YRKESHÖGSKOLAN
NOVIA

Styrsystem för skogsskördare

Kevin Linqvist

Examensarbete för ingenjör (YH)-examen

Utbildningsprogrammet för el- och automationsteknik

Vasa 2020

EXAMENSARBETE

Författare: Kevin Linqvist
Utbildning och ort: El- och automationsteknik, Vasa
Inriktningsalternativ: Informationsteknik
Handledare: Roger Mäntylä

Titel: Styrsystem för skogsskördare

Datum 5.5.2020 Sidantal 28 Bilagor 1

Abstrakt

Examensarbetet utfördes åt Syketec Oy AB som främst tillverkar skogsskördaraggregat. Målet med arbetet var att utveckla ett fullt fungerande styrsystem för ny hårdvara som inskaffats.

Arbetet utgörs av en genomgång av det befintliga programmet för planering och översättning för den nya hårdvaran, samt utveckling av nya funktioner. Den nya hårdvaran har två olika skärmenheter, vilket kräver två separata användargränssnitt och ändringar i funktionaliteten.

Resultatet blev ett fungerande program som utvecklats med verktyget Guitu. Programmet används nu för alla aggregat som säljs av Syketec.

Språk: svenska Nyckelord: PLC, automation, skogsskördare

OPINNÄYTETYÖ

Tekijä:	Kevin Linqvist
Koulutus ja paikkakunta:	Sähkö- ja automaatiotekniikka, Vaasa
Suuntautumisvaihtoehto:	Tietotekniikka
Ohjaaja:	Roger Mäntylä

Nimike: Metsäharvesterin ohjausjärjestelmä

Päivämäärä 5.5.2020 Sivumäärä 28 Liitteet 1

Tiivistelmä

Opinnäytetyö on tehty Syketec Oy AB:lle, joka valmistaa pääasiassa metsäharvestereitä. Opinnäytetyön tavoitteena oli kehittää täysin toimiva ohjausjärjestelmä uudelle vasta hankitulle laitteistolle.

Työ koostuu kaikkien vanhojen ohjausjärjestelmien läpikäynnistä ja kääntämisen suunnittelusta uuteen laitteistoon sekä uusien toimintojen kehittämisestä. Uudessa laitteistossa oli kaksi näyttömoduulia, jotka vaativat kaksi eri käyttöliittymää ja muutoksia toiminnallisuuteen.

Tuloksena on toimiva ohjelma, joka on kehitetty Guitu-työkalulla. Ohjelmaa käytetään nyt kaikissa Syketecin myymissä yksiköissä.

Kieli: ruotsi Avainsanat: PLC, automaatio, metsäharvesteri

BACHELOR'S THESIS

Author:	Kevin Linqvist
Degree Programme:	Electrical Engineering and Automation
Specialization:	Information Technology
Supervisor:	Roger Mäntylä

Title: Control System for Forest Harvester

Date: May 5, 2020 Number of pages 28 Appendices 1

Abstract

The thesis was made for Syketec Oy AB which mainly produces forest harvesting machines. The goal of the thesis was to develop a fully functioning control system for the new hardware that had been purchased.

The work consists of going through the existing control system for planning and translation for the new hardware with newly developed functions. The new hardware has two display modules which require two separate user interfaces and changes to functionality.

The result was a functioning program that has been developed with the tool Guitu. The program is now used for all units sold by Syketec.

Language: Swedish Key words: PLC, automation, forest harvester

---

Innehållsförteckning

Bilagor

Bilaga 1: Kopplingsschema

Ordförklaring

PLC	Programmable Logic Controller
CPU	Central Processing Unit
I/O	Input/Output
Tidrelä	Tidsbaserad funktion, som kan fördröja start eller stopp av I/O
CAN-Buss	Controller Area Network, databuss som kan hittas överallt men främst i fordon
TCP/IP	En samling protokoll för datakommunikation i nätverk
MID/CCD/CCM	Programmerbara styrenheter av Exertus
Guitu	Programmeringsverktyg för Exertus enheter
Canto	Verktyg för att läsa av CAN-buss
FBD	Function Block Diagram, en metod för att programmera PLC
Debugging	Att hitta och lösa fel i programkod
Assembler	Låg-nivå kodspråk, det närmaste man kan komma maskinkod

1 Inledning

I det här examensarbetet ges information om PLC och programmeringen av dessa. Det beskriver utvecklingen av ett nytt och förbättrat styrsystem åt Syketec för en ny PLC, utgående från det existerande program som är i användning. Första kapitlet ger introduktion till arbetsgivaren och uppdraget.

1.1 Uppdragsgivare

Uppdragsgivaren är Syketec Oy AB i Replot, Korsholm. Företaget är grundat 2011 av Börje Fågelklo. Deras verksamhet består av design, tillverkning, montering, försäljning av skogsskördare och frontpumpar. Dessutom ges service och underhåll för dessa produkter [1]. Huvudprodukten Syketec tillverkar och säljer är stegskördaren JOBO ST75. Stegskördaren kan monteras på till exempel en traktor eller grävmaskin och går att montera på de flesta kranar. Det finns nu en mindre stegskördare, ST50. Med ett kortare steg så kan den monteras på en tre tons grävmaskin och mindre traktorkranar [2]. Den ursprungliga styrningen för JOBO ST75 var utvecklat för en Siemens LOGO! PLC.

1.2 Uppdrag

Mitt uppdrag var att byta ut PLC-systemet från Siemens LOGO! till hårdvara av Exertus, genom att programmera ett program likt det befintliga. Det gamla systemet var opålitligt då reläerna inuti modulerna var i högfrekvent bruk och slutade fungera efter en tid. Även vibrationerna av maskinen kunde orsaka löskontakt i kablarna. Det fanns inte heller mera utvecklingsmöjligheter för systemet då in- och utgångar inte gick att utöka och programmet hade uppnått gränsen för hur stort det kunde vara. Efter att ha fått en offert på dessa nya styrsystem var det tvunget att så snabbt som möjligt utveckla en fungerande version av ett styrprogram för de nya enheterna. Projektets syfte är att erbjuda användarna en likvärdig, mer utvecklad och prisvärd produkt, som även kan vidareutvecklas.

2 Teori

Teorin tar upp information om de olika delarna av hårdvaran samt vilka program som används för programmering av en PLC samt hur kommunikationen sker mellan de olika enheterna.

2.1 Programmerbar logik

Maskiner kan styras manuellt med olika spakar, knappar och brytare. Med en PLC är det möjligt att automatisera dessa funktioner, beräkningar, sekvenser och kommunicera eller indikera till användaren genom meddelanden med LED-lampor eller genom att visa det på en skärm. Med hjälp av en PLC behövs alltså inte komplicerade kretsar och kabeluppkopplingar, utan PLC-systemets mikroprocessor kopplas till alla in- och utgångar och instruerar därefter i vilka situationer allting ska reagera. Tack vare denna flexibilitet så behövs ingen omkoppling eller tillsättning av nya kablar, istället programmeras nya instruktioner för systemet [3, p. 3].

PLC:er introducerades för produktionslinjer inom bilindustrin. Den tillverkades för att ersätta hårdvara som reläer med lättare programmerbara alternativ. Det kan minska på antalet komponenter som behövs för en styrprocess. Utvecklingen av PLC-system inleddes på 60-talet. Det var nu lättare att felsöka när enheter slutade fungera eller någon komponent har givit upp. Det var således inte nödvändigt att söka på lika många ställen eller byta ut lika många komponenter. Det blev även lättare att göra ändringar i processen [4].

Ett PLC-system består av delarna CPU, minne, ingångar och utgångar. Vid programkörning läser en CPU av läget på ingångarna samt gör beslut utgående från detta och kan då ge olika beräkningar och resultat i form av att ändra läget på utgångar eller skriva ut meddelanden. Mellan programcykler kan den lagra data i minnet [3, pp. 4-8]. Med dessa funktioner i systemet kan det anpassas för olika ändamål. Om automatiseringen är tillräckligt bra så kan arbetstid sparas samt arbetskraft. När ändringar måste göras i programflödet så är det ofta möjligt att enbart göra justeringar i programmet och inte alls inskaffa eller byta ut komponenter.

Beroende på ändamålet finns det många tillverkare och varianter av PLC:er att välja mellan. Exempel på krav som kan begränsa alternativen är vilka temperaturer den kommer att utsättas för, om den ska vara i inomhus- eller utomhusmiljö, mätområdets omfattning, antalet I/O, prestanda och beräkningskapacitet. Det finns ingen begränsning till en enda enhet, utan PLC:na kan kommunicera med varandra genom diverse bussar och nätverk.

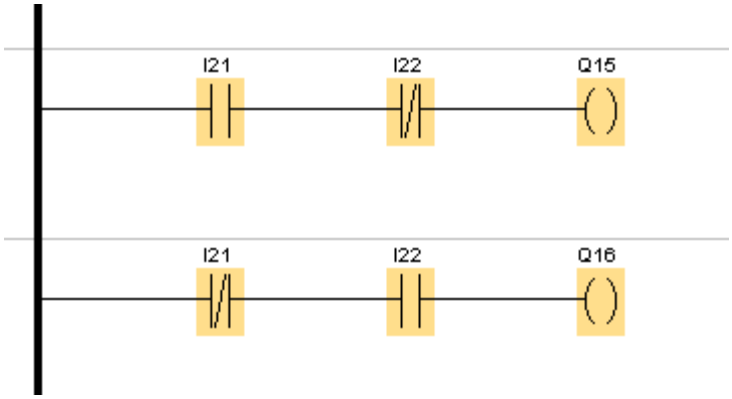
2.2 Programmering

Programmerbar logik, som namnet antyder så kan hårdvaran programmeras för anpassning av olika situationer samt arbeten. In- och utgångar kan definieras för olika ändamål. Genom att läsa av värden på sensorer kan användare meddelas samt varnas om situationer i programcykeln.

PLC-system programmeras med hjälp av en PC. Programvaran överförs därefter till enhetens minne. Nu finns enskilda varianter av programmeringsspråk men även högre nivå av språk som C. Enligt standarden IEC61131-3 så finns fem programmeringsspråk för programmering av PLC-system [5]. Dessa är:

- Function block diagram
- Ladder diagram
- Sequential function chart
- Structured text
- Instruction list

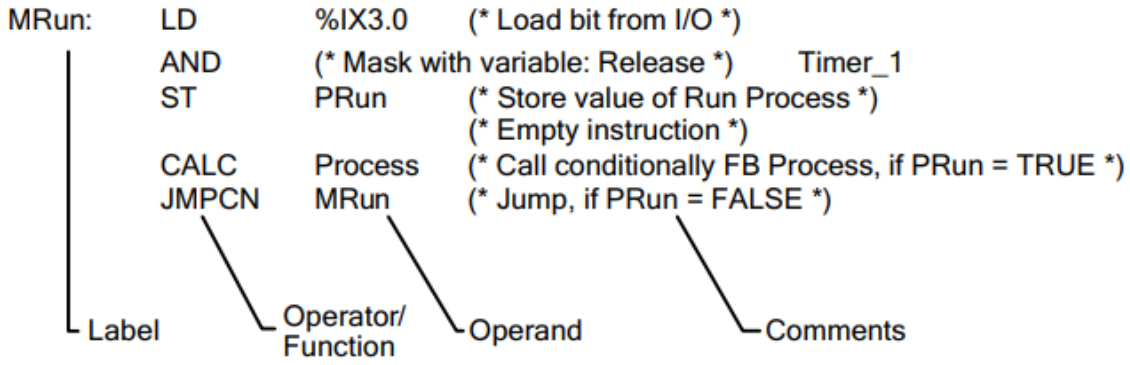
Av dessa programmeringsspråk är Function block diagram och Ladder diagram grafiska. De övriga är textbaserade språk. Sequential function chart kan även göras grafiskt. [6, p. 98] IEC61131 standarden fungerar som en riktlinje för PLC programmering. Den måste inte strikt följas. Det som krävs för ändamålet implementeras, därefter dokumenteras andelen av standarden som inkluderats. [6, p. 12]



Figur 1 Ett Ladder diagram som visar en tilt funktion med två ingångar och två utgångar.

(gjort med LOGO!Soft Comfort)

Inledningsvis programmerades PLC-system med ladderdiagram på grund av likheten i kartläggning av relälogik. Det är ännu ett av de vanligaste programmeringsalternativen.



Figur 2 Exempel på kod med Instruction List [6, p. 101]

Instruction list är textbaserat. Kodens första kolumn är en etikett. Med hjälp av den görs sektioner i koden för att navigera till utsatta positioner. I den andra kolumnen sätts alternativt Boolesk algebra, jämförelse, avläsning, lagring samt matematiska funktioner. Instruction list är lik Assemblerkod som kan anses svårt jämfört med andra

metoder. Det producerar mest kompakt kod av metoderna. Nackdelen är att den är svår att felsöka.

```
FUNCTION Xyz: MulVar
    VAR_INPUT Factor1, Factor2:INT; END_VAR
    VAR_TEMP Tmp: MulVar; END_VAR

    Tmp.Var1 := 10 * Factor1;
    Tmp.Var2 := 4.5 * INT_TO_REAL(Factor2);
    Xyz := Tmp;
END_FUNCTION
```

Call of function Xyz:

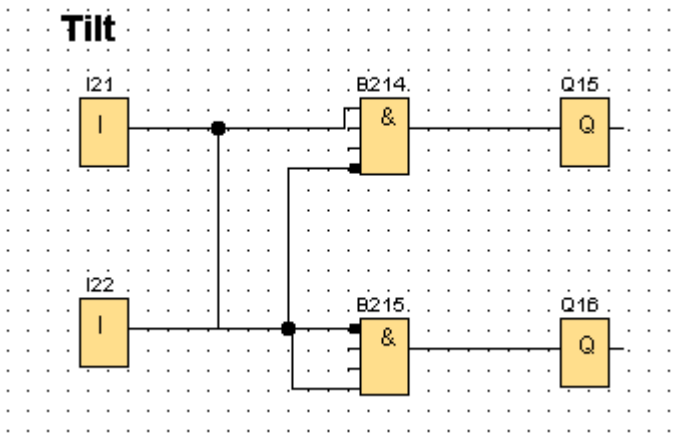
```
...
VAR i: INT; z: MulVar; END_VAR

z := Xyz (20, 3);
i := z.Var1;
...
```

Figur 3 Exempel på Structured Text med en funktion för multiplikation [6, p. 123]

Structured text är ett högre-nivå språk, kodsatser underlättas med hjälp av ”IF” eller ”SWITCH CASE”. Funktioner kan definieras, vilka kan ta emot värden när de anropas. Språket är mera läsbart för människor än Instruction list.

I sequential function charts bryts programmet i mindre delar och bygger upp ett flödesschema. Till skillnad från ett flödesschema så kan sequential function chart använda sig av förgreningar. Detta är ett tydligt och läsbart sätt att programmera.



Figur 4 Function block diagram. (gjort med LOGO! Soft Comfort)

Function block diagram är visuellt lättare att uppfatta. Istället för att forma funktioner så befinner de sig i block. Block har in- och utgångar som ansluts med hjälp av linjer. Programmet läses från vänster till höger och topp till botten [3, p. 94]. Utöver befintliga block kan egna definieras. En nackdel i metoden är risken för opassande placering av block vilket gör större program svårlästa. För detta arbete användes FBD metoden.

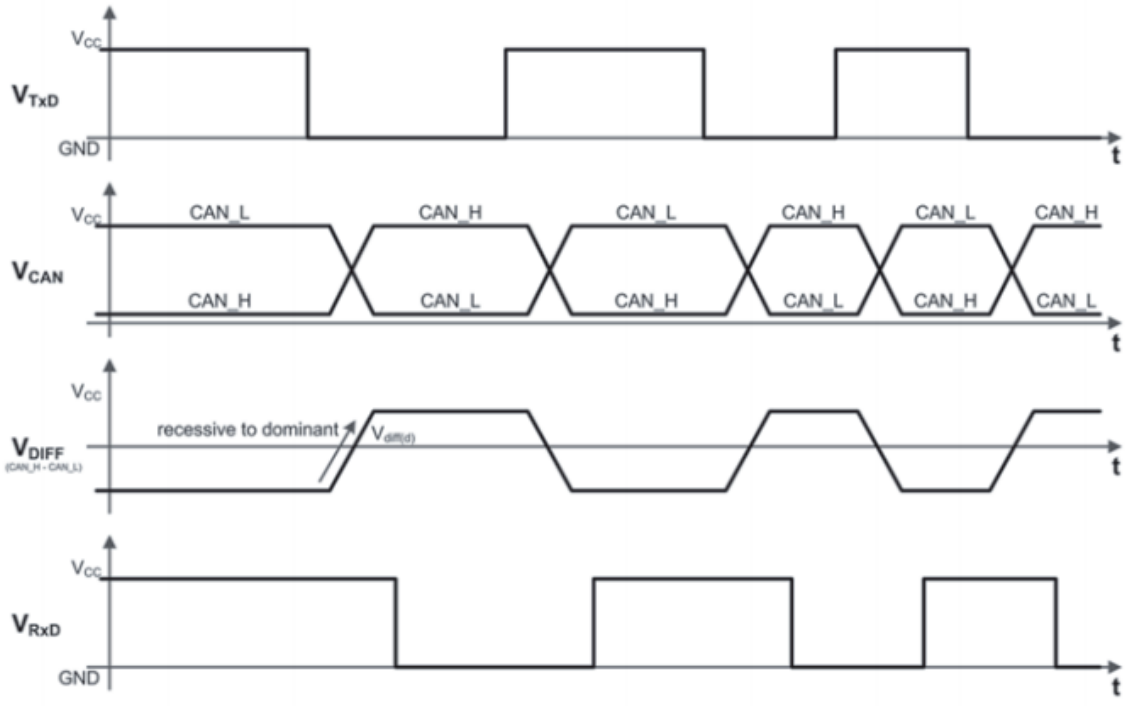
Under en programläsningscykel läser PLC-systemet av I/O, därefter exekverar programmet utgående som lästs. Detta resulterar i att utgångarna uppdateras [3, p. 80].

2.3 CAN-buss

Följande stycke summerar CAN-bussen som det beskrivs av Lawrenz [7, pp. 1-15]

I början av 80-talet hade elektroniken blivit mer komplex inom bilautomationen. När alla system såsom speglar, luftkonditionering, lås och motorrelaterade uppkopplingar gjorts så är den elektriska installationen svår. En lösning söktes för en kontroll som kan hantera kommunikationen samt minska antalet kablar. Ett protokoll som utvecklades var CAN-bussen.

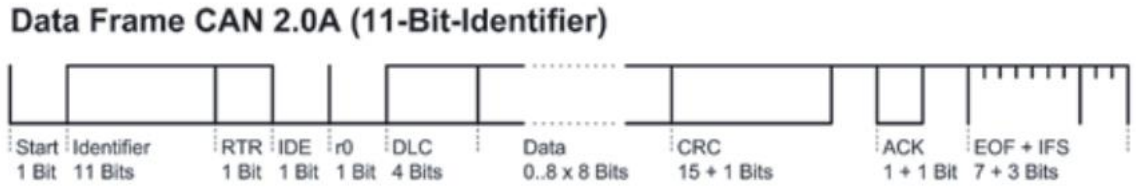
CAN-bussen utvecklades 1983 av Bosch och blev en standard 1993. Ursprungligen var den utvecklad för bilar men den finns nu överallt i transportmedel samt automation. Första implementationen var i en Mercedes S-Class 1992. CAN-bussen består av ett tvinnat kabelpar som kopplas till noderna i nätet. Noderna skickar och mottar meddelanden samtidigt. Bussen är därför snabb och kan användas till system där snabb handlingstid krävs som i motorstyrningar samt krocksystem. Ska större mängder data överföras längre distanser där snabbheten inte gör lika stor skillnad så används TCP/IP. Den maximala kabellängden för CAN-bussen är 40 meter.



Figur 5 Typisk CAN signal [7, p. 147].

Bussen fungerar med differentialsignalering. Signalen utgörs av spänning som ökar på ena kabeln och faller på den andra med motsvarande mängd. Detta gör att det är säkrare mot störningar. Det har fem olika typer av felhantering för att kunna hitta så många fel som möjligt i kommunikationen, dessa är: Bit-, Stuff-, CRC-, Form- och Acknowledgement-Error. När en nod på bussen upptäcker ett fel så reses en flagga för att ett fel har uppstått så noderna kan kassera meddelandet.

Paket sänds till alla enheter samtidigt. Huvuddelarna i ett CAN-buss paket är följande. Identifierare som innehåller information om noden ska utföra eller behandla meddelandet. I ett paket så är också en del data. Det finns fyra olika meddelandetyper: Datameddelande, meddelandebegäran, felmeddelande, kömeddelande.



Figur 6 Innehållet i ett CAN meddelande [7, p. 5].

Ett grundläggande meddelande kan ses i figuren ovan. Först är Start of frame som anger att meddelandet börjar. Identifier anger meddelandets prioritet. Remote transmission request avgör om det är data eller begäran efter data. Identifier Extension Flag definierar om meddelandet är 11-bitar eller förlängt till 29-bitar. Data length code beskriver meddelandets längd, vilket sedan följs upp av själva meddelandet. Cyclic redundancy check markerar fel, Acknowledge informerar om meddelandet nått mottagarna. End of frame avslutar meddelandet.

Utöver CAN-bussen finns Modbuss och Profibuss. Modbussen är det äldsta protokollet för PLC-system och är lika simpelt att sätta upp som CAN-bussen. Modbussen är långsammare då huvudenheten bara sänder ett meddelande åt gången. Profibussen kan skicka större meddelanden mellan enheterna med hjälp av en större bandbredd. Detta gör huvudenheten dyrare. Det lämpar sig bättre över längre distanser där det inte krävs att meddelanden skickas och tas emot på direkten.

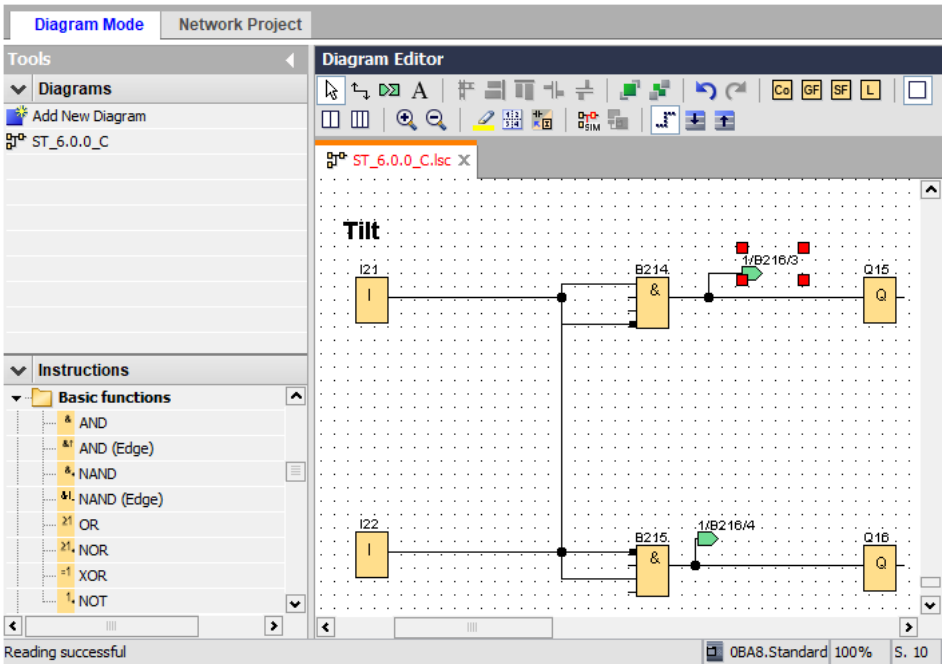
Alla dessa protokoll har en Ethernet-variant för mer komplexa system. Detta innebär högre kostnader men ger möjligheten för större nätverk med högre bandbredd.

2.4 Program och verktyg

Detta kapitel behandlar alla de program och verktyg som använts för att utföra examensarbetet.

2.4.1 LOGO! Soft Comfort

Programmering för Siemens PLC utförs med programmet LOGO! Soft Comfort. Programmeringsmiljön fungerar på Windows, Linux och Mac OS X. Det följer IEC61131-3 standarden för PLC. Av programmeringsspråken som standaren definierar, så stöds bara två av de grafiska alternativen, Ladder Diagram och Function Block Diagram.

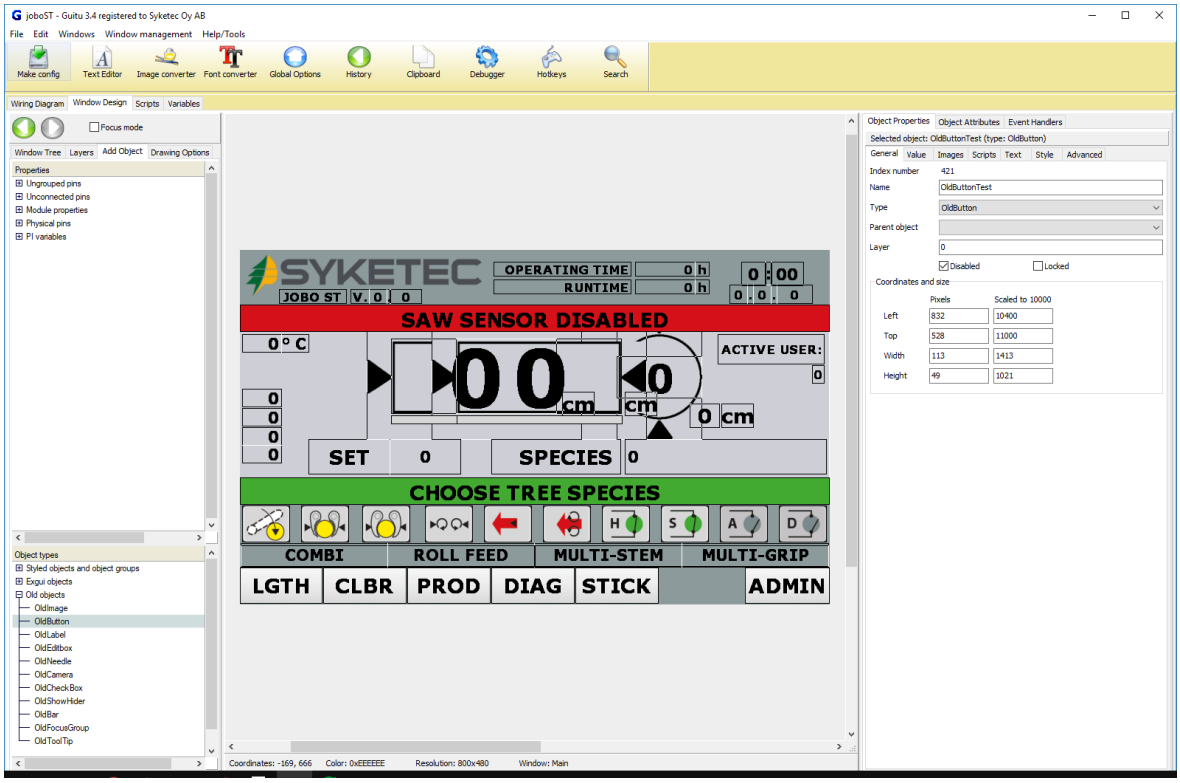


Figur 7 Exempel på en enkel funktion i LOGO! Soft Comfort.

Skördaraggregatets program är skapat med FBD metoden. Programmeringen består av olika funktionsblock som ansluts till varandra, början på en linje och slutet måste vara av samma typ. Detta bildar ett flöde i programmet som går från vänster till höger. Basen för funktionsblock är boolesk algebra vilket har block som AND, OR, NOT, NAND, NOR, XOR. Utöver dessa finns också färdigdefinierade block för att underlätta programmeringen som matematiska funktioner och tidsreläer [8].

2.4.2 Guitu

Guitu är ett program av Exertus. Det är avsett för deras hårdvara för att både designa grafiska användargränssnitt och programmera funktionaliteten och kommunikationen för alla deras moduler.



Figur 8 Fönsterdesignvy ur Guitu.

Programmeringen i Guitu är liknande till LOGO! Soft Comfort men följer inte strikt IEC61131-3 standarden då det endast stöder metoden för funktionsblock. Programmeringsmiljön fungerar endast på Windows vilket kan vara en nackdel. I botten på den grafiska programmeringen så är det språket C som används så utöver blocken kan satserna ”if”, ”switch” och ”while” användas. Inledningsvis var programmet designat för att skapa grafiska användargränssnitt. Det finns moduler med pekskärmar, men det går också att bygga menyer som navigeras med hjälp av piltangenterna som är inbyggda i skärmmodulen. Det tillåter också att sätta till flera språk för menyerna.

Guitu tillåter användargränssnittdesign med insättning av olika objekt. Dessa består av knappar, etiketter, krysslådor, ställbara lådor, mätare och bilder. Objekten anpassas genom att bindas till variabler eller ingångar för att visa värden, köra skript samt gå till andra skärmvyer. Guitu hanterar från grunden allting relaterat till CAN-bussen,

noderna tillsätts bara på bussen. Egna funktioner kan göras för bussen, som att verifiera om alla noder är anträffbara och därpå meddela användaren.

Verktyg finns för att mata in texter i programmet. Det går att konfigurera fonter och typsnitt. Flera språk kan definieras. En tabell ges med alla textrader som finns i programmet. Tabellen kan matas ut till Excel-fil för översättning till ett annat språk. Filen kan därpå matas in tillbaka för att inkludera det nya språket. Det finns även verktyg för import av bilder i korrekt färg samt format.

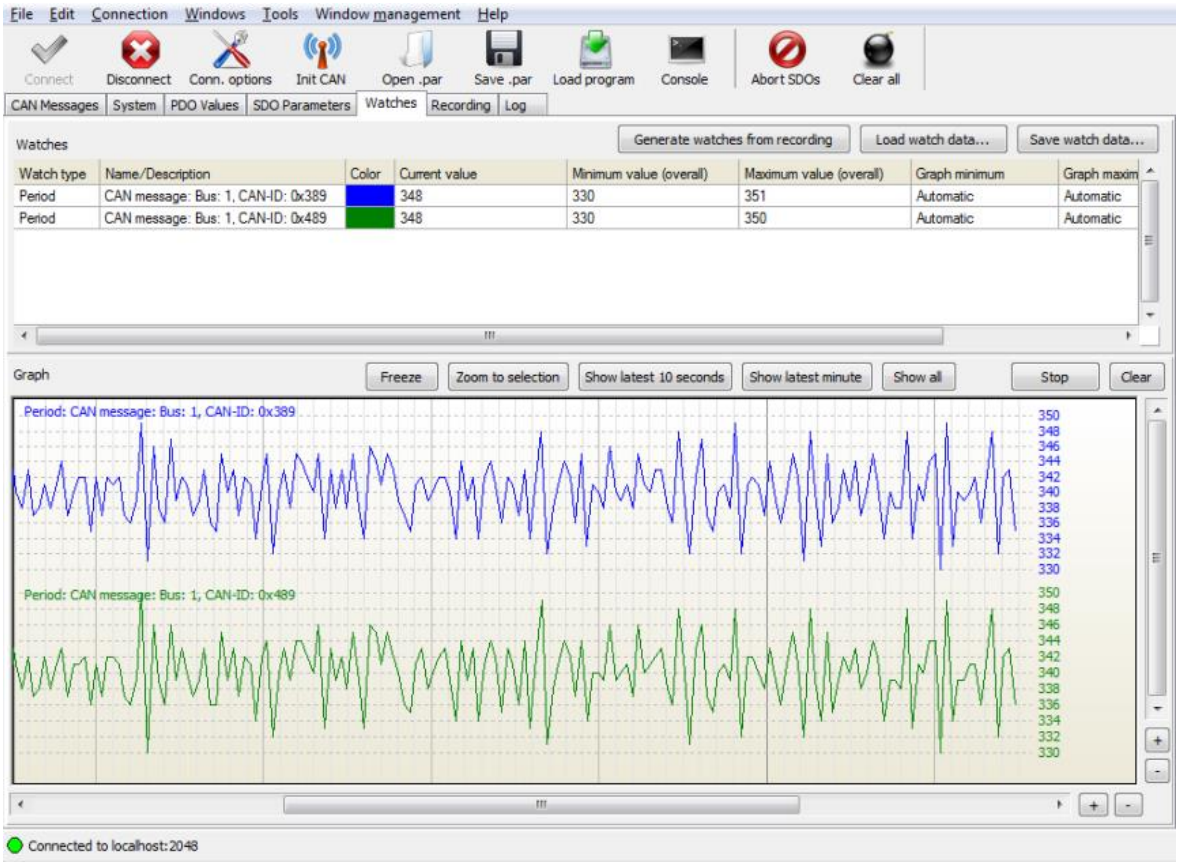
En meny finns för definiering av variabler samt deras typ. Dessa kan användas i funktioner eller så kallade scripts. Till skillnad från LOGO! Soft Comfort där programmet förekommer främst på bara en yta så delar Guitu upp dessa funktioner som ett mera traditionellt skrivet programmeringsspråk, vilket ger läsligare och lättare organiserad kod. Funktioner kan ta emot variabler och därefter ger den ett resultat från det som utförts.

Inbyggda funktioner för att felsöka över CAN-bussen tillåter att sätta in brytpunkter i programkoden för att stoppa och läsa in olika värden, alternativt stegvis gå igenom programmet [9].

2.4.3 Canto

Canto är ett monitoreringsprogram av Exertus för CAN-bussen. Med en CAN till USB adapter kopplas en dator med Windows in för monitorering av CAN-bussen. Specifikt används CANopen standarden. Programmet är avsett för Exertus moduler. Med verktyget kan enheternas alla parametrar avläsas.

En programkörning kan spelas in och därefter kan den noggrannare analyseras om den agerar avvikande. Nodernas identifieringsnummer kan omdefinieras ifall flera av samma modul existerar på CAN-bussen. Då programmet i Guitu kompileras till en config fil så överförs programvaran till modulerna med Canto.



Figur 9 Canto visar signaler som läses av från CAN-buss.

Canto underlättar service på maskiner som fungerar abnormt. En graf kan fås på alla ingångar och korrelerande utgångar för att säkerställa att en signal skickas samt att programflödet sker i rätt ordning [10].

3 Projektets genomförande

Detta kapitel beskriver aggregatets utseende och funktioner, samt utförandet av programmeringen för styrsystemet. Med en kort tidsplan på några månader så valdes det att översätta LOGO programmet så det fungerar på MID- och CCD-enheterna. Genom att först analysera LOGO programmets funktionalitet ansågs det genomförbart. Ett testsystem behövs för verifiering av att allt potentiellt fungerar innan testning mot ett riktigt aggregat. Utöver detta så fanns inget användargränssnitt för det befintliga systemet utöver förvald längd och utförd matningslängd. Det krävdes att ett användargränssnitt skapas från grunden. Två olika gränssnitt behövdes för skärmenheterna på grund av olika skärmstorlekar och prestanda.

Det fanns krav för styrsystemet som valdes. Det måste stöda 12V och 24V då maskiner kör alternativt med dessa som huvudspänningar. PLC-systemet måste vara tåligt mot temperaturer i kalla vintrar samt regn, damm och fukt om maskinen står utomhus över längre perioder. I aggregatet behövs en enhet för att kommunicera med buss och inte enskilda kablar till varje ventil och givare. Detta minskar både tillverkningstid och felsökningstid. Det ger även mera in- och utgångar än vad LOGO erbjuder, vilket bidrar till mera utvecklingsmöjligheter. Programmeringsverktyget för PLC-systemet ska likna Siemens funktionsblock i LOGO! Soft Comfort så kunskaperna i det befintliga systemet kan utnyttjas. Användarupplevelsen av styrsystemet skall fungera likadant som LOGO-systemet. Användarna måste ges möjligheten att ersätta LOGO-systemet med det nyutvecklade.

3.1 Aggregatet

Stegskördaraggregatet som arbetet behandlar är JOBO ST75. Skördaraggregatets mekaniska delar består av klor, knivar, såg och potentiellt rullor för rullmatning och en tiltfunktion. I bommen finns det en cylinder för stegutskjutet. Den rekommenderade maximala diametern på träd som aggregatet kan ta är 30 cm. Det finns även en mindre variant passar på en traktors mindre skogskran eller en tre tons grävmaskin.

Figur 10 Undersidan av stegskördaraggregatet JOBO ST75.

Aggregatet har tre till fem givare beroende på konfiguration. Vid stegmatning så känner en givare av skruvar för att mäta längder på bommen, för rullmatningen utvecklades rullmätning som ger noggrannare resultat. En givare finns som känner av att sågen är i startposition så att inte automatmatningen startas och bryter av sågsvärdet. När steget är i ett ändläge så känner en givare av positionen. Slutligen mäter en diametergivare trädets diameter med vilken matningen kan avbrytas för att såga eller byta längd när stammens diameter minskat. Aggregatet har fyra till sex ventiler på ventilbordet.

Till aggregatet löper en kabel på 18-poler som kopplas till varje ventil. Detta byts nu till en mindre kabel tack vare busskommunikationen till enheten i aggregatet. CAN-bussen var inte ett val i sig, utan det är vad som Exertus hårdvara använder.

3.2 Hårdvara

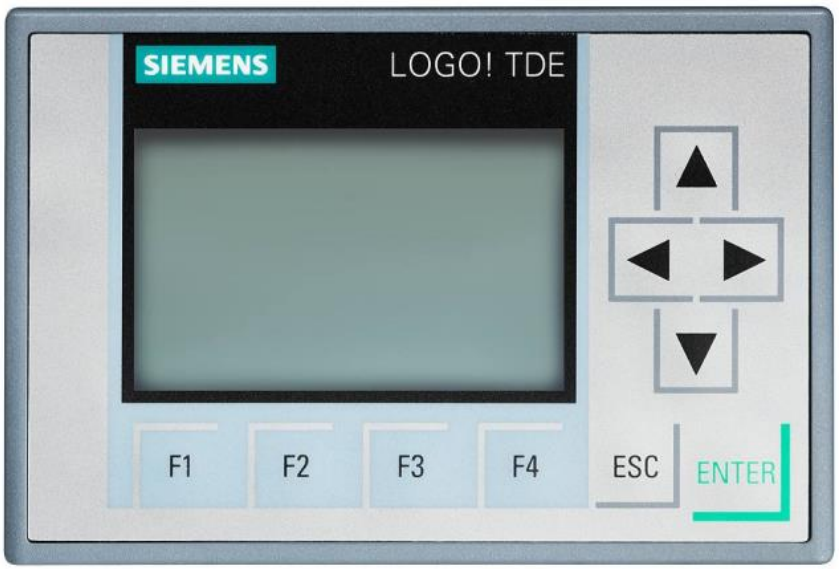
Detta kapitel beskriver det befintliga systemet av Siemens samt de motsvarande enheter av Exertus som det byts till.

3.2.1 Siemens LOGO!

Siemens LOGO! Hade valts på grund av lågt pris och användarvänlighet både för programmering och användaren i slutsyftet. Det lämpar sig bättre för mindre projekt med dess begränsningar, främst antalet in- och utgångar. Det har begränsningar på blockmängden i ett program innan minnet tar slut. Det är inte heller lämpat för tunga beräkningar, därför var inte mätningen så exakt när det gjordes mekaniskt genom att med givare känna av skruvar längs med bommen för att veta var den befinner sig.

Figur 11 Siemens LOGO! Logikenhet [14].

Skördaraggregatets styrsystem består av en LOGO! huvudenhet med två till tre expansionsmoduler beroende på maskinkonfiguration. En TDE bildenhet används för att välja automatikprogram, längd för automatiken, avläsning på nuvarande längder samt ge meddelanden åt användaren. Det har TCP/IP-kommunikation mellan enheterna som även används vid programöverförning och uppdatering. Styrsystemet har två knappsatser samt skärmenhetens knappar. Senare styrsystem hade dedikerade knappar monterade för val av längd. Knappsatsernas knappar är för såg, starta automatik, framåt- och bakåtmatning samt öppna och stäng grip, knivar och aggregat. Knappar utökas ifall maskinen har ventiler för tilt- och rotatorfunktioner.



Figur 12 Bildenheten TDE som monterades i hytten på maskinen [13].

Styrsystemet rymdes inte i aggregatet och det tål inte heller köld och fukt. Därför användes en 18-polig kabel till uppkopplingen. Denna kabel lindades in med resten av slangarna och den var utsatt för att skadas. Kabeln är mycket tidskrävande att tillverka för hand och kontakterna dessutom väldigt dyra.

3.2.2 Exertus styrsystem

I LOGO! Soft Comfort programmeringsmiljön så användes FBD. Efter granskning av några styrsystem så valdes Exertus hårdvara, eftersom det efterliknade vad Siemens använde. Systemet har bättre specifikationer och formfaktor än LOGO. Det har använts för liknande maskiner i samma bransch.

Figur 13 Skärmenheten CCD1200S [12].

För vårt bruk rekommenderades CCM1200S som modul för montering i aggregatet för att styra ventilerna och känna av givarna. Skärmenheten CCD1200S kompletterar med montering i hytten på maskinen. För avancerade användare så valdes dessutom en större skärmenhet, MID070S, med mera funktionalitet som alternativ till CCD-enheten. MID-enheten kör embedded Linux, med detta operativsystem är den närmare en dator med högre prestanda än CCD-enheten. Kommunikationen mellan dessa enheter sker med CAN-buss. Programöverföringen går till CCD-enheten över CAN-bussen. MID-enheten uppdateras med ett USB-minne. Detta innebär att användarna själva kan uppdatera enheten.

3.3 LOGO! program

LOGO! Programmet har fyra lägen för körning av automatisk matning. Stegmatning, Rullmatning, Multi-stem för att ta grepp om flera små träd, och Multi-grip om extra grip eller rullor monterats för bättre grepp om större stammar. Stegmatning kan även kombinera matning för att tillfälligt använda rullor i situationer där det lämpar sig.

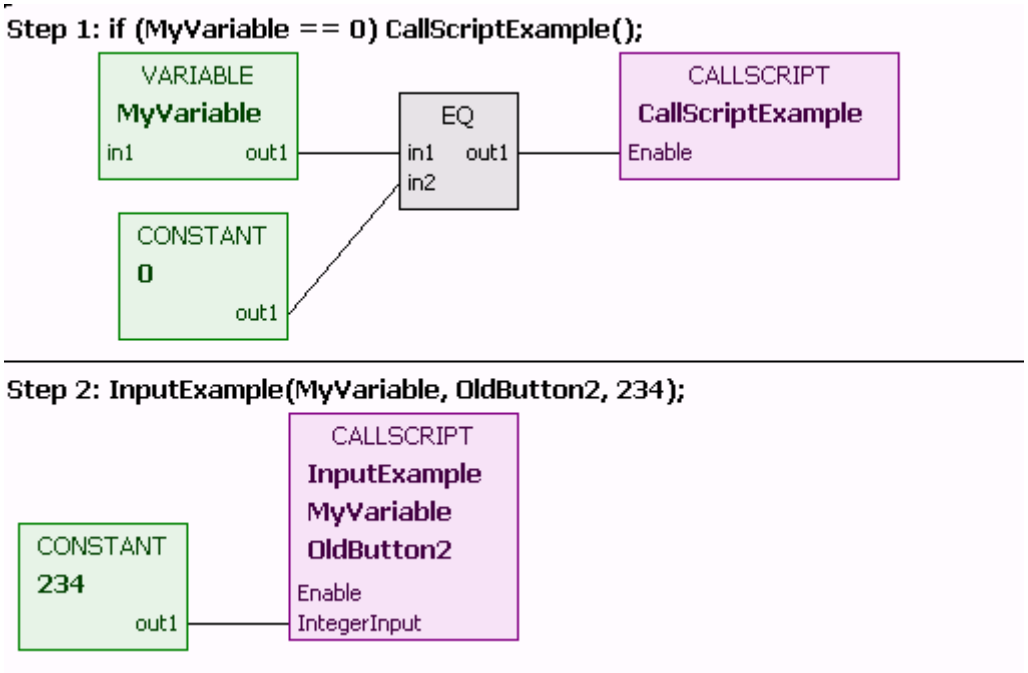
Skärmen visar valda längder, nuvarande längd samt hur längden som är uppnådd. Det finns en meny för att läsa av produktion i form av hur kapningar som gjorts per trädslag och längd. I brist på menyer och knappar så fanns knappkombination att trycka för att nollställa produktionen samt avaktivera diameterspärrning.

Det var skilda program för olika språk och dessutom skilda program för vissa maskinkonfigurationer. Detta försvårade arbetet när det fanns flera versioner.

3.4 CCD/MID-program

Detta kapitel behandlar utförandet av planeringen, programmeringen, testningen och ett underkapitel som presenterar användargränssnittet.

Det inleddes med en kurs vid Exertus i två dagar med en introduktion till enheterna, programmeringsmiljön Guitu och verktyget Canto. Fokus var på design av användargränssnitt, inte alls på programmeringen. Resurser och information gavs om hur programvara överförs samt manualer för dessa program.

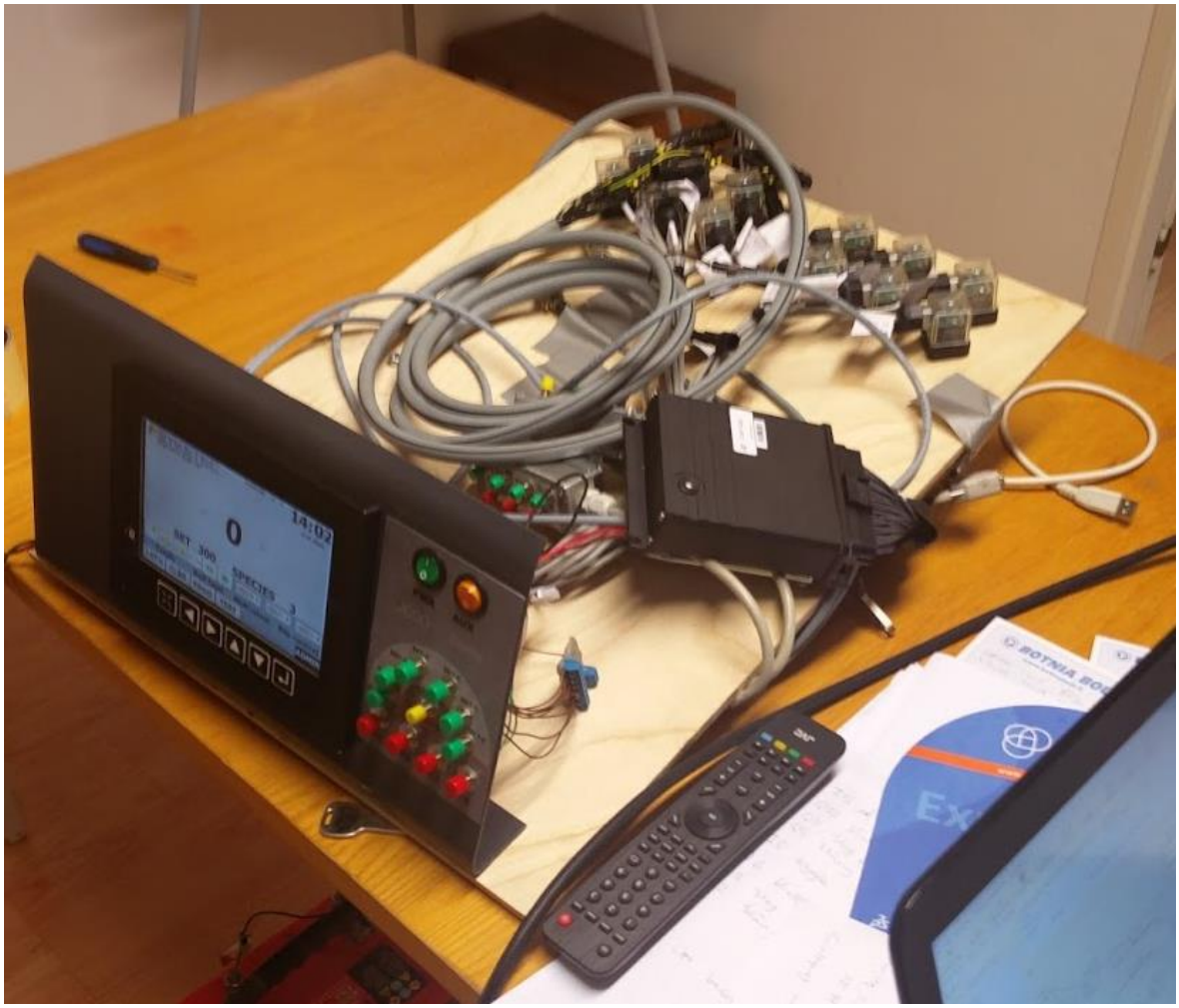


Figur 14 Ett exempel ur Guitu som visar hur skript kallas på.

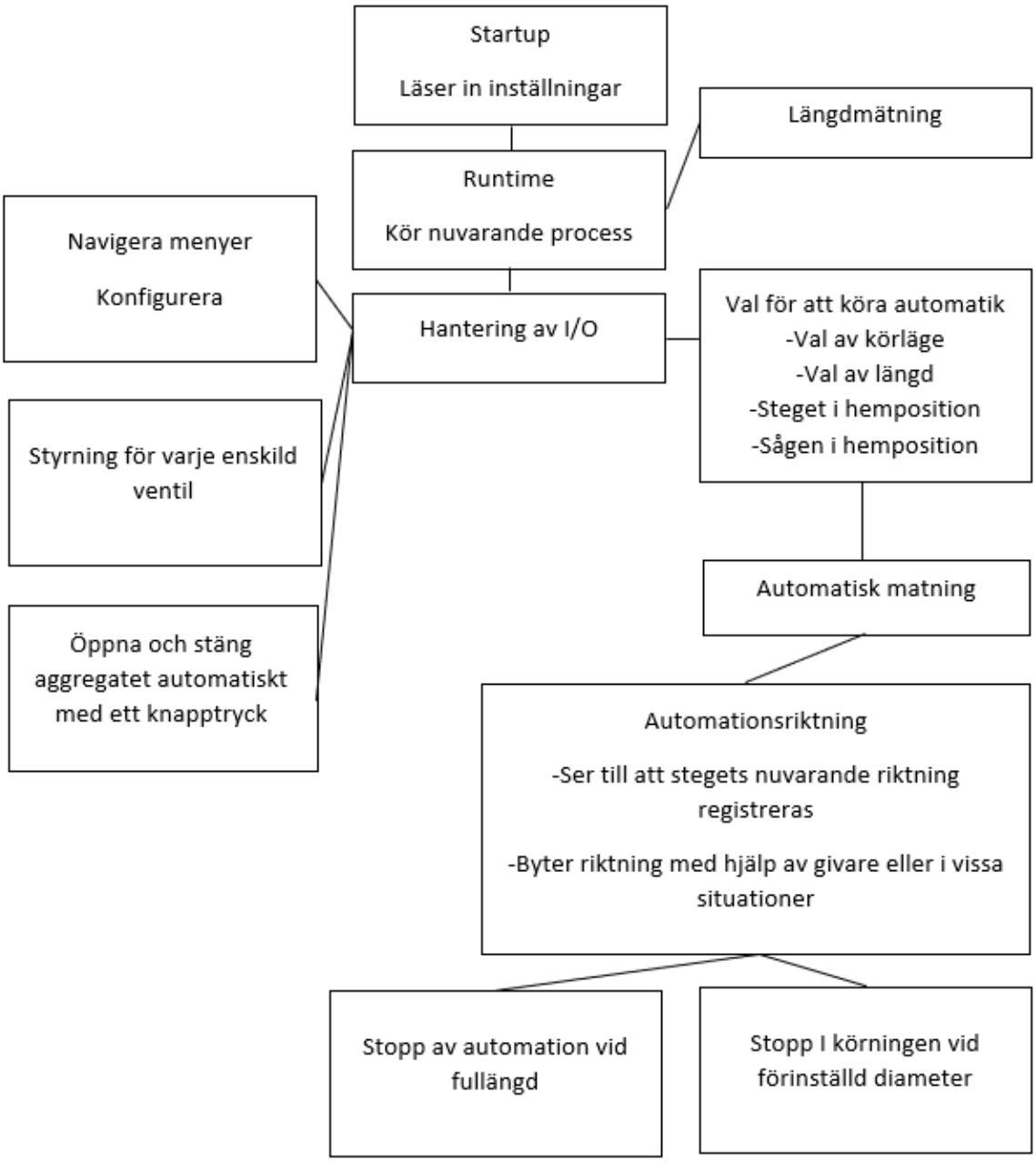
Arbetet inleddes med att bekanta sig med Guitu med hjälp av manualen. Det använder skript för programkörningen, där det i grunden finns Startup och AtRunTime. Egna skript definieras som köra genom AtRunTime eller händelser som knapptryck.

LOGO! Soft Comfort har verktyg för att analysera programflödet mot riktig hårdvara och även simulera programkörningar. Guitu har inte denna möjlighet att simulera och kräver därför ett testsystem. Det fanns redan definierat antalet in- och utgångar på det befintliga systemet samt om de var digitala eller analoga. Alla I/O sattes på lämplig pinne i enheterna vilket kan ses i bilagan med kopplingsschemat. Det kopplades upp på ett testsystem med ventilkontakter som hade LED-lampor så att aktiva utgångar kunde observeras.

Alla knappar kopplades mot respektive utgång samt givarna kopplades upp. Hela programmet kan med detta system teoretiskt testas genom knapptryck och att sätta ett metallföremål mot givarna.



Figur 15 Testsystemet som enheterna kopplas i och simulerar programmet.



Figur 16 Funktionerna när de brutits ut i delar (gjort i Word).

Efter en genomgång av det befintliga programmet kunde alla funktioner brytas ut i mindre delar för att klargöra hur de fungerade. Dessa mindre delar var lättare att hantera. När Guitu öppnas med ett nytt projekt så väljs en huvudenhet. När skärmenheten har valts så skapas funktionerna Startup och AtRuntime samt ett tomt fönster för att placera objekt. Enheten har ett förinställt nodnummer på CAN-bussen. Då ytterligare enheter sätts till så krävs unika nummer. Enheterna visualiseras vertikalt mot den horisontella bussen. Det går att anpassa noderna på egen hand och

byta överföringshastigheter samt välja standard eller utökade meddelanden. De förinställda valen fungerar utan ändringar. Programmet befinner sig endast i skärmenheten. Aggregatets enhet fungerar som hantering av I/O. Första funktionen var hantering av in- och utgångar. En testmeny byggdes för att prova skärmenhetens knappar och sedan kopplades en knapp till varje ventil. Därefter kunde detta testas mot testsystemet. Av testmenyn byggdes en diagnostikmeny så att skärmenheten kan visa att allting ger signal på knapptryck och givare, samt att dessa signaler når utgångarna.

Automatiken får inte starta om inte ett körläge valts. De fyra körlägena går sina egna spår. Automatiken får inte starta om ingen längd valts. Om inte sågen återgår till hemposition så startar det inte, annars bryts sågen när trädet matas fram. Utgångsläget för att automatiken ska starta är när steget är helt inskjutet.

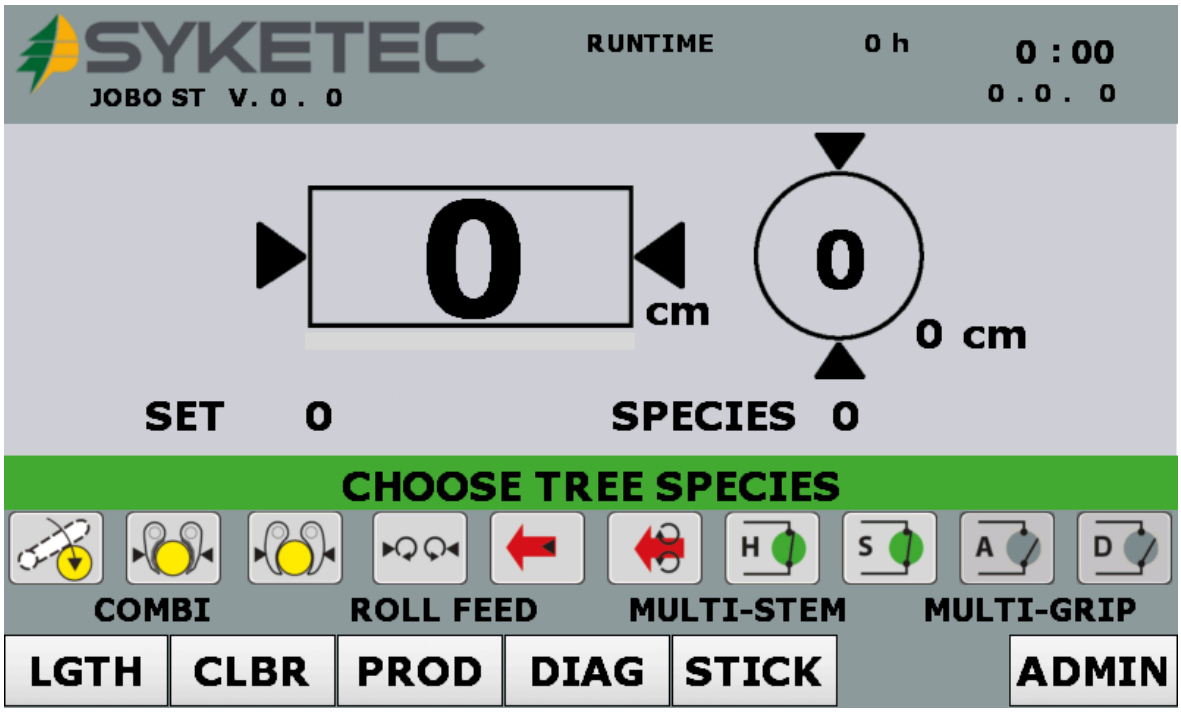
Längdmätningsfunktionen går kontinuerligt och mäter längder oavsett körläge. Det finns en funktion för att stoppa frammatning när den förinställda längden eller diameter gränsen har uppnåtts, i detta läge finns valet att såga eller att byta längd. Körs rullmatning så matar den funktionen fram tills en gräns uppnås. I stegmatningsläget behövs en funktion som håller koll på vilken riktning som matningen ska ske. När givaren signalerar att den är i ändläget så ska det gå in och ta ett nytt tag. När full längd uppnås går steget in och lämnar där.

Efter några tester så krävdes ännu att mäthjulet kunde kalibreras noggrant. Det var tvunget att göra en funktion så att det gick att stoppa körningen lite före full längd uppnås då hydrauliken inte reagera genast utan att göra mätfel.

I grunden är programmet det samma för båda skärmenheter, med några tillägg endast för MID-enheten. Användargränssnittet skiljer då CCD-enheten har mindre upplösning, minne för programkod samt lägre prestanda. Det gör att färre bilder används och måste beskrivas med text. CCD-enheten har mindre navigeringspilar på panelen vilket gör det svårare att navigera skärmarna.

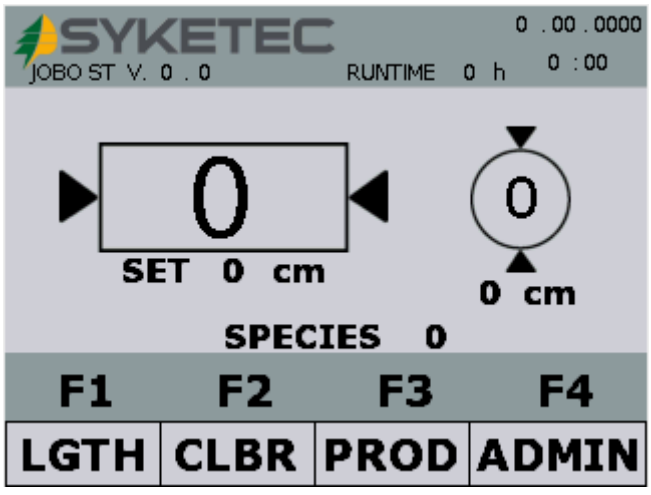
För överföring av programmet till CCD-enheten behövs en adapter för en Windowsdator med USB till CAN-buss. Överföringen sker med programmet Canto. För MID-enheten skrivs programmet till ett USB-minne och sätts i enheten.

3.4.1 Användargränssnitt



Figur 17 Huvudskärm för användargränssnittet på MID-enhet.

När användargränssnitt designas kan en mall bildas av att testa eller se på implementationer från liknande system, samt från orelaterade gränssnitt för riktlinjer. Flera styrsystem användes och prövades för att göra en bild över hur det skall fungera i praktiken. Det gav en inblick på element som behövs, hur stor del av skärmens yta som element använder, samt vilka alternativ och inställningar som användaren får anpassa.



Figur 18 Samma skärm som i figuren ovan men på CCD-enheten.

Några grundläggande principer för ett bra användargränssnitt: Ge överblick på alla alternativ så att allt går snabbt att ögna igenom. Sätt inställningar och information i kategorier för lättare navigering. Gruppera stora mängder innehåll. Använd så lite text som möjligt, designa det lätt så att manual inte är nödvändig. Visa tydligt vad som är möjligt att klicka på [11].

Upplägget på skärmarna planerades genom illustrering på papper, med information om vilka element som behövs. Det hölls lätt så att den mindre skärmenheten kan visa en liknande bild. Designen gick genom en iterativ process där användare frågas vad som behövs. När designen var utförd så testas programmet en tid, därefter avlägsnas buggar och användaren frågas igen om åsikter.

Huvudskärmen måste innehålla relevant information då maskinen körs. Det måste visa körläget, förvald längd samt nuvarande längd. Endast om diametermätning finns så ska den visas. Under förvald längd visas en meddelanderad som ger varningsmeddelanden. På denna rad förekommer också läget på aggregatets enskilda komponenter. Från huvudskärmen finns en meny för att navigera till resten av programmet.

Det finns en skärm med inställningar för enheten. Där kan ställas språk, klocka återställa allt till grundinställningar, ljusstyrka, mätmetod, kalibrering av mätrulle och diametermätning samt inställningar för olika programbeteenden. Det går att ställa om de fördefinierade längderna och tidreläer och fördröjningar i programmet ifall maskinen har högre eller lägre oljeflöde än det standardvärde som de anpassats för.

Under produktionsmenyn så kan antalet kapningar per trädslag ses. Det finns en grov kalkyl på hur många kubik träd som sågats om diametermätning finns i aggregatet.

För felsökning finns bilder av enheterna och dess in och utgångar. Det går då att kontrollera om utgångar aktiveras eller om knappsatserna har slutat att fungera. Det ger också varningsmeddelande när enheterna förlorar kontakten med varandra.

Slutligen finns en meny för inställning av styrspakar. De måste kalibreras så att spaken har en mittpunkt och så att en hel rörelse registreras. Strömmarna som skall gå till ventilerna kalibreras. Det kan finjusteras med tidsramper och progression. Dessa inställningar går att göra för flera användare för lagring i olika profiler. Vid behov kan en riktning väljas på styrspakarna för en hjälpfunktion, med det kan en knapp hållas inne en knapp för att den riktningen sedan styra en extra utgång.

3.5 Testning

Testning utfördes under utvecklingen med testsystemet. I det finns ingång för båda skärmenheter, aggregatets enhet, kranens enhet, knappsatser och styrspakar, utgångar med LED-indikator, mätrulle och alla resten givare. När programkörningar simuleras så kan fel upptäcka som är farliga att göra mot ett riktigt aggregat. Det går också att köra koden stegvis för noggrannare genomgång.

Funktioner som är baserade på automatik och längdgivare så måste ändå testas på ett riktigt aggregat för att ge realistiska testkörningar med längder och grepp om trädstammar. Testning gjordes med flera användare för att säkerställa att allt designats rätt.

Under programmets utveckling lämnades testmenyn kvar så att användare själva kan testa alla funktioner om fel uppstår.

4 Resultat

Det första målet med arbetet var att utveckla en ny produkt som hade all funktionalitet av vad den föregående programvaran hade. Det underlättade att skördaraggregatet i sig var helt färdigutvecklat så det inte fanns några frågetecken med hårdvaran. Målet uppnåddes därför ganska fort och det kunde därför vidareutvecklas för att ge en ännu bättre produkt. Resultatet av examensarbetet är ett fullt fungerande styrsystem programmerat med Guitu. Systemet uppfyller alla de krav som utsatts. Det består av två program för modulerna CCD1200S och MID070S som använder sig av nästan samma kod. Den mindre enheten har några funktioner borta och mindre användargränssnitt. Enheterna använder CAN-buss för kommunikation, det ger möjligheten att utöka modulernas antal vid expansion.

Programmet har nu ett fullt användargränssnitt för att visa status, varningar och även ställa in språk, längdförval och tidsvariabler. Det har också möjlighet att via inställningarna välja alternativ för olika maskinkonfigurationer utan att behöva flera programversioner. Det fanns många funktioner som inte hade gjorts och skjutits upp då de inte fungerade med LOGO och som nu kunde utföras i det nya, som till exempel möjligheten att använda styrspakar, mät hjul och diametermätning.

Programmet är i dagsläget helt komplett, tills aggregatet i sig utvecklas mera. Det finns också möjlighet att bryta ut delar av programmet, som till exempel styrspakarna. Det går att göra endast en kranstyrning av det med bara en modul för att skära ned på priset.

Sedan det blev klart så är styrprogrammet för Exertus moduler det som har använts och sålts för stegskördarna av Syketec.

5 Diskussion

Det har varit ett långt och ibland utmanande projekt, men samtidigt var det också väldigt intressant. Från början var det meningen att det skulle vara som en rak översättning från LOGO-programmet, men programmeringsmiljöerna var ändå så olika att det var tvunget att göra programmet lite annorlunda. Dessutom kom det mera tillägg och funktioner efterhand att sätta till i programmet.

Ser man tillbaka så hade det varit bättre att planera allt i detalj och göra hela programmet helt från grunden programmeringen ändå inte var helt likadan som i LOGO! Soft Comfort. Vissa funktioner kan ännu förenklas eller göras om för bättre läsbarhet och framtida utveckling.

Det mest utmanande var att komma igång med projektet, kursen vi var på i två dagar gick främst igenom design av användargränssnittet och lite grundläggande instruktioner om verktygen. Manualen blev en viktig del av utvecklingen då den beskrev alla funktioner för att komma igång. I programmet fanns också beskrivning på alla funktionsblock och ett kort exempel för blocket i användning.

Framförallt så har projektet varit lärorikt för mig. Jag fick göra på egen hand, så det blev mycket som jag testade att göra på eget initiativ och om det såg ut som en bra idé så fick det komma med i programmet. Efter att det hade testats en tid i skogen så kom det feedback på olika saker som kunde ändras eller om buggar som uppstått, då var det att göra de ändringarna och repetera processen igen tills allt fungerade.

Jag är nöjd med resultatet av projektet. Med mycket arbete så har det förbättrats så mycket från vad som erbjöds i LOGO-programmet. Det är en mera prisvärd och mera lättanvänd produkt än det som fanns förut. Trots allt detta arbete så finns det alltid rum för mera uppdateringar, genom att förbättra användargränssnittet, göra små tillägg enligt förslag eller till och med sätta till helt nya funktioner.

6 Källförteckning

[1] ”Syketec,” [Online]. Available: http://www.sykeharvesteri.fi/kontakt/. [Använd 2019].

[2] ”Stegmatare i kompakt utförande, passar små maskiner,” 2019. [Online]. Available: http://www.sykeharvesteri.fi/product-portfolio/harvesters/stegmatare-i-kompakt-utforande-passar-sma-maskiner-en-US/. [Använd 2019].

[3] W. Bolton, Programmable Logic Controllers, 4th red., Newnes, 2006, pp. 3, 4-8, 80, 94.

[4] ”Programmable logic controller - Wikipedia,” 10 Oktober 2019. [Online]. Available: https://en.wikipedia.org/wiki/Programmable_logic_controller. [Använd 2019].

[5] ”IEC 61131-3 - Wikipedia,” Wikimedia foundation, 12 Augusti 2019. [Online]. Available: https://en.wikipedia.org/wiki/IEC_61131-3. [Använd 2019].

[6] K.-H. John och M. Tiegelkamp, IEC 61131-3 : programming industrial automation systems : concepts and programming languages, requirements for programming systems, decision-making aids, 2nd red., New York: Hiedelberg, 2010, pp. 12, 98, 101,123.

[7] W. Lawrenz, CAN System Engineering, New York: Springer, 2013, pp. 1-15, 147.

[8] ”LOGO! Soft Comfort Online Help,” 2017. [Online]. Available: https://support.industry.siemens.com/cs/attachments/100782807/Help_en-US_en-US.pdf?download=true. [Använd 2019].

[9] Exertus, Guitu User Manual, 2016.

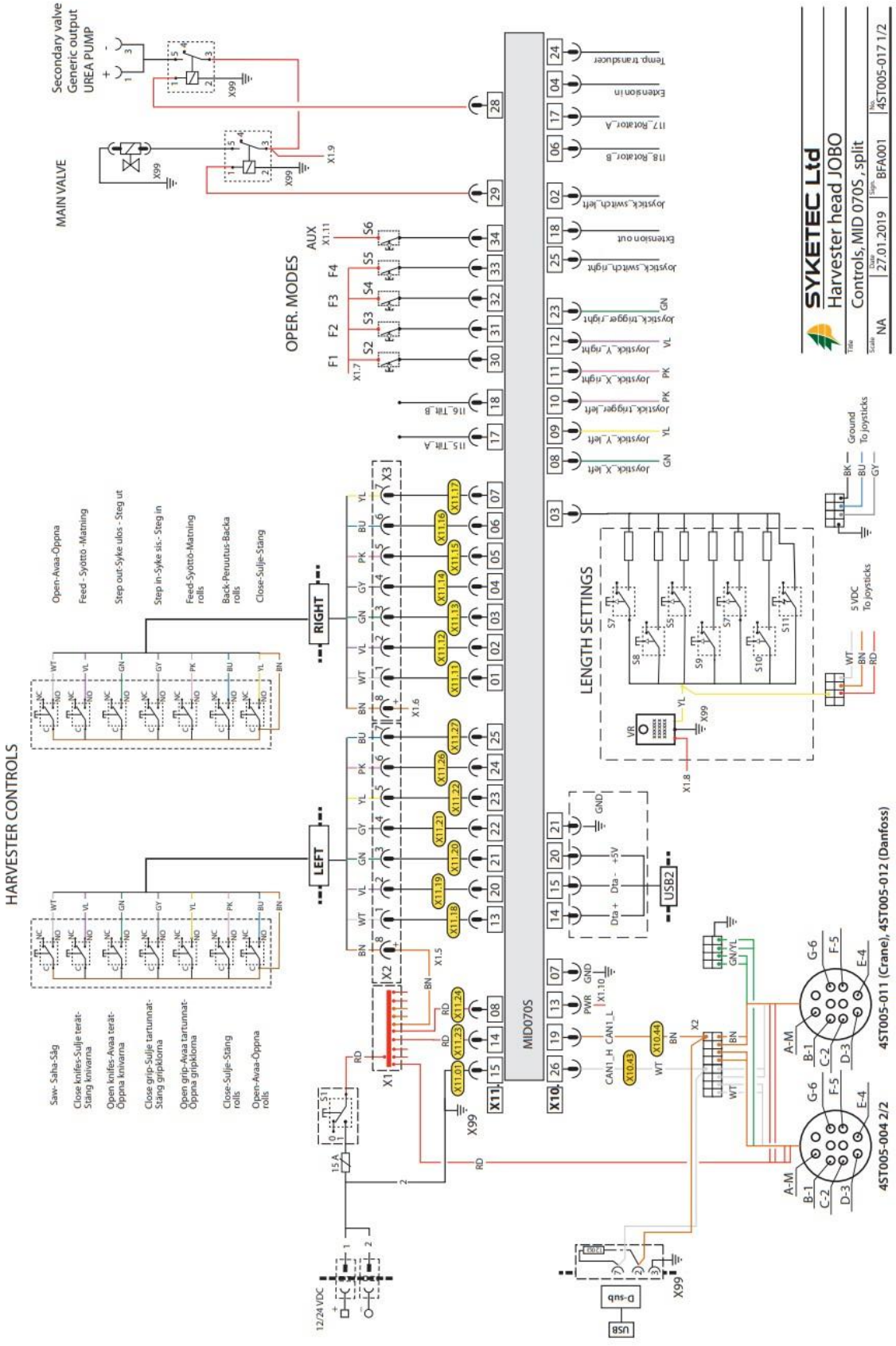
[10] Exertus, CANTO2 User Manual, 2014.

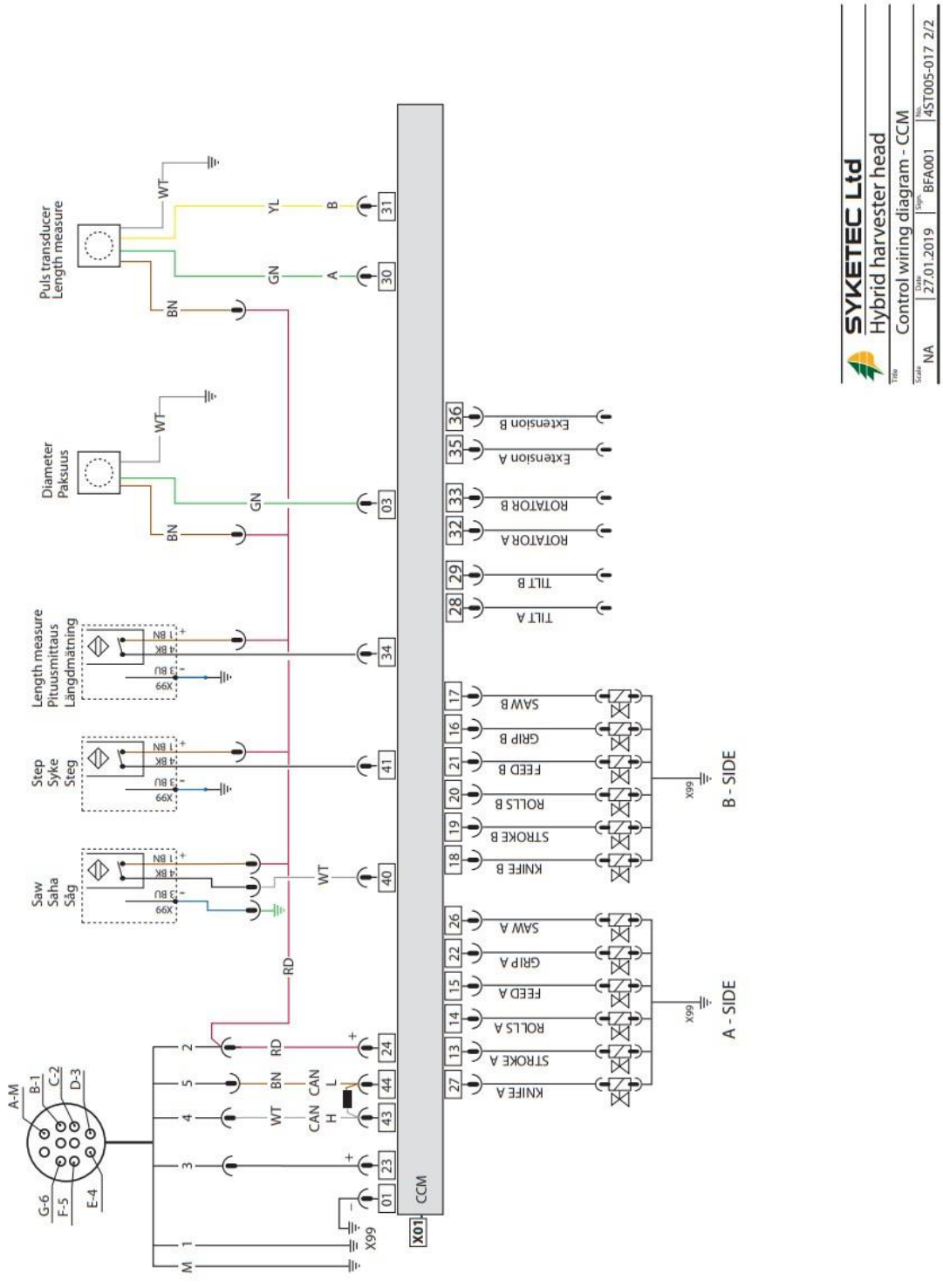
[11] S. Krug, Don't make me think, revisited : a common sense approach to web usability, Third edition red., San Fransisco: New Riders, 2014.

[12] ”CCD1200S, figur,” [Online]. Available: https://assets.website-files.com/5c545c15041cbc3e59a8ff56/5c9df88fa9898d33d6c8552d_CCD1200S_front%201000px-p-800.png. [Använd 29 10 2019].

[13] ”Siemens LOGO! TDE, figur,” [Online]. Available: https://assets.new.siemens.com/siemens/assets/api/uuid:30ab9cd80ab4956d830d60cec7722ad64eeae5aa/width:640/quality:high/version:1547043812/p_st70_xx_06993p.png. [Använd 29 10 2019].

[14] ”Siemens LOGO!, figur,” [Online]. Available: https://assets.new.siemens.com/siemens/assets/api/uuid:2afe8d4f-c80c-4c0d-8c7e-fda7371400e4/width:640/quality:high/version:1558090969/logo-8-24rceo-ac-basismodul-ohne-display-grey.png. [Använd 26 10 2019].


SYKETEC Ltd
Harvester head JOBO
Controls, MID 070S, split
Scale NA
Date 27.01.2019
Sign. BFA001
No. 4ST005-017 1/2
HARVESTER CONTROLS
MAIN VALVE
Secondary valve
Generic output
UREA PUMP
OPER. MODES
AUX
LENGTH SETTINGS
LEFT
RIGHT
MID070S
USB2
Saw- Saha- Såg
Close knifes-Sulje terät-Stäng knivarna
Open knifes-Avaa terät-Öppna knivarna
Close grip-Sulje tartunnat-Stäng gripklorna
Open grip-Avaa tartunnat-Öppna gripklorna
Close-Sulje-Stäng rolls
Open-Avaa-Öppna rolls
Open-Avaa-Öppna
Feed - Syöttö -Matning
Step out-Syke ulos - Steg ut
Step in-Syke sis.- Steg in
Feed-Syötto-Matning rolls
Back-Peruutus-Backa rolls
Close-Sulje-Stäng
Joystick_X_left
Joystick_Y_left
Joystick_trigger_left
Joystick_X_right
Joystick_Y_right
Joystick_trigger_right
Joystick_switch_left
Joystick_switch_right
Extension out
Extension in
Temp. transducer
Ground
To joysticks
5 VDC
To joysticks
12/24 VDC
4ST005-004 2/2
4ST005-011 (Crane), 4ST005-012 (Danfoss)


SYKETEC Ltd
Hybrid harvester head
Control wiring diagram - CCM
Scale NA
Date 27.01.2019
Type BFA001
No 4ST005-017 2/2
Puls transducer
Length measure
Diameter
Paksuus
Length measure
Pituusmittaus
Längdmätning
Step
Syke
Steg
Saw
Saha
Såg
CCM
X01
X99
A-M
B-1
C-2
D-3
E-4
F-5
G-6
WT
YL
GN
BN
RD
CAN H
CAN L
A
B
Extension A
Extension B
TILT A
TILT B
ROTATOR A
ROTATOR B
KNIFE A
STROKE A
ROLLS A
FEED A
GRIP A
SAW A
KNIFE B
STROKE B
ROLLS B
FEED B
GRIP B
SAW B
A - SIDE
B - SIDE