroniska kretsscheman. Den tittar på ett system i termer av flödet av signaler mellan bearbetningsselement, figur 10. [13]



Figur 10: Figuren visar fyra olika språk som beskriver samma enkla program. [13]

2.4.2 ABB:s programmeringsspråk

Enligt ABB's systemguide funktionsbeskrivning för 800xA system så är Control Builder-redigerarna kompatibla med följande IEC 61131-3 programmeringsspråk:

- FunktionsBlock Diagram (FBD).
- Ladder Diagram (LD).
- Strukturerad Text (ST).
- Sekventiell Funktion Diagram (SFC).
- InstruktionsLista (IL). [8].

2.4.3 ABB-specifika verktyg

Följande extra verktyg ingår för att optimera konfigurationsprocessen för styrlogik, den ena är Funktionsdesigner och den andra är Styrmoduler.

Function Designer (FD): Användning av detta verktyg ger en betydande ökning av effektiviteten. För mer information hur den fungerar läs avsnittet 2.2.4. [8]

Control Modules (CM): Verktyget kontrollmodul som finns i 800xA-systemet är unikt jämfört med många andra automationssystem. Det är utvecklat för att underlätta konstruktion och hanteringen av kontrollslinga i komplexa industriella processor.

En kontroll loop är en sluten regleringsstruktur där en processvariabel (till exempel temperatur, tryck eller flöde) mäts, jämförs med ett börvärde och justeras automatiskt av ett styrsystem för att hålla processen stabil. För att detta ska fungera effektivt krävs att signalerna överförs både framåt (från mätvärde till regulator och vidare till styrsignal) och bakåt (till exempel information om begränsningar, status eller spårning).

Det är en objektorienterad byggsten som kombinerar logik och ett standardiserat gränssnitt som kallas för ControlConnection (CC) för att kommunicera med andra moduler.

I figur 11 nedan visas skillnaden mellan den moderna lösningen med ControlConnection och den traditionella. Där i traditionell kaskadkoppling mellan två regulatorer som på vänstra sidan visas att varje signal måste kopplas manuellt, vilket kräver mycket erfarenhet för det är komplext. I moderna lösningar används ControlConnection som är standardiserat gränssnitt där den automatiskt hanterar alla signaler mellan moduler. [14]

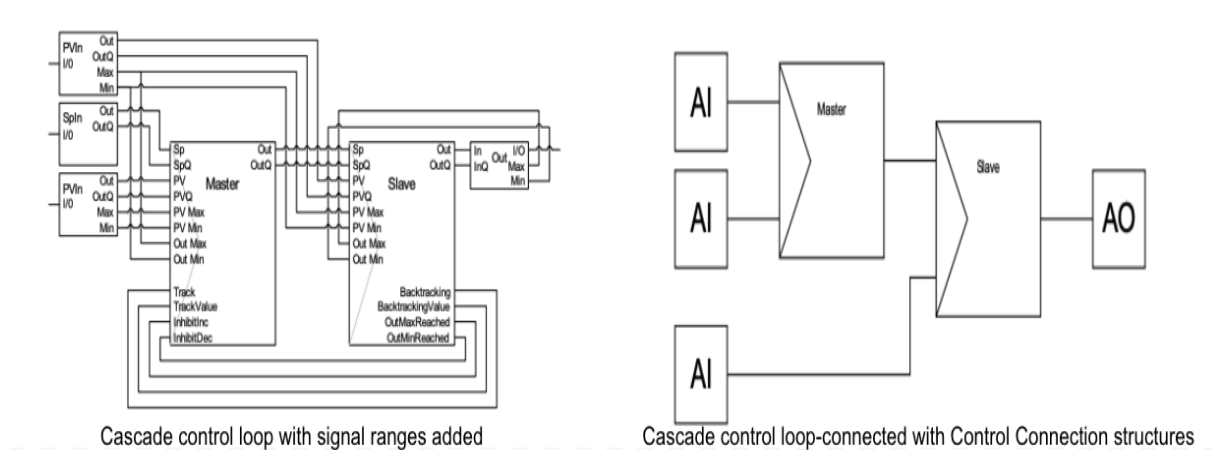


Figure 11: Skillnaden mellan en kaskadreglering med och utan ControlConnection. [14]

2.5 Andra aspekter nödvändiga vid programmering

I detta avsnitt förklaras vilka andra verktyg som en automationsingenjör använder för att skriva en bra programmeringskod. För att kunna uppnå det behövs ett tydligt beteckningssystem, processförståelse och förmåga att förstå el-ritningar.

2.5.1 Beteckningssystem

DOC80- Beteckningssystem:

Tidigare användes DOC80 ABB's namngivning och dokumentationsstandard från ABB.

KKS-Beteckningssystem (Kraftwerk- KennzeichenSystem):

Kraftverksidentifikationssystemet (KKS) är ett system för att identifiera anläggningar, system, delsystem, apparater, el- och styrskåp samt byggnader och rum enligt perspektivet hos den som driver ett kraftverk. KKS utgör också grunden för tillförlitlig anläggningsdokumentation.

KKS använder 15- till 17-siffriga bokstavs- och sifferkombinationer för att identifiera systemet. Särskilt i de stora kraftverken måste strukturen finnas med beteckning av instrument. Bokstäverna som finns används för att klassificera system och enheter i kraftverket. Siffrorna används för uppräknings.

Till exempel kan en beteckning vara så här (ES-XA10AA001), dessa bokstäver och siffror innebär följande: ES– Elkraftsystem, XA10– Huvudtransformator (XA refererar till transformator, 10 är identifiering av plats eller funktion) och AA001 betyder första transformatorn i systemet. Några av de vanligaste prefixen enligt KKS- beteckningssystem och deras betydelse visas i tabell 1.

Tabell 1: beskrivning av några av KKS-prefix

KKS prefix	Beskrivning
ES	Elektrisk System
CE	Kontroll och Styrsystem
TE	Temperaturmätning

Analys av standarder: Det aktuella kraftverket 'Energiknuten' använder beteckningssystemet ibland, men inte helt och hållet i panna 3 och 4. Anläggningen använder sig av sitt eget beteckningssystem, vilket beror mest på att hela systemet är gammalt. Några exempel på benämning som användes ses i tabell 2.

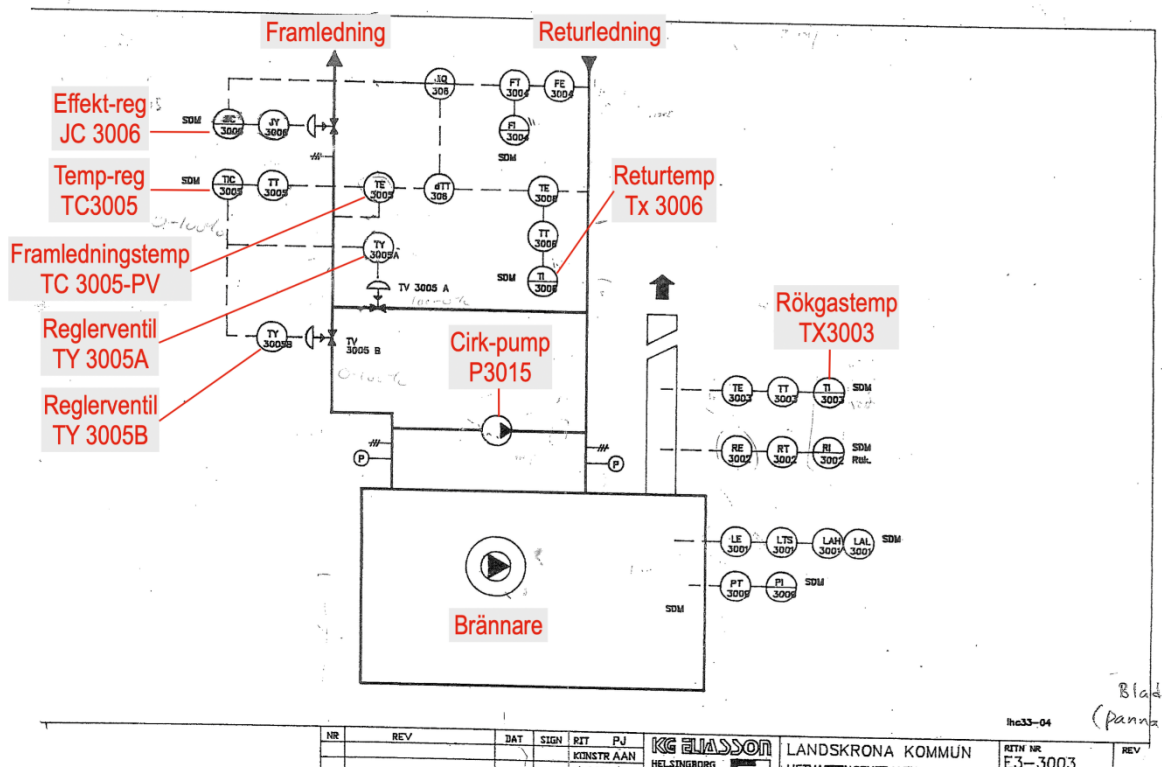
Tabell 2: beskrivning av några av KVV2012 prefix.

Prefix	Beskrivning
FI	Flöde Instrument
TT	Temperatur Transmitter
V	Ventil
PT	Tryck Transmitter
UA	Unit Alarm (Larmcentral)
X	Plint

2.5.2 Processförståelse

PI-schema står för (Process and Instrument schema), och är ett viktigt "verktyg" vid programmering för förståelse av anläggningens funktioner. Schemat är nödvändigt för att förstå processen och vägleda kring logiken, i det ingår processkunskap om hur en anläggning är uppbyggd. Det visar styrning, flöde, signalvägar och var utrustning är placerad.

För att tolka PI-schema behövs en instrumentförteckning som medföljer schemat. Dokumentet innehåller detaljerad beskrivning av instrumentfunktioner såsom ID på instrument, vilket objekt det är, gradering, samt vilka processer som måste implementeras i PLC. Instruktioner visar hur olika signaler ska hanteras, samt hur PLC-programmering ska struktureras, se figur 12.



Figur12: PI-schema på panna 3. (Landskrona Energi AB egen dokumentation)

Ett exempel på beskrivning av ett instrument:

TIC 3005 regulator SattControl ECA40-0000
 Ingångar: 4st analoga 4–20 mA
 Binära in- och utgångar
 Utgångar: 2st analoga 4–20 mA
 Inställning: direkt verkan

Det beskriver en regulator med namnbeteckning TIC3005 av fabrikat SattControl och modell 40-0000, regulatören tar emot analoga signaler med 4–20 mA intervall och skickar ut en strömsignal 4 till 20 mA. Inställning på direktverkan innebär att utsignalen ökar i takt med mätsignalens ökning.

Instrumentförteckningen innehåller också interna namnbeteckningar för varje instrument som används i anläggningen. Inom HVC-anläggningen används inte ett fullständigt KKS-system.

2.5.3 Förmåga att förstå elritningar

El-ritningar eller el-schemaförståelse är en värdefull färdighet för att ordna arbetet. Scheman redovisar komponenter, strömkällor och ledningsdragning, anslutningar, säkerhetskretsar och skydd.

Några metodiska genomgångar som används för att analysera en el-ritning är först och främst att kunna identifiera huvudkomponenter, följa strömmens väg, förstå kontrollkretsar, analysera

säkerhetskretsar och att tolka hur de mätvärdena används i systemet. Färdighet att läsa ett el-schema sparar tid och resurser vid felsökning men även vid integrering av ett gammalt styrsystem till ett nytt. Nedan på figur 13 visas en el-ritning för en brännarautomatik.

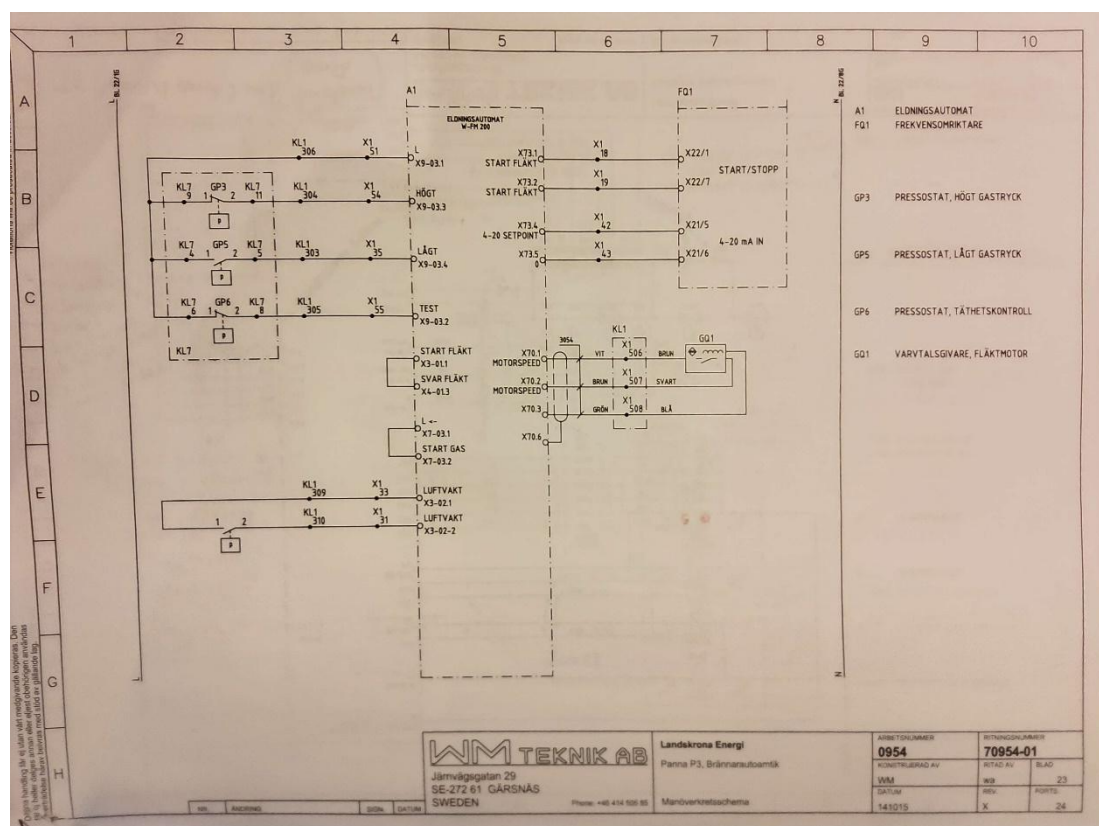


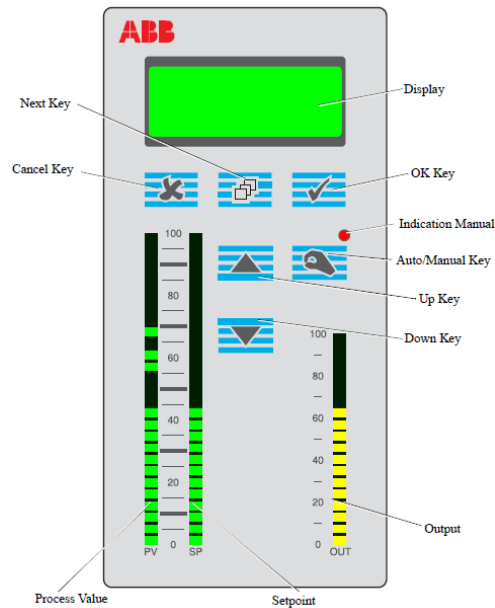
Figure 13: Elritningar på brännarautomatik. (Landskrona Energi AB egen dokumentation)

2.6 ECA 60

ECA 60 är en elektronisk processregulator från ABB, se figur 14, avsedd för industriella styr- och reglerapplikationer. Enheten installeras vanligtvis i panel och kommunicerar med övriga delar av styrsystemet via analoga och digitala I/O, samt seriella protokoll som Modbus RTU eller COMLI. Regulatorn erbjuder en flexibel struktur med stöd för olika typer av regleringar såsom en tvåstegsreglering, PID-reglering och logiska funktioner.

Det kan hantera både analoga och digitala signaler och innehåller ett brett utbud av funktionsblock såsom:

- PID-reglering (med stöd för Gain Scheduling och Autotuning)
- Analoga och digitala I/O (inkl. Kommunikationsblock AOC och DOC)
- Spårning (tracking), två- och trestegutgångar.



Figur 14: ECA 60 regulator (fysisk enhet), kontroller och indikatorer. [16]

2.7 SDM2

SDM2 är ett äldre PLC-verktyg som har en omfattande programvara. I denna ingår en mängd förprogrammerade funktionsblock, vilka kan användas för att ge systemet önskade egenskaper. De olika blocken som finns kan kombineras fritt vid konfigurationen.

Konfigurationen av de analoga och digitala signalerna som ska hanteras och behandlas av systemet, görs från ett alfanumeriskt tangentbord. Detta tangentbord ansluts till funktionstangentbordet som svarar för signalöverföringen till centralenheten. Informationen om systemet och de olika blocken fås via bildskärmen.

Ett SDM2 grundsystem består av följande delar:

- En centralenhet innehållande CPU-kort och bussmodem-kort
- En eller två Operatörsplatser med tillhörande operatörstangentbord samt teckenskrivare
- Anslutningsmoduler

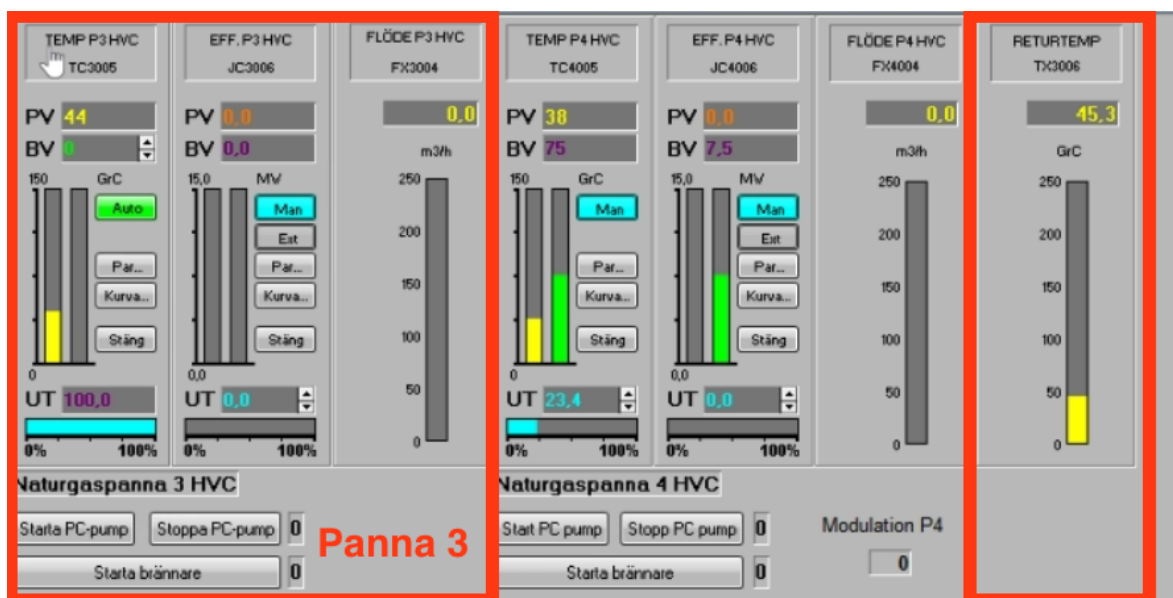
Liksom andra moderna styrsystem behöver SDM2 anslutningsmoduler för att kunna kommunicera med fältutrustning.

2.8 Panna 3 i Uniview

I Uniview-systemet delade Panna 3 och Panna 4 en gemensam processbild (se figur 15) som fungerade för övervakning och manövrering av båda pannorna. Denna processbild innehåller två separata regulatorer – en effektregulator och en temperaturregulator för respektive panna, där samtliga mätvärden presenteras. Temperaturregulatorn kan dock endast hantera

automatiskt driftläge och tillåter därmed ingen manuell hantering, vilket begränsar operatörernas kontroll.

I gränssnittet visas även övriga signaler såsom flöde och returtemperatur tillsammans med indikatorer (1 = starta, 0 = stoppa) som representerar cirkulationspumpens och brännarens status. Designen i Uniview är funktionellt uppdelad men saknar flexibilitet vad gäller individuell styrning, utökad signalvisning och modern användaranpassning.



Figur 15: Processbild på Panna 3 och Panna 4 i Uniview

Dessa begränsningar påverkar processoperatörernas arbete, särskilt eftersom övriga pannor i anläggningen styrs via ABB:s 800xA-system medan Panna 3 och 4 fortfarande hanteras i Uniview. Vidare i metodavsnittet, avsnitt 3.7, redogörs för vilka funktioner och förbättringar som införts i det nya systemet tillsammans med processoperatörernas förbättringspunkter för att uppnå en enhetlig, mer flexibel och operatörsanpassad lösning.

3 Metod

Syftet med detta kapitel är att redogöra för den arbetsgång och de metoder som använts för att nå syftet under examensarbetets gång, från utbildning till slutliga tester och driftsättning. Material som ligger till grund för arbetet har tillhandhållits av Landskrona Energi, bland annat projektmanual, I/O-lista och processbilder på reservpanna 3 och 4 och HVC som visar hur de ser ut i Uniview.

I början av examensarbetet bestämdes att funktionsblocksdiagram skulle användas som programmeringsspråk för de flesta delar eftersom det var slutkundens önskemål, medan till exempel PID-regulator och andra mer komplexa funktioner skulle programmeras med strukturerad text i Control Builder.

En central del i examenarbetet var att utveckla ett användarvänligt operatörsgränssnitt i 800xA-miljön. Gränssnittet utformades så att det efterliknar andra processbilder i anläggningen, till exempel Panna 6, men samtidigt är specifikt för Panna 3 och de signaler som är kopplade till den.

3.1 Förberedelse och utbildningar

Förberedelserna inför examensarbetets genomförande inleddes med ett möte med slutkunden. Under mötet presenterades examensarbets syfte, omfattning och vilka moment som ingick. Kunden gick igenom vad som skulle göras och vilka material som skulle lämnas in för att göra det möjligt att genomföra arbetet. Materialet som lämnades in var projektmanual för slutkundens första 800xA-projekt 2011 (som presenterar hur HVC-anläggningen är uppbyggt), elritningar för Brännarautomatik, I/O-lista för SDM signaler, processbilder på Panna 3 och 4 och HVC från SCADA-systemet Uniview.

Därefter hölls vid två tillfällen utbildningar (workshop) av uppdragsgivaren för att tydliggöra vad 800xA och PG2 handlar om. Det första tillfället handlade om 800xA allmänt och vilka fördelar och svagheter den har. Workshopen sträckte sig över 4 dagar och där presenterades plattformens struktur och vilka verktyg som skulle arbetas mest med såsom Control Builder, SoftController, Plant Explorer och Function Designer.

Den andra workshopen fokuserade på Process Graphics 2 (PG2) som används för att skapa operatörsgränssnitt. Eftersom en viktig del i examensarbete handlar om att bygga och utveckla ett gränssnitt för panna 3, var workshopen till stor nytta för hur det skulle göras. Under denna interna workshop visades hur grafik byggs upp, hur signaler kopplas till objekt och hur en processbild designas som båda är funktionell och enhetlig med övriga delar av anläggningen.

Dessutom erhöles mer information om och förklaringar av hur nätverket ser ut och fungerar vid anläggningen. För att ytterligare fördjupa företåelsen anordnades ett särskilt tillfälle tillsammans med uppdragsgivare, där genomgång av elritningar samt en praktisk visning av ett konkret exempel i verklig miljö genomfördes. Förutom utbildningstillfällena gavs ett

kontinuerligt lärande under examensarbetets gång som visade sig vara mycket värdefullt, särskilt vid programmering av styrlogiken där förklaringar och stöd gavs av mentorn.

3.2 Fysisk förändring av kommunikationsvägar

All kommunikation mellan fältutrustningen och det gamla styrsystemet hade redan flyttats över till det nya systemet av slutkunden innan examensarbetet påbörjades. Denna förändring gjordes utan att dokumenteras fullt ut, och av detta skäl uppstod svårigheter att efteråt följa upp hur det gamla systemet var uppbyggt och fungerade. Det innebar att i vissa fall fick extra tid läggas på att förstå tidigare kommunikationsvägar manuellt och att tolka signalflöden.

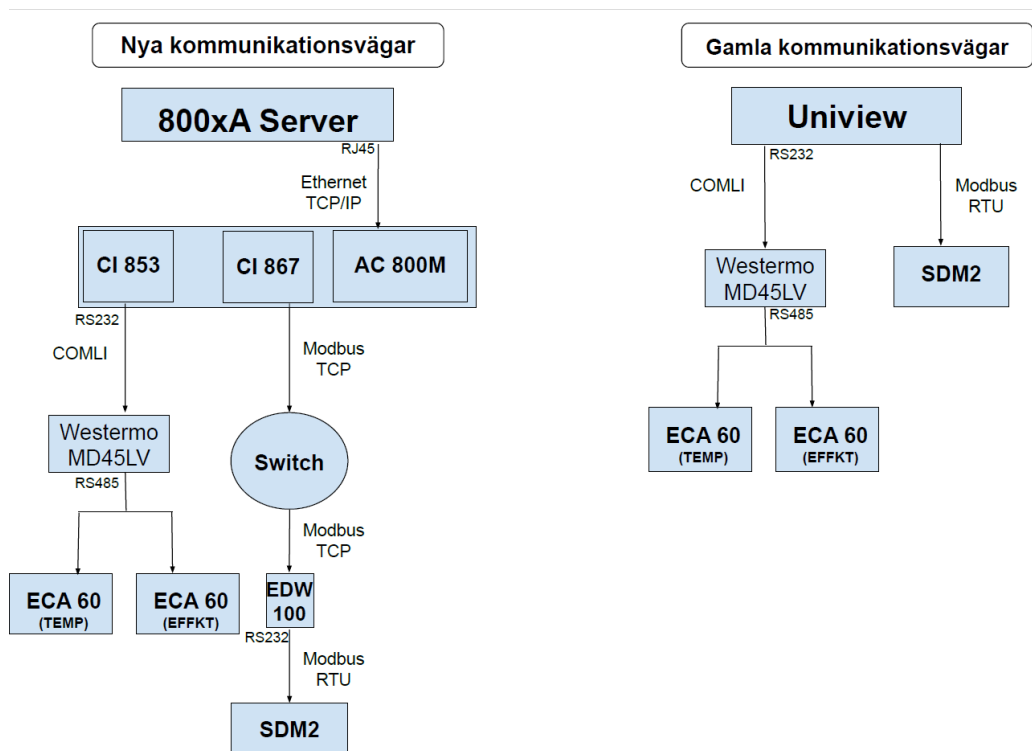
I Figur 16 nedan visas nuvarande systemuppbyggnad jämfört med den gamla, inklusive vilka kommunikationsprotokoll som användes. För den ursprungliga systemstrukturen med Uniview var kommunikationen baserad på ett äldre serieprotokoll (COMLI), som är i huvudsak punkt-till-punktförbindelse. Uppgraderingar samt felsökning var utmanande på grund av låg standardisering och dokumentation. Strukturen har moderniserats efter utbyte till 800xA systemuppbyggnad.

Figur 16 visar att kommunikationen sker mellan 800xA Server och PLC:n (AC800M) med hjälp av ett nätverksbaserat protokoll Ethernet TCP/IP, vilket förbättrade tillförlitlighet och ökade överföringshastigheten tillräckligt mycket. Kommunikationen använder de vanliga fysiska gränssnitt RJ45 i ett Ethernet-baserat nätverk. PLC:n har två kommunikationsmoduler:

- CI867 för nätverkskommunikation mot Server och Switch.
- CI853 för seriell kommunikation mot ECA60 via COMLI-protokollet.

Vidare är PLC:n ansluten till en nätverksswitch, som i sin tur fördelar kommunikationen. Modbus TCP-protokoll användes för att kommunicera med Gateway EDW-100, protokollet översätts till Modbus RTU seriell kommunikationen, i den andra ändan av gateway, som är kopplad till SDM2 via det fysiska gränssnittet RS232.

Förändringen i kommunikationsstruktur skedde under övergången från Uniview till ABB 800xA. Från början var kommunikationskopplingen från Uniview och direkt till SDM2. Kopplingen var en seriell kommunikation via protokollet Modbus RTU och det fysiska gränssnittet RS232. Denna direkta koppling avvecklades, och i stället baserades den nya kommunikationskedjan där PLC:n kommunicerar via Modbus TCP över Ethernet till Gateway EDW-100, som sedan kopplades till SDM2, som förklarades innan. Denna lösning fördelar var att den antog Ethernet-baserad arkitektur snarare än punkt-till-punkt-förbindelse. Den andra fördelen var att den gamla SDM-strukturen kunde återanvändas utan stora förändringar.



Figur 16: Visar kommunikation mellan olika komponenter efter och före utbyte

3.3 Programmeringsmetodik och verktyg

Här beskrivs metodik och de verktyg som använts vid utvecklingen av styrsystemet i 800xA. Fokus ligger på valet av programmeringsspråk och standarder, användningen av I/O-listan som grund för signalstrukturerings samt de två programmeringsmiljöerna - Function Designer och Control Builder.

3.3.1 Signalstruktur och I/O-lista

Programmeringen av styrsystemet i Panna 3 utgick från en I/O-lista som skickades från slutkunden. Denna lista visade vilka signaler som pannan har och vilka som skulle kommuniceras med SDM2 (den gamla PLC:n) och den nya styrningen. Informationen i I/O-listan låg till grund för hur funktionsblocksdiagrammen strukturerades, samt hur varje instrument kopplades till rätt adress.

Vid genomförandet visade det sig dock att den lista som levererades i början inte var komplett (se figur 17), vissa signaler saknades helt. Ett konkret exempel är signalen för att starta pannan (**Signalnamn 3050, IO-referens DO519**) som inte var med i andra versioner av listan heller, vilket upptäcktes av en processoperatör under FAT:en.

I detta examensarbete löstes många frågor genom direkt kontakt med slutkunden, ofta via telefon eller Teams för att fråga om I/O-listan eller PI-schema osv.

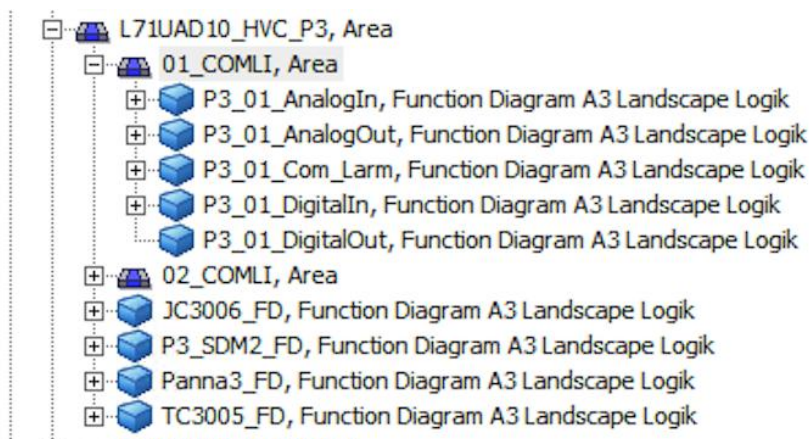
Signal Namn	Undercentral	IO referens	Beskrivande text	Värde	Enhet	Undre	Övre	Decimaler
JC3006.DIN	131	AI15	DIG. LÅSREG. JC3006	0	MW	0	50200	0
JC3006.B	131	AO17	BÖRVÄRDE JC3006	0	MW	0	15	1
JC3006.DOUT	131	AO31	DIG. SKRIVREG. JC3006	0	%	0	65535	0
JC3006	131	MO1	EFFEKT PV	0	MW	0	15	1
JC3006-Man	131	CR1	Man	1				
TC3005.DIN	132	AI15	DIG. LÅSREG. TC3005	0	°C	0	65535	0
TC3005.DOUT	132	AO31	DIG. SKRIVREG. TC3005	0	%	0	65535	0
TC3005	132	MO1	FRAML. TEMP. P3 HVC	63	°C	0	150	0
BR3051	101	DI242	DRIFTSVAR BRÄNNARE P3					
P3015	101	DI243	DRIFTSVAR PC PUMP P3					
P3015.T	101	CR119	STARTA PC PUMP P3	DS				
P3015.F	101	CR120	STOPPA PC PUMP P3	DS				
TX3006	101	AI147	RETUR TEMP P3		°C	0	150	0
Analoga								
JC3008	101	AI127	EXT BÖRVÄRDE EFF. P3 P4	0				
FX3004.calc	101	AI145	Flöde P3	0	m ³ /h			
TX3006	101	AI147	RETURTEMP. P3	72	°C			
PX3009	101	AI148	ELDSTADSTRYCK P3	-10	kPa			
RX3002	101	AI149	RÖKTÄTHET P3	0	%			
TX3003	101	AI150	RÖKGASTEMP. P3	27	°C			
Digitala								
BR3051	101	DI242	BRÄNNARE P3	0				
P3015	101	DI243	PC PUMP P3	0				
LARM P3	101	DI246	SUMMALARM P3	0				

Figur 17: I/O-listan från slutkunden

3.3.2 Programmering i Function Designer (FBD)

Programmering för Panna 3 utfördes på ett strukturerat sätt för att uppnå hög tydlighet. Samtliga signaler delades in och grupperades enligt deras funktionella tillhörighet. De delades enligt figur 18 och genom att:

- Alla signaler som tillhör effektreulatorn (*JC3006*) och temperaturregulatorn (*TC3005*) är placerade i Area *01_COMLI* respektive *02_COMLI*. Dessa områden inkluderar alla signaler som är kopplad till regulatorerna, det vill säga Analoga in/utgångar (AI, AO), Digitala in/utgångar (DI, DO) samt kommunikationslarm (COM-Larm).
- Alla signaler från SDM2 i FD (*P3_SDM2_FD*), bland annat Rökgastemperatur, flöde och alla larm som kommer från SDM:en.
- Cirkulationspump och brännare är också samlade i ett gemensamt funktionsdiagram (*Panna3_FD*).



Figur 18: Uppdelning i Function Diagram beroende på signaler (DCAB, utvecklingsmiljö)

3.3.3 Exempel på programmering i Function Designer

I detta avsnitt presenteras tre praktiska exempel som illustrerar hur programmering har genomförts i Function Designer för styrsystemet. Syftet med exemplen är att visa hur olika typer av signaler och logiken hanteras i projektet, både vad gäller kommunikation med externa enheter, säkerhetsstyrning samt anpassningar till äldre fältutrustning.

Dessa fall visar hur olika funktionella block från ABB:s bibliotek har använts, i kombination med anpassad logik för att uppfylla både tekniska krav och operatörernas önskemål. Exempelen omfattar:

Det första exemplet visar hur AI-sigaler har programmerats i *01_COMLI*, som tillhör effektregulator (*JC3006*). I figur 20 presenteras hur samtliga pages har sammanställts:

För att läsa in de analoga signalerna från det externa styrsystemet **SDM2** används *COMLI*-protokollet. I **page 1** av signalflödet som visas i figur 20 används blocket **COMLIConnect** för att påbörja kommunikationen. Anslutningen styrs av en boolesk aktiveringssignal, och utgångarna från COMLIConnect används för att övervaka kommunikationen och vidare skicka data till **COMLIRead-blocket** i **page 2**. **Read-blocket** används för att läsa signalvärden från den externa enheten SDM2, där Rd [1] till [5] lagras i interna variabler i Double Integer format och skickas vidare till konvertering och skalning. Blocket aktiveras med ett definierat läsintervall och rapporterar sin **Status**, **Error**, och **Ndr** (New data received) för att övervaka kommunikationen.

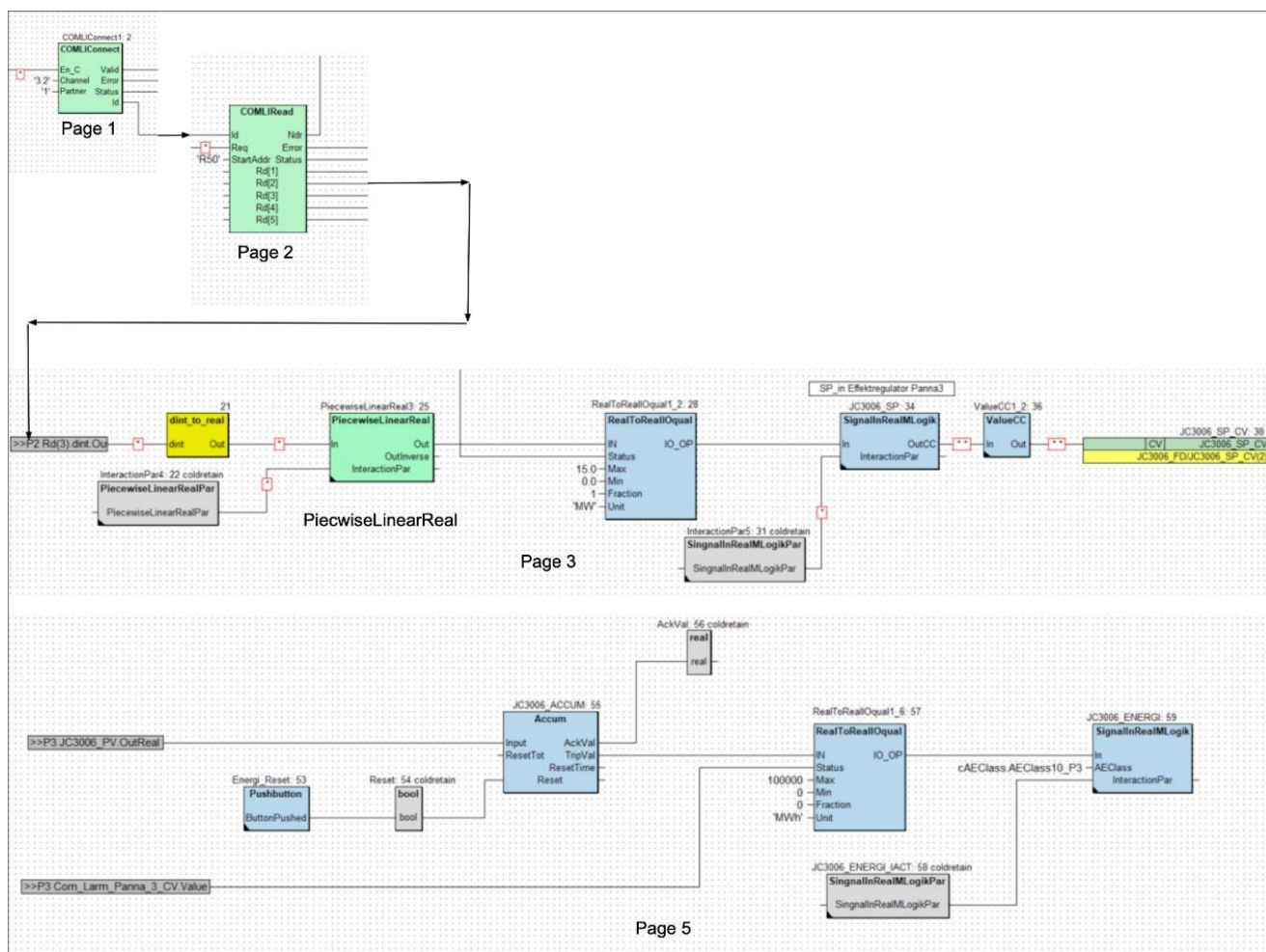
De råa Double Integer signalerna skickas vidare för att bearbetas, blocket **dint-to-real** används för att konvertera formatet. Och därefter till ett **PiecewiseLinear-block** som används för att linjärt skala om signalens värde från ett 16-bitars heltalsintervall (0 - 65535) till ett realvärde till exempel (0 - 15); ett XY-skalningsblock. Blocket har ett tillhörande gränssnitt (Interaction)

Window), vilket gör det möjligt att justera parametrar och gränser direkt (se figur 19) utan att behöva ändra i källkoden.



Figur 19: Interaction window från Control Builder vid online-läge för *PiecewiseLinear*.
(DCAB, utvecklingsmiljö)

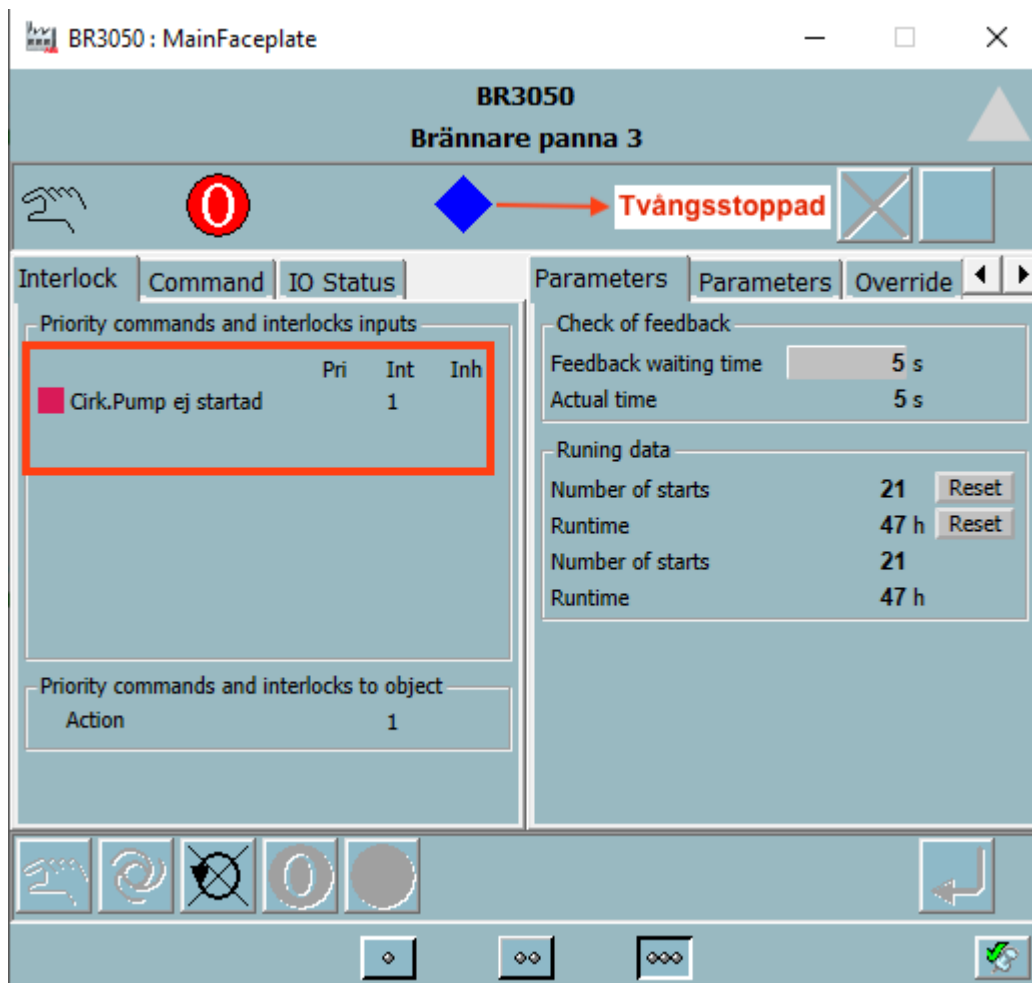
Det linjäriserade värdet skickas därefter vidare till ett *RealToRealIOqual*-objekt (se page 3 i **figur 20**) som används för att paketera flyttal med ytterligare attribut och metadata. *RealToRealIOqual* är en kopia av *RealToRealIO* från ABB:s bibliotek *UtilityLib* men kompletterad med till exempel statusfunktion. Detta möjliggör korrekt presentation av värdet i operatörsgränssnittet när det skickas som färdigbehandlat realvärde till *SignalInRealMlogik* som också är ett objekt från ABB:s bibliotek.



Figur 20: Ett exempel visar hur alla 5 pages (utan page 4 för det är samma som 3:an) i Analoga IN (AI) signaler för effekregulatorn är programmerad. För tydligare figur och mer info om analoga in, se Appendix Figur (1–5). (DCAB, utvecklingsmiljö)

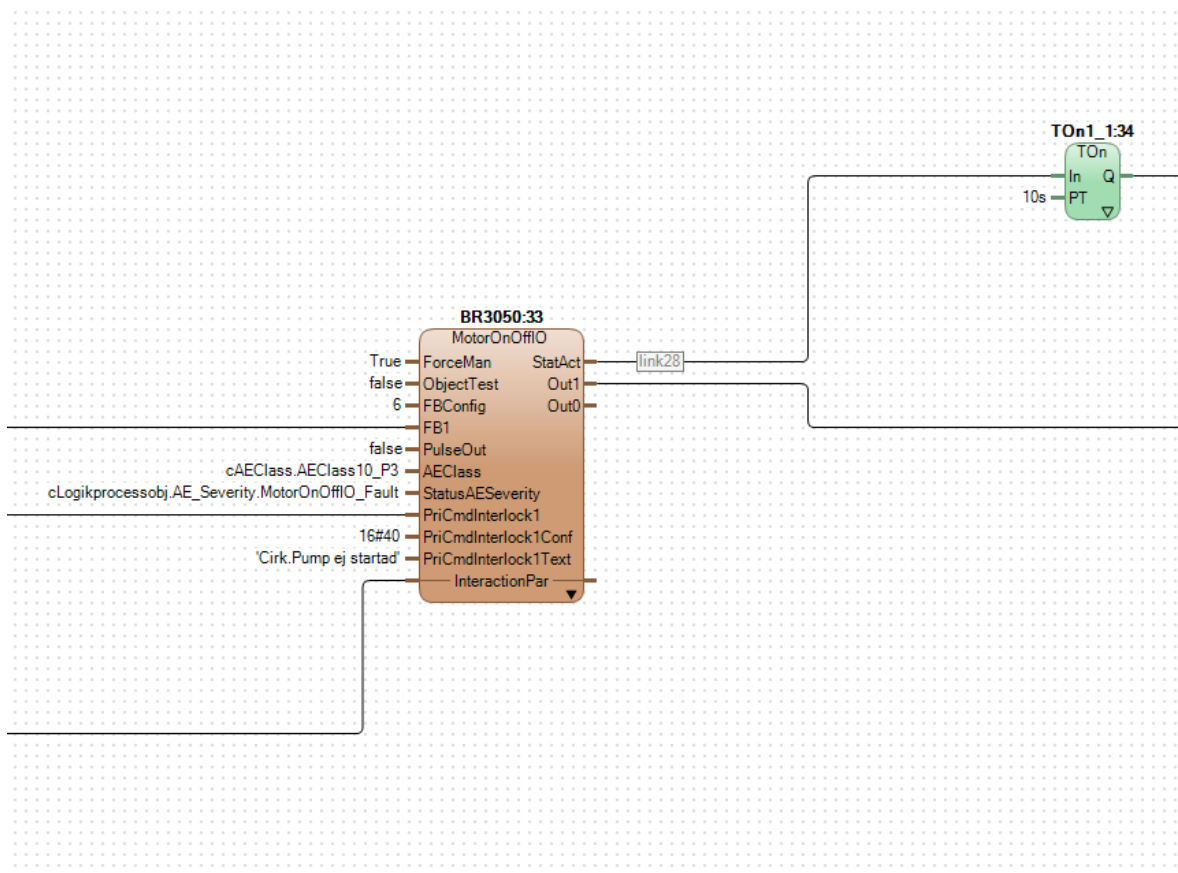
Det andra exemplet handlar om styrning av brännare *BR3050*, se figur 22.

Styrlogiken **MotorON/OFF** är ett objekt från ABB:s bibliotek som har signalnamn enligt I/O-listan *BR3050*. Det används för att starta/stoppa brännare och har några speciella funktioner för att kunna uppfylla slutkundens önskemål. Funktionen är utformad för att säkerställa att brännaren endast kan startas när samtliga villkor är uppfyllda, i synnerhet att cirkulationspumpen är i drift. Styrlogiken mottar en statussignal från cirkulationspumpen, som indikerar om pumpen är aktiv eller inte. Om cirkulationspumpen inte är i drift, visas en blå romb i faceplate (se figur 21), som beskriver att den är tvångsstoppad.



Figur 21: Faceplate för Brännare **BR3050** (DCAB, utvecklingsmiljö)

Dessutom används en extra input som **ObjectTest** i blocket och i andra block till exempel i cirkulationspumpen (*P3015*). Det är en testsignal som används för att aktivera simuleringsläge eller testlogik, vilket används för FAT:en.

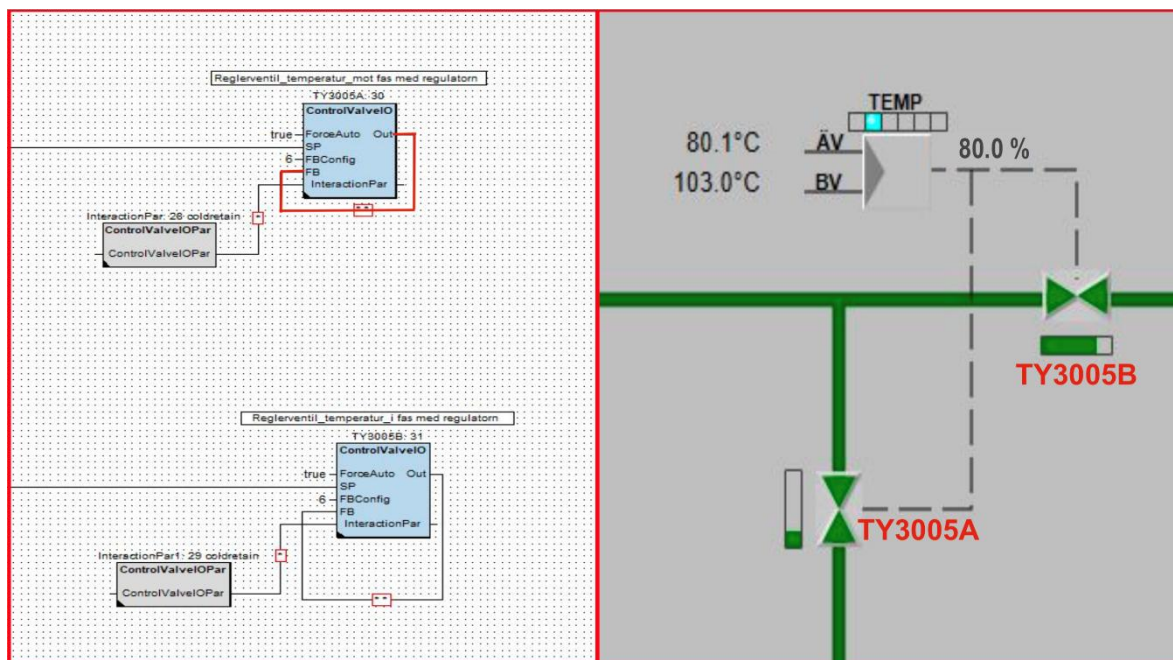


Figur 22: Motorobjektet **BR3050** för att starta/stoppa brännare som får **StartAct** från Cirkulationspumpen. (DCAB, utvecklingsmiljö)

Det tredje exemplet beskriver hur ventilerna har programmerats.

Vid programmeringen av ventilerna noterades att dessa är av äldre modell och saknar signalåterföring om läge. På grund av detta har ventilernas utsignaler istället kopplats direkt till ingångar (FB, återföring) i styrsystemet. Lösningen innebär att ventilernas tillstånd hanteras via logik i funktionsblocket snarare än genom externa återkopplingssignaler från fältutrustningen, se figur 23 (vänster).

Detta medför att när en ventil visas som öppen i processbilden se figur 23 (höger), behöver det inte nödvändigtvis betyda att den faktiskt är öppen i fältet. Trots denna begränsning är lösningen vanligt förekommande inom industrin, särskilt i samband med integrering av äldre fältutrustning i moderna automationsplattformar. I sådana fall bedöms kompromissen mellan teknisk verklighet och systemlogik som acceptabel, så länge den tekniska personalen är medveten om begränsningen och eventuellt kompenserad med andra övervakningsrutiner vid behov.



Figur23: Temperaturreglerventilerna B (i fas med regulator) och A (i motfas med regulator). (DCAB, utvecklingsmiljö)

3.3.4 Programmering i Control Builder (ST)

För att säkerställa korrekt funktionalitet och flexibilitet i styrningen av regulatorerna av typen *ECA60*, programmerades och utvecklades en särskild **Control Module** i Control Builder med strukturerad text (ST), se figur 24. Regulatorerna är PID-regulator med egen intern styrlogik och konfiguration, den överordnade styrlogiken i 800xA programmerades i strukturerad text för att lättare kunna styra *ECA60*-regulatorerna utifrån operatörernas behov. Denna logik bestämmer hur regulatorerna ska fungera när det växlas mellan automatiskt och manuellt läge, och hur in- och utsignaler ska behandlas beroende på det aktuella läget.

Vid uppstart säkerställs att systemet börjar i ett stabilt och definierat tillstånd genom att initiera signalvärden och driftlägen, programmet innehåller även en logik som utvärderar avvikelser mellan processvärdet och börvärdet.

Beroende på vilket driftläge som är aktiverat styrs utsignalen till regulatorn via två olika vägar:

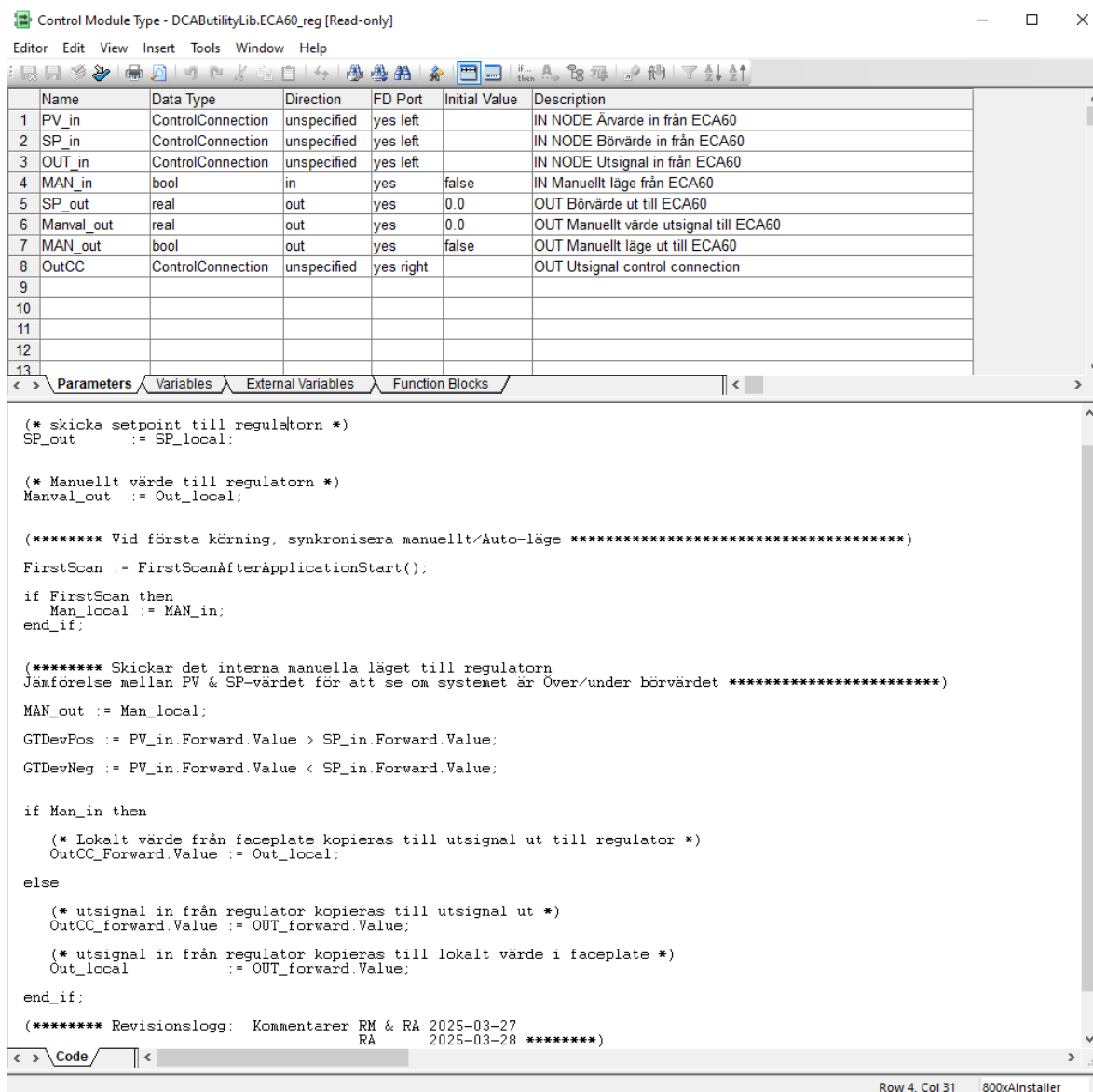
I manuellt läge används ett lokalt utsignalvärde (*Out_local*), som kan sättas direkt från faceplate, och skickas till OUTCC (Output Control Connection).

I automatiskt läge kopieras utsignalen från regulatorn (*OUT_forward.Value*) vidare till OUTCC, samtidigt som den lagras i *Out_local* för senare användning i manuellt läge.

De viktigaste signalerna i regulatorn är *SP_in* (börvärdet), *PV_in* (processvärdet) och *Out_in* OUTCC. I koden hanteras dessa signaler som parametrar av typen *ControlConnection*. Det är en särskild ABB-datatype som gör det enkelt att koppla signaler mellan t.ex. faceplate, regulatorer och andra programblock i **Control Builder**. Det innehåller flera egenskaper såsom:

- **.Value** (värdet som används i beräkningen)

- **.Status** (viser om signalen är aktiv eller inte)
- **.Quality** (anger datakvalitet)
- **.Unit** (anger enheten)



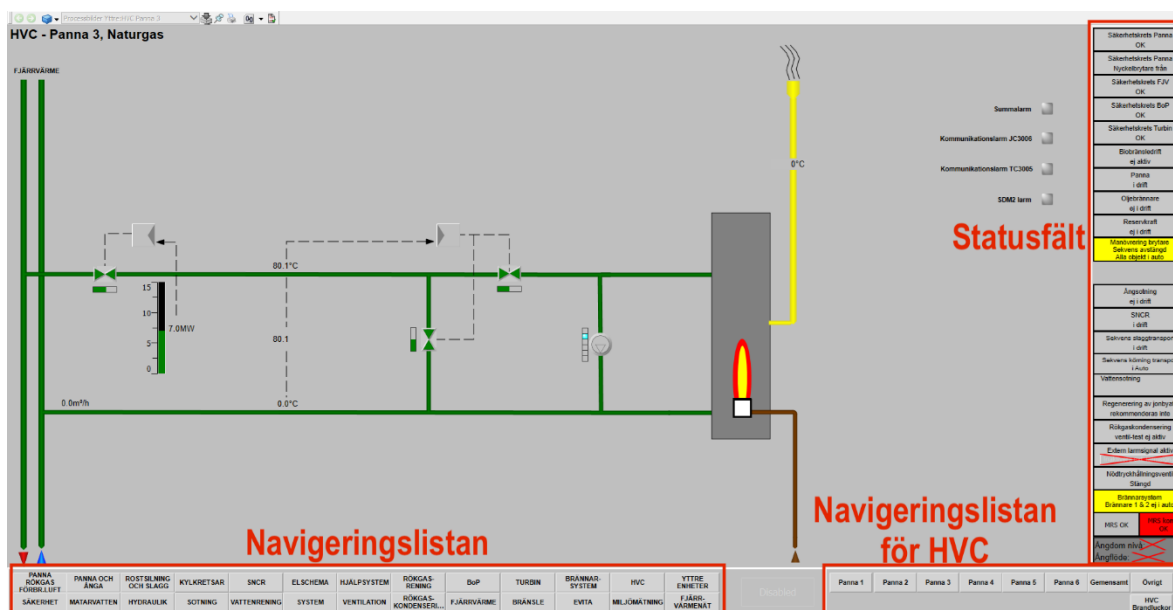
Figur 24: Control Module programmerad i Control builder för regulatorerna.
(DCAB, utvecklingsmiljö)

3.4 Operatörens gränssnitt och Faceplate

Vid utvecklingen av operatörsgränssnittet för Panna 3 i PG2 utgick projektgruppen från redan existerande lösningar inom anläggningen, specifikt den tidigare utbytta pannan (Panna 6). Detta bidrog till att effektivisera utvecklingsarbetet samt säkerställa en enhetlig och konsekvent användarupplevelse. Genom att återanvända layouten från Panna 6 kunde samma grafik, storlek, statusfält, fasta navigeringslista och navigeringsfält för HVC-anläggningen användas som standard, se figur 25. Det underlättade grafikarbetet mycket eftersom många saker var desamma, såsom färg på processbilden, rör och panna osv.

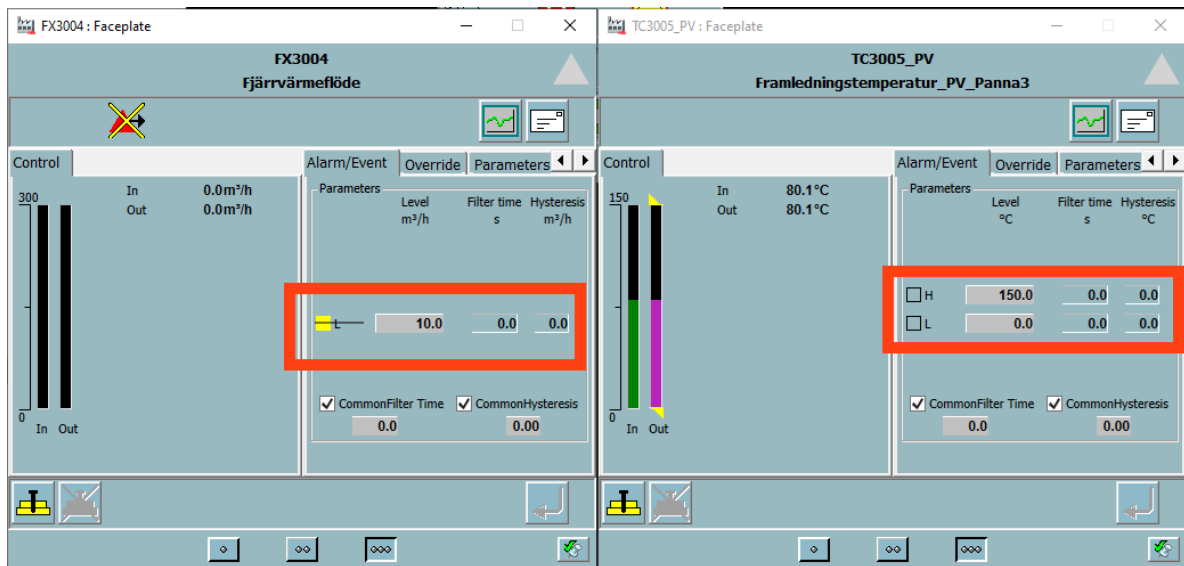
Allt som tillhörde Panna 6 rensades och istället lades relevanta komponenter in för Panna 3. Arbetet utgick från ett PI-schema (figur 12) för Panna 3, där samtliga instrument i schemat hade dubbelkontrollerats i fält för att säkerställa placering runt pannan. PI-schemat låg till grund för hur processbilden Panna 3 utformades, där alla instrument placerades i enlighet med detsamma.

Utöver detta anpassades operatörsgränssnittet ytterligare efter processoperatörens och slutkundens förbättringssynpunkter vid FAT:en. För en detaljerad beskrivning av vilka ändringar som genomfördes, hänvisas till avsnitt 3.7.



Figur 25: Processbild för Panna 3 under IFAT och FAT.
(DCAB, utvecklingsmiljö)

Alla processobjekt som tillkom, såsom givare, pumpar, ventiler och regulatorer integrerades med tillhörande signaler och Faceplates. I det kopierades även faceplaterna för det är samma över hela anläggningen, men alla modifierades och anpassades för att passa de specifikationer som tillhör just det instrumentet och panna 3. Såsom att det programmerades hantering av larmnivåer, där operatörerna själva kan ange värden på både hög och låg nivå direkt i Faceplaten, (se figur 26). För detta ger ökad flexibilitet i driftövervakningen och möjliggör en enkel anpassning av larmgränser utifrån aktuella driftförhållanden.

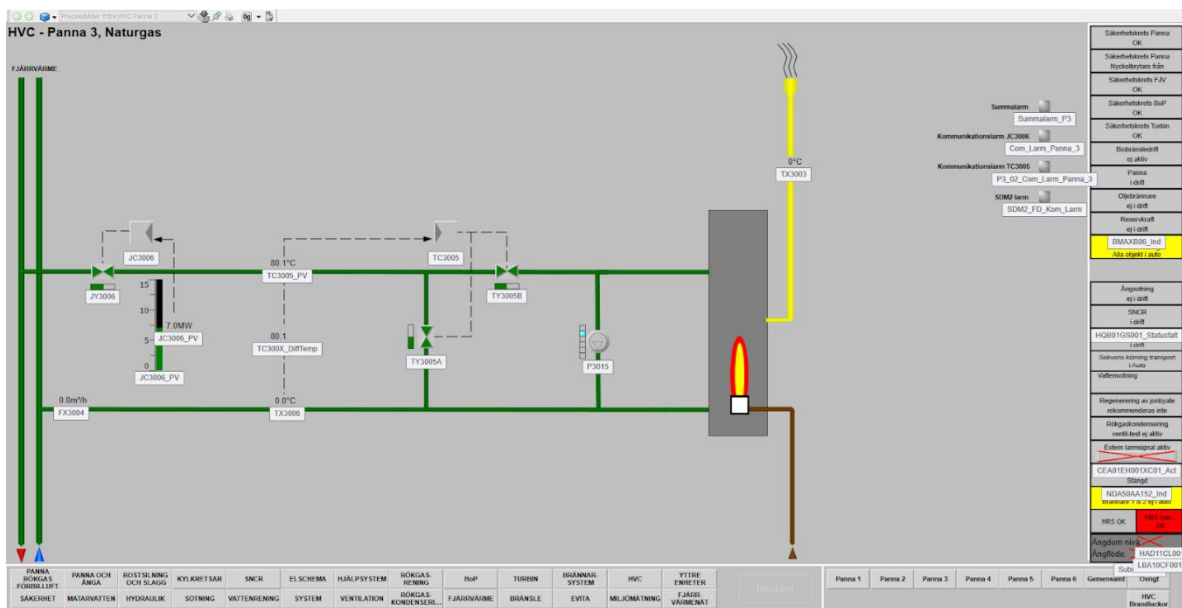


Figur 26: Faceplate på Fjärrvärmeflöde och Framledningstemperatur.
(DCAB, utvecklingsmiljö)

3.5 IFAT och FAT

Två huvudsakliga tester genomfördes under examensarbete för att verifiera funktionalitet, systemstruktur och användargränssnitt innan driftsättningen kunde påbörjas: **IFAT** (Internal Factory Acceptance Test) och **FAT** (Factory Acceptance Test). Båda genomfördes på DCAB:s kontor, och i deras utvecklingsmiljö.

Under det interna testet tillsammans med uppdragsansvarig granskades systemets funktionalitet och struktur. Arbetet godkändes med vissa anmärkningar. Fokus låg på att säkerställa att gränssnittet var tydligt, att koden var dokumenterad samt att benämningar stämde överens med I/O-listan (se figur 27). Dessa ändringar åtgärdades efter testet, och banade väg för ett FAT-test med slutkunden Landskrona Energi AB.



Figur 27: Processbild med beteckningar vid IFAT och FAT.
(DCAB, utvecklingsmiljö)

En icke-protokollförd **FAT** hölls med deltagande av både en styr-och reglertekniker och en processoperatör från slutkunden. Det omfattade ett grundligt test av systemets samtliga funktioner, där särskilt fokus låg på regulatorfunktioner, processbilden och larmhantering. Även detta test godkändes med några förbättringspunkter från processoperatören, vilka ansågs nödvändiga för att optimera systemets användbarhet i verkliga drift. En sammanställning av dessa förbättringar, samt jämförelser med det tidigare Uniview-systemet belyses närmare i avsnitt **3.7**.

Under både IFAT och FAT visades den ursprungliga processbilden som låg till grund för testerna, se figur 27. Efter genomförda ändringar uppdaterades processbilden och funktioner för att motsvara slutkundens och processoperatörernas önskemål. Den nya processbilden utformades enligt operatörernas behov och innehöll både tydligare presentation av information och förbättrad funktionalitet. Dessutom ville operatören se alla värden som kommer in och går ut från regulatorn såsom är markerat i figur 34, som liknar andra regulatorer i anläggningen. Dessa ändringar resulterade i en mer användarvänlig och processanpassad grafisk bild, som sedan godkändes av slutkundens operatörer.

3.6 Utcheckning och driftsättning

Utcheckningen av systemet inleddes med att importera det utvecklade projektet från utvecklingsmiljön i DCAB till det skarpa systemet i Landskrona. Den initiala importen stötte på vissa komplikationer, framför allt på grund av namnkrock på vissa objekt. Efter åtgärd av dessa krockar kunde importen slutföras, men det upptäcktes att samtliga funktionsdiagram inte längre var allokerade och det krävdes manuell återallokering av samtliga diagram innan arbetet kunde fortsätta.

Efter den lyckade importen påbörjades integration av det äldre styrsystemet till den nya genom användning av en EDW100-modul. Denna fungerar som en protokollomvandlare mellan seriell kommunikation (RS485) och Ethernet. Kommunikationslänken etablerades, men under testfasen uppstod ett ytterligare fel: COMLI-slaven kunde inte upprätta en stabil kommunikation, vilket resulterade i återkommande kommunikationsbortfall.

Nästa steg i processen var att justera skalningen i alla **PieceWiseLinear**-block. Detta säkerställde att analoga signaler som mottogs från SDM-enheten tolkades korrekt efter konverteringen från 16-bitars heltal till flyttal.

När kommunikationen stabiliserats och signalomvandlingen verifierats, kunde den faktiska signalutcheckningen påbörjas enligt utcheckningslistan som var förebred tidigare, se figur 28. I och med att det gamla styrsystemet fortfarande var i drift, utnyttjades detta för att tvångställa signalvärden såsom flöde, rökgas-och returtemperatur, vilket underlättade valideringen av signalvägarna.

För digitala signaler utfördes ytterligare verifiering genom att granska elritningar och identifiera de specifika plintnummer där kablar var anslutna. Spänningsmätning gjordes direkt på specifika plintar enligt listan för att säkerställa att signalerna korrekt gick från SDM till fältutrustning, exempelvis signal om att starta brännare. Andra signaler såsom start av cirkulationspumpen verifierades genom så kallad “virtuell utcheckning”, där fysiken till exempel kontrollerades genom att en pumps fläkt snurrade vid aktivering av pumpen (fläkten var synlig och kopplad på samma axel som pumpen – som i sig ej var synlig).

Kort	Kanal	Name	StartAdd	IO-referens	Beskrivning
3.2 CI853	0	JC3006_PV	R50	Se Name	Fjärrvärme effekt.PV_in-P3
3.2 CI853	0	JC3006_SP			Effektregulator SP_in
3.2 CI853	0	JC3006_OUT	R50		Effektregulator utsignal (Out_in)
		JC3006_Man			signalen anger om styrningen är Auto (or-Man_in)
		JC3006_DOC1			Signalen anger om styrningen är Man (or-Man_in)
		JC3006_SP_out	R52		Out Börvärde ut till ECA60
		Manval_out			Out Manuellt värde utsignal till ECA60
		MAN_out			Out manuellt läge ut till ECA60
		OutCC			Utsignal till reglerventil(JY3006)

Figur 28: En del av utcheckningslistan som användes under utcheckning och driftsättning. Hela listan visas i Appendix.

Därefter inleddes driftsättningen i samarbete med processoperatör, driftingenjör och styr-och reglertekniker. Tillsammans kontrollerades hela systemets funktion i båda manuella och automatiska lägen innan det slutligen godkändes för drift.

3.7 Förbättringar och ändringar

I detta avsnitt sammanfattas de förbättringar som identifierades under FAT samt generella förbättringar jämfört med det tidigare SCADA-baserad Uniview-systemet (se avsnitt 2.8).

Tabellen nedan sammanställer centrala förbättringar som genomfördes under examensarbets gång och belyser skillnader mellan det gamla och det nya systemet. Förbättringsområdena sträcker sig från operatörsgränssnitt och regulatorstyrning till larmhantering, signalöversikt, energimärkning och processvisualisering.

Tabell 3: Förbättringar och skillnader jämfört med tidigare system

Förbättringsområde	Tidigare lösning Uniview	Ny lösning ABB:s 800xA med processoperatörs synpunkter
Processbildstruktur	Panna 3 och 4 visades tillsammans i en gemensam processbild	Separat, strukturerad och tydlig processbild specifikt för Panna 3
Regulatorfunktioner	Temperaturregulator endast i automatiskt läge	Möjlighet att växla mellan Auto/Man-läge för både regulatorerna
Larmantering	Inga specifika larmnivåer	LL- och HH-larmnivå implementerad för flöde, framledningstemp och returtemp. Vid kommunikationsfel läggs ett (X) över signalerna.
Signalöversikt	Begränsad synlighet	Alla signaler synliga direkt i processbilden, samt vilket mode det har (Larm, Manuel, Forced (wait for ACK), local, Action och Disabled/Inhibited)
Energihantering	Ingen visning eller återställning av total energi	Funktion för totala energivisning samt Reset-knapp
Trend/loggning	Begränsad eller ingen trendvisning	Log configuration och Trend Display implementerat för alla signaler.
Navigering	Begränsad, få genvägar	Intuitiv struktur med snabb åtkomst till regulatorer, larm och signaler.

Utöver de visuella och användarmässiga förbättringar som redovisas i tabellen ovan, genomfördes även ett antal viktiga förbättringar i styrlogiken inom ABB:s 800xA-systemet. Dessa förbättringar omfattade bland annat implementering av tydliga larmnivåer (LL/HH),

spärrar/blockeringar av signaler samt logisk koppling mellan olika funktioner. Dessa förändringar utgör förbättringar av styrlogiken inom 800xA-plattformen för att öka driftsäkerheten och tydligheten i systemet.

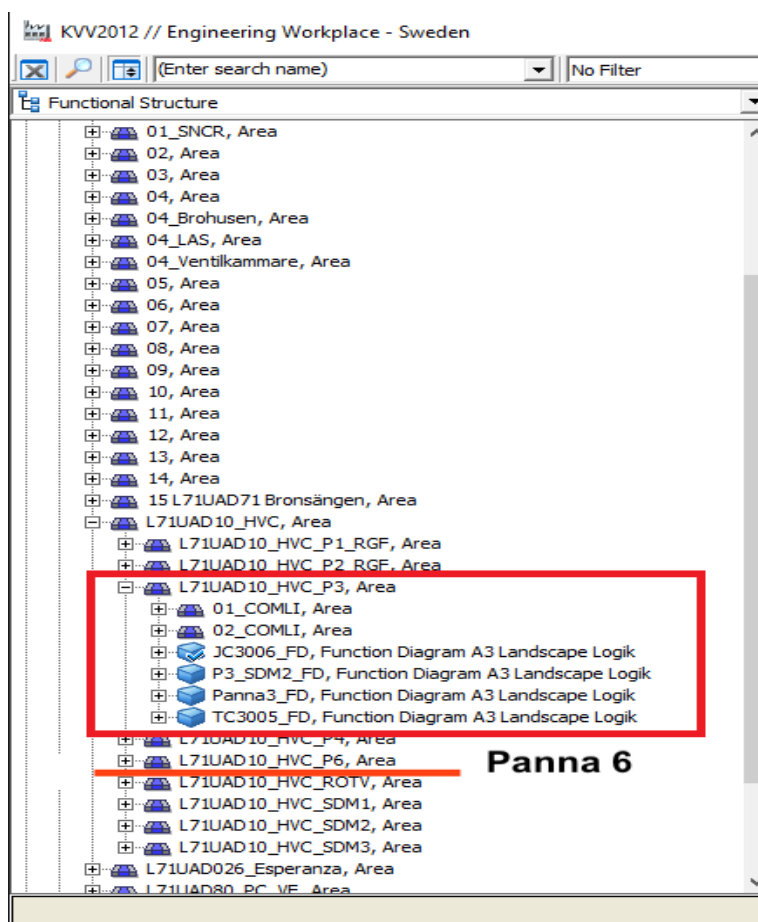
Ingen förändring skedde däremot med styrlogiken i SDM, pga. det bygger på ett äldre teknik och hårdvara och det saknas både tillgång till systemet och erfarenhet. Alla förbättringar implementerades ovanpå den befintliga logiken, med syftet att anpassa styrsystemet funktion till de standarder och behov som råder för övriga pannor och anläggningsdelar.

4 Analys

4.1 Olika val av programmeringsspråk

I början av examensarbetet diskuterades vilket programmeringsspråk som skulle användas för styrsystemet till Panna 3. Det konstaterades att valet av språk utgår från slutkundens önskemål och den standard som redan är implementerad. Eftersom tidigare pannor var programmerade med Funktionsblocksdiagram (FBD), beslutades det att även detta projekt skulle följa samma struktur. Detta för att säkerställa att slutkundens personal kan förstå, vidareutveckla systemet och felsöka det vid behov.

I figur 29 visas att programmeringen för Panna 3 har utförts i Function Designer, vilket sker inom Plant Explorer på samma sätt som tidigare pannor är programmerade såsom till exempel panna 6. Detta för att alla pannor ska vara samlade och placerade på ett gemensamt ställe när hela anläggningen är färdigkonverterad till 800xA. Figuren visar även hur samtliga signaler är grupperade utifrån signaltyp, vilket innebär att alla analoga signaler (AI/AO), digitala signaler (DI/DO), kommunikationslarm samt regulatorer är logiskt uppdelade och strukturerade i olika sektioner.



Figur 29: Function Designer, HVC_P3.
(DCAB, utvecklingsmiljö)

Utöver programmeringen som gjordes i Function Designer har även en Control Module (se avsnitt 3.3.3) skapats i Control Builder. Denna modul är specifikt gjord för regulatorerna (Effekt och Temperatur) och innehåller hela regleralgoritmen och hantering av analoga signaler samt växlingen mellan manuell och automatisk drift. Regulatorerna byggdes som Control Module för att få en standardiserad och återanvändbar struktur som kan användas för liknande regulatorer i andra pannor.

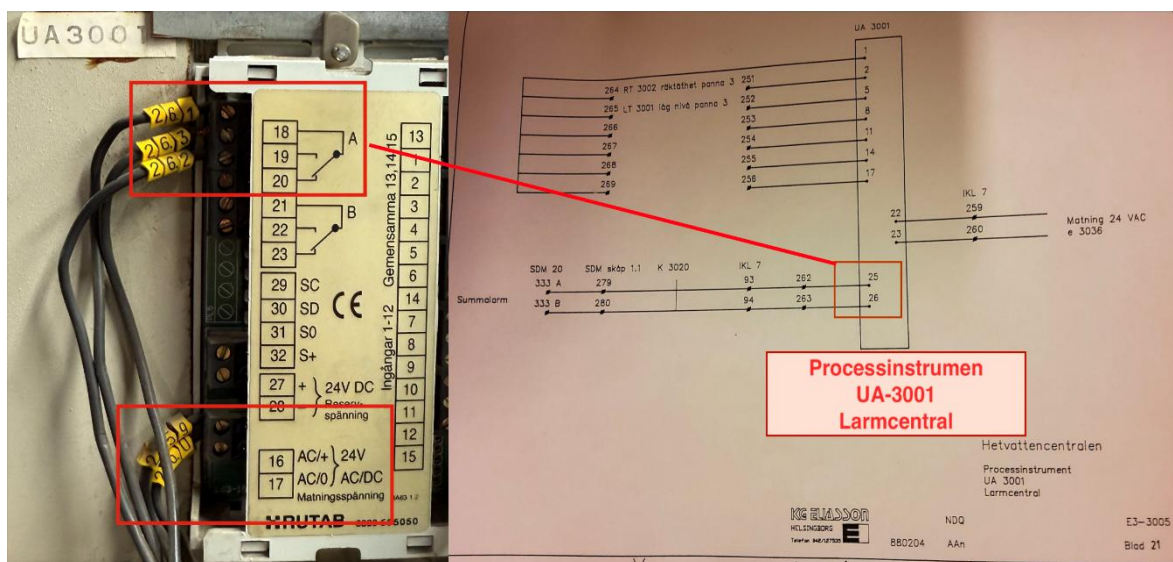
4.2 Utmaningar och lösningar

I detta avsnitt förklaras några av de utmaningar som orsakade fördröjningar i arbetet samt vilka åtgärder som vidtogs för att lösa dem.

4.2.1 Felaktig dokumentation

För det första uppmärksammades ett farligt problem, att vissa elektriska inkopplingar av komponenter i fält ej var i överensstämmelse med dokumentationen. Problemet upptäcktes under FAT-testet när larmfunktioner skulle testas. Det här problemet berodde till stor del på anläggningens föråldrade struktur och otillräckligt uppdaterad ritningsdokumentation.

Ett konkret exempel på detta syns i figur 30 från elskåpet där UA3001 representerar larmcentral i fältet, enligt figuren så är relän anslutna till plintarna 18,19 och 20. I ritningar visar att samma funktionella kopplingar i stället är anslutna till plint 25 och 26 på enheten UA3001. Detta ledde till att signalerna inte kommer till korrekt ingångar i SDM, vilket i sin tur innebär att larmet inte kommer att fungera enligt specifikation.



Figur 30: Larmcentral UA3001 och el-ritning på processinstrument UA3001

Detta problem löstes genom visuell inspektion av signaler och att anteckna de ändringar på vilka signalvägar som ledde till reläer, samt genom att elektriska mätningar gjordes mellan komponentens ändpunkt och förväntad plint.

Detta var ett av flera fall där signalkablar var felkopplade, även andra signaler hade hamnat i liknande situationer där den fysiska inkopplingen inte överensstämde med ritningar. Ytterligare tid krävdes därför för felsökning, verifiering av signalvägar och korrigering av felaktiga kopplingar i fält.

4.2.2 Synkroniseringsproblem vid Modbus-läsning

Det uppstod även ett problem med kommunikationen under signalutcheckning. Alla ReadBlock som signalen skickas till för att läsa kan inte läsa order från PLC:n. En första felsökning gjordes genom att koppla om kablar till EDW100, vilket hjälpte till att säkra att inga fysiska kommunikationsfel fanns som påverkade signalerna. Sedan det visat sig att felet låg i själva programmeringen, där Readblocket endast är aktivt under skanning som tar cirka en millisekund, och när signalen når Realblocket så har den millisekunden redan gått ut, vilket skapar synkroniseringsproblem. Normalt kollar Readblocket om den ska läsa och i detta fall hittar den ingenting att läsa.

PLC:n skickar en puls för att initiera läsning och det tar en viss tid innan den når blocket. PLC:n kan skicka en order i taget, dock är Realblocken konfigurerade att bara skanna en gång per millisekund. PLC:n har många tasks/uppgifter att köra och detta ledde till att pulsen aldrig fångades upp eller någon läsning genomfördes.

För att lösa detta användes ett *T-OFF-block* (figur 31) som gör att PLC:n's order ligger hög under en längre tid. Testning på 2s och intervallet till 5s gjordes så att Readblocket skulle hinna ta emot order, men det gav för lång fördröjning för operatörerna. En bättre lösning gjordes så att totalt responstid var cirka 4s för order och feedback fram och tillbaka från operatörens faceplate.



Figur 31: Förlängning på pulstiderna med 0.5s.
(DCAB, utvecklingsmiljö)

4.2.3 Saknad historik

En miss orsakade att historiken inte sparades pga. att Log Configuration inte var aktiverad i historikservern "Information Manager" (IM). Detta orsakade att ingen historik lagrades efter driftsättningen och av den anledningen kunde inga trendkurvor (Trend Display) användas förrän två veckor efter driftsättningen, se avsnitt 6. Trendkurvorna ansågs nödvändiga för att till exempel kunna se hur regulatorerna arbetade.

4.3 Begränsningar och stabilitetsproblem i utvecklingsverktygen

Under utvecklingsarbetet identifierades flera begränsningar och stabilitetsproblem, särskilt i **Function Designer**. En av de största begränsningarna var att felmeddelanden ofta var otydliga, även om verktyget har en funktion för att visa var felet finns, så var informationen bristfällig och programmeraren tvingades felsöka hela diagrammet manuellt. Detta var särskilt besvärligt när diagrammen innehåller många funktioner.

En annan begränsning som påträffades var att systemet enbart visade ett fel även om det fanns många samtidigt. Det felet som visades var enbart det mest allvarliga, vilket visserligen var bra men gjorde att felsökningsarbetet var tidskrävande eftersom de andra inte kunde hanteras förrän den första var löst.

Utöver dessa begränsningar inträffade även flera stabilitetsproblem i **Function Designer**. Vid vissa tillfällen kopplades länkarna mellan objekts-block bort utan tydlig anledning och vissa andra block flyttades till andra sidor i diagrammet. Detta resulterade i att logiken blev felaktig, vissa signaler "hängde i luften" utan anslutning och arbetet behövdes göras om.

I jämförelse erbjuder Control Builder en stabilare miljö för utveckling och felsökning. Vid nerladdningen visades tydliga meddelande som beskriver och pekar ut var felet finns i koden. Det visas även vilken applikation och vilket diagram felet tillhör, vilket gör felsökningen både enklare och mer effektiv.

4.4 Källkritik

Referenserna [1] och [2] kommer från energibolag och industriföretag med teknisk expertis inom sina respektive områden. Landskrona Energi är en etablerad aktör inom fjärrvärme, och informationen bedöms vara saklig. Referensen från Deterministic Control AB hänvisas till deras branschknytning och erfarenhet inom automation, vilket gör den till en trovärdig källa.

Referensen [3] har ingen formell granskning bekräftad. Den är användbar men artikeln är från 2014, vilket innebär att informationen kan vara föråldrad. Skulle ha varit bättre om den stöddes av nyare källor. Referensen [4] Autic är ett företag som säljer SCADA-system, vilket innebär att det finns möjlighet att de har ett kommersiellt syfte. Marknadsföringssyfte minskar deras tillförlitlighet på grund av risk för selektivt urval av fakta. Referensen [5] är en akademisk och teknisk referens som rekommenderas att läsas som extra material inom Automationskursen, vilket innebär att den är objektiv. Boken är tryckt 2008, en viss risk finns att viss teknik och praxis har utvecklats sedan dess.

Referenserna [6]-[11], [13] och [14] utgörs av ABB:s egna tekniska manualer och dokumentation, vilka anses vara pålitliga källor. ABB är tillverkare av de system och produkter som analyserats i rapporten, vilket stärker informationsvärdet i dessa källor. Dokumenten är i Pdf-format och publiceras via ABB:s officiella bibliotek, där de inte kan redigeras av externa parter.

5 Reflektion och kritik

Under examensarbetets gång har flera utmaningar uppstått som är viktiga att belysa, trots att arbetet i sin helhet genomfördes framgångsrikt och resulterade i ett modernt styrsystem med förbättrad funktionalitet.

I början av examensarbetet var det önskvärt av uppdragsansvarig att lämna en uppdragsbeskrivning med tydliga instruktioner om hur processbilden skulle göras, samt riktlinjer för hur programmering skulle ske, vilket inte skedde. Detta ledde till att alla krav och förväntningar behövdes förklaras i efterhand och i dialog med mentor under arbetsgången. Eftersom uppdragets omfattning var oklar från början, och funktionaliteten av det tidigare systemet (Uniview) inte är väl dokumenterat, blev det en betydande utmaning att klargöras/utforska omfattningen steg för steg.

För att få en effektiv arbetsgång i skrivandet av examensarbetet så var det nödvändigt att få en korrekt lista innan programmering påbörjades. Ett problem var dock en brist i I/O-listan som slutkunden levererade. Några signaler i listan var felaktiga i de kopior som fått av slutkunden. Detta ledde främst till ett ineffektivt tidsutnyttjande i väntan på slutgiltig och korrekt lista. Arbetet blev oorganiserat på grund av förseningen i kommunikationen med slutkunden och tvingades emellanåt pausas tillfälligt. Ibland flyttades det fram till veckan efter, ibland behövdes kontakt med mentor för att lösa problemet.

Kommunikationskedjan som finns nu ser ut såhär:

SDM2 → EDW100 → Ethernet → Switch → CI867 → 800xA server.

Problemet med denna lösning är att EDW100 är en konverterare som omvandlar Modbus RTU till Modbus TCP, och varje extra omvandlingssteg kan orsaka ökad risk för kommunikationsfel (bit fel) samt att det ställs krav på att omvandlaren är korrekt konfigurerad och underhållen. En bättre och säkrare kommunikationslösning hade varit en direkt integration av SDM2 via modern gateway med stöd för OPC UA istället för att gå genom flera steg. Detta medför att hela kommunikationsvägar göras mer robusta och framtidssäkra.

Seriekortet CI853 hade kunnat uteslutas om ECA 60 varit direkt kopplad till 800xA genom en serieportsserver, dvs att den lokala PLC:n (AC800M) hade försvunnit. Den här lösningen har några för- och nackdelar. Fördelarna är att det blir en enklare struktur, lägre systemkostnad och flexibilitet för förändringar. Dessa fördelar är bara av betydelse om processen inte är tidskänslig eller kritisk. Å andra sidan innebär lösningen nackdelen att om 800xA kraschar eller startas om, så kommer hela styrlogiken till pannan försvinna, logiken skulle vara beroende av nätverk och server, och därför finns inga möjligheter för lokal styrning.

Panna 3 är en kritisk process och därför är det rekommenderat att ha en lokal PLC (AC800M) som kör styrlogiken, medan servern hanterar larm, övervakning och operatörs-styrning.

6 Resultat

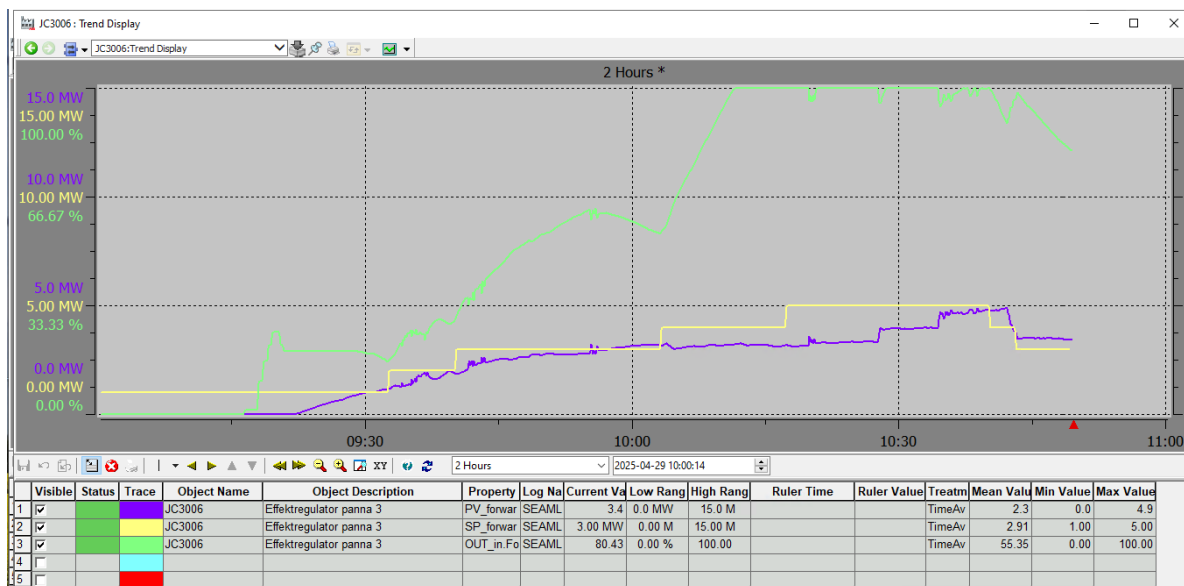
För att uppnå ett resultat som motsvarade slutkundens krav och förväntningar genomfördes arbetet i flera faser av tester och valideringar för att säkerställa att reservpanna 3 uppfyllde samtliga krav. Den första fasen utgjordes av ett internt test, **IFAT**. Syftet med detta test var att göra en övergripande granskning av systemets funktionalitet och säkerställa att samtliga delar fungerade innan tid för FAT bestämdes med slutkunden.

Under **FAT**:en, som genomfördes i samarbete med slutkund och en processoperatör, testades de praktiska funktionerna utifrån operativa krav. Alla förbättringspunkter från operatören dokumenterades och åtgärdades, se avsnitt 3.7. Resultatet från FAT visade att systemet i stort uppfyllde användarens krav men också behövde lite förändringar för att efterlikna resten av anläggningen.

Den mest kritiska fasen i examensarbetet var **driftsättningen**, där det färdigutvecklade styrsystemet implementerades i Landskronas arbetsmiljö. Resultatet visade att brännare kunde startas enligt förväntningar och systemets prestanda verifierades i praktiken.

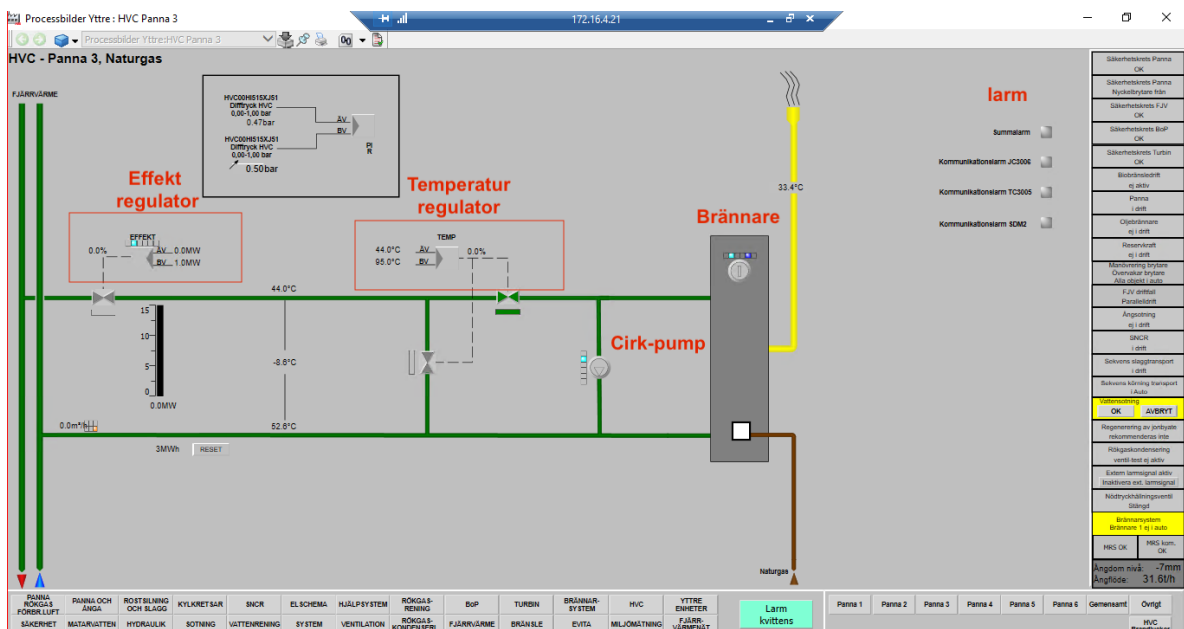
En särskilt viktig del av systemverifieringen rörde effekt-och temperaturregulatorerna, vilka testades i tidigare faser (IFAT och FAT) under simulerade förhållanden. Dessa tester visade att regulatorernas logik och signalflöde fungerade korrekt, men det var först under faktisk drift som prestandan kunde bedömas fullt ut. Under provkörningen, som skedde efter driftsättningen, genomförde processoperatörerna en stegvis ökning av börvärdet (SP) för effektreulatorn upp till 5 MW. Systemet reagerade som förväntat, och efter en stabiliseringstid nådde processvärdet (PV) det inställda värdet, se figur 32.

Resultatet av driftsättningen och provkörningen visar att hela kedjan - från kommunikationslösning (som gjordes av slutkunden) till styrlogik, gränssnitt och reglerfunktioner fungerade enligt slutkundens specifikationer. Regulatorernas prestanda i driftmiljö visar att implementeringen av ABB:s 800xA-plattform för reservpanna 3 gav ett lyckat slutresultat.



Figur 32: Trend Display på effektregulatorn vid driftsättningen.
(Landskrona arbetsmiljö)

Slutligen kan resultatet även illustreras genom en jämförelse mellan den nya processbilden i ABB:s 800xA-systemet (Figur 33), och den tidigare processbilden från Uniview (Figur 15). I det äldre systemet presenterades Panna 3 och Panna 4 tillsammans i samma processbild, utan en tydlig struktur för hur instrumenten var placerade eller hur anläggningen var uppbyggd. I det nya systemet erbjuds en mer detaljerad och överskådlig visualisering, där varje instrument och funktion är tydligt representerad. Utöver grafiska förbättringar så har den nya lösningen integrerat flera funktioner som stödjer operatörens arbete, exempelvis bättre övervakning, enklare navigering och larmhantering.



Figur 33: Processbild på Panna 3 i 800xA.
(Landskrona arbetsmiljö)

7 Slutsats

I detta avsnitt redovisas examensarbetets huvudsakliga resultat samt insamlade erfarenheter och lärdomar. Avslutningsvis ges rekommendationer för framtida förbättringar för anläggningen.

Syftet med detta examensarbete var att byta det gamla Uniview-baserade styrsystemet med ABB:s moderna 800xA-plattform för reservpanna 3. Arbetet avklarades framgångsrikt genom programmering, design av operatörsgränssnitt och i slutändan driftsättning av ett nytt styrsystem.

Vid examensarbetets början uppsattes några mål (§1.3) som i slutändan uppnåddes. Arbetsgången började med överföring av styrlogik med dess funktionalitet från det gamla systemet utan förlust av prestanda, sedan utvecklades ett nytt operatörsgränssnitt som anpassades efter användarkrav och i slutet testades och driftsattes systemet med slutkunden. Problemformuleringens frågor kan besvaras genom att analysera examensarbetets utförande.

De tekniska utmaningar som uppstod vid utbytet från Uniview till ABB:s 800xA är från början integrering av ett gammalt kommunikationsprotokoll som Modbus RTU och COMLI med ett modernt system som 800xA, samt som nämndes bristfällig dokumentation av den gamla anläggningen. Några utmaningar löstes av slutkunden innan examensarbetet påbörjades, vilket skapade brister i informationen och krävde extra undersökningar gällande möjliga kommunikationssätt. Problemet löstes genom användning av rätt kommunikations-moduler och signalhantering i 800xA. Ofullständig dokumentation hanterades genom en aktiv dialog med kontaktpersonen på Landskrona Energi, jämförelse med en annan liknande reservpannas (Panna 4) dokumentation samt med hjälp av mentorn.

Styrlogiken för panna 3 (inklusive effekttreglering, temperaturreglering, brännare, pumpar osv) som fanns- och fortfarande finns- i SDM2 har inte ändrats, däremot har systemets tillgänglighet förbättrats genom implementering i 800xA. Några förbättringar som behövde göras i 800xA var att skapa en tydligare signalstruktur, att effektivisera skalning av analoga signaler och att skapa några block (typkretsar). Detta omfattar ett *RealToRealIOQual-block* som kan visa status på signaler för att underlätta felsökning och ett annat block som skapades är ett *Motor ON/OFF-block* för att starta brännare som kan startas bara om *Cirkulationspumpen* är på, detta förlopp ökar driftsäkerheten.

Ytterligare skapades regulatorblocket *ECA60-reg* som programmerades i **Control Builder** för att reglera både temperatur och effekt. Blocket används för att hantera växling mellan manuellt och automatiskt driftläge och säkerställer att utsignalen anpassas beroende på läge. Dessutom inkluderar den nya logiken initiering vid uppstart samt utvärdering av avvikelser mellan börvärde och processvärde. Förbättringar är för att ge operatörerna flexibilitet, att öka systemets tillgänglighet samt att underlätta övervakningsprocess i kontrollrummet.

Det nya gränssnittet (figur 26) designades i **Process Graphics 2 (PG2)** för att möta operatörens krav såsom att alla in- och utgångars värde ska synas i regulatorerna på processbilden. Gränssnittet skapades så att det imiterar andra pannor som redan uppgraderats, exempelvis Panna 6. Planen var att åstadkomma enhetlighet, att underlätta arbetsflödet för operatörerna och att anpassa gränssnittet till de specifika signaler och funktioner som Panna 3 har.

Moderniseringen av Panna 3 har lett till tydliga förbättringar i operatörstillgänglighet och användargränssnitt. Utbyte av den tidigare gemensamma faceplaten med en processbild som är strukturerad och har separat faceplate för varje signal har ökat tydlighet och användarvänlighet. Signalers loggar och trender och larmhantering har förbättrats. Genom denna lösning har hela anläggningen närmat sig mot en enhetlig systemplattform med en mer säker och effektiv drift.

Sammanfattningsvis har examensarbete gett stora lärdomar inom projektstyrning, problemlösning samt systemintegration, stärkt programmeringskompetensen avseende särskilt FBD-programmeringsspråk, felsökning i ABB:s 800xA-miljö och utvecklat grafisk förmåga i PG2.

7.1 Vidare arbete

Här anges några förslag och rekommendationer för att förbättra anläggningen:

- Förnyelse av kommunikationsutrustningar, exempelvis kan Ethernet/IP-baserad kommunikation användas i anläggningen för snabb och tillförlitlig kommunikation som kan ersätta de äldre seriella protokollen.
- Komplettera historikloggning och trender för andra viktiga signaler såsom till exempel flöde, tryck, och temperatur - detta förbättrar analysmöjligheter särskilt vid felsökning av driftstörningar.
- Sist men inte minst att införa ett beteckningssystem för hela HVC. Detta kräver mycket resurser i form av ekonomi och tid men kan leda till att framtida dokumentation, underhåll och drift underlättas. Ändringen kommer att garantera att anläggningen följer industristandarder.

Att följa dessa vägledningar kan stärka anläggningens effektivitet, driftsäkerhet samt utvecklingsmöjligheter.

8 Terminologi

COMLI: Communications Link

DCS: Distributed Control System

FAT: Factory Acceptance Test

FBD: Function Block Diagram

HMI: Human Machine Interface

HVC: Hetvattencentral

IEC: International Electrotechnical Commission

IFAT: Internal Factory Acceptance Test

KKS: Kraftwerk-Kennzeichen-System

MMS: Manufacturing Message Specification

OPC: Object Linking and Embedding (OLE) for Process Control

PG2: Process Graphics 2

PLC: Programmable Logic Controller

SCADA: Supervisory Control and Data Acquisition

ST: Structured Text

9 Källförteckning

- [1] Landskrona Energi AB. Fjärrvärme, 2025. Tillgänglig: <https://landskronaenergi.se/privat/fjarrvarme/>. Hämtad: 29-01-2015.
- [2] Deterministic Control AB. Referenser, 2025. Tillgänglig: <https://www.deterministic.se/referenser/>. Hämtad: 29-01-2015.
- [3] Raza, S. & Ikram, M., 2014. *Review of Remote Terminal Unit (RTU) and Gateways for Digital Oilfield Deployments*. [online] ResearchGate. Tillgänglig: https://www.researchgate.net/publication/267403801_Review_of_Remote_Terminal_Unit_RTU_and_Gateways_for_Digital_Oilfield_deployments Hämtad: 24-04-2015.
- [4] Autic, u.å. *Vad är SCADA?* Autic Systems AB. Tillgänglig: <https://www.autic.se/vad-ar-scada/>. Hämtad 22- 03- 2025.
- [5] Zhang, P., 2008. *Industrial Control Technology: A Handbook for Engineers and Researchers*. Amsterdam: Elsevier. Tillgänglig: <https://www.sciencedirect.com/ludwig.lub.lu.se/science/article/pii/B9780815515715500074> Hämtad 3-03-2025.
- [6] ABB. 3BSE035980 -600 A, 2016. Tillgänglig: https://library.e.abb.com/public/0a5240681b35410d8e24ac55c75ee89c/3BSE035980-600_A_en_System_800xA_Control_6.0_AC_800M_Configuration.pdf Hämtad 12-03-2025.
- [7] ABB. 3BSE041880-610 A, 2019. Tillgänglig: <https://www.scribd.com/document/782986046/3BSE041880-610-A-en-Control-AC-800M-Getting-Started> Hämtad: 01-03-2025.
- [8] ABB. 3BSE038018-600. 2014. Tillgänglig: https://library.e.abb.com/public/d9ce45e2f627f75dc1257dbd0043816c/3BSE038018-600_-_en_System_800xA_6.0_System_Guide_Functional_Description.pdf Hämtad:13-03-2025
- [9] ABB. 3BSE030322-600 C. Tillgänglig: https://library.e.abb.com/public/d024aacf218c421ba78e03d1bc9421a6/3BSE030322-600_C_en_System_800xA_Operations_6.0_Operator_Workplace_Configuration.pdf Hämtad:15-03-2025
- [10] ABB. 3BDS100968-600C, 2016. Tillgänglig: https://library.e.abb.com/public/1781629a5cdf4770b8ad427c663aed29/3BDS100968-600_C_en_System_800xA_Engineering_6.0_Engineering_Studio_Function_Designer_Getting_Started.pdf Hämtad: 17-03-2025
- [11] ABB. 3BSE049230-600 B Tillgänglig: https://library.e.abb.com/public/1bfa63553a6b4640a5a521685cf3036b/3BSE049230-600_B_en_System_800xA_Engineering_6.0_Process_Graphics.pdf Hämtad: 19-03-2025

- [12] Rockford Qatar, u.å. *Westermo EDW-100 Serial to Ethernet Converter*. Tillgänglig på: <https://rockford-qatar.com/Westermo-EDW-100-Serial-to-Ethernet-Converter>. Hämtad: 25-04-2025.
- [13] ABB, u.å. *Overview of the IEC 61131 Standard – XSeries Products (DS/2101127-EN)*. [pdf] Tillgänglig: <https://search.abb.com/library/Download.aspx?DocumentID=DS%2F2101127&LanguageCode=en&DocumentPartId=&Action=Launch> Hämtad 25-04-2025.
- [14] ABB. 2001. *PLUG AND PLAY IN CONTROL LOOP DESIGN*. Tillgänglig: https://library.e.abb.com/public/324f9de3cfe84c0398e3762993e6db8e/3BSE028758_-_en_PLUG_AND_PLAY_IN_CONTROL_LOOP_DESIGN.pdf Hämtad 03-03-2025
- [15] i-SCOOP. u.å. *SCADA systems (Supervisory Control and Data Acquisition)*. Tillgänglig på: <https://www.i-scoop.eu/industry-4-0/scada-supervisory-control-data-acquisition/>. Hämtad: 01-05-2025.
- [16] Prega. (2002). *ECA Process Controllers*. Tillgänglig på: https://prega.pl/download/Pliki_do_pobrania/ECA_en.pdf. Hämtad: 05-05-2025.

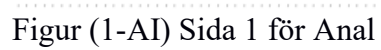
10 Appendix

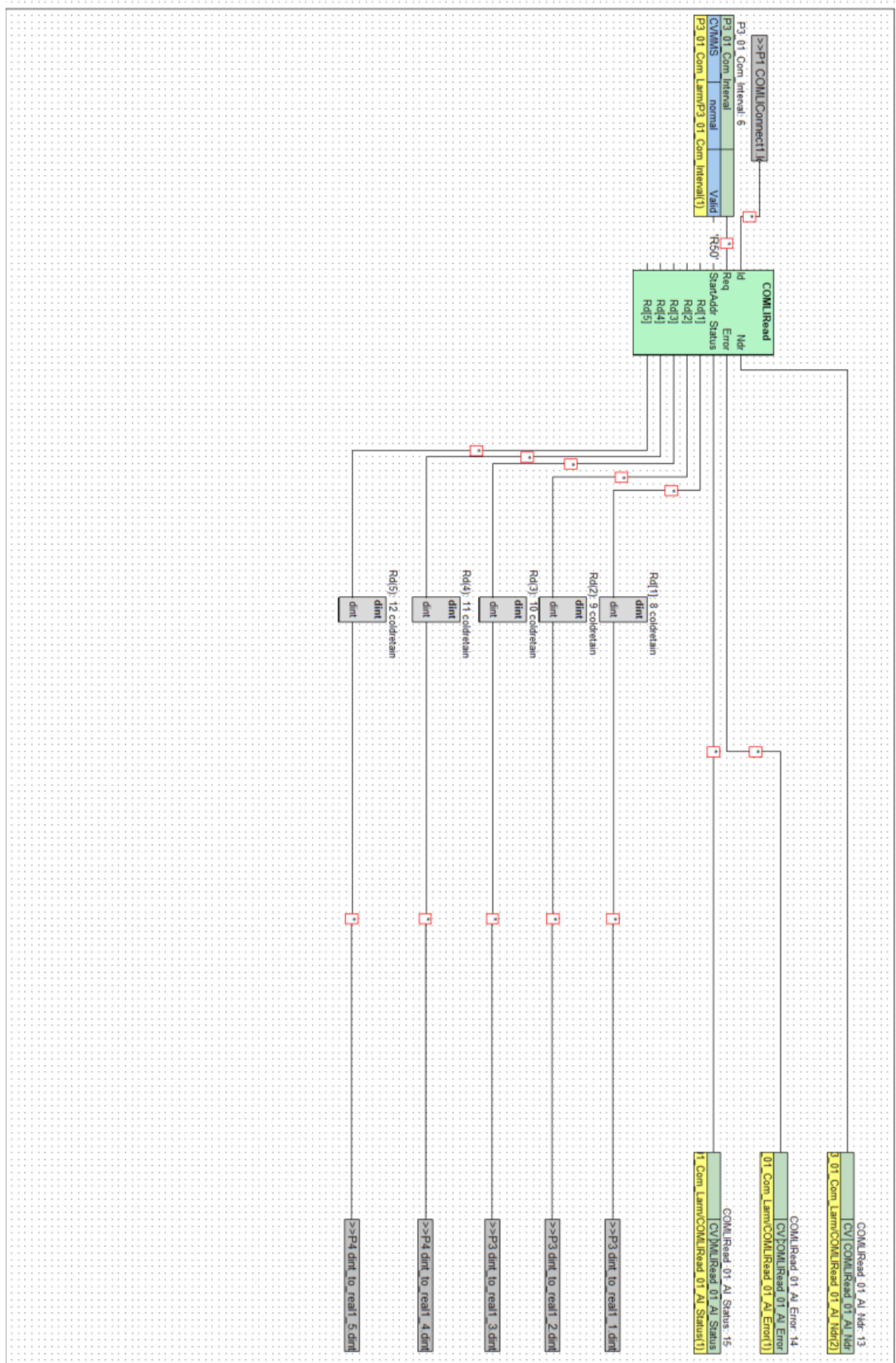
Gantt schema för examensarbetet:

https://docs.google.com/spreadsheets/d/16D47U9FSsQ_6YMV-tLY5G2-Xamn91sm9XB1B2r83oVs/edit?usp=sharing

Utckeckningslistan:

https://lunduniversityo365-my.sharepoint.com/:x:/g/personal/ra0627mu-s_lu_se/EeOH7aRz4JZEv4TOPgmvQNIBHrqT7EM_aRyIAoQOigF_CA?e=wxm





Figur (2-AI) Sida 2 för Analoga IN (AI)

[illegible]

