



CHALMERS
UNIVERSITY OF TECHNOLOGY

Implementering och övergång till AVL Flexicart i en växellådstrigg

Ersättning och utveckling av befintligt styrsystem till AVL Flexicart som styrsystem i en testrigg för växellådor

Examensarbete inom högskoleingenjörsprogrammet Mekanik

MAX KÄRRMAN
SIMON WARRINGER HEDÉN

Institutionen för Elektroteknik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2025

Examensarbete 2025

Implementering och övergång till AVL Flexicart i en växellådstrigg

Ersättning och utveckling av befintligt styrsystem till AVL Flexicart som styrsystem i en testrigg för växellådor

MAX KÄRRMAN
SIMON WARRINGER HEDÉN



CHALMERS

Institutionen för Elektroteknik
Chalmers Tekniska Högskola
Göteborg 2025

Implementering och övergång till AVL Flexicart i en växellådstrigg
Ersättning och utveckling av befintligt styrsystem till AVL Flexicart som styrsystem i en testrigg för
växellådor

Examensarbete inom högskoleingenjörsprogrammet Mekanik
MAX KÄRRMAN
SIMON WARRINGER HEDÉN

© Max Kärrman, 2025
© Simon Warringer Hedén

Handledare: Stefan Englund, Volvo GTT
Examinator: Bengt Lennartsson, Institutionen för elektroteknik, Chalmers tekniska högskola

Institutionen för elektroteknik
Chalmers tekniska högskola
SE-412 96 Göteborg
Telefon: +46-(0)31 772 10 00

Förord

Examensarbetet utfördes tillsammans med Volvo Group Trucks Technology (GTT) under våren 2025 som en del av högskoleprogrammet Mechatronik på Chalmers tekniska högskola. Arbetet täcker områden inom mekatronik, datorteknik, signalhantering och automation, och därför var det viktigt att det fanns en bred kompetens med många hjälpsamma kollegor på plats.

För det utförda arbetet vill vi speciellt tacka Stefan Englund på Volvo GTT, för hans stöd, handledning och expertis. Vi vill även tacka andra medarbetare på Volvo för deras stöd med tekniska kunskaper och vänliga inställning. Slutligen vill vi tacka Bengt Lennartsson för hans feedback och stöd som examinator.

Göteborg, maj 2025

Max Kärman

Simon Warringer Hedén

Sammanfattning

Projektet behandlar moderniseringen av ett styrsystem för en testrigg avsedd för test och utveckling av växellådor. Tidigare har ett egenutvecklat, Python-baserat styrsystem använts i kombination med en Siemens PLC. Kommunikationen mellan Python och PLC skedde via Modbus. För att förbättra testprocessens effektivitet och driftsäkerhet ersätts Python-systemet med AVLS "Flexicart", en avancerad testplattform anpassad för drivlinesimuleringar.

Arbetet omfattar integrationen av Flexicart i den befintliga testmiljön, inklusive elektrisk installation av givare samt anpassning av styrsystemet för att säkerställa kompatibilitet, optimal kommunikation och en högre säkerhetsnivå.

Resultatet visar att övergången till Flexicart har resulterat i ett mer robust och användarvänligt system som möjliggör mer tillförlitliga tester och en effektivare arbetsprocess. Utöver dessa förbättringar har dessutom säkerheten förstärkts för testriggen genom implementering av nya givare. Den installerade temperaturgivaren möjliggör övervakning av värmeutveckling på testriggens objekt eller direkt på riggkomponenter. Samtidigt ser dörrgivaren till att körningen automatiskt avbryts ifall en dörr öppnas in till riggen.

Den förbättrade driftsäkerheten bidrar till en stabilare och säkrare miljö för tester och personal. Dessutom ger den modulära uppbyggnaden av Flexicart och PUMA goda förutsättningar för att återanvända lösningen i framtida riggar, samtidigt som det nya systemet ger en långsiktig plattform för fortsatt utveckling hos Volvo.

Abstract

This project focuses on the modernization of a control system for a gearbox test rig at Volvo Group Trucks Technology (GTT). The existing system, based on a custom Python program communicating with a Siemens PLC via Modbus, was replaced with AVL's Flexicart platform and its PUMA software to enhance efficiency, reliability, and modularity. The implementation included hardware integration, such as installing temperature sensors and door switches, and adapting PLC programming to ensure compatibility. Communication was transitioned from Modbus to EtherCAT to resolve protocol conflicts and improve real-time performance. The results demonstrated improved system stability, user-friendliness through a graphical interface, and enhanced safety features. The project successfully established a foundation for future rigs, highlighting the benefits of standardized, modular systems in industrial testing environments.

Innehållsförteckning

Förord	4
Sammanfattning	5
Abstract	6
Innehållsförteckning	7
Förkortningar	8
1. Inledning	9
1.1 Bakgrund.....	9
1.2 Syfte.....	9
1.3 Avgränsningar.....	9
1.4 Precisering av frågeställningen.....	10
1.5 Inledande sammanfattning.....	10
2. Teoretisk bakgrund	11
2.1 Programmable logic controller (PLC).....	11
2.2 Testtrigg.....	12
2.3 Kommunikation.....	13
2.4 Befintligt Styrsystem.....	14
2.5 Övergång till AVL-baserat Styrsystem.....	14
2.5.1 Flexicart.....	15
2.5.2 PUMA.....	15
2.6 Givare.....	15
2.6.1 Temperaturgivare.....	15
2.6.2 Dörrbrytare.....	16
2.7 Teknisk sammanfattning.....	16
3. Metod	17
3.1 Omfattning och förberedelse.....	17
3.2 Arbetsprocess, planering och prioritering.....	17
3.3 Säkerhetsaspekter.....	18
3.4 Sammanfattning av metod.....	19
4. Genomförande	20
4.1 Projektplanering och förarbete.....	20
4.2 Implementering av dörrgivare.....	20
4.3 Implementering av temperaturgivare.....	21
4.4 Implementering av Flexicart.....	24
4.4.1 Kommunikation.....	25
4.4.2 Funktionstestning.....	25
4.5 Sammanfattning av genomförande.....	26
5. Resultat	28
5.1 Förbättrad systemstabilitet och tillförlitlighet.....	28
5.2 Användarvänlighet och modularitet.....	28
5.3 Säkerhet.....	30
5.4 Summering av resultat.....	30
6. Slutsats och diskussion	31
6.1 Reflektion över genomfört arbete.....	31

6.2 Effekt på användning och framtida tillämpning.....	31
6.3 Hållbarhet.....	32
6.4 Framtida arbete och utveckling.....	32
6.5 Sammanfattande slutsatser.....	32
7. Referenser.....	33
8. Bilagor.....	34

Förkortningar

- PLC: Programmable logic controller
- I/O: Input / Output
- Testtrigg - Provningsanläggning för kontrollerad utvärdering av komponenter/system
- Drivlina - Kraftöverföringssystem i fordon (växellåda, axlar etc.)
- Fältenheter - Sensorer/styrdon direkt kopplade till processen
- PUMA 2: Mjukvaran i Flexicarten som för enkelhets skull benämns som endast PUMA.
- GUI: Graphical User Interface - ett grafiskt gränssnitt användaren interagerar med
- Dyno - Dynamometer (Mätanordning för vridmoment och effekt)
- EtherCAT - Industriellt kommunikationsprotokoll
- Modbus - Äldre industriellt kommunikationsprotokoll
- SFC - Sekvensdiagram (programmeringsmetod)
- FBD - Funktionsblocksdiagram (PLC-programmering)

1. Inledning

Lastbilsindustrin är under konstant utveckling, något som är ännu mer sant idag än någonsin. Med ett ökande intresse för elektrifiering och effektivisering av drivlinor, krävs rigorös testning och produktutveckling för att uppnå satta mål för elektrifiering och sticka ut bland konkurrenter. Volvo är idag ett av världens största företag i branschen, och en av de största arbetsgivarna i Göteborgsområdet. De är just nu i en expansiv fas av sin forsknings- och utvecklingsavdelning, och genomför därför uppgraderingar och förbättringar bland sina testmiljöer.

1.1 Bakgrund

Volvo Group Trucks Technology (GTT) är i de första stegen av att byta ut ett styrsystem till en av deras testriggar. I dagsläget styrs riggen av ett egenutvecklat Pythonprogram i kombination med en Siemens PLC, där kommunikationen mellan dem sker genom kommunikationsprotokollet Modbus. På grund av riggens komplexitet har Pythonprogrammets kodbas blivit svår att hantera och felsöka, och både uppstart och aktiv styrning av tester är komplext och svårt att felsöka. Således är Volvo intresserade av att byta styrsystemet mot AVLS automationssystem PUMA som körs på deras produkt Flexicart. Framöver väntas allt fler riggar hos Volvo övergå till PUMA och planerade kommande riggar tas in med systemet i åtanke direkt från AVL.

1.2 Syfte

Volvo GTT är just nu i en fas av att expandera sina testanläggningar, och därför är det i deras intresse att förbättra och utveckla deras automationssystem och programpaketet i mjukvaran PUMA som styr många av deras testriggar. De väntar sig att deras riggar framöver kommer bli mer komplicerade och vill därför lägga en stabil grund med modularitet, familjaritet och pålitlighet hos testriggar.

1.3 Avgränsningar

På grund av pågående tester och tidsbegränsningar kommer inte det planerade automationssystemet att kopplas mot skarpt testobjekt. Istället kommer fokus främst ligga på att konstruera programpaketet för ett flertal olika komponenter som är nödvändiga för testriggens funktion, som temperatursensorer, säkerhetsfunktioner, tryckluft och temperaturreglage.

Gruppen hoppas även på att lägga grunden för ett större automationsprogram, där dessa signaler testas för att till exempel bryta pågående test om en dörr öppnas eller att ett annat larm går. Då kommer signaler från testobjekt att simuleras, exempelvis lyckad eller misslyckad växelsekvens.

1.4 Precisering av frågeställningen

1. Hur ser kopplingsschemat mellan dator och testrigg ut?
2. Varför blir riggen mer modulär av PUMA?
3. På vilket sätt blir systemet mer pålitligt tack vare PUMA?
4. Vilka säkerhetsåtgärder måste beaktas när en testrigg ska köras?
5. Hur sker datakommunikationen mellan Flexicart och testrigg?
6. Vilka tekniska utmaningar kan uppstå vid migrering från det egenutvecklade Pythonprogrammet till PUMA?
7. Hur skiljer sig simuleringen från verklig körning?

1.5 Inledande sammanfattning

Detta kapitel beskriver varför övergången till ett nytt styrsystem är relevant för Volvo GTTs fortsatta utveckling av testmiljöer. Genom att definiera syfte, avgränsningar och frågeställningar skapas en stabil grund för projektets genomförande. Med projektets mall utlagd fokuserar nästa kapitel på den teoretiska bakgrunden och de system som påverkar styrningen av testriggen.

2. Teoretisk bakgrund

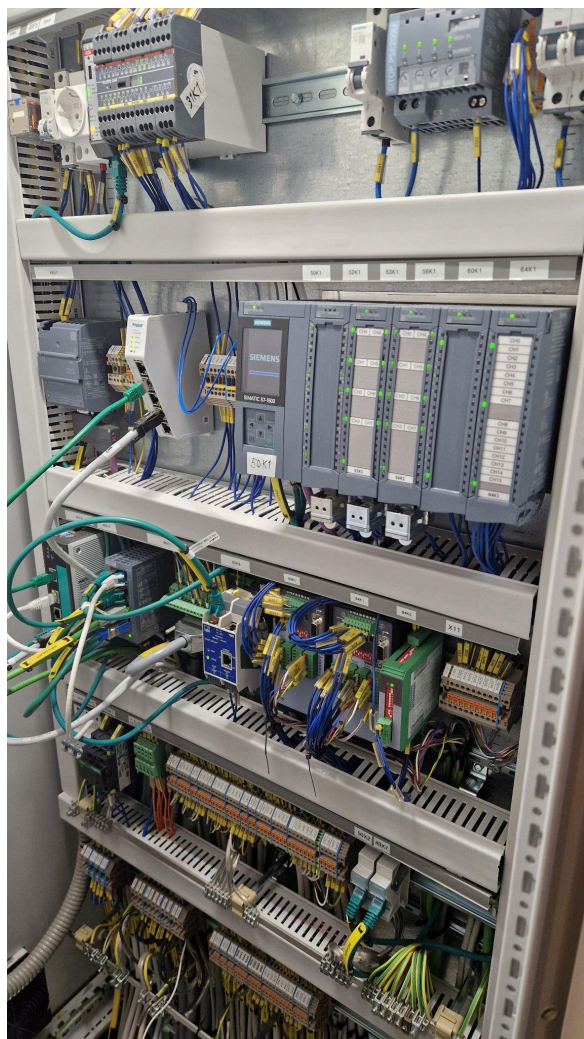
För att få en bättre uppfattning om projektet som utförts, är det viktigt att förstå vilka enheter och komponenter som använts. Styrssystem är en bred term där många olika enheter är sammankopplade till ett komplett system. Vanligtvis ser man styrssystem inom industribranschen där automation är en central del av arbetet, men man kan se styrssystem i nästan alla branscher.

2.1 Programmable logic controller (PLC)

Enligt William Bolton [1] är en PLC (Programmable Logic Controller) en sorts mikroprocessors-baserad kontrollenhet som använder ett programmerbart minne för att lagra instruktioner och utföra funktioner som logik, sekvensering, timing, räkning och aritmetik i ordning, för att styra maskiner. De är inte designade för att endast programmerare ska kunna använda dem, utan ska kunna justeras genom användning av en intuitiv form av programmering.

Vanliga användningsområden för enheten är tillverkningsindustrin och testanläggningar, men en av de stora fördelarna med PLC:n är att den grundläggande styrenheten har ett mycket stort användningsområde, då den kan kopplas samman med flera typer av I/O-enheter (Input/Output), som styrdon, givare, sensorer, ventiler, eller ytterligare styrenheter. Dessa I/O-enheter gör det möjligt för PLC:n att samla information från flera enheter och skicka styrsignaler mellan dem. Vanligtvis sker kontakt mellan enheter genom kommunikationsprotokoll som exempelvis Modbus och EtherCAT, vilket gör det möjligt att integrera PLC:n i större styrssystem [1].

Siemens PLC:n som användes vid projektet var en Simatic S7-1500 styrenhet tillsammans med flera I/O-enheter. I ett större system var denna styrenhet en central del eftersom den gav signaler till många andra enheter, som exempelvis en säkerhets PLC, en D/A-omvandlare (digital-to-analog-converter) och direkt till dynos som fysiskt kör riggen (se figur 2.2 nedan).



Figur 2.2: Simatic S7-1500 styrenhet med flera I/O-moduler.

2.2 Testrigg

Under utveckling av nya produkter krävs kontrollerade miljöer där prestanda och funktionalitet kan utvärderas med upprepbara tester. Dessa miljöer, som här benämns som testriggar, består av en kombination av sensorer, styrdon och mjukvara som samverkar under testprocessen. Detta projekt fokuserade på utvecklingen av ett nytt styrsystem för en specifik testrigg inom Volvo GTTs transmissionsavdelning. Riggens syfte är att testa hela drivlinor, och den benämns därför internt som Växellådsprovrigg 4 (förkortat Växel 4).

Från testriggsingenjörens perspektiv innehåller Växel 4 ett antal kritiska komponenter som är centrala för riggens funktionalitet. De främsta är: två elektriska dynos, tryckluftsreglage, kyloljekonditioneringssystem samt provobjektet (växellådan). Här skiljer det mellan riggfasta komponenter (dynos, tryckluft, konditionering), och provobjekt.

Eftersom testriggen enbart är avsedd för att testa växellådan som enskild komponent, används elektriska dynos för att eliminera motorn som en okänd variabel. Dynoserna har en dubbel funktion: dels att generera rotationsrörelse, dels att simulera körningsmotstånd genom reglerbar lastpåverkan.

Tryckluften ansvarar för växlingsmekanismen, medan konditioneringsystemet ansvarar för att säkerställa stabila testförhållanden.

Provobjektet är instrumenterat med en omfattande uppsättning sensorer och mätgivare. Dessa, tillsammans med styr- och återkopplingssignalerna från riggens övriga komponenter samlas kontinuerligt in av styrsystemet. Här analyseras data i realtid för att både reglera testprocessen och övervaka växellådans prestanda enligt definierade testparametrar.

2.3 Kommunikation

För att säkerställa tillförlitlig och centraliserad kommunikation mellan riggfasta komponenter och styrsystemet används en PLC. PLC:n fungerar som ett gränssnitt mellan styrsystemets högnivåkommandon (insatta av riggingenjören eller testsekvenser) och de specifika enheterna som ska utföra dessa kommandon.

Styrsystemets huvudsakliga funktion begränsas här till hantering av önskvärda tillstånd (börvärden) och visualisering av aktuella tillstånd (ärvärden) för riggens fasta komponenter. Den faktiska styrlogiken implementeras i PLC:n som:

- Tolkar och processar börvärden från styrsystemet.
- Övervakar enheternas tillstånd genom kontinuerlig sensoravläsning.
- Reglerar komponenterna enligt sin inbyggda programlogik.
- Skickar tillbaka aktuella ärvärden till styrsystemet för övervakning.

Kommunikationen till och från PLC:n skedde via Modbus-protokollet. Modbus är ett kommunikationsprotokoll som utvecklades av Modicon år 1979 specifikt för användning med PLC-system [3]. Protokollet överför data via seriella länkar mellan industriella elektronikheter. I denna implementering fungerade PLC:n som klient (master), och hämtade data från fältenheterna (server/slaves). Även styrsystemet agerade server. Protokollet stödjer både värden på en bit (coil) och värden på 16 bitar (register). PLC:n i Växel 4 använde sig enbart av coils. Meddelandena i protokollet innehåller:

- Data adress
- Läs/Skriv-flagga
- Det faktiska värdet

Eftersom Modbus-protokollet saknar inbyggda mekanismer för kommunikationsövervakning, implementerades detta direkt i PLC:n. För att övervaka den distribuerade I/O:n som är ansluten till PLC:n har Siemens mjukvara använts, som säkerställer en tillförlitlig kommunikation. För kommunikation mellan styrsystem och PLC:n har en så kallad Watchdog implementerats. Denna fungerar enligt följande princip:

1. PLC:n skickar ett värde till styrsystemet.
2. Styrsystemet skickar samma värde till en adress monitorerad i PLC:n.
3. PLC:n inverterar det mottagna värdet och skickar det på nytt.
4. En tidsgräns på 100ms används för att detektera kommunikationsbrott.

Vid överskriden tidsgräns initieras ett omedelbart nödstopp av testprocessen för att säkerställa systemets säkerhet.

2.4 Befintligt Styrssystem

Det befintliga styrsystemet bestod av en komplex, egenutvecklad lösning baserad på ett Pythonprogram i kombination med en Siemens PLC. Pythonprogrammet producerade en GUI där testsekvenser kunde laddas in, alternativt vissa börvärden som vilken växel provobjektet skulle befinna sig i och vilka varvtal och moment det skulle utsättas för. Den visade även aktuella ärvärden för relevanta parametrar. Ytterligare en GUI fanns även för styrning av vissa funktioner implementerade direkt i PLC:ns logik, exempelvis uppstart av riggens dynos och konditioneringssystem.

Pythonprogrammet kan beskrivas som den del som ansvarat för hantering av parametrar vilka relaterar till styrning av provobjektet. Det var den del av styrprogrammet som direkt kommunicerade med provobjektet, genom interna verktyg Volvo utvecklat för att styra växling för deras växellådor. Andra parametrar relaterade till provobjektet hanterades även här, exempelvis de givare och sensorer som anslutits till provobjektet för att övervaka det under aktiv testning, som interna temperaturer och upplevda moment och varvtal. Pythonprogrammet ansvarade därför även för övervakning av gränsvärden och felhantering av växling, där det var designat för att stoppa pågående testning om skador på provobjekt kunde uppstå.

PLC:ns roll är istället att skydda riggen och personal som utför provning, alltså ett lager djupare än Pythonprogrammet. Då testtriggarna är medvetet överdimensionerade gentemot objekten som ska testas i dem, kan riggfasta objekt som elektriska dynos och kardaner klara mycket högre moment och varvtal än växellådorna som testas där. Gränsvärden för dessa objekt behövs ändå övervakas, då risken för katastrofala utfall, som plötsliga spikar i moment eller oväntade fel hos testobjekt kan ske, oavsett hur osannolikt det är. Dessa funktioner övervakas av PLC:n då den har direkt kommunikation med testtriggens fasta objekt och därför kan bryta pågående provning snabbare än Pythonprogrammet. Dessa gränsvärden är dessutom oföränderliga oavsett provobjekt, och är mycket mer kritiska för säkerheten av de som utför proverna och det har därför tagits ett beslut att de ska vara svårare att ändra, med en högre access-nivå krävd för att ändringar ska genomföras.

Kombinationen av två GUI:s, och Pythonprogrammets kodstruktur resulterade i en extremt komplex uppstart av riggen, och en mängd data och fönster som var svårtolkade under körning. Vidare krävdes det ofta stora omskrivningar av koden för att anpassa den efter nya provobjekt, något som väntades ske regelbundet. För en riggingenjör med erfarenhet av Växel 4 kunde detta innebära en uppstartsprocess på cirka 20 minuter, om inga misstag eller felklick skett under starten vilket ofta krävde att processen initierades på nytt. För riggingenjörer som inte besatt erfarenhet med riggen kunde denna process ofta ta mycket längre tid, och eventuell felsökning ansågs vara alldeles för svår.

2.5 Övergång till AVL-baserat Styrssystem

För att förbättra styrsystemet i Växel 4 valde Volvo att gå vidare med ett styrsystem utvecklat av företaget AVL. AVL är ett företag baserat i Österrike, som erbjuder en mängd tekniska lösningar för utveckling av drivlinor genom simulering, virtualisering och testautomation [3].

2.5.1 Flexicart

AVL erbjuder lösningar för fullständiga testtriggare och testceller, men då det i detta fallet redan fanns en testtrigg beslutades det här att använda sig av en produkt de kallar Flexicart. Flexicart är ett datorsystem som levereras med en Windows-installation tillsammans med AVLs mjukvara installerad. Vid sidan om Windows installationen är ett antal av processorns kärnor reserverade för ett realtidssystem som övervakar signaler från ett antal moduler som valts till Flexicarten. Dessa moduler kan fylla en rad funktioner, som analog I/O, digital I/O, olika kommunikationsprotokoll som CAN, EtherCAT och Modbus, eller annat som krävs av en specifik testtrigg [4]. Detta ger kunden möjligheten att specialanpassa varje Flexicart för varje unik rigg eller testmiljö, utan att behöva utbildas för en ny mjukvara och GUI varje gång.

2.5.2 PUMA

PUMA är namnet på den programmiljö AVL använder för sina styrsystem. PUMA erbjuder många möjligheter för styrning av riggar som kan kombineras med varandra för att bygga komplexa styrsekvenser. Mindre funktioner kan byggas i ett programmeringsspråk, men det finns också stöd för tillståndsmaskiner och SFC-programmering. Även den GUI som riggingenjören interagerar med kan designas från grunden.

Då Volvo GTT har använt sig av PUMA som testmiljö för flera av sina testtriggare under ett antal år har de byggt upp ett programbibliotek och en designfilosofi för användningen av systemet. Målet för Volvo är att det ska vara så lätt som möjligt för deras testingenjörer att flytta sig mellan olika testtriggare, trots deras i vissa fall enorma olikheter i riggkomponenter. Därför designas GUI enligt samma riktlinjer för alla deras implementationer, och gemensamma funktioner för till exempel dynostyrning och växling av testobjekt har tagits fram.

2.6 Givare

För att säkerställa noggrann övervakning och styrning av testtriggen används givare för att mäta relevanta parametrar såsom temperatur, tryck, vridmoment, oljeflöde mm. Dessa givare spelar en central roll i styrandet av testtriggen. Givarna är integrerade i styrsystemet genom elektriska kopplingar, där deras signaler, både digitala och analoga, behandlas av antingen Siemens PLC:n eller direkt i AVL:s Flexicart. Givarna möjliggör en hög nivå av automation samt säkerhet för personal, testobjektet och testtriggen självt.

Givarna som används i testtriggen har varit installerade sedan långt innan projektets start, och en stor del av arbetet är att se till att användningen sker felfritt vid ersättningen av styrsystem istället för att installera nya givare, men två nya givare ska installeras i testtriggen.

2.6.1 Temperaturgivare

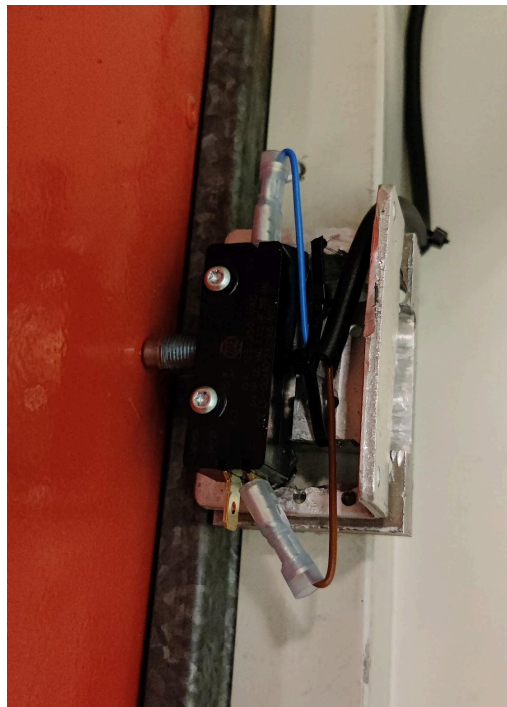
Temperaturgivaren Optris CS LT används för att kontinuerligt mäta temperaturen på testobjektet med hjälp av infraröd teknik. Givaren fungerar genom att sända ut infraröd strålning (8–14 µm) och analysera det reflekterade ljuset, vilket möjliggör noggranna mätningar inom intervallet –50 °C till 1030 °C med en precision på ±1,5 °C.

För projektet konfigurerades givaren med en analog 0–10 V-utsignal, vilken kopplades till Flexicart via en D-sub-kontakt. En skalningskurva implementerades i PUMA för att omvandla den analoga

signalen till korrekta temperaturvärden. Givarens snabba responstid (14 ms) och justerbara emissivitet (0,100–1,100) säkerställer tillförlitliga mätningar även under dynamiska förhållanden. [6]

2.6.2 Dörrbrytare

Den första givaren som installerades var en enkel "normally open" brytare, där en knapp på givaren bröt kretsen när den inte var intryckt (se figur 2.4.2). Givaren gav en digital signal till Flexicarten som informerade ifall dörren som den är fäst mot var öppen eller stängd. Kretsen bryts då dörren öppnas, och Flexicarten läser då en 0:a på sin digitala ingång, vilket resulterar i värdet FALSE hos variabeln bundet till dörrsensorn. En kortare programmeringssekvens ser till att systemet nödstoppas när denna variabeln får värdet FALSE, och på så vis nödstoppas hela testriggen när dörren öppnas.



Figur 2.4.2: Dörrbrytare i form av en "normally open" brytare.

2.7 Teknisk sammanfattning

Detta kapitel har introducerat de komponenter och kommunikationssystem som är betydelsefulla för projektets genomförande. Det beskriver funktionalitet hos både det befintliga och det nya styrsystemet och jämför dem mot varandra, vilket även leder till att behovet av förändring tydliggjorts. Denna tekniska grund möjliggör en mer konkret beskrivning av hur projektet planerades och genomfördes, vilket behandlas i nästkommande kapitel.

3. Metod

I detta kapitel beskrivs de metoder som användes för utförandet av projektet. Kapitlet kan delas upp i tre delar, som alla har ett eget ändamål. Djupdykning av dokumentation och en komplett förståelse av komponenter som behandlas i 3.1, planeringsschemat och planeringen som beskrivs i 3.2, och säkerhetsöversikten som krävde en översiktlig analys vilket beskrivs i 3.3.

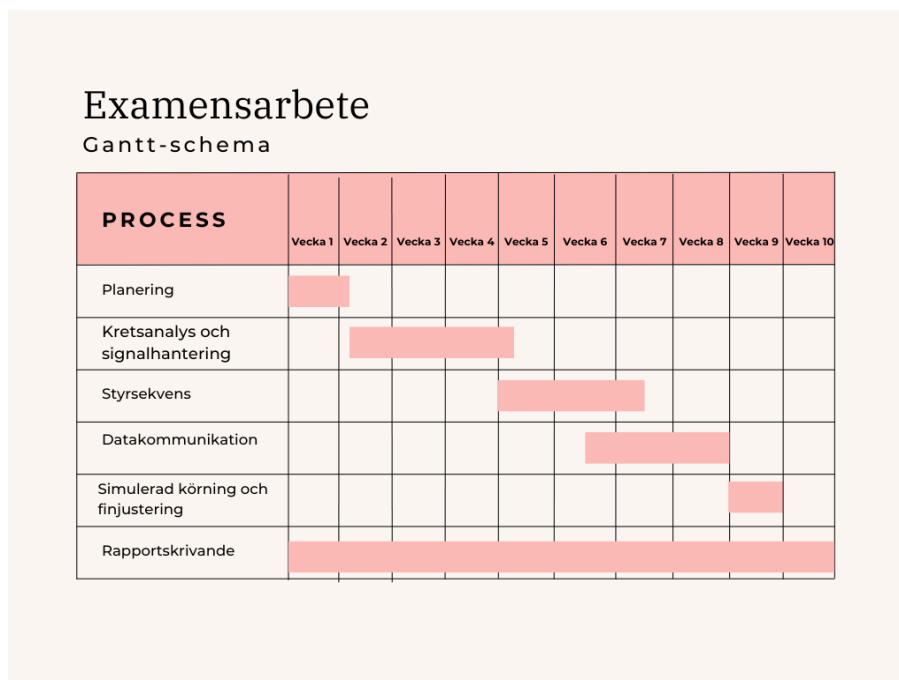
3.1 Omfattning och förberedelse

Omfattningen av projektet blev en av de stora frågorna vid arbetets start. Många olika enheter skapar det styrsystem som ska ersättas och utvecklas, och en grundläggande förståelse för det kompletta systemet blev det första steget. Därför blev en stor del av den inledande fasen att sätta sig in i den givna dokumentationen och göra en närmare granskning av de relevanta komponenterna i systemet.

3.2 Arbetsprocess, planering och prioritering

I planeringsfasen togs ett Gantt-schema fram för att ge en tydlig överblick över projektets struktur och för att underlätta planeringen inom den tillgängliga tidsramen, se figur 3.2 nedan. Schemat gav gruppen en mer konkret bild kring hur arbetet och tiden kunde delas upp och med överlappande delmoment blev övergångarna mellan momenten naturliga. Utöver de givna momenten i schemat var säkerhetsaspekter en stor del av projektet, som gavs tid och utrymme under varje delmoment att analyseras och påverka arbetet.

Gantt-schemat fungerade främst som ett stöd för att hålla arbetet på rätt spår, men eftersom många moment överlappade var det svårt att följa en exakt tidslinje. Trots detta gav schemat en bra överblick och gjorde det enklare att planera och justera arbetet längs vägen vid behov.



Figur 3.2: Gantt-schema över projektets arbetsprocess.

3.3 Säkerhetsaspekter

På Volvo är säkerhet av största vikt och därför var arbetet med säkerhet en av de centrala aspekterna för det utförda projektet. Redan i planeringsfasen utfördes en riskanalys där risker identifierades och deltagarna i projektet skapade en genomgående utredning kring hur de kunde förebyggas (se tabell 3.3). De identifierade riskerna delades upp i följande kategorier:

- Mekaniska risker: Rörliga delar som kan skada personal.
- Elektriska risker: Kortslutningar, felkommunikation och oskyddat kablage.
- Felsignaler: Givare som inte registrerar korrekt eller nödsignaler som ej fungerar.
- Systemfel: Programkraschar och skadat styrsystem.
- Manuella misstag: Riggingenjör eller mekaniker som gör misstag.
- Bristande dokumentation: Ej markerade nödstopp eller givare.

Under projektets gång blev det tydligt att vissa säkerhetsrelaterade kategorier inte låg inom gruppens ansvar att förebygga, eftersom dessa redan var utredda och hanterade av olika avdelningar inom Volvo. Mycket kablage byggdes av Volvos elverkstad, mekanikerna var väl medvetna om att hålla avstånd från aktiva rigger, och vissa tekniska delar från det tidigare styrsystemet skulle fortsatt användas. Detta innebär att sannolikheten för flera av de identifierade riskerna bedöms som låg. Samtidigt gav dessa kategorier projektgruppen en tydlig struktur kring vilka säkerhetsaspekter som behövde beaktas, samt vad som skulle vara färdigställt inför projektets avslutning.

Varje identifierad risk gavs en sannolikhet (1-5) och en konsekvens (1-5), vilket tillsammans gav dem en risknivå ($S * K$). De risker med högst risknivå gavs störst prioritet, men i verkligheten åtgärdades riskerna parallellt längs arbetets gång (se tabell 3.3 på nästa sida).

När riskerna identifierats behövde de också hanteras på ett praktiskt sätt. Ett system sattes upp som automatiskt stänger av riggen om dörren öppnas under drift så att ingen av misstag kan befinna sig där inne när ett test pågår. Kablar och givare kopplades på ett säkert och tydligt sätt och funktionerna testades noggrant för att allt skulle fungera som det skulle.

Detta säkerhetsarbete bidrog inte bara till att uppfylla Volvos säkerhetskrav, utan även att skapa en tydlig struktur för några av projektets lösningar. Genom det nära samarbetet mellan projektgruppen och Volvos interna avdelningar kunde ett integrerat, pålitligt och säkert system tas fram.

Tabell.3.3: Riskbedömning

Risk	Sannolikhet (1–5)	Konsekvens (1–5)	Riskenivå (S × K)	Förebyggande åtgärder
Operatör befinner sig i riggen vid teststart	1	5	5	Säkerhetsgivare monterad i dörr, nödstopp kopplad till dörrkontakt, automatiskt stopp via mjukvara, larmsignal när testtriggen aktiveras
Givare registrerar inte att dörren är öppen	2	4	8	Funktionstest av givare före varje körning, visuell kontroll vid behov
PLC eller Flexicart hänger sig under körning	2	5	10	Watchdog-funktion implementeras
Kodfel i testskript (t.ex. felaktig logik)	3	4	12	Kodgranskning, simulering före användning, versionshantering
Operatör gör fel vid manuell hantering	4	4	16	Tydliga instruktioner och utbildning, skyltning vid riggen, kontrollerade gränsvärden

3.4 Sammanfattning av metod

Tillsammans skapade de använda metoderna en bra helhetsbild kring projektet och ett bra tillvägagångssätt för utförandet. Tydliga prioriteringar, en förståelse av viktiga komponenter tillsammans med en planerad arbetsgång är viktigt för omfattande projekt, och med förberedda metoder kunde det utföras som planerat.

4. Genomförande

Genomförandet av arbetet genomfördes stegvis, för att gruppen under arbetets gång skulle utveckla en djupare förståelse för systemet i sin helhet. Inledningsvis analyserades det tidigare systemet tillsammans med testtriggen för att kunna skilja ut vad som påverkas av projektet och vad som kommer att behålla sin funktionalitet. Efter detta påbörjade gruppen arbete med mindre komponenter för att bekanta sig med Flexicartens I/O, både digitalt och analogt. Slutligen fokuserade arbetet på att förstå kommunikationen mellan Flexicarten och riggen, och hur PUMA behandlar detta.

4.1 Projektplanering och förarbete

Projektet inleddes med att skapa en helhetsbild av vad som behövde göras och vilka krav det nya systemet skulle uppfylla. För att få en tydlig överblick granskades det befintliga styrsystemet, som bestod av ett egenutvecklat Pythonprogram i kombination med en Siemens PLC. Det var viktigt att förstå hur styrningen av testtriggen tidigare hade hanterats, för att kunna få en bild kring vilka delar som kunde behållas, och vilka delar som behövde ersättas.

Särskilt vikt lades vid att förstå vilka funktioner i det tidigare systemet som var avgörande för riggens drift, som exempelvis nödstopp, signalövervakning och den generella styrningen av testtriggen. Dessa delar hade en direkt påverkan på styrningen av testtriggen och dess säkerhet, vilket ansågs som generellt, men viktigt att planera. Genom att identifiera funktionerna blev det enklare att hitta en struktur kring hur arbetet skulle utföras, och hur man kunde avgränsa varje delmoment i processen.

4.2 Implementering av dörrgivare

Den första givaren som installerades var en enkel "normally open" brytare, där en knapp på givaren bröt kretsen när den inte var intryckt (se figur 1). Givaren monterades på en metallskiva, som i sin tur fästes vid dörrkarmen tillsammans med ett metallfäste. När dörren var stängd var knappen intryckt, vilket slöt kretsen. För att ansluta givaren drogs kablar från dess plus- och minuspol längs dörrkarmen och väggen fram till Flexicarten. Kablarna fästs genom lödning på en nio-polig D-Sub-kontakt ansluten till de pinnar som PUMA strömförsörjer och använder för att hantera digitala insignaler. Denna installation säkerställde en stabil och pålitlig anslutning mellan givaren och styrsystemet. Nästa steg var att definiera en variabel för den digitala insignalen, som implementerades i testtriggen genom ett kortare skript skrivet direkt i PUMAs mjukvara. Skriptet övervakar variabeln och reagerar om kretsen bryts, det vill säga när dörren öppnas. Vid detta tillfälle skickas en signal som omedelbart stoppar riggens drift (se fig 4.2.1, 4.2.2).



Figur 4.2.1: Dörrbrytare och kabeln till Flexicart.

```
if (isvalid(I_safety_door_01) or isvalid(I_safety_door_02) or isvalid (I_safety_door_03)) and SYSState >=5 then
  if not I_safety_door_01 or not I_safety_door_02 or not I_safety_door_03 then
    DO_enable_warning_lamp = 1

  else
    DO_enable_warning_lamp = 0
  endif
elseif SYSState < 5 then
  DO_enable_warning_lamp = 0
endif
```

Figur 4.2.2: Kod för nödstopp samt varningslampa.

De implementerade säkerhetsåtgärderna kan vara extrema, eftersom det kan vara mycket farligt att befinna sig inne i testriggen under ett pågående test. Genom implementeringen säkerställdes en pålitlig nödstoppsfunktion, vilket minimerade risken för olyckor och skapade en säkrare arbetsmiljö.

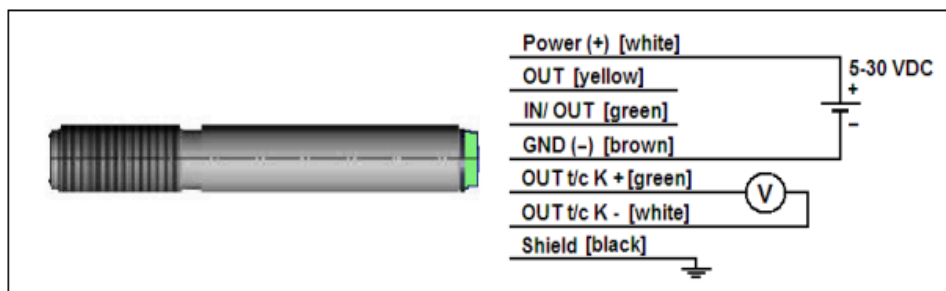
4.3 Implementering av temperaturgivare

För att säkerställa att objektet som provas i testriggen inte överhettar, eller att det ska hålla en viss temperatur under testet, installerades en temperaturgivare. Implementeringen av denna givaren är mer komplicerad då den ger en analog signal istället för en digital.

Givaren hade till en början bara en kabel med ett USB-C-port uttag, och efter all dokumentation för givaren hade granskats togs beslutet att klippa kabeln och istället skapa en D-sub kontakt till givaren så kopplingen till Flexicart kunde ske. Under granskningen fann gruppen att en termoelement-koppling kunde skapas samtidigt, och kunde vara användbar för att snabbt och enkelt kunna använda givaren utan att behöva koppla den till Flexicarten. På så vis kunde givaren testas och valideras innan den slutgiltiga installationsfasen. Termoelement-kopplingen är den kontakt som sticker ut från D-sub-kontakten (se fig 4.3.1, 4.3.2).



Figur 4.3.1: Temperaturgivare och dess kontakter.

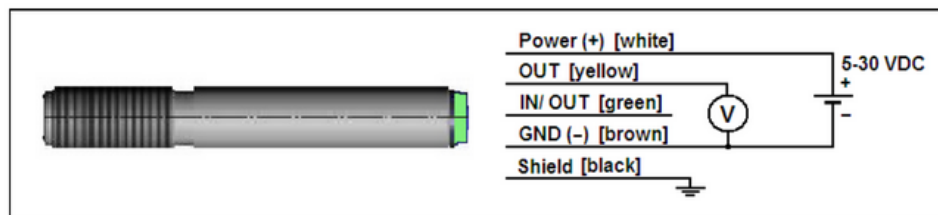


Figur 4.3.2: Givarens kontaktkrets för termoelementkoppling.

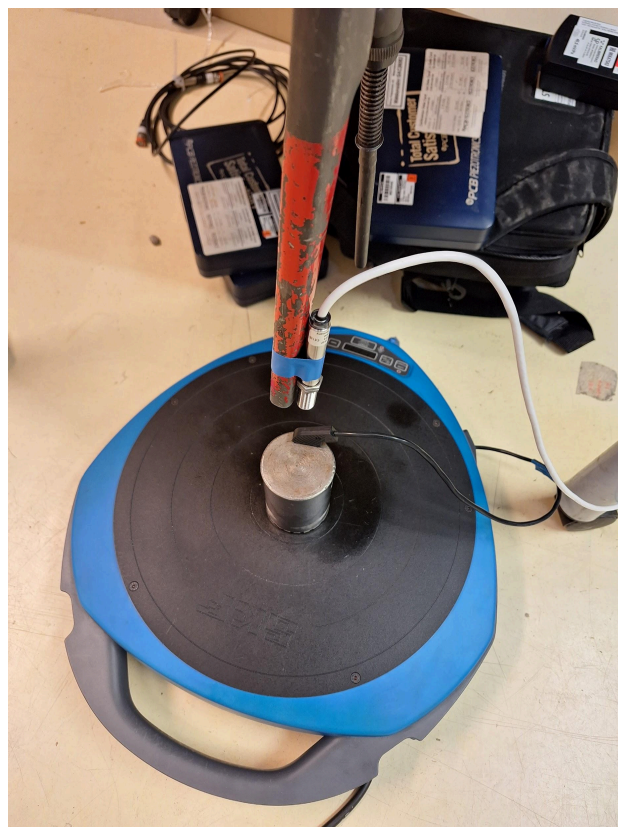
Valideringen av givaren utfördes genom att hetta upp en metallcylinder med en värmeplatta som har en integrerad givare som fästs på metallen och håller den i den valda temperaturen. Temperaturgivaren kopplades in i en M5stack, vilket är en liten processorenhet med en display som kunde kopplas med termoelementkopplingen för att se den uppmätta temperaturen. När metallcylindern nått sin temperatur pekades givaren mot metallcylindern och det kunde konstateras att givaren fungerar, då den angav samma temperatur som värmeplattans givare var.

Nästa steg var att koppla givaren direkt till Flexicarten via D-sub-kontakten, kopplad enligt givarens dokumentation för analog mätning, för att få den att fungera tillsammans med styrsystemet. En

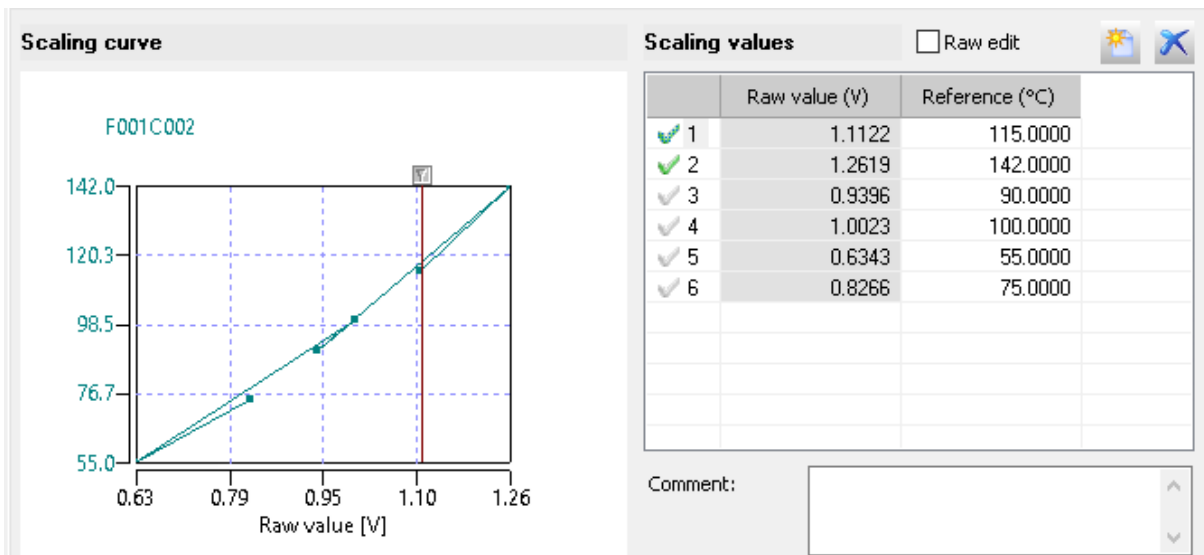
skalningskurva krävdes för att få givaren att fungera med en hög grad av noggrannhet. Den skapades med en liknande metod för den tidigare valideringen, men för minimera felmarginalen fästes givaren i en position direkt mot metallcylindern, för att se till att den inte rörde sig mellan de olika punkterna i skalningskurvan. Skalningskurvan skapades i PUMA, där metallcylindern först hettades upp till 55° Celsius och det angavs som givarens första referenspunkt. Ytterligare 5 gånger ändrades temperaturen på metallcylindern med värmeplattan, och varje gång den nått sin temperatur, skapades en ny referenspunkt för givaren i PUMA. På så vis skapades en skalningskurva och givaren var nu redo att fästas i testriggen (se fig 4.3.3, 4.3.4, 4.3.5).



Figur 4.3.3: Givarens kontaktkrets för analog signal.



Figur 4.3.4: Fastsatt givare mot upphettad metallcylinder på en värmeplatta.



Figur 4.3.5: Skalningskurvan för temperaturgivaren.

Nästa steg var att skapa en förlängningskabel för att se till att allt kablage kunde fästas längs väggar och tak, så att det inte är i vägen för mekaniker, ingenjörer, städare eller för det pågående provet. Den skapades genom att på samma sätt som kablarna kopplats till D-sub-kontakten från givaren, går kablarna i en förlängningskabel med en D-sub-kontakt i båda ändar (se figur 4.3.6).



Figur 4.3.6: Förlängningskabel med dubbel D-sub-kontakt.

4.4 Implementering av Flexicart

Innan den faktiska inkopplingen av Flexicarten till testriggen kunde påbörjas analyserades kommunikationen mellan Pythonprogrammet och PLC:n för att bygga förståelse för hur riggens automationssystem skulle kunna bytas ut med så få ändringar som möjligt. Få ändringar var önskat då man delvis redan var nöjd med PLC:ns hantering av riggfasta komponenter, och delvis på grund av att det var önskvärt att få testriggen i ett fullt fungerande skick så snabbt som möjligt för att kunna

genomföra väntande prov. Efter detta delades övergången in i två steg; kommunikation mellan Flexicart och PLC, och funktionstestning av parametrar samt testsekvenser.

4.4.1 Kommunikation

Inledningsvis skapades en tabell över samtliga parametrar som skickades och togs emot via Modbus i det redan implementerade Pythonprogrammet. Denna tabell implementerades därefter i PUMA-systemet och kopplades till nya parametrar för att säkerställa full funktionell kompatibilitet mot det nya systemet. Under testfasen upptäcktes dock en problematik kring kommunikationsprotokollet Modbus: både PLC:n och PUMA-systemet kräver att agera som master (klient) i protokollet, vilket skapar en konflikt. Eftersom en omfattande omdesign av kommunikationsarkitekturen ändå krävdes, beslutades det att utvärdera alternativa protokoll och man beslutade sig för att övergå till EtherCAT istället.

EtherCAT är ett klient/server-protokoll precis som Modbus, och även här behöver både PLC och PUMA agera klient. Även EtherCAT bygger på en klient/server-modell, och även här måste både PLC och PUMA agera som klienter. Dock erbjuder EtherCAT flera fördelar jämfört med Modbus, främst:

1. Realtidsprestanda, med deterministisk kommunikation,
2. Enklare infrastruktur, eftersom det enbart kräver Ethernet-anslutning utan dedikerad hårdvara,
3. Kostnadsbesparingar, genom att befintliga Ethernet-portar kan användas istället för att investera i extra Modbus-moduler för Flexicarten.

[5]

För att åtgärda problemet med två enheter som agerar master bryggas kommunikationen mellan dem med en EtherCAT bridge terminal. Denna fungerar som en server mellan de två klienterna – den läser data från PUMA-systemet och skriver sedan om dessa till de register som PLC:n kan läsa från. På så sätt uppnås en stabil och synkroniserad datakommunikation utan konflikter.

Slutligen valde man att kopiera tabellen över de register PLC:n läser ifrån och placera den på ett register med annan adress än den tidigare Modbus-implementeringen. Detta gjordes så att man lättare skulle kunna återgå till det föregående systemet ifall senare problem som visar sig svårare att lösa skulle uppstå.

4.4.2 Funktionstestning

Med kommunikationen färdigställd påbörjade man nu verifiering för att säkerställa att den fungerade som väntat. Testningen inleddes med en valideringsprocedur där en enskild bit (värdet '1') skrevs till ett känt register från PUMA för att se att detta lästes in korrekt av PLC:n. Denna initiala verifiering visade snabbt att den nya kommunikationsimplementationen fungerade som avsett.

Därefter implementerades en watchdog-mekanism i PUMA för att övervaka kommunikationens tillförlitlighet. Watchdoggen bygger på samma princip som i den tidigare lösningen: PUMA speglar ett booleskt värde som skrivs av PLC:n, medan PLC:n kontrollerar detta värde med en timer för att upptäcka eventuella kommunikationsavbrott. Denna design säkerställer att båda systemen kan detektera och hantera felaktigheter i realtid.

Med watchdogens funktionalitet säkerställd gick man vidare till att försöka styra komponenter till riggen, inledningsvis en början lufttryckssystem och dynos. För att undvika möjliga skador gjordes detta utan provobjekt eller monterade kardaner.

Testproceduren innebar att:

1. Skriv ett önskat varvtal eller tryck i PUMA.
2. Visuellt och auditivt bekräftade att dynon roterade och att tryckluftssystemet aktiverades som förväntat.
3. Verifiera att de uppmätta värden från fältenheterna monterade i riggen (till exempel sensorer för varvtal och tryck) stämmer överens med önskat värde och därmed säkerställa korrekt datainhämtning

Efter att grundfunktionaliteten säkerställts monterades provobjektet in i riggen, och testningen utvidgades till mer komplexa styrsekvenser.

Första fasen fokuserade på korta och isolerade sekvenser, såsom:

- Växling.
- Applicerad rotation på ingående axel mothåll av inställt moment på utgående axel.
- Oljekonditionering (flöde och temperatur)

Därefter introducerades längre och mer integrerade sekvenser, där flera funktioner testades. Exempelvis testades växling under belastning: Riggen utförde en växlingsprocedur med samtidig rotation och momentapplicering, med krav på att den korrekt:

1. Varvar ner motorn.
2. Slutar applicera moment.
3. Utför växlingen.
4. Återgår till drift med rotation och moment utan avvikelser.

Denna stegvisa metodik säkerställde att systemet hanterade både grundläggande och avancerade scenarier kritiska för riggens fulla funktionalitet.

I samband med att testning av alla funktioner för testriggen genomfördes upptäcktes det att viss funktionalitet sedan tidigare inte var implementerat enligt den standard Volvo utgår från idag. Ett antal funktioner för att skydda provobjektet var programmerade djupt in i PLC:ns logik, och då denna består av långa kedjor med FBD kod, samt att den kräver en högre access-nivå, gör det svårt att upptäcka och ändra av riggingenjör. När detta upptäcktes flyttades de funktioner som påverkats ut ur PLC:n och integrerades istället fullständigt in i PUMA i form av dess SFC-programmering. Det ger resultatet att det är lättare för riggingenjörer och provningsingenjörer att felsöka programmen och se vart i koden provningen har stoppats, och åtgärda eventuella fel.

4.5 Sammanfattning av genomförande

Genomförandet visade hur planeringen tillämpats i praktiska moment, där installation och validering av givare samt integration av Flexicart genomfördes stegvis. Den tekniska övergången hanterades med

både försiktighet och flexibilitet, särskilt vid kommunikationen mellan styrsystem, för att inte riskera en lång tidsperiod där testriggen var tagen ur bruk. Med alla centrala delar på plats kunde tester påbörjas och förbättringar identifieras, vilket utvärderas i nästa kapitel.

5. Resultat

Efter genomförd implementering av AVL:s Flexicart och nödvändiga givare, har flera förbättringar av systemet fastställts jämfört med det tidigare styrsystemet som byggde på ett Pythonprogram. Det avslutade projektets resultat kan delas upp i 3 delar: systemets stabilitet och tillförlitlighet, dess användarvänlighet och modularitet, samt dess ökade säkerhetsnivå.

5.1 Förbättrad systemstabilitet och tillförlitlighet.

Ett av målen med implementeringen av det nya styrsystemet var att öka robustheten och därmed minska antalet driftfel jämfört med det tidigare Python-baserade systemet. Det tidigare systemet var egenutvecklat av Volvo och hade genomgått flera ombyggnationer, vilket gjorde det svårt att felsöka, särskilt när det gällde att avgöra om fel berodde på provobjekt eller själva mjukvaran. Genom att ersätta detta med det mer standardiserade och dedikerade Flexicart- och PUMA-systemet har ett stabilare och mer komplett styrsystem skapats. De mjukvaruproblem som tidigare orsakade driftstörningar och påverkat testresultaten har därmed eliminerats, och styrsystemet behöver inte längre beaktas som en okänd variabel.

5.2 Användarvänlighet och modularitet

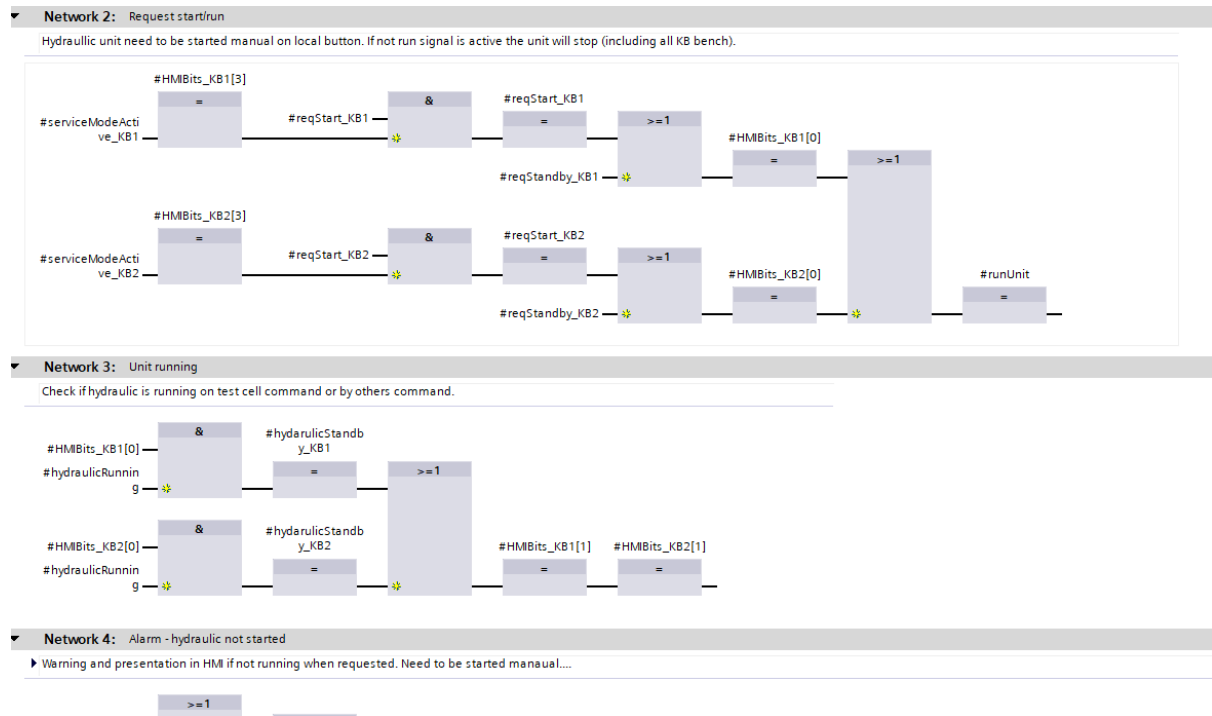
En väsentlig skillnad efter införandet av Flexicart är användargränssnittet som kommer med PUMA. Tidigare upplevdes Pythonprogrammets GUI som svårt att hantera, men med det nya styrsystemet har ett grafiskt gränssnitt liknande det som används i andra PUMA-riggat införts, vilket underlättar konfiguration och drift av testriggen. Med det tidigare systemet var uppstarten för testriggen mycket tidskrävande, samt krävde erfarenhet, medan med det nya systemet kan användaren med enkelhet ändra testparametrar, starta sekvenser och övervaka signaler.

Den modulära designen av Flexicart och PUMA ger möjligheten till ett styrsystem som kan anpassas till en mängd olika tester, då Flexicartens moduler lätt kan bytas ut, och erbjuder stöd för ett brett utbud av in- och utsignalsmoduler. Denna flexibilitet gör det enkelt att implementera nya komponenter som givare och kommunikationsgränssnitt, utan att behöva göra stora förändringar på systemet. I kombination med ett uppbyggt programbibliotek i PUMA som kan expanderas ytterligare, minskar behovet av specialanpassningar och dessutom påskyndar implementeringsprocessen för nya testmiljöer. Denna standardisering lägger en stabil grund för Volvo GTTs testanläggningar och säkerställer att systemet kan anpassas till framtida behov utan omfattande omkonstruktion. Det underlättar dessutom underhåll och felsökning eftersom enskilda komponenter snabbt kan bytas ut utan att påverka hela systemet.

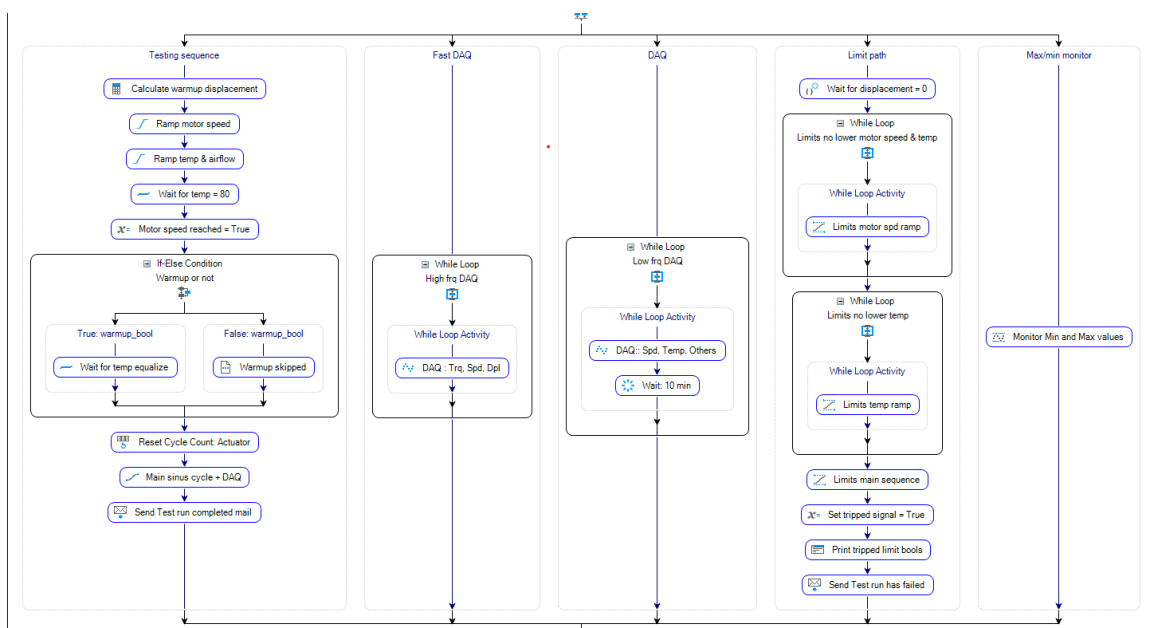
Slutligen blev även testsekvenser mycket lättare för användare att följa och felsöka, då PUMAs SFC-kod går att följa under körning. Detta är en enorm skillnad mot Pythonprogrammet, då det endast bestod av kodrader och vid debuggning var man tvungen att utgå från felmeddelanden och aktuell rad i sekvensen för att sedan arbeta bakåt därifrån. Även de sekvenser som lyftes ut från PLC:ns FBD-kod blev lättare att följa, då det kan vara svårt att få en visualisering av vilken ordning blocken har exekverats i och hur det har påverkat sekvensen.

På grund av sekretesskäl kunde inte kod som var aktuell för projektet användas, utan bilderna nedan är exempel från andra riggar. FBD-nätverket startar ett hydraulsystem och kontrollerar att det är igång. Utan att känna hela nätverket kan det vara svårt för en operatör att snabbt se när nätverket har

exekverats. SFC-koden ansvarar för rotation av ett testobjekt och konditionering av kammaren det befinner sig i, samt monitorering av gränsvärden. Operatören kan snabbt se vilket block programmet befinner sig i och var det har brutit om så sker (se fig 5.2.1 samt 5.2.2).



Figur 5.2.1: Exempel på FBD-kod.



Figur 5.2.2: Exempel på SFC-kod.

5.3 Säkerhet

En viktig förbättring efter det utförda projektet var att testriggen blev säkrare att använda, eftersom det uppgraderades med en temperaturgivare och en dörrsensor. Temperaturgivaren möjliggjorde övervakning av temperaturen på objektet, och därmed minskade risken för överhettning. Samtidigt ökade dörrgivaren säkerheten genom att automatiskt avbryta körningen ifall dörren öppnades under drift, vilket minskar risken för olyckor och ger en tryggare arbetsmiljö för personalen. Innebörden av detta var att systemet har en bättre kontroll över både driften och arbetsmiljön.

Även styrsystemet i sig bidrog till en säkrare lösning. Det tidigare styrsystemet var ofta instabilt och kunde orsaka oplanerade avbrott och andra driftfel. Med övergången till Flexicart och PUMA blev styrningen mer driftsäker, vilket i sin tur minskade risken för oförutsedda fel som kunde påverka personal och utrustning.

Efter utförandet av projektet blev riggen inte bara användarvänligare, stabilare och mer modulär, utan även säkrare.

5.4 Summering av resultat

Resultatet visar tydligt att bytet till Flexicart och PUMA medförde en stabilare, säkrare och mer användarvänlig tesmiljö. Den ökade modulariteten och förbättrade felsökningsmöjligheten skapar ett system som är mer framtidssäkert. Dessa resultat bildar grunden för de slutsatser och reflektioner som behandlas i de kommande och avslutande kapitlet, där projektets fulla påverkan diskuteras i ett bredare sammanhang.

6. Slutsats och diskussion

Detta kapitel analyserar projektet i sin helhet efter att arbetet är avslutat. Här drar gruppen slutsatser kring de frågeställningar som inledningsvis ställdes, de resultat arbetet förväntades ge och möjliga framtida utvecklingar. Det behandlar även den påverkan projektets resultat kan ge ur ett hållbarhetsperspektiv.

6.1 Reflektion över genomfört arbete

En stor del av arbetet gick åt till att läsa dokumentation hos de olika komponenterna. Ett styrsystem är ett samlingsord för många enheter som skall fungera i harmoni. Därför blev det en stor utmaning att förstå alla olika komponenter samt deras mjukvara. Som mer utförligt beskrivet i den teoretiska bakgrunden, består systemet först av ett Pythonprogram som kommunicerar med en Siemens PLC, som vidare skickar signal till en distribuerad I/O-enhet som i sin tur styr dynos. Inledningsvis var det därför av stor vikt att förstå det befintliga styrsystemet för att underlätta implementeringen av det nya styrsystemet.

Nästa utmaning var att förstå det nya styrsystemet, alltså Flexicart och dess mjukvara PUMA, samt de givare som skulle fungera tillsammans med PUMA. En bred kompetens skapades därför av gruppen för att kunna hantera projektet, vilket ledde till att det fysiska arbetet många gånger fick stannas upp medan gruppen hanterade all dokumentation och instruktioner som kom med projektet. En viktig lärdom har varit vikten av tydlig dokumentation och strukturerad felsökning.

En kort tid efter att gruppen påbörjat arbetet fick projektet högre prioritering från Volvos sida. Det innebar att fler resurser tilldelades, och att tidsramen flyttades fram. Till följd av detta fick gruppen strukturera om sitt arbete och planering, och fokusera mer på koncepten kring styrsystem och kommunikation, istället för att även implementera och bygga upp styrsekvenser och program. Det innebar också att skarpa provobjekt tilldelades, vilket möjliggjorde testning på en verkligare nivå istället för provning på enskilda riggkomponenter och simulerade objekt som planerades från början.

6.2 Effekt på användning och framtida tillämpning

PUMAs miljö öppnar möjligheten för mer avancerad och djupgående testning, med styrsekvenser som är lättare för provningsingenjörer att strukturera och riggingenjörer att genomföra. Standardiseringen i linje med många av Volvo GTTs andra riggar gör det även möjligt med större kunskapsdelning mellan anställda som i sin tur kan leda till nya idéer och lösningar. Det öppnar även möjligheten för prover där sekvenserna är uppbyggda genom en kombination av tidigare teoretisk bakgrund och tester, och data insamlad från användning av provobjekten i praktiska tillämpningar i form av körning med lastvagn. Alla dessa förbättringar förväntas leda till produkter som har genomgått testning mer likt det objekten kommer utsättas för i verkligheten efter de gått ut i produktion, och därigenom bli effektivare, med ökad kvalitet, och med vidare eliminering av möjliga fel som annars inte upptäckts i tid.

På lång sikt väntas man tjäna igen den kostnad som den initiala investeringen inneburit på grund av minskad tid där provning inte pågår på grund av omprogrammering eller fel inom styrsystemet. Möjligheterna som PUMAs system erbjuder med mer avancerad testning och större förmåga till automation av tester, och därmed mer testtid, väntas också spara pengar genom förbättrade produkter och tidig upptäckt av möjliga felkällor.

6.3 Hållbarhet

Det tidigare styrsystemet krävde ofta speciallösningar och anpassningar för varje nytt testobjekt, vilket ofta resulterade i engångsprogrammering och överflödiga kod. Med PUMAs standardiserade bibliotek och återanvändbara moduler underlättas byte av objekt och det väntas inte längre vara nödvändigt att genomföra prover där enbart programmet testas, och på så sätt förbrukas mindre energi. De förbättrade möjligheterna för provning kan också förväntas minska materialförluster, delvis genom förminskning av provobjekt som behöver ersättas efter ett misslyckat prov och delvis på grund av eventuell tidig upptäckt av fel hos produkter.

PUMAs modularitet och den förbättrade provningsmiljö den erbjuder, öppnar för möjligheten att genomföra utförlig provning på energiförluster genom hela drivlinor, och eliminera onödiga förluster. Då Volvo satsar på mer hållbara produkter och minskade utsläpp väntas detta leda till ett grönare utbud av produkter och en mer hållbar framtid inom fordons- och transportindustrin.

6.4 Framtida arbete och utveckling

Som ett förslag på framtida arbete anser gruppen att det är möjligt att ersätta Siemens PLC:n och utföra all logik med PUMA i Flexicarten. Gruppen startade efter utfört projekt ett kortare arbete för att undersöka hur implementeringen av detta kan ske, men vid undersökningen ansåg gruppen att detta kan vara ett omfattande projekt självt, eftersom den logik som utförs i PLC:n har en hög grad av komplexitet. Det var få ingenjörer på avdelningen som förstod hur funktionerna i enheten fungerade, samtidigt som mängden funktioner var så omfattande att det skulle krävas en djupgående genomgång bara för att få en överblick över enhetens uppbyggnad. I kombination med ett begränsat dokumentationsunderlag, är slutsatsen att en fullständig migrering till PUMA skulle kräva en omfattande analys, samt ett noggrant testarbete.

6.5 Sammanfattande slutsatser

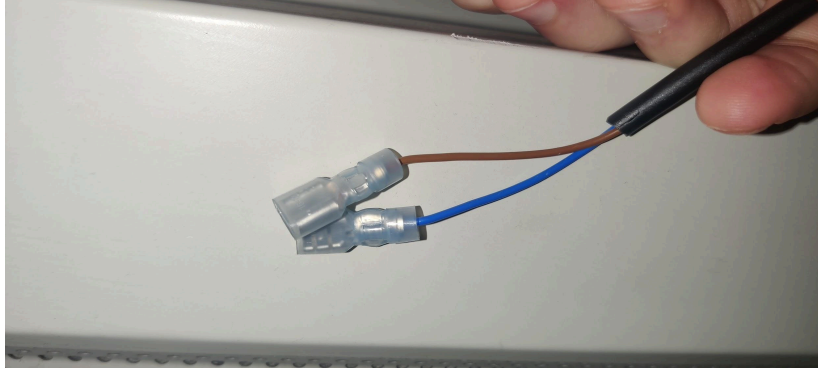
Målet med projektet var att modernisera det befintliga styrsystemet för en testtrigg som används för att utföra växellådtester hos Volvo GTT. De resultat som utförandet av projektet har lett fram till är uppskattade av Volvo GTT, samtidigt som gruppen är nöjd över det framtagna resultatet. Utförandet av projektet gick mestadels utefter GANTT-schemat som skapades i planeringsfasen, där den största skillnaden från schemat var att installation av Flexicart och PUMA gick smidigare än förväntat. När kommunikationen väl fungerade som den skulle, stod det endast kvar att testa styrsignaler till testtriggen för bekräfta att bägge dynos kunde svara på styrsignalerna, vilket de gjorde direkt.

7. Referenser

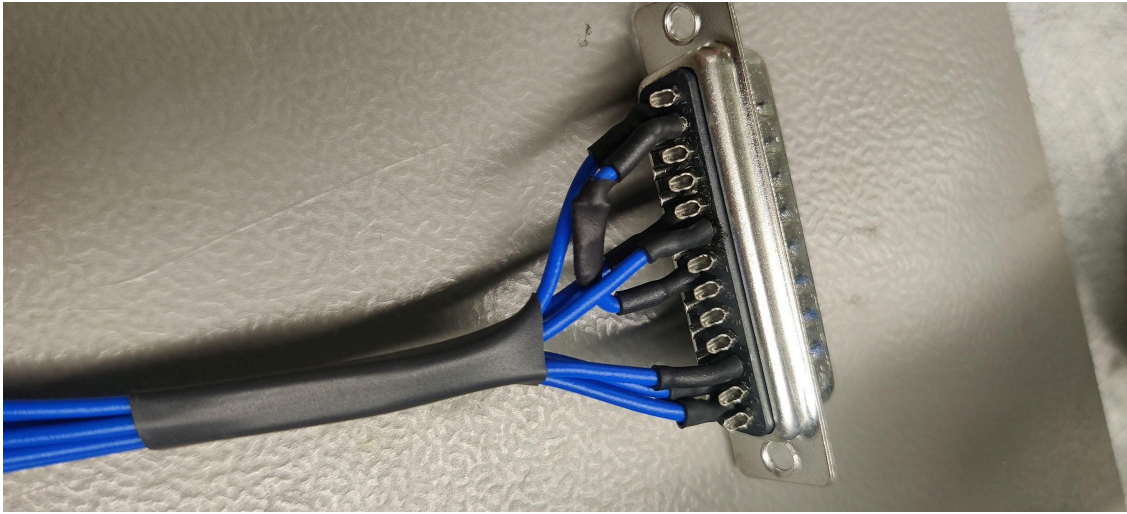
- [1] W. Bolton, Programmable Logic Controllers, 2. Uppl., Burlington, USA: Elsevier Inc, 2009. [Online]. Tillgänglig: <https://ebookcentral.proquest.com/lib/chalmers/detail.action?docID=535058>, (hämtad 07 maj, 2025)
- [2] “What is Modbus and How does it work? | Schneider Electric Sverige,” www.se.com, Mar. 19, 2013. <https://www.se.com/se/sv/faqs/FA168406/> (hämtad 21 maj, 2025).
- [3] “About AVL | AVL,” www.avl.com, Tillgänglig: <https://www.avl.com/en/about-avl> (hämtad 21 maj, 2025)
- [4] “AVL PUMA Open” avl.com. Tillgänglig: https://www.avl.com/html/static/ITS_ONLINE/docs/pdfs/K_Pruefstandsautomatisierung/Tabelleneuersicht_ITS_AutomationTestbed_E_2011_iP_2.pdf (hämtad 24 maj, 2025).
- [5] “EtherCAT -The Ethernet Fieldbus” Tillgänglig: https://www.beckhoff.com/media/downloads/information-media/etg_brochure_en.pdf (hämtad 24 maj, 2025)
- [6] “CS LT” Tillgänglig: <https://optris.com/products/infrared-thermometers-pyrometers/cs-series/cs-lt/> (hämtad 24 maj, 2025).

8. Bilagor

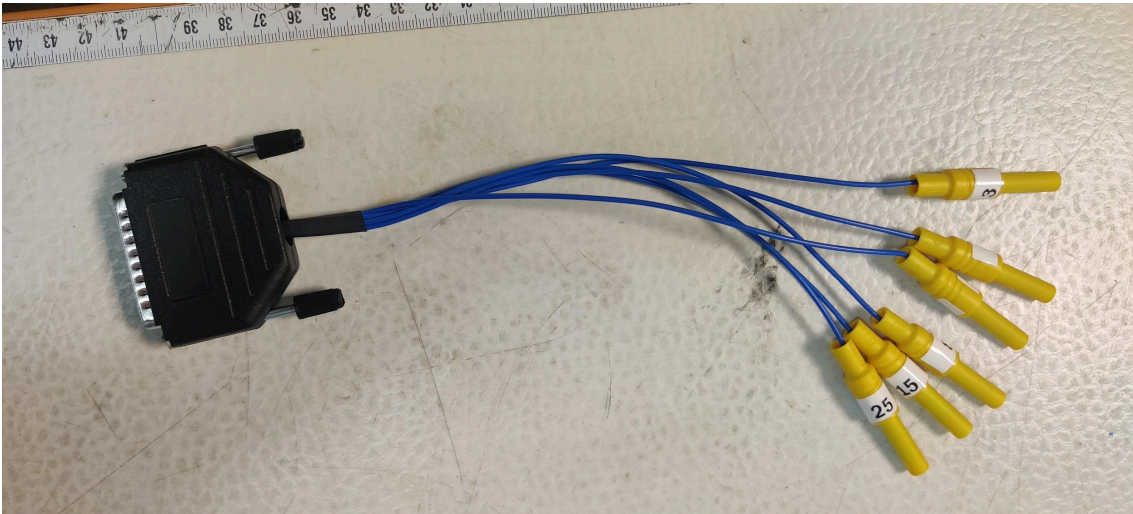
A. Kontakter för plus och minuspol till dörrsäkerhetsgivaren.



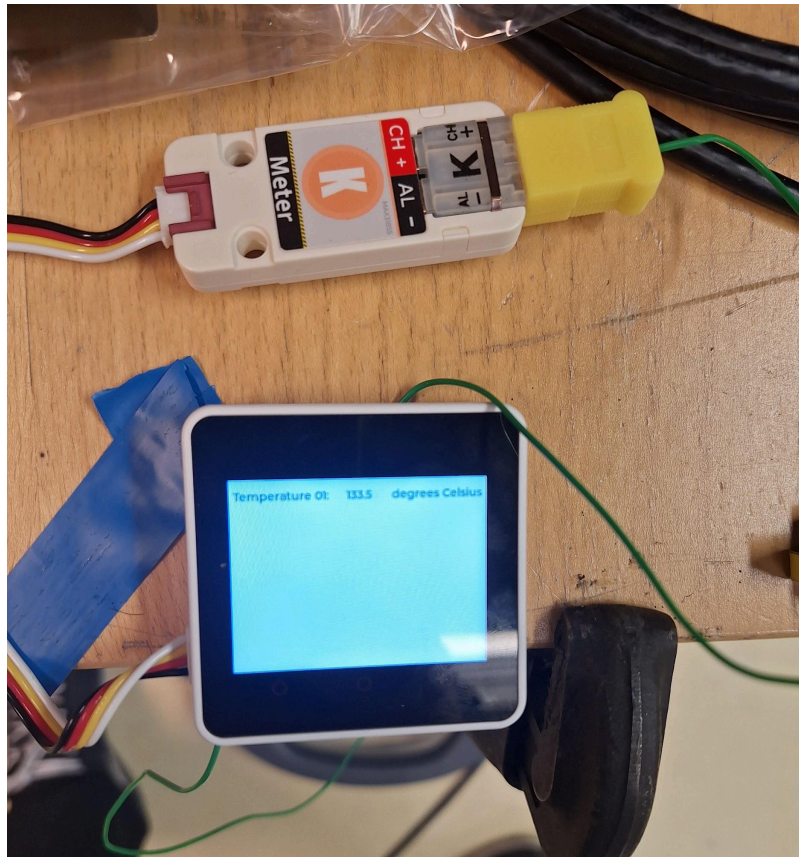
B. Insidan av D-Sub kontakt där gruppen lödde fast kablage



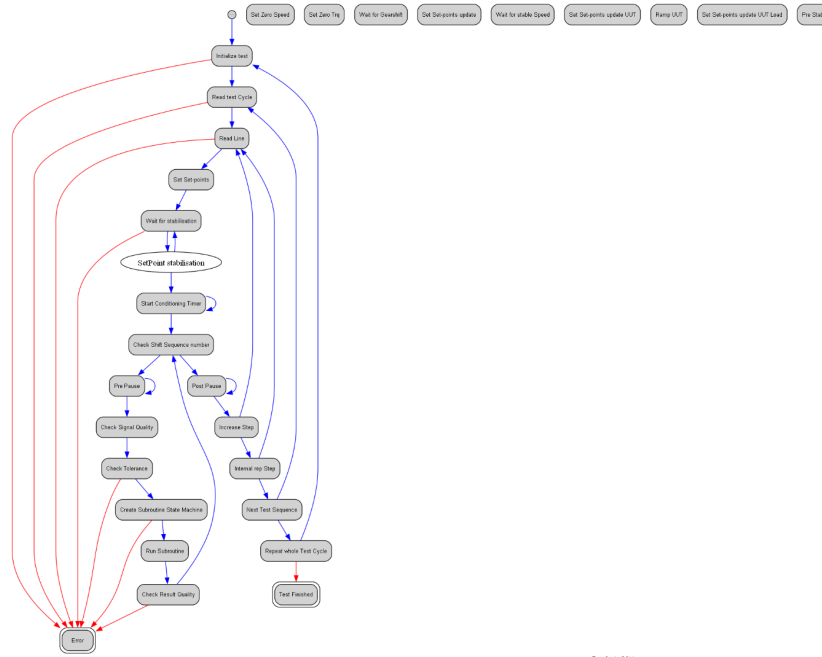
C. D-Sub anslutning med banankontakt för testning av kablage och signaler som anslöts till Flexicarten



D. M5Stack med termoelement-anlutning



E. Programflöde för exempelsekvens i Pythonprogrammet



G. Modbus-tabell för adresser i PUMA

Modbus (MBS)

Modbus signal tar från PLC

General

Connection type: TCP/IP

IP address: 0 . 0 . 0 . 0

IP port: 502

ZeroBasedRegisterAddresses

Error reactions

Line error reaction: Error message

Status channel:

Line error timeout: 1000 ms

Analog inputs | Analog outputs | Digital inputs | Digital outputs

No.	Register type	Register number	Device ID	Quantity	Frequency	Channel description	Length	Bit group	Byte order
1	Holding	30041	1	1	1	adrSetCtrlAirPressure	1	1111 1111 1111 1110	Motorola
2	Holding	30052	1	1	1	adrSetCtrlCond1	1	1111 1111 1111 1110	Motorola
3	Holding	30055	1	1	1	adrSetCtrlCond2	1	1111 1111 1111 1110	Motorola
4	Holding	30006	1	1	1	adrSetCtrlModeDynaFr	1	1111 1111 1111 1100	Motorola
5	Holding	30012	1	10	10	adrSetCtrlModeDynaFr	1	0000 0000 0000 0000	Motorola

Quantity Browser

Name	Description	Unit	Data type	Systemname
adrSetClutchSlip	Allowed clutch slip	%	Long	adrSetClutchSlip
adrSetCtrlAirPress	Compressed air - Control	-	Long	adrSetCtrlAirPressure
adrSetCtrlCond1	Conditioning 1 - Control	-	Long	adrSetCtrlCond1
adrSetCtrlCond2	Conditioning 2 - Control	-	Long	adrSetCtrlCond2
adrSetCtrlModeDynaFr	Input shaft (front dynamic)	-	Long	adrSetCtrlModeDynaFr
adrSetCtrlModeDynaRear	Output shaft (rear dynamic)	-	Long	adrSetCtrlModeDynaRear
adrSetCtrlOilCond1	Oil conditioning - Control 1	-	Long	adrSetCtrlOilCond1
adrSetCtrlOilPump	Oil conditioning - Control	-	Long	adrSetCtrlOilPump
adrSetCtrlPS	DC Supply - Control	-	Long	adrSetCtrlPS
adrSetCtrlRetarderCooler	Retarder - Control	-	Long	adrSetCtrlRetarderCooler
adrSetCurClutchState	UUT - State of Clutch	-	Long	adrSetCurClutchState
adrSetCurGbxState	Current Gearbox State	-	Long	adrSetCurGbxState
adrSetCurNeutralState	UUT - Neutral mode	-	Long	adrSetCurNeutralState
adrSetCurMaxPS	DC Supply - Limit	A	Long	adrSetCurMaxPS
adrSetCurSeqStep	External - Step in sequence	-	Long	adrSetCurSeqStep

75 items

OK Cancel

H. Modifierad USB-C till D-Sub kabel för användning av temperaturgivare i telefon/dator



I. Distribuerad I/O

