



Høgskulen på Vestlandet

Bacheloroppgave

ELE350

Predefinert informasjon

Startdato:	08-05-2023 09:00 CEST	Termin:	2023 VÅR
Sluttdato:	22-05-2023 14:00 CEST	Vurderingsform:	Norsk 6-trinns skala (A-F)
Eksamensform:	Bacheloroppgave		
Flowkode:	203 ELE350 1 O 2023 VÅR		
Intern sensor:	Olav Gerhard Haukenes Nygaard		

Deltaker

Naun:	Johannes Arnesen Eidsvik
Kandidatnr.:	221
HVL-id:	591452@hvl.no

Informasjon fra deltaker

Egenerklæring *: Ja
**Inneholder besvarelsen
konfidensielt
materiale?:** Nei
**Jeg bekrefter at jeg har
registrert
oppgavetittelen på
norsk og engelsk i
StudentWeb og vet at
denne vil stå på
vitnemålet mitt *:** Ja

Gruppe

Gruppenavn: BO23EB-31
Gruppenummer: 24
**Andre medlemmer i
gruppen:** Josef Nicolai Osland Hellesen-Heimset, Daniel Valestrand Klepsvik

Jeg godkjenner autalen om publisering av bacheloroppgaven min *

Ja

Er bacheloroppgaven skrevet som del av et større forskningsprosjekt ved HVL? *

Nei



Høgskulen
på Vestlandet

BACHELOROPPGAVE:

BO23EB-31

Sammenkobling av CIROS og TIA-Portal
som undervisningsverktøy

Daniel Klepsvik

Johannes Eidsvik

Josef Hellesen-Heimset

21.05.2023

Dokumentkontroll

<i>Rapportens tittel:</i> BO23EB-31 Sammenkobling av CIROS og TIA-Portal som undervisningsverktøy	<i>Dato/Versjon</i> 21.05.2023/1.0
	<i>Rapportnummer:</i> B023EB-31
<i>Forfatter(e):</i> Daniel Klepsvik Johannes Eidsvik Josef Hellesen-Heimset	<i>Studieretning:</i> AUTB20
	<i>Antall sider m/vedlegg</i> 92
<i>Høgskolens veileder:</i> Olav Gerhard Nygaard	<i>Gradering:</i> Åpen
<i>Eventuelle Merknader:</i> Vi tillater at oppgaven kan publiseres.	

<i>Oppdragsgiver:</i> Fagskulen på Vestlandet	<i>Oppdragsgivers referanse:</i>
<i>Oppdragsgivers kontaktperson(er) (inkludert kontakinformasjon):</i> Guttorm Lyngvær - Guttorm.Lyngver@vlfk.no	

Revisjon	Dato	Status	Utført av
0.1	22.03.2023	Strukturering av rapport	Josef
0.2	29.03.2023	Startet utfylling forskjellige punkt.	Johannes og Daniel
0.3	19.04.2023	Lagt inn flere delkapittel som setter beskriver kronologisk prosess av	Johannes, Daniel og Josef
1.0	20.05.2023	Endelig versjon av rapport	Daniel, Josef og Johannes

Forord

Denne rapporten utgjør den avsluttende delen av bacheloroppgaven som ble skrevet ved Høyskolen på Vestlandet i Bergen. Oppgaven tar for seg utviklingen av Festo CIROS og Siemens TIA-Portal som undervisningsverktøy for Fagskolen på Vestlandet. Arbeidet med bacheloroppgaven har vært svært lærerikt og krevende.

Vi ønsker å takke professor Olav Gerhard Nygaard fra institutt for datateknologi, elektroteknologi og realfag for å ha veiledet oss gjennom oppgaven. Vi setter pris på din interesse og tette oppfølging, som var avgjørende for oppgavens utførelse. Vi vil også takke Guttorm Lyngvær, lærer ved Fagskolen på Vestlandet og kontaktperson hos oppdragsgiver, for å ha opprettholdt god kommunikasjon og for å ha bidratt til at oppgaven var tydelig. Til slutt ønsker vi å takke alle andre som har bistått oss i arbeidet med oppgaven.

Sammendrag

Bacheloroppgavens hensikt er å utvikle et samspill mellom Siemens TIA-Portal og simuleringsprogrammet Festo CIROS for Fagskolen på Vestlandet på Nordnes. Fagskolen på Vestlandet eier en Festo-levert minifabrikk. Kostnadene ved å reparere den fysiske fabrikken er veldig høye, og det kreves at en ingeniør fra Tyskland kommer til Norge for å utføre reparasjoner. Derfor ønsker Fagskolen på Vestlandet å kunne simulere programmet virtuelt for å se hvordan det vil oppføre seg, og for å detektere eventuelle feil før koden lastes opp til den fysiske fabrikken. Dette vil gi studentene mulighet til å simulere minifabrikken og få en bedre forståelse av hvordan koden vil oppføre seg på den fysiske fabrikken. Dette vil også skape en sikkerhetsbarriere for å unngå feil på den fysiske fabrikken, hvis koden lastes opp til dens fysiske PLS.

Målsettingen for bacheloroppgaven er å programmere 3 moduler av fabrikken i Festo CIROS: transportbåndet «CP-L-LINEAR» [1], drillmodulen «CP-AM-iDRILL» [2] og den pneumatiske pressa «CP-AM-PRESS» [3]. De er allerede modellert i CIROS sitt bibliotek, men det kreves applikasjoner og programmering for å kunne koble dem sammen med TIA-Portalen. Det vil også bli utarbeidet lab-oppgaver med økende vanskelighetsgrad for å gi studentene en god forståelse i TIA-Portalen. Det er også utviklet brukerveiledninger for forskjellige operasjoner som vil gjøre det enklere for studenter og faglærere å lære seg prosessen.

Bacheloroppgaven har oppfylt målene og kravene fra oppdragsgiver og vi har utviklet et samspill mellom Siemens TIA-Portal og simuleringsprogrammet Festo CIROS for Fagskolen på Vestlandet på Nordnes. Videre er det utviklet lab-oppgaver og brukerveiledninger som vil gi studentene en god forståelse i TIA-Portalen.

Summary

The purpose of the bachelor's thesis is to develop an interface between Siemens TIA Portal and the simulation program Festo CIROS for the Vocational School of Western Norway at Nordnes. The Vocational School of Western Norway owns a Festo-supplied mini factory. The costs of repairing the physical factory are exceedingly high, and it requires an engineer from Germany to travel to Norway to perform the repairs. Therefore, the Vocational School of Western Norway wants to be able to simulate the program virtually, to see how it will behave and to detect any errors before uploading any code to the physical factory. This will provide students with the opportunity to simulate the mini factory and gain a better understanding of how the code will perform in the physical factory. It will also create a safety barrier to avoid errors in the physical factory if the code is uploaded to its physical Programmable Logic Controller (PLC).

The objective of the bachelor's thesis is to program 3 modules of the factory in Festo CIROS: the conveyor belt "CP-L-LINEAR" [1], the drilling module "CP-AM-iDRILL" [2], and the pneumatic press "CP-AM-PRESS" [3]. These were already modeled in CIROS's library, but further programming was required to connect them to the TIA Portal. In addition, lab exercises with increasing levels of difficulty will be developed to provide students with a good understanding of TIA Portal. User guides for various operations have also been developed to make the learning process easier for students and instructors.

The bachelor's thesis has met the objectives and requirements of the client, and we have developed an interface between the Siemens TIA Portal and the simulation program Festo CIROS for the Vocational School of Western Norway at Nordnes. Furthermore, lab exercises and user guides were developed and tested to provide students with a good understanding of the TIA Portal.

Innholdsfortegnelse

DOKUMENTKONTROLL	2
FORORD	3
SAMMENDRAG	4
SUMMARY	5
1 INNLEDNING	9
1.1 OPPDRAGSGIVER	9
1.1.1 Oppgavens formål.....	9
1.2 CIROS	10
1.3 TIA-PORTAL	10
1.4 DIGITAL TVILLING	10
1.5 PROBLEMSTILLING	11
1.6 HOVEDIDÉ FOR LØSNINGSFORSLAG	11
2 KRAVSPESIFIKASJON	12
2.1 LISTE OVER KRAVSPESIFIKASJON	12
2.1.1 Ønskelige spesifikasjoner	12
3 ANALYSE AV PROBLEMET	12
3.1 UTFORMING AV MULIGE GRENSESNIITT MELLOM PLS-IDE OG CIROS	14
3.1.1 Fysisk PLS kommunikasjon via EasyPort og EzOPC eller OPC UA	14
3.1.2 CODESYS og EzOPC.....	16
3.1.3 PLCSIM Advanced og TIA-portalen	17
3.2 VALGT LØSNING.....	17
4 REALISERING AV VALGT LØSNING	18
4.1 FORARBEID	18
4.1.1 Festo LX.....	18
4.1.2 Forberedende pedagogisk arbeid.....	18
4.1.3 Nedlastning og installasjonsveiledninger.....	18
4.1.4 Valg av moduler.....	19
4.2 INNLEDENDE ARBEID I CIROS OG TIA	20
4.2.1 Høynivåspråk.....	21
4.2.2 Brukerveiledninger.....	21
4.3 FERDIGUTVIKLEDE LAB-OPPGAVER	22
4.3.1 Lab 1 - CP Lab Linear (CP-L-LINEAR).....	22
4.3.2 Lab 2 - Press (CP-AM-PRESS).....	24

4.3.3	Lab 3 - iDrilling (CP-AM-iDRILL).....	25
4.4	LAB 4 – HIGH BAY STORAGE UNIT (CP-F-ASRS32-P)	28
5	TESTING.....	31
5.1	TEST AV LAB 1	31
5.2	TEST AV LAB 2	33
5.3	TEST AV LAB 3	34
5.4	REVISJON AV LAB-OPPGAVER OG NEDLASTNINGSGUIDE	35
6	DISKUSJON.....	35
6.1	TEKNISK GRENSESNIITT	35
6.2	LAB-OPPGAVER.....	36
7	KONKLUSJON OG VIDERE ARBEID.....	37
7.1	TEKNISK GRENSESNIITT	37
7.2	LAB-OPPGAVER.....	37
7.3	VIDERE ARBEID.....	37
	REFERANSER.....	39

APPENDIKS A LISTER 41

A.1	FORKORTELSER OG ORDFORKLARINGER	41
A.2	FIGURLISTE	FEIL! BOKMERKE ER IKKE DEFINERT.

APPENDIKS B PROSJEKTLEDELSE OG STYRING 43

B.1	PROSJEKTORGANISASJON	43
B.2	PROSJEKTFORM	43
B.3	FREMDRIFTSPLAN	44

APPENDIKS C BRUKERDOKUMENTASJON 45

C.1	NEDLASTNINGSGUIDE CIROS	45
C.2	NEDLASTNINGSGUIDE TIA-PORTALEN	59
C.3	GUIDE FOR Å LEGGE TIL MODULER I CIROS	61
C.4	GUIDE FOR Å KOBLE SAMMEN TIA-PORTALEN OG CIROS	63

APPENDIKS D LAB OPPGAVER 68

D.1	LAB 1	68
D.2	LAB 2	74
D.3	LAB 3	79

APPENDIKS E PROGRAMKODE 85

E.1	LAB 1	85
E.2	LAB 2	86
E.3	LAB 3	88

1 Innledning

1.1 Oppdragsgiver

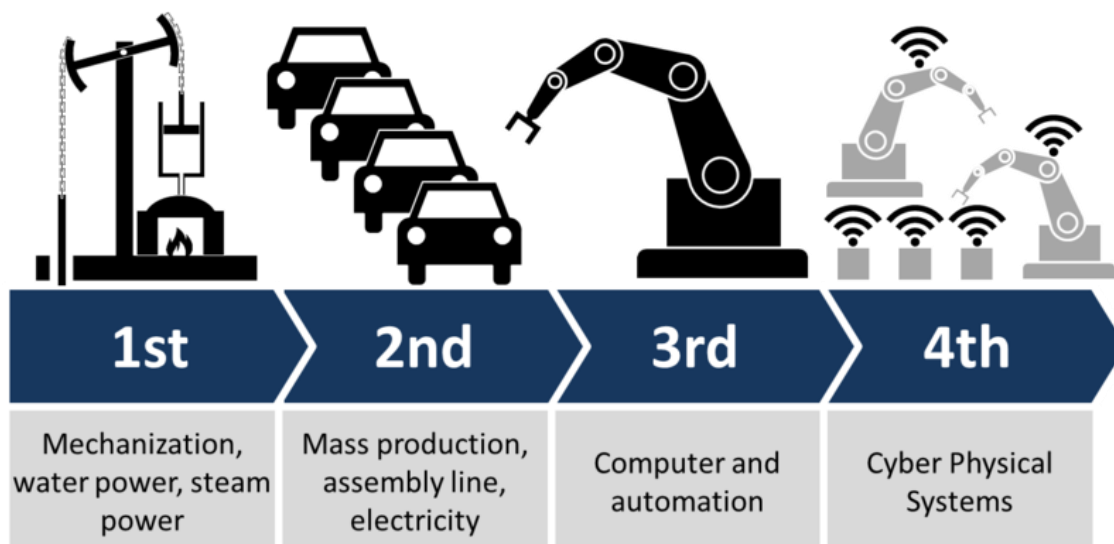


Figur 1: Logo Fagskolen på Vestlandet

Fagskolen på Vestlandet er blant de største i landet med omtrent 1900 studenter og åtte ulike studiesteder i Vestland fylke. Fagskolen tilbyr studier innen tekniske fag, maritime og marinefag, helsefag, petroleumsfag, jordbruksfag, arboristfag og lokal matkultur. Flere av studiene kan fullføres i kombinasjon med jobb [4]. Oppdragsgivers kontaktpersoner Guttorm Lyngsvær og Bjørnar Seim underviser begge i tekniske fag ved Fagskolen på Vestlandet.

1.1.1 Oppgavens formål

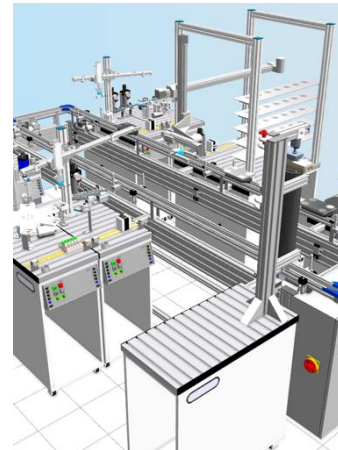
Oppdragsgivers ønske er å utvikle en digital tvilling som kan integreres med TIA-Portalen og CIROS. Digital tvilling er et stadig mer relevant verktøy i industrien i forbindelse med den forestående fjerde industrielle revolusjonen. «Den fjerde industrielle revolusjon er en beskrivelse av en økt digitalisering av industrielle systemer der maskiner, utstyr, og mennesker samarbeider i et integrert digitalt nettverk.» [5]. Med bakgrunn i dette ønsker oppdragsgiver å modernisere opplæringen av studenter.



Figur 2: Den fjerde industrielle revolusjonen. Kilde:[6]

1.2 CIROS

CIROS er et virtuelt simuleringsprogram utviklet av Festo. Programmet gir deg muligheten til å modellere fabrikkkomponenter fra bunn, og kommer med et stort bibliotek av premodellerte komponenter. Noen av de er modellert fra minifabrikken Fagskolen eier, og er derfor svært relevante. Festo er et stort tysk selskap som er blant de ledende leverandørene innen automasjonsindustrien, og som legger stor vekt på opplæring og bruk av sine produkter [7]. Derfor har de utviklet CIROS som er kompatibelt med flere programmer, inkludert Siemens PLS og TIA-Portal. CIROS kan brukes som et enkeltstående simuleringsprogram eller utvides til å fungere som en digital tvilling.



Figur 3: Eksempel på MPS-fabrik i CIROS

1.3 TIA-Portal

TIA-Portalen er programmeringsverktøy utviklet av Siemens for programmering av PLS-ene de leverer. Siemens er Europas største industrielle produksjonsselskap [8]. TIA-Portalen lar deg simulere programmene du lager med en virtuell PLS og laste de ned til fysiske PLS-er.

1.4 Digital tvilling

«En digital tvilling er en virtuell modell designet for å nøyaktig reflektere et fysisk objekt.» [9]. Ved å montere sensorer på det fysiske objektet kan dataene kontinuerlig mates inn i den digitale tvillingen. Den digitale tvillingen bruker dataen til å forutse fremtidige problemer, som slitasje og eventuelle feil.



Figur 4: Digital tvilling av olje rigg. Kilde:[10]

1.5 Problemstilling

Opprinnelig ble prosjektet beskrevet som utvikling av en digital tvilling ved hjelp av CIROS og TIA-Portalen. Det ble imidlertid klart at dette ikke var riktig og prosjektet ble justert til å modellere digitale modeller av minifabrikkmoduler. Dette er fordi oppdragsgiver ikke ønsket å speile simuleringene i sanntid. Fagskolen har investert i en minifabrikk for å gi studentene bedre opplæring i automasjonsfag. Målet er å modellere og programmere flere av modulene i CIROS og samkjøre dem med TIA-Portalen for å gi studentene muligheten til å se virtuelt hva koden deres gjør. I tillegg skal det utvikles lab-oppgaver for modellene, samt detaljerte brukerveiledninger for faglærere slik at de kan utvikle egne undervisningsoppgaver.

CIROS har imidlertid ikke hovedfunksjon å samkjøres med TIA-Portalen, og dette vil utgjøre en utfordring for mer avanserte moduler på grunn av komplekse kommunikasjonsprotokoller som ikke er egnet for TIA-Portalen. Videre har studentene begrensninger innen matematikk og andre teoretiske fag, og det er derfor viktig å velge moduler som passer deres nivå. Målet er å utvikle lab-oppgaver med passende vanskelighetsgrad som gir studentene en god læringsopplevelse. Det vil dermed ikke være hensiktsmessig å velge moduler som krever avanserte utregninger for styring.

1.6 Hovedidé for løsningsforslag

Prosjektets hovedmål er å samkjøre TIA-Portalen og CIROS for å oppfylle kravene til Fagskolen på Vestlandet. En detaljert brukerveiledning er nødvendig for å oppnå dette. Først velger vi hvilke moduler fra fabrikken som skal modelleres. Det er avgjørende å velge moduler som ikke er for utfordrende for studentene, men samtidig gir ønsket læringsutbytte. En brukerveiledning for installasjon av programvarene vil også bli levert. Faglærerne på fagskolen har en egen protokoll å følge ved installering av programvare for å ivareta sikkerheten på enhetene deres. De ønsker derfor at vi lager en guide for studentene, som ikke trenger å følge disse protokollene.

Fagskolen krever 3-4 lab-oppgaver som passer til studentenes kompetansenivå og kan brukes på forskjellige stadier i utdanningsløpet. Ingeniørutdanningen vår inkluderer ikke pedagogikkfag, og det forventes ikke at lab-oppgavene er feilfrie sett fra et pedagogisk perspektiv. De vil hovedsakelig brukes som inspirasjon og guide for videreutvikling av nye lab-oppgaver laget av Fagskolen. Vi vil imidlertid gjøre en stor innsats for å sikre at lab-oppgavene kan gjennomføres på den måten de er laget.

2 Kravspesifikasjon

I starten av dette prosjektet ble kravspesifikasjonene drøftet med oppdragsgiver, der hovedfokuset var å bevise samspeillet mellom CIROS og TIA-Portalen for bruk i undervisningen. I tillegg ble det drøftet noen ekstra spesifikasjoner som kan utføres dersom det ble tid til dette før ferdigstilling av prosjektet.

2.1 Liste over kravspesifikasjon

- Få det digitale simuleringsmiljøet CIROS til å simulere mot TIA-Portalen
- Lage utfyllende brukerveiledninger for oppsett av CIROS mot TIA-Portalen og eventuelt brukerveiledning for bruk av CIROS ved utførelse av lab-oppgaver.
- Anvend modeller i CIROS og programmer funksjonaliteten til 4 moduler. De 4 modellene speiler noen av modulene i den fysiske fabrikken slik at oppdragsgiver kan benytte disse i undervisningen.
- Lage 4 lab-oppgaver om PLS-programmering som benytter CIROS for testing av PLS-programmet studentene har laget. Oppgavene skal også være relevant for pensum til elevene

2.1.1 Ønskelige spesifikasjoner

- Teste mulighet for å laste opp programkode til en fysisk PLS kan brukes til CIROS simuleringer.
- Implementering av BUS-løsning slik at modulene kan samkjøre i CIROS som en hel eller delvis fabrikk.
- Integrering av Manufacturing Execution System (MES) i et system som inneholder noen fabrikkmoduler.

3 Analyse av problemet

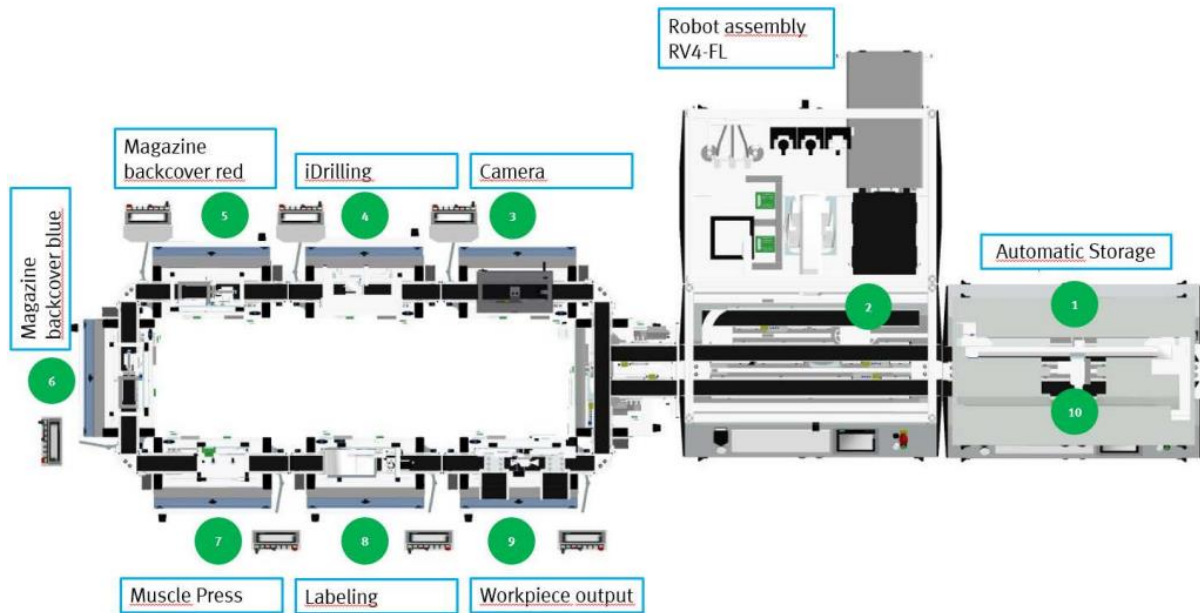
CIROS er et digitalt, tredimensjonalt simuleringsprogram for industrimaskiner laget av Festo. Hovedfunksjon til CIROS i vår oppgave er å simulere fysiske komponenter, sensorer og PLS-er internt. Problemet med CIROS er at det ikke er mulig å skrive PLS-program internt i programvaren. Dette har CIROS løst ved å lage en funksjon for å motta I/O fra en ekstern PLS, dette kan gjøres på et par forskjellige måter. Første metoden er å koble til en fysisk PLS til en datamaskin via EasyPort-enhet og USB-inngang. Deretter kan du skrive I/O til CIROS fra den fysiske PLS-en [11], dette blir drøftet i 3.1.1.

Neste metode for å oppnå dette samspillet er ved å benytte Open Platform Communications (OPC) i et utviklingsmiljø som CODESYS via OPC Unified Platform (UA), eller EzOPC som er Festos egen versjon av OPC-programvaren [12]. CODESYS er et integrert utviklingsmiljø (IDE) for programmering av kontrollapplikasjoner og er basert på industristandarden IEC 61131-3 [13]. OPC UA er en kommunikasjonsstandard som gjør det mulig for utstyr og systemer fra forskjellige leverandører å kommunisere. UA noterer at det kan operere på flere operativsystem enn bare Windows [14]. I vårt tilfelle er dette kommunikasjon mellom CIROS og CODESYS.

Den siste og mest brukervennlige metoden er å benytte Siemens PLCSIM Advanced. Med PLCSIM Advanced kan vi simulere en PLS (soft PLC) på datamaskinen vår der vi har muligheten til å skrive kode selv. Dermed kan PLCSIM Advanced snakke direkte med CIROS. Med denne løsningen håndterer CIROS oversettelsen av I/O-verdiene og vi slipper da å benytte en OPC-server for oppsettet. Fagskolen har lisenser på PLCSIM Advanced og de bruker TIA-Portalen som undervisningsverktøy for studentene sine. Studentene har lite erfaring med TIA-portalen, det kan derfor være utfordrende å bruke programvarene i starten. Derfor føler oppdragsgiver det er nødvendig å utvikle en brukerveiledning for oppsett av programvaren, hvis denne løsningen skal brukes.

Vi ønsker å finne den metoden som gir mest læringsutbytte til studentene på fagskolen, derfor har vi undersøkt læreplanen deres. «Robotteknologi og digital produksjon er et studium utviklet for å møte industriens behov innen elektromekanisk produksjonsprosesser. Dette innebærer en flerfaglig kompetanse innenfor maskinfag, elektrofag og informasjonsteknologi. Denne tverrfagligheten skal gi studenten teoretisk og praktisk systemforståelse av produksjonsprosesser. Studenten skal etter endt utdanning kunne lede utvikling av nye produksjonsprosesser, automatisere produksjonsprosesser og vedlikeholde produksjonsprosesser og maskiner» [15]. PLS er sammenkoblingsleddet mellom maskin, elektro og informasjonsteknologi, og er veldig relevant for utdanningen.

Tidligere ble det diskutert muligheten for å bruke en fysisk PLS koblet sammen med simuleringer i CIROS. Løsningen tilbyr arbeid med fysiske komponenter som fagskolen mener er viktig for å forberede studentene på arbeidslivet. De vil at vi skal se nærmere på denne løsningen både med EasyPort-kommunikasjon og/eller OPC UA-kommunikasjon.



Figur 5: Illustrasjon oppsett av fabrikken. Kilde: [16]

Vi har også et ønske om å lage en fullstendig kopi av den fysiske fabrikken i CIROS. Vi vil programmere funksjonaliteten til modulene for så å koble de sammen med den interne BUS-løsningen i CIROS. Videre kan vi implementert et MES slik at det blir en fullverdig kopi av den fysiske fabrikken.

3.1 Utforming av mulige grensesnitt mellom PLS-IDE og CIROS

Vi har diskutert de forskjellige løsningene for å oppnå sammenkoblingen av CIROS og TIA-Portalen, eller andre PLS-programmeringsverktøy. I dette kapitlet vil vi gå dypere inn på hvordan disse sammenkoblingene kan oppnås, deres egnede bruksområder, samt fordeler og ulemper ved de ulike løsningene.



Figur 6: Bilde av EasyPort enhet, Kilde: [16]

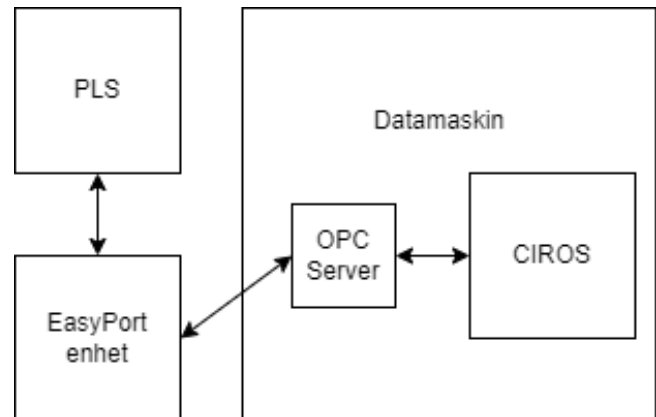
3.1.1 Fysisk PLS kommunikasjon via EasyPort og EzOPC eller OPC UA

Et alternativ for å kjøre eksternt PLS-kode mot CIROS er å laste ned koden til en eksternt fysisk PLS. Deretter kobles denne til datamaskinen via en EasyPort-enhet (Figur 6). Ifølge Festo skal de fleste PLS-ene kunne benyttes med EasyPort. EasyPort benytter USB-tilkobling til datamaskinen og en 24-pin SysLink socket for tilkobling til PLS-en. Datamaskinen kjører en simulering i CIROS og kommuniserer med PLS-en via EasyPort. På datamaskinen oversettes signalet fra EasyPort-enheten over til leselig data for CIROS, via en OPC-server. Festo har utviklet en OPC-server kalt EzOPC, og denne OPC-serveren har predefinerte

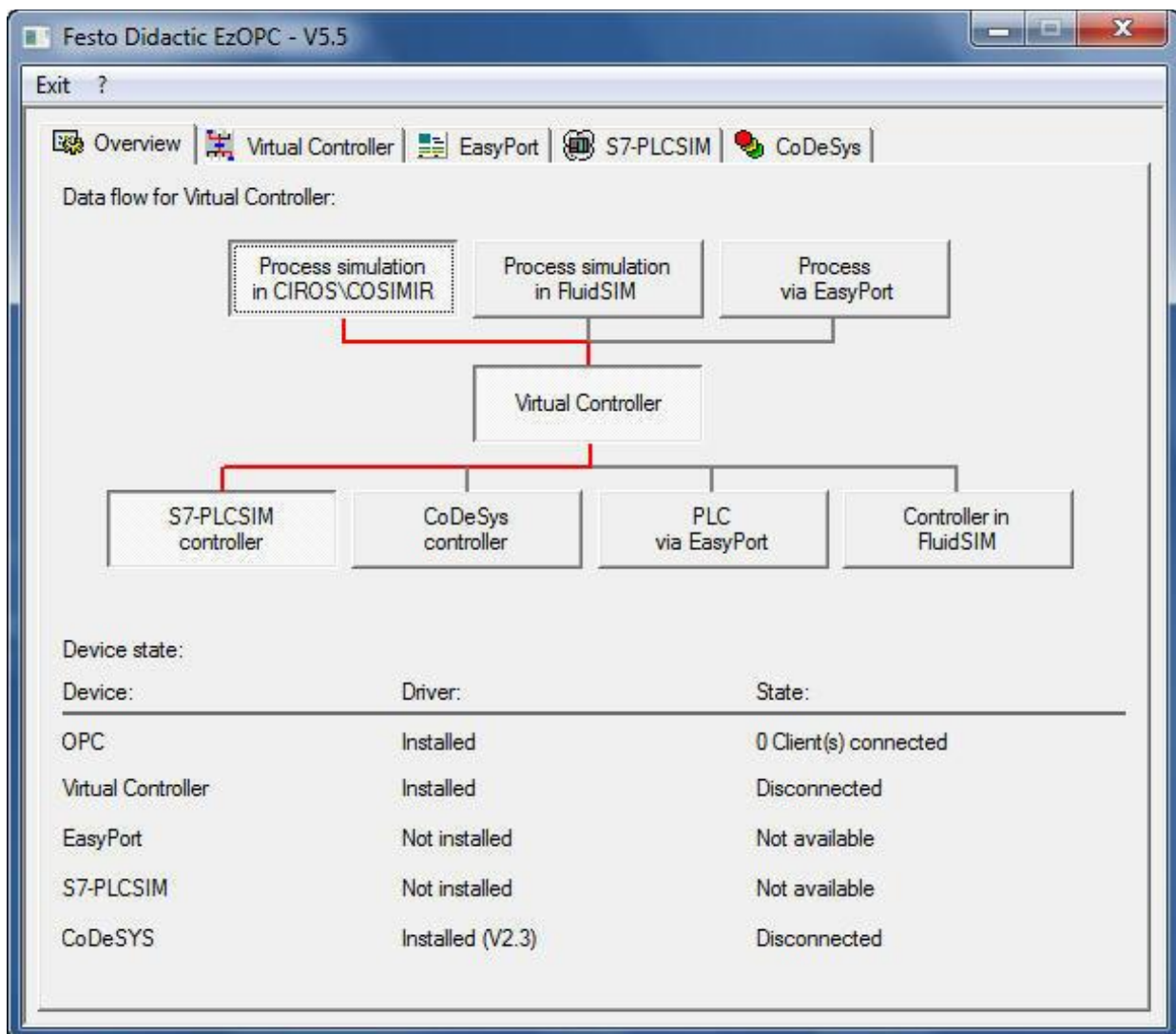
parameter for oversetting av data inn og ut. Parameter blir valgt via et brukergrensesnitt, og illustrert i Figur 8 ser du brukergrensesnittet med parameter «S7-PLCSIM controller» til

«Process simulation in CIROS\COSIMIR». I tilfellet diskutert i dette delkapittelet vil parameterne være «PLC Via EasyPort» til «Process simulation in CIROS\COSIMIR».

Denne kommunikasjonen kan også oppnås via en egendefinert OPC UA-server. I dette tilfellet er det unødvendig da Festo sin løsning er bedre.



Figur 7: Blokkdiagram av sammenkobling med fysisk PLS



Figur 8: Skjerm bilde av EzOPC oversikt tab. Kilde: [17]

EzOPC må konfigureres basert på I/O fra PLS-en, men dette blir ikke diskutert i denne rapporten.

Fordeler

Studentene får praktisk erfaring med en fysisk PLS.

Relativt enkel konfigurasjon av EzOPC.

Mer stabil kommunikasjon enn ved en simulert kontrollenhet.

Enklere å teste koden på simulator før faktisk modell i fysisk fabrikk.

Ulemper

Må ha tilgang til fysisk PLS og EasyPort enhet

Mer kostbart oppsett

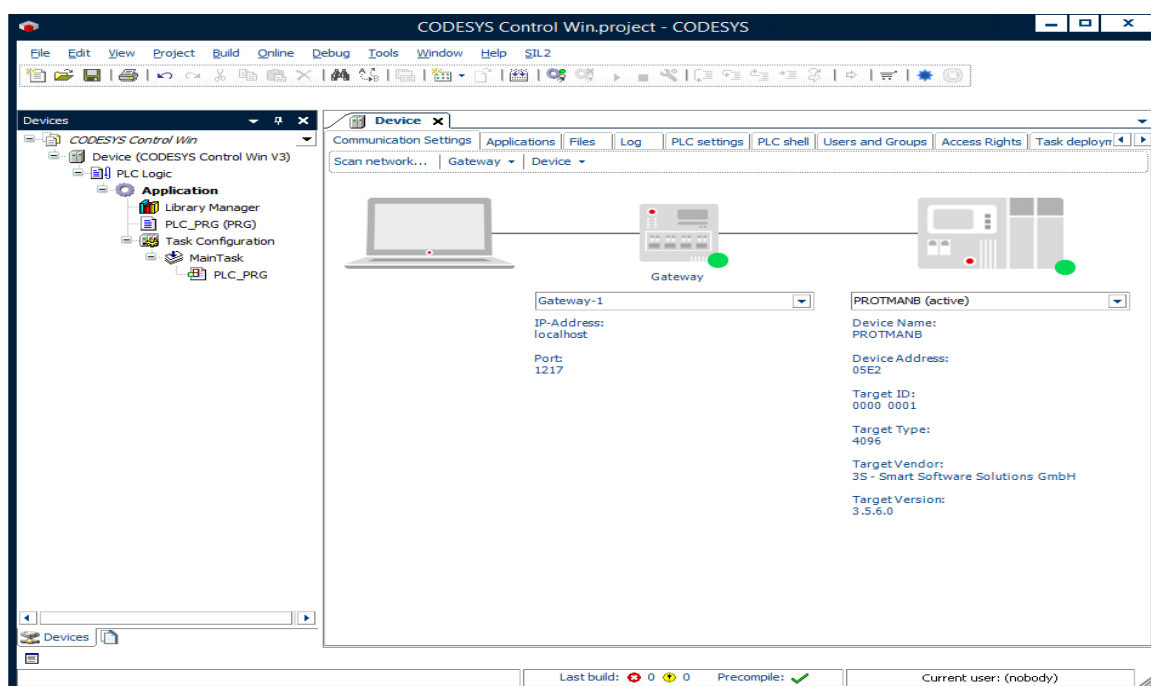
Begrenset av antall PLS-er for hvor mange studenter kan bruke oppsettet samtidig

Ikke mulighet for å utføres hjemme på egenhånd av studentene

3.1.2 CODESYS og EzOPC

Samspillet med CIROS kan oppnås med flere PLS IDE-er enn bare TIA-Portalen. CODESYS Control for Windows er en soft PLS styringsenhet som Festo har integrert i EzOPC.

CODESYS-enheten får PLS kode fra CODESYS Development System som er CODESYS PLS-IDE. OPC-serveren mottar da denne koden fra styringsenheten og oversettes til kode CIROS kan lese. Disse parameterne vil i Figur 8 være «CODESYS Controller» til «Process simulation in CIROS/COSIMIR».

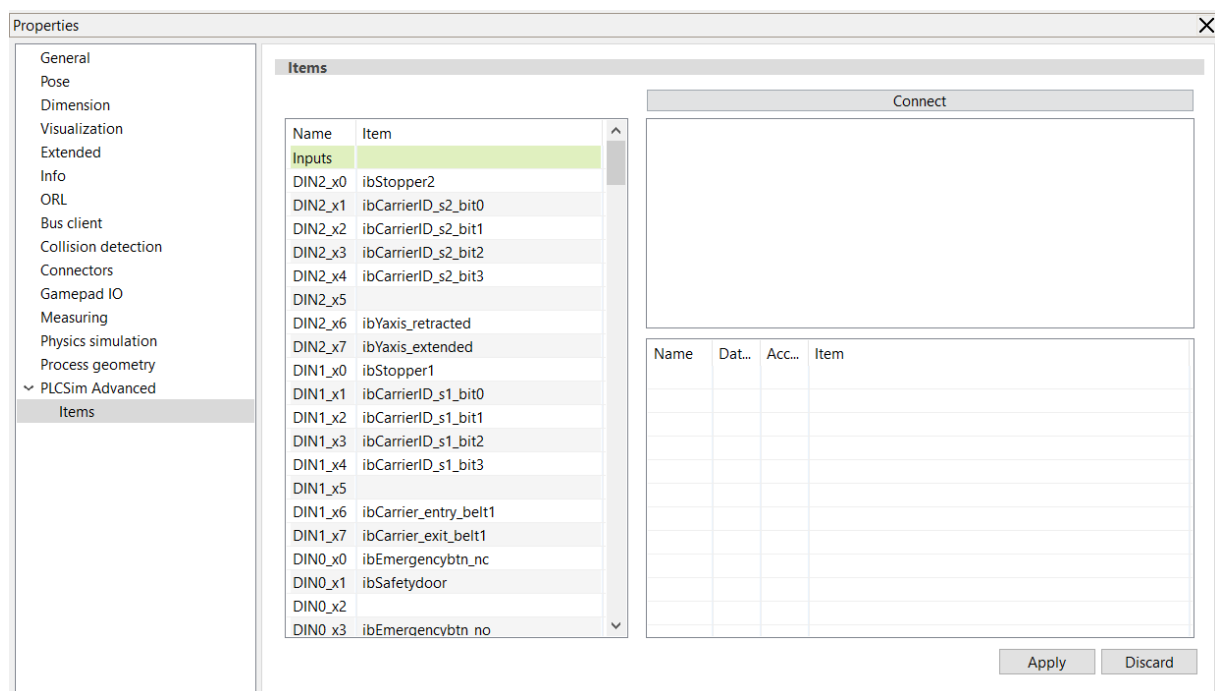


Figur 9: Eksempel brukergrensesnitt CODESYS Control Windows. Kilde: [18]

Denne kommunikasjonen kan også oppnås for TIA-Portalen via EzOPC, men som vi skal se i 3.1.3 er det en enklere og mer brukervennlig måte å oppnå denne kommunikasjonen for TIA-Portalen. Siste steg i sammenkoblingen er å definere grensesnittet mellom I/O fra CODESYS i CIROS, se Figur 10 for eksempel på dette utført i grensesnitt mellom PLCSIM Advanced og CIROS.

3.1.3 PLCSIM Advanced og TIA-portalen

Det siste grensesnittet vi skal diskutere er kommunikasjon mellom CIROS og TIA-Portalen ved hjelp av PLCSIM Advanced. PLCSIM Advanced er et verktøy som simulerer en eller flere PLS-er så nøyaktig som mulig. Det har et kraftig Application Programming Interface (API) som gir deg muligheten til å simulere med andre program både lokalt på PC, og over en virtuell Ethernet-adapter med andre enheter. Programmet kan simulere opp mot 16 forskjellige PLS-er til enhver tid, men du kan kun simulere PLS-er av typen S7-1500 og ET 200SP. Programmet er oppdatert flere ganger siden det ble utviklet og er nå på versjon 5, her fokuserer vi på versjon 2 ettersom det er den versjonen oppdragsgiver har lisenser på.



Figur 10: Eksempel grensesnitt til TIA i CIROS

3.2 Valgt løsning

Vi har diskutert flere grensesnitt for kommunikasjon mellom CIROS og TIA-Portalen. Vi har derimot ikke muligheten til å velge hvilket grensesnitt vi skal bruke selv. Fagskolen på Vestlandet har satt bruken av PLCSIM Advanced 2.0 som et krav for prosjektet ettersom de allerede eier lisenser for programmet.

4 Realisering av valgt løsning

4.1 Forarbeid

Arbeidet startet med å få en oversikt over den fysiske fabrikken, og vurdere hvilke moduler som ville vært interessante å modellere i CIROS. Ettersom det tok litt tid å få tilgang til fagskolens nettverk og lisenser som kreves for å bruke nødvendige programmer, fortsatte vi med å ta kurs på internett gjennom opplæringsportalen Festo LX. Når vi fikk tilgang til programmene arbeidet vi fortløpende med å lage instruksjoner for installering og oppsett. Etter flere møter med oppdragsgiver bestemte vi oss for hvilke moduler som ville være mest relevante å modellere.

4.1.1 Festo LX

Festo LX er en nettportal studenter benytter for å få tilgang til læringsstoff og en tilpasset læringsopplevelse [19]. Nettportalen inneholder mange nettkurs som varierer i lengde og vanskelighetsgrad, og som omhandler Festo sine programmer, produkter samt generelle emner.



Figur 11: Festo LX logo. Kilde: [19]

4.1.2 Forberedende pedagogisk arbeid

Oppgaven krever at vi lager lab-oppgaver. Som studenter har vi derimot ingen pedagogisk erfaring. Vi begynte derfor med å lese eldre lab-oppgaver gitt til andre studenter på fagskolen. Deretter satt vi oss inn i læreplanen deres. «Nærmere 60% av fagskolestudentene er 26 år eller eldre, og hele 27% er over 35 år.» [20]. Mange av studentene er i arbeid ved siden av deltidsstudier. Dette betyr at det meste av arbeidet skjer ved fjernundervisning. Det er også mulig å gi lab-oppgaver som skal gjennomføres digitalt. Flere studenter gjennomfører også studiet på fulltid. Lab-oppgavene er derfor utarbeidet slik at de kan gjennomføres både hjemme og på skolen. Studentene på fagskolen har som oftest ikke mer formell utdanning enn yrkesutdanning på videregående opplæringsnivå, og har ikke det samme teoretiske nivået som på en bachelorutdanning. Derfor var det viktig at lab-oppgavene ikke hadde høyt teoretisk nivå, men heller gav mer yrkesfaglig forståelse.

4.1.3 Nedlastning og installasjonsveiledninger

Detaljerte nedlasting- og installasjonsveiledninger var et av kravene fra fagskolen. Vi brukte derfor god tid på denne prosessen for å dokumentere stegene våre, samt utfordringer

underveis. Studentene opplevde noen av de samme utfordringene under testing av lab-oppgavene, og dokumentasjonen gjort i forkant ble derfor viktig.



Figur 12: Feilmelding møtt på i installasjon, skjermdump av Windows PC



Figur 13: Filene etter problem er fikset, skjermdump av Windows pc

4.1.4 Valg av moduler

Når vi bestemte oss for hvilke modeller vi ville utvikle som lab-oppgaver fulgte vi fagskolen sine retningslinjer. Modellene burde være de samme som de fabrikkmodulene fagskolen eier fysisk, det burde være ulikt nivå på kompleksiteten deres, og de må være relevante for pensumet til studentene. I regi med oppdragsgiver kom vi fram til at de 4 listet i tabell 1 var optimale.

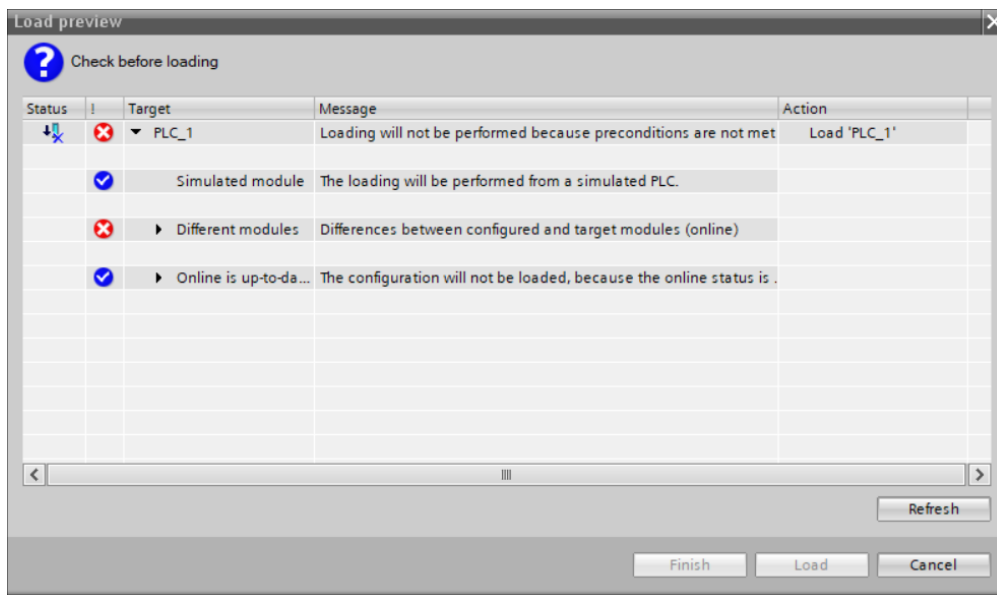
Navn på modul	Kort beskrivelse	Lenke til datablad
CP-L-LINEAR	Enkelt transportbånd	CP linear
CP-AM-PRESS	En pneumatisk presse for å presse sammen mobildeksel	CP-AM Press
CP-AM-iDRILL	Drill som applikasjonsmodul plassert oppå transportbåndet	CP-AM-iDrill
CP-F-ASRS32-P	Lager for mobildeleer, plassert på hyller og hentet av en kartetisk robot [21]	CP-F-ASRS32-P

Tabell 1: oversikt av CIROS moduler benyttet

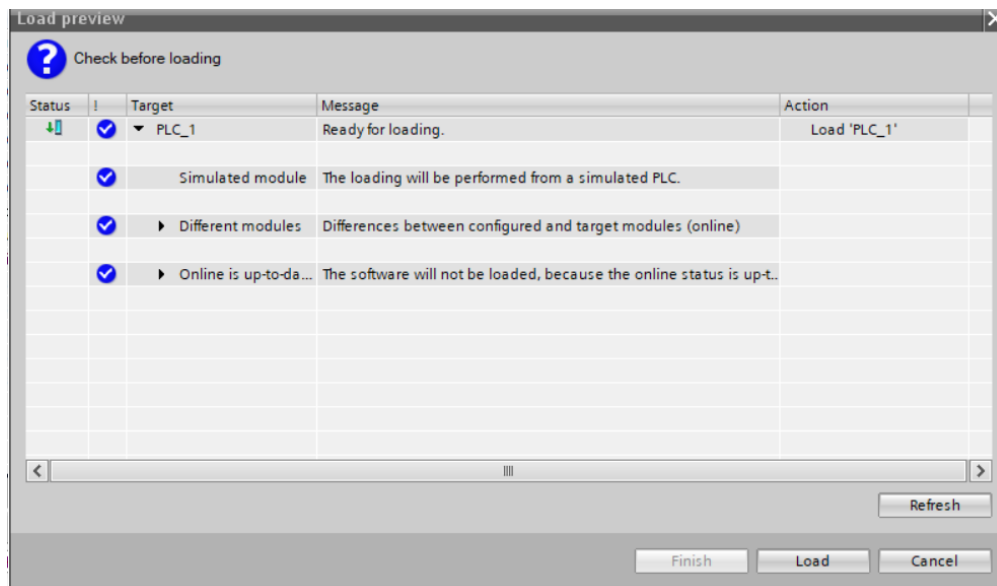
Etter at funksjonaliteten i modellene ble programmert fikk vi bekreftet at modellene hadde passende vanskelighetsgrad for studentene, og at de var relevante for pensumet deres. Vi begynte derfor å utvikle lab-øvelsene sentrert rundt modellene.

4.2 Innledende arbeid i CIROS og TIA

For at programmene skal kommunisere må det skapes en bro mellom dem. Dette gjøres ved bruk av PLCSIM Advanced [22], som er et tilleggsprogram til TIA-portalen. Programmet simulerer den fysiske koblingen mellom PLS-en og modellen i CIROS. Når programmene kommuniserer blir det sendt digitale signaler frem og tilbake, så det er mulig å endre verdier fra begge sider. Vi valgte å fokusere mye på dette i forkant, da vi mente det var et viktig punkt. Selv om det var relativt enkelt å lage den første koblingen opplevde vi PLCSIM Advanced 2.0 som ustabil, og opplevde ved flere tilfeller at programmet stoppet. Etter gjentatte forsøk klarte vi å lage en protokoll som medførte at programmet ble mer stabilt.



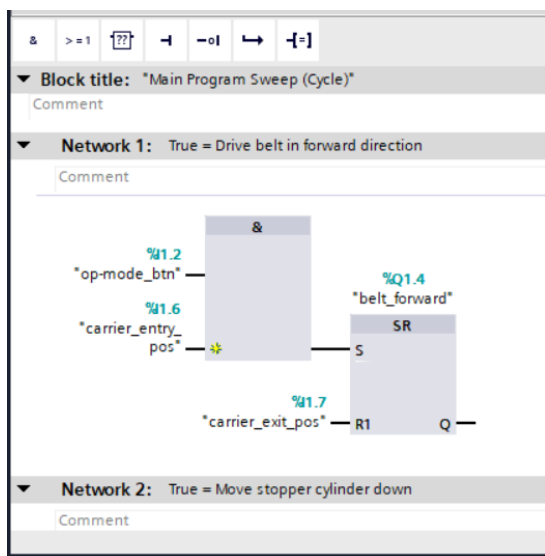
Figur 14: Eksempel på feilmelding etter ustabil PLCSIM



Figur 15: Eksempel på koblingen fikset

4.2.1 Høynivåspråk

Vi diskuterte med oppdragsgiver om vi skulle lage lab-oppgaver med Structured Control Language (SCL), eller om vi skulle bruke ladder eller funksjonsblokkdiagram. «SCL (Structured Control Language) er et tekstbasert programmeringsspråk på høyt nivå som er basert på PASCAL. Det er også basert på industristandarden for PLS-er (programmerbare logiske styringer)» (Oversatt fra engelsk) [23]. Det var ønsket å bruke SCL siden det kan argumenteres for at språket er mer framtidsrettet. Etersom studentene får lite kunnskap om SCL i deres utdanningsløp, ønsket oppdragsgiver å lage to versjoner. Det ble derfor besluttet å lage en versjon med SCL, og en med funksjonsblokkdiagram.



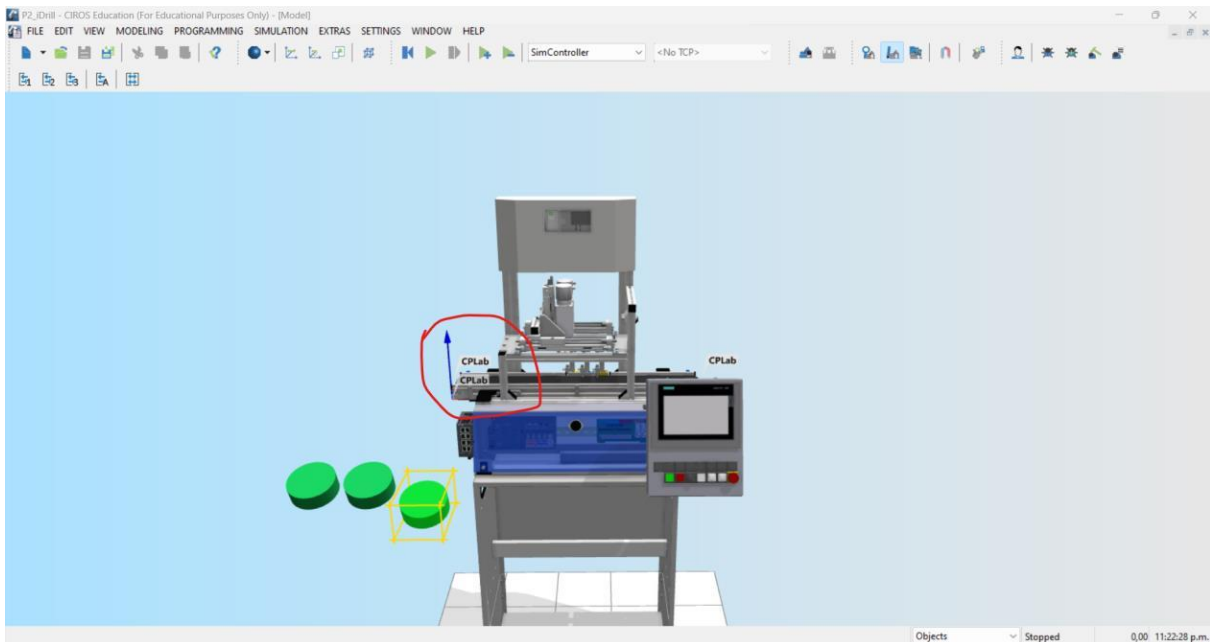
Figur 16: Eksempel på funksjonsblokkdiagram.

4.2.2 Brukerveiledninger

Utvikling av brukerveiledninger var et viktig punkt for oppdragsgiver. Dette innebar å lage en brukerveiledning for sammenkobling av programmene, og hvordan de skal brukes.

Brukerveiledning ble knyttet tett opp til lab 1, men vil også være relevant for alle lab-oppgavene som leveres. Brukerveiledningen vil være relevant både for studenter og lærere.

Vi utarbeidet en veiledning for hvordan forskjellige moduler kunne kobles sammen, dette var for å vise lærere hvordan de enkelt kan utvikle sine egne modeller. Å legge til modeller er veldig intuitivt, det vil komme frem vinduer som skal legges opp på hverandre, se eksempel «Figur 17». Utfordringene starter hvis modellene skal kobles sammen. Det er ikke alltid mulig å koble modellene til transportbåndet, som ofte brukes som basemodell. Grunnen til dette er at flere av modellene som er ferdiglaget av Festo ikke alltid har et komplett I/O panel, men har andre grensesnitt som pneumatiske styringsenheter.



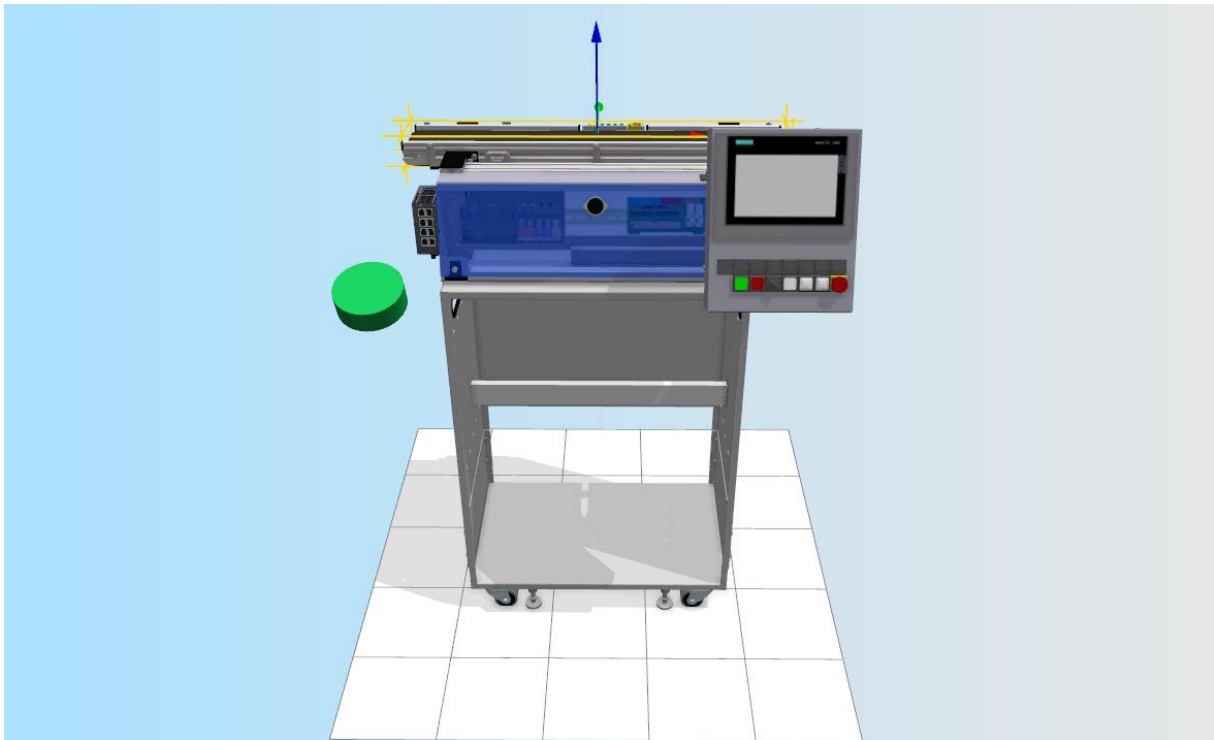
Figur 17: Eksempel på hvordan man legger til iDrilling modulen til transportbånd modulen

4.3 Ferdigutviklede lab-oppgaver

Som beskrevet i avsnittet om «Forberedende pedagogisk arbeid» vil studentene ha begrenset teoretisk bakgrunn, og det er derfor viktig å tilby en intuitiv introduksjon til programmene. På denne måten kan en gradvis økning i vanskelighetsgraden skapes, og det vil danne en sammenhengende progresjon. Lab 1 vil fungere som en veiledning i å koble sammen programmene, og navigere i TIA-Portalen. Lab 2 vil gi studentene en enkel programmeringsoppgave for å forberede dem på Lab 3. Dette vil være den mest utfordrende lab-oppgaven, og vil utfordre studentene i prosessen med å lage et PLS-program.

4.3.1 Lab 1 - CP Lab Linear (CP-L-LINEAR)

Denne lab-oppgaven har vi valgt å bruke mest som en introduksjon til CIROS og TIA-Portalen. Studentene begynner med å laste ned de ferdiglagde programmene våre, for å så følge en detaljert brukerveiledning som viser hvordan de to programmene skal kobles sammen. Hensikten med dette er at studenten skal få kunnskap i å navigere seg rundt i de to programmene, før de begynner å programmere selv. De får se en fungerende modell og kode, slik at de kan visualisere programmets funksjon.



Figur 18: CP-L-Linear i CIROS

I denne lab-oppgaven er det transportbåndet som håndteres. I koden studentene får utlevert har vi laget en tidtaker som skal stoppe arbeidsstykket på midten i 3 sekunder, dette detekteres av en sensor i CIROS. Modellen har en bryter som brukes til å skifte mellom manuell og automatisk modus. Studentene gis tilgang til en fil der kun funksjonaliteten til det automatiske modus er ferdig.

IF...	CASE... OF...	FOR... TO DO...	WHILE... DO...	(*...*)	REGION
1	IF				#"Xop-mode_btn" = TRUE THEN
2					IF #Xcarrier_entry_pos = TRUE THEN
3					#Ybelt_forward := TRUE;
4					END_IF;
5					IF #Xcarrier_exit_pos = TRUE THEN
6					#Ybelt_forward := FALSE;
7					END_IF;
8					//ELSE
9					END_IF;
10					"IEC_Timer_0_DB".TON(IN:=#Xcarrier_stopper_pos,
11					PT:=T#3S,
12					Q=>#Ymove_cylinder_down);

Figur 19: Programkode for styring av transportbånd i TIA-Portalen

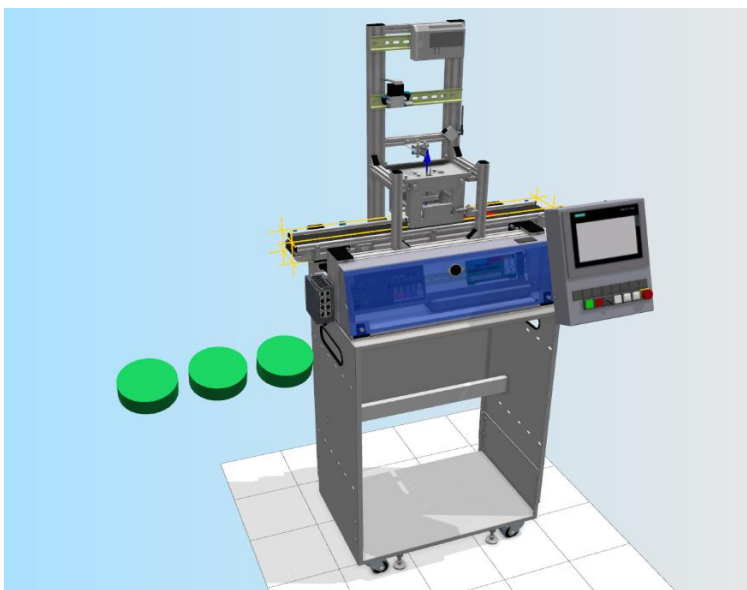
Etter studentene har samkjørt programmene, skal de gå gjennom oppgaveteksten og svare på noen teoretiske spørsmål. Når dette er fullført, skal de prøve å programmere manuell modus selv. Funksjonaliteten til manuell modus er forklart i et flytskjema.

IF...	CASE... OF...	FOR... TO DO...	WHILE... DO...	(*...*)	REGION
1					"CP-L-Conveyor_DB" (Xstartbtn:="start_btn",
2					Xstopbtn:="stop_btn",
3					Xop-mode_btn:="op-mode_btn",
4					Xreset_btn:="reset_btn",
5					Xcarrier_stopper_pos:="carrier_stopper_pos",
6					Xemergency_stop_btn:="emergency_stop_btn",
7					Xcarrier_entry_pos:="carrier_entry_pos",
8					Xcarrier_exit_pos:="carrier_exit_pos",
9					Xcarrier_ident_0:="carrier_ident_0",
10					Xcarrier_ident_1:="carrier_ident_1",
11					Xcarrier_ident_2:="carrier_ident_2",
12					Xcarrier_ident_3:="carrier_ident_3",
13					Xswitch_S1:="switch_S1",
14					Xswitch_S2:="switch_S2",
15					Xstop_cylinder_low_pos:="stop_cylinder_low_pos",
16					Ystart_btn_light=>"start_btn_light",
17					Yreset_btn_light=>"reset_btn_light",
18					Yfunc_Q1_light=>"func_Q1_light",
19					Yfunc_Q2_light=>"func_Q2_light",
20					Ybelt_forward=>"belt_forward",
21					Ybelt_reverse=>"belt_reverse",
22					Ybelt_speed=>"belt_speed",
23					Ymove_cylinder_down=>"move_cylinder_down",
24					Ycoupling_signal_left=>"coupling_signal_left",
25					Ycoupling_signal_right=>"coupling_signal_right");
26					

Figur 20: Implementering av funksjonsblokk i Main.

4.3.2 Lab 2 - Press (CP-AM-PRESS)

Lab 2 bygger videre på arbeidet som er gjort i lab 1. Vi bruker det samme transportbåndet og kobler til en presse som applikasjonsmodul. Pressen er en enkel modul med få innganger og utganger, som vi meinte var en passende økning i vanskelighetsgrad. Lab-oppgaven fokuserer på god programforståelse som et gjennomgående tema. Vi introduserer bruk av tidtakere i programmet og flere av oppgavene er sentrert rundt dem.



Figur 21: Transportbånd med pneumatisk presse i CIROS.

I oppgave 2 og 3 får studentene teste programforståelsen sin. I oppgave 2 skal de drøfte måten vi har brukt tidtakere og signal. Her lagde vi en mer åpen oppgave hvor studentene må tenke godt gjennom svaret sitt. I dette tilfellet kan det være flere riktige svar basert på hvordan studentene selv ville ha brukt programmet. De får muligheten til å argumentere for løsningene sine, og kan lære av løsningene til andre studenter. Oppgave 3 er i likhet med oppgave 2 en veldig åpen oppgave. Den kan gjennomføres på ulike måter, så lenge de opprettholder funksjonaliteten i programmet. Denne oppgaven belønner god programforståelse, og den kan gjennomføres med en enkel endring i koden.

```

IF... CASE... FOR... WHILE... (*...*) REGION
OF... TO DO... DO...

1 IF #sbPressDown = TRUE THEN
2     #Ycylinder_down      := TRUE;
3     #Ycylinder_up       := FALSE;
4     #sbPressUp          := FALSE;
5 ELSE
6     #sbPressDown := FALSE;
7     #Ycylinder_down := FALSE;
8     #Ycylinder_up := TRUE;
9 END_IF;
10 IF #sbPressUp = TRUE THEN
11     #sbPressDown      := FALSE;
12     #Ycylinder_down   := FALSE;
13     #Ycylinder_up    := TRUE;
14 END_IF;
15 IF #Xcarrier_stppoer_pos = FALSE THEN
16     #sbPressUp := TRUE;
17     #sbPressDown := FALSE;
18 END_IF;
19 IF #Xcarrier_stppoer_pos = TRUE THEN
20     "IEC_Timer_0_DB_1".TP(IN := #Xcarrier_stppoer_pos,
21                          PT := T#1500MS,
22                          Q => #sbPressDown);
23 ELSE
24     RESET_TIMER("IEC_Timer_0_DB_1");
25 END_IF;
26

```

Figur 22: Programkode for styring av pneumatisk presse.

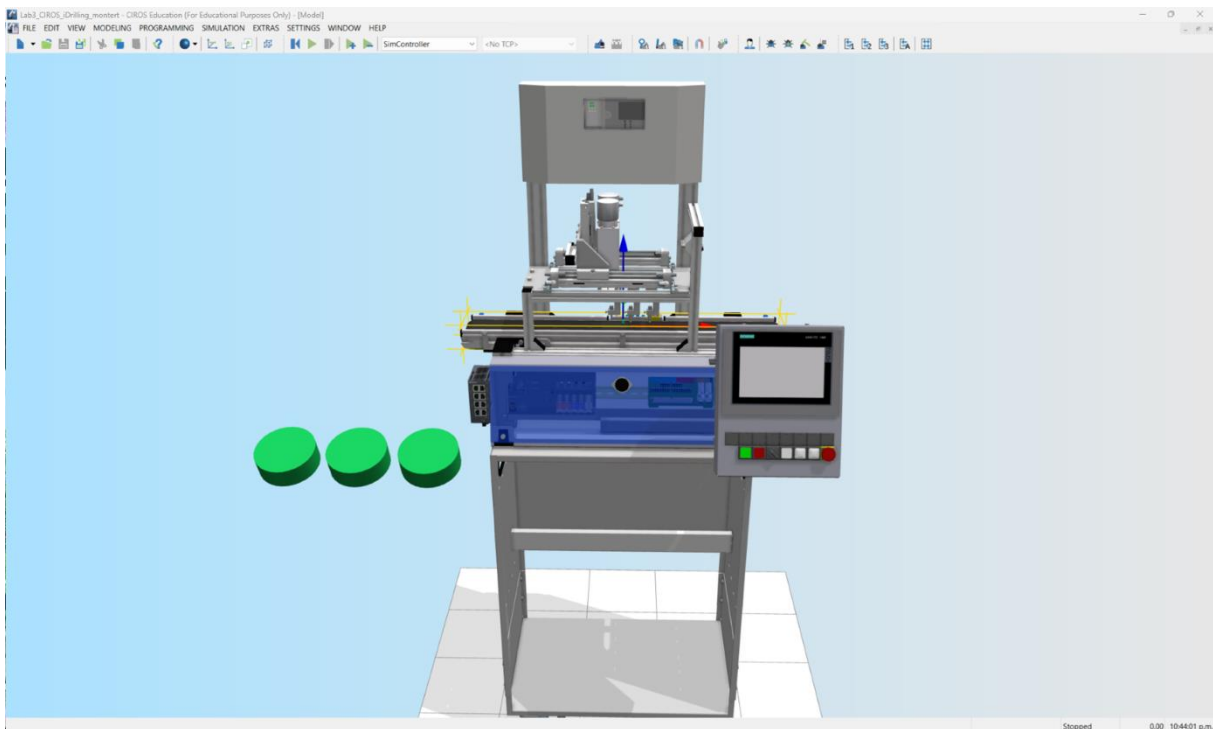
4.3.2.1 Pneumatikk og virkemåte av applikasjonsmodul

Pressen vi koblet til transportbåndet i lab 2 er en pneumatisk presse som er styrt av 3 inngangssignal og 2 utgangssignal. De 3 inngangssignalene forteller oss om hvilken posisjon pressen er i, og om arbeidsstykket som kommer inn har et bakdeksel. Med de 2 utgangssignalene styrer du posisjonen til pressen, den ene utgangen får pressen til å gå opp og den andre vil presse den ned. Når du aktiverer pressen vil du sende et signal til luftventilen som kontrollerer lufttrykket i systemet. Ventilen har kun to modi der den ene vil kjøre pressen opp og den andre ned.

4.3.3 Lab 3 - iDrilling (CP-AM-iDRILL)

I lab 3 skal studentene jobbe med iDrilling-modellen. Når arbeidsstykket står på midten av transportbåndet, skal drillen automatisk komme ned og drille to hull i arbeidsstykket. Utføringen av oppgaven krever den komplette koden til transportbåndet, og studentene får derfor utlevert løsningsforslaget til lab 1. Hensikten bak denne lab-oppgaven er å gi

studentene en bredere forståelse av koblingen i TIA-Portalen. Studentene starter med å legge inn I/O-en til iDrilling-modellen i tag-tabell. Dette er en relativt enkel oppgave, men det må være nøyaktig samme navn som vi har gitt for at det skal kommunisere med CIROS-modellen. Videre skal de begynne å programmere iDrilling-modellen, som de nå for første gang skal begynne helt fra start på. Selv om det ikke er veldig avansert programmering som skal til, er det et steg opp i vanskelighetsgrad. Studentene skal først programmere startposisjonen til drillen, deretter skal drillen aktiveres når arbeidsstykket er på midten av transportbåndet.



Figur 23: iDrilling modulen i sammenkobling med transportbånd modulen.

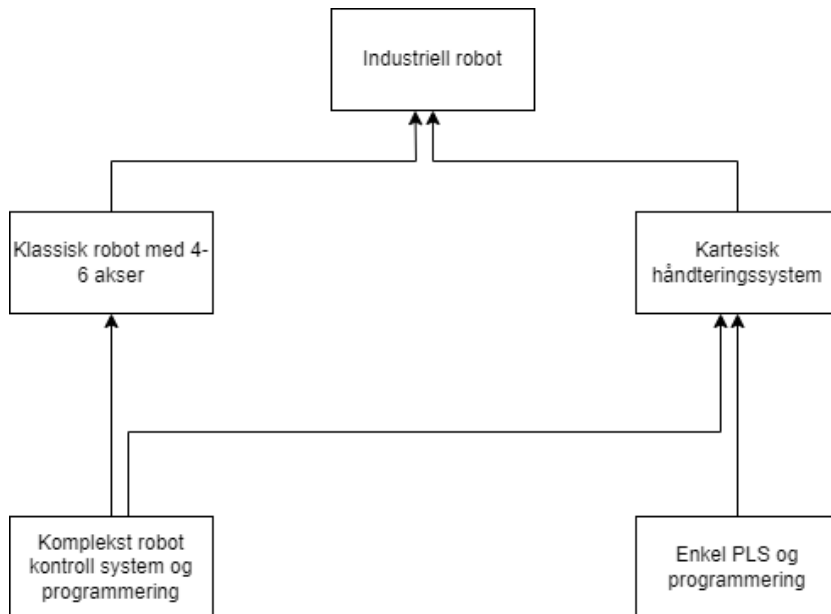
4.3.3.1 Lineær gantry robot

Drillen i denne modulen fungerer som en robot som forflytter seg langs x-aksen, og beveger seg opp og ned i z-aksen. Dette gjør den til en lineær gantry robot, som beveger seg i 2D.



Figur 24: Festo lineær gantry robot lignende til drillen. Kilde: [24]

Festo er spesialister på gantry roboter. De velger å bruke roboter som kan styres av enkle PLS programmer, i stedet for roboter som krever kompleks kode for å skape samme resultat. [25].



Figur 25: Illustrasjon av hvordan Festo sine roboter bruker enkle systemer for å lage avanserte systemer.

Laget selv i diagrams.net, oversatt versjon av kilde: [25]

Roboten bruker også en pneumatisk motor for å bevege seg både i x-aksen og z-aksen, som gjør at den er energieffektiv [2].

4.3.3.2 Oppkobling

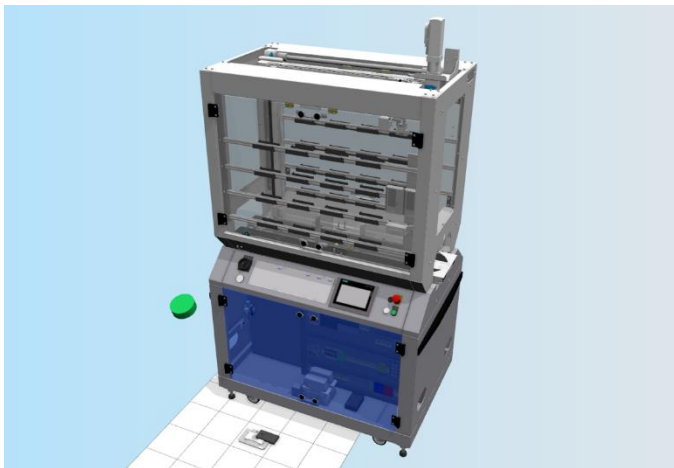
iDrilling-modellen bruker en annen tilkobling enn de andre modellene. Denne bruker CECC-LK tilkobling som er et program som baserer seg på CODESYS. Denne styringsenheten kobler sammen pneumatikken og de elektriske aktuatorene som gjør at vi klarer å styre modulen med en PLS [26]. Dette er ikke noe problem i sammenkoblingen med CIROS ettersom den leser dette som enkle I/O signaler. Dermed vil den lese signalene som en ordinær PLS tilkobling.



Figur 26: Bilde av CECC-LK fra datablad. Kilde: [26]

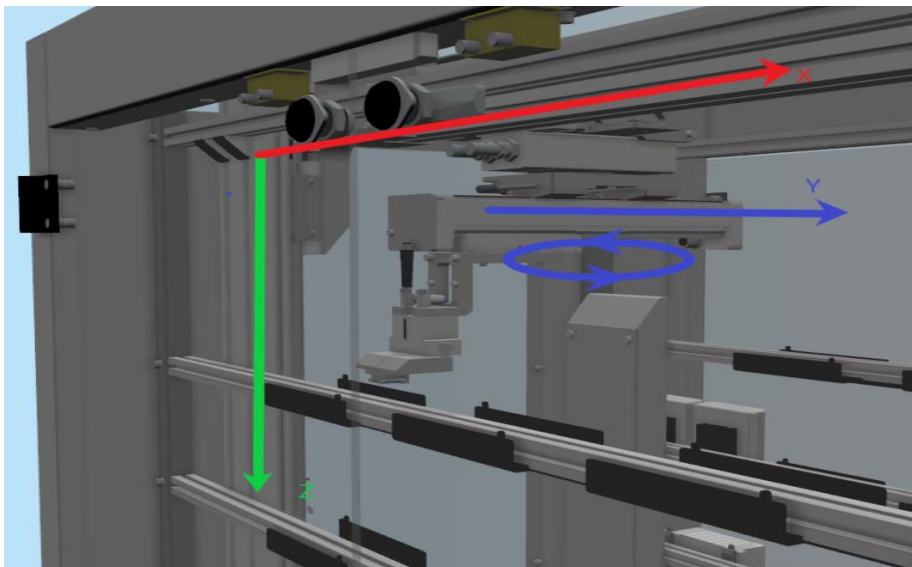
4.4 Lab 4 – High Bay Storage Unit (CP-F-ASRS32-P)

Vi vurderte flere forskjellige metoder for gjennomføring av oppgaven. Vi tenkte først at studentene kunne få prøve å programmere hele oppgaven selv. Etter vi startet å programmere modellen for å teste den skjønte vi derimot at dette ville bli for utfordrende for studentene. Deretter vurderte vi å programmere alle funksjonsblokkene ferdig slik at oppgaven til studentene blir først å implementere de på riktig måte. Deretter måtte de ha programmert en funksjon for å lagre pallene på riktig plass. Dessverre ble lab 4 aldri ferdigutviklet. Dette på grunn av tekniske feil ved programmeringen av modellen som vi ikke klarte å løse.



Figur 27: Highbay storage modell i CIROS

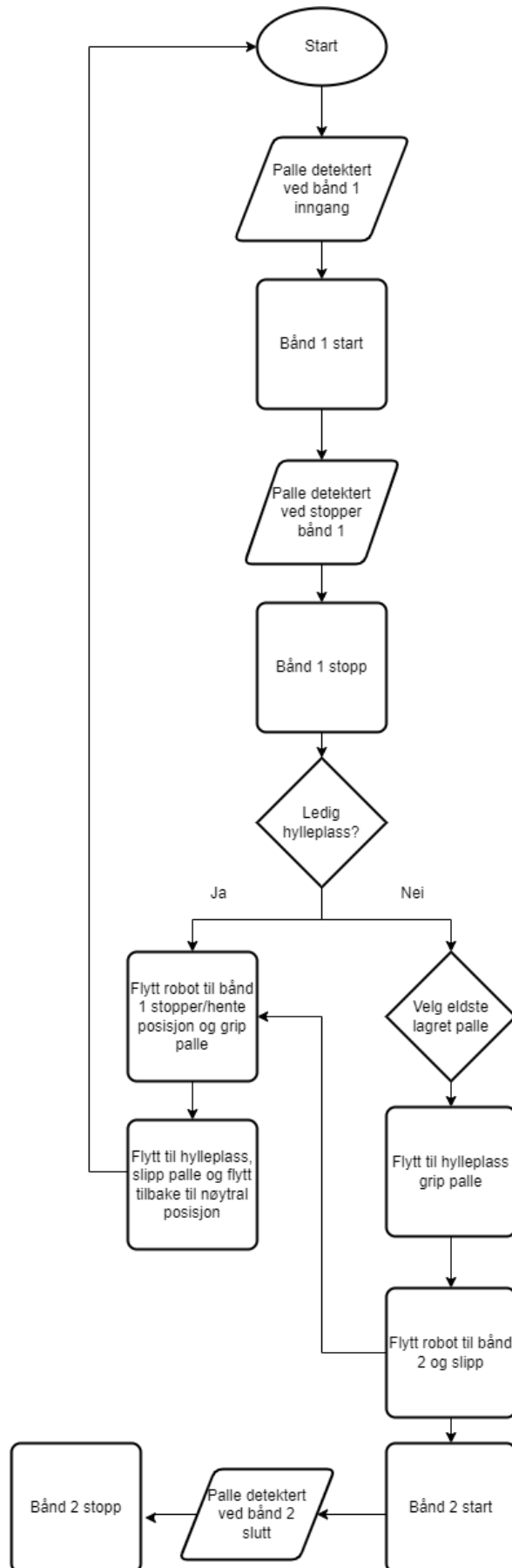
Modellen lab-oppgaven omhandlet er en lagerenhet-robot vist i Figur 27. Modellen mottar paller og lagre disse på hylleplasser i enheten. Hoveddelen av modellen er den kartesiske roboten på innsiden av enheten. Denne roboten flytter seg langs x og z-aksen ved hjelp av 2 lineære servomotorer, og y-aksen lineært og roterende ved hjelp av 2 pneumatiske aktuatorer illustrert i Figur 28.



Figur 28: kartesisk robot med markerte akser

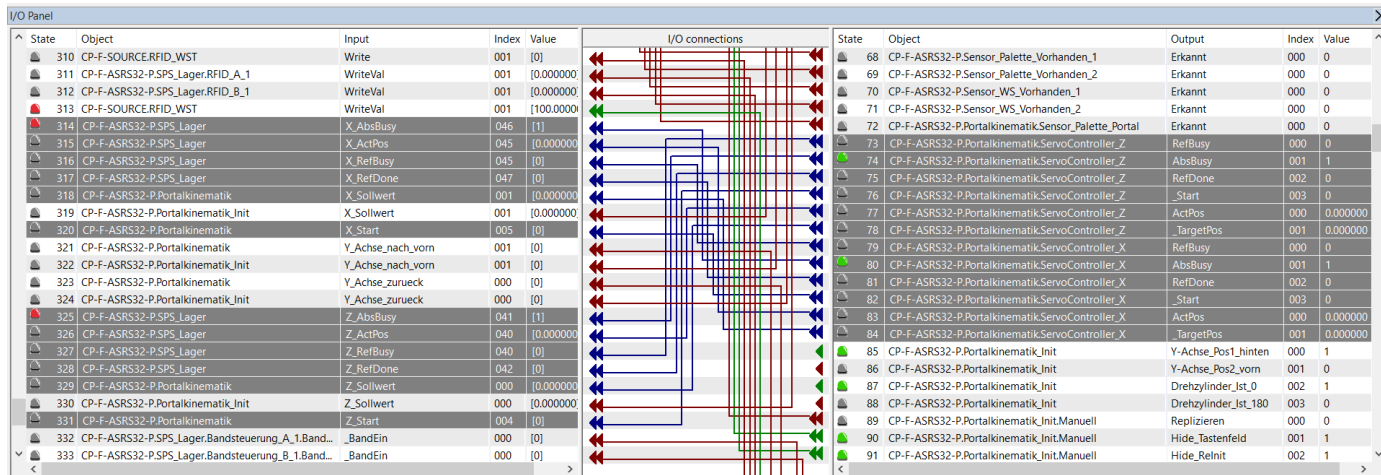
Utviklingen av lab 4 ble påvirket av flere utfordringer i forsøket på å programmere modellen. En primær problemstilling oppsto da Festo ikke hadde definert noen I/O for styring av den kartesiske roboten, med unntak av y-aksen i databladet [27]. Etter gjennomgang av manualen avdekket vi at roboten ble kontrollert via en servokontroller, og to servomotorer som var definert i I/O-panelet i CIROS. På bakgrunn av dette identifiserte vi de nødvendige I/O-signalene, som vist i Figur 30. Vi implementerte de i I/O-grensesnittet mot TIA-portalen. Da kunne vi styre roboten ved hjelp av kode fra TIA-portalen.

Deretter utviklet vi testkode for å bevege roboten langs de ulike aksene. Vi klarte å flytte og rotere roboten uten problemer langs y-aksen. Imidlertid støtte vi på nye utfordringer når det kom til x- og z-aksene. Selv om vi klarte å flytte roboten jevnt til en spesifikk koordinatverdi i begge aksene, oppdaget vi at den ikke fortsatte bevegelsen utenfor et visst område. Dette er vist i illustrasjonene for x-aksen i Figur 31, og Figur 32 hvor en kan se verdien aksens låser

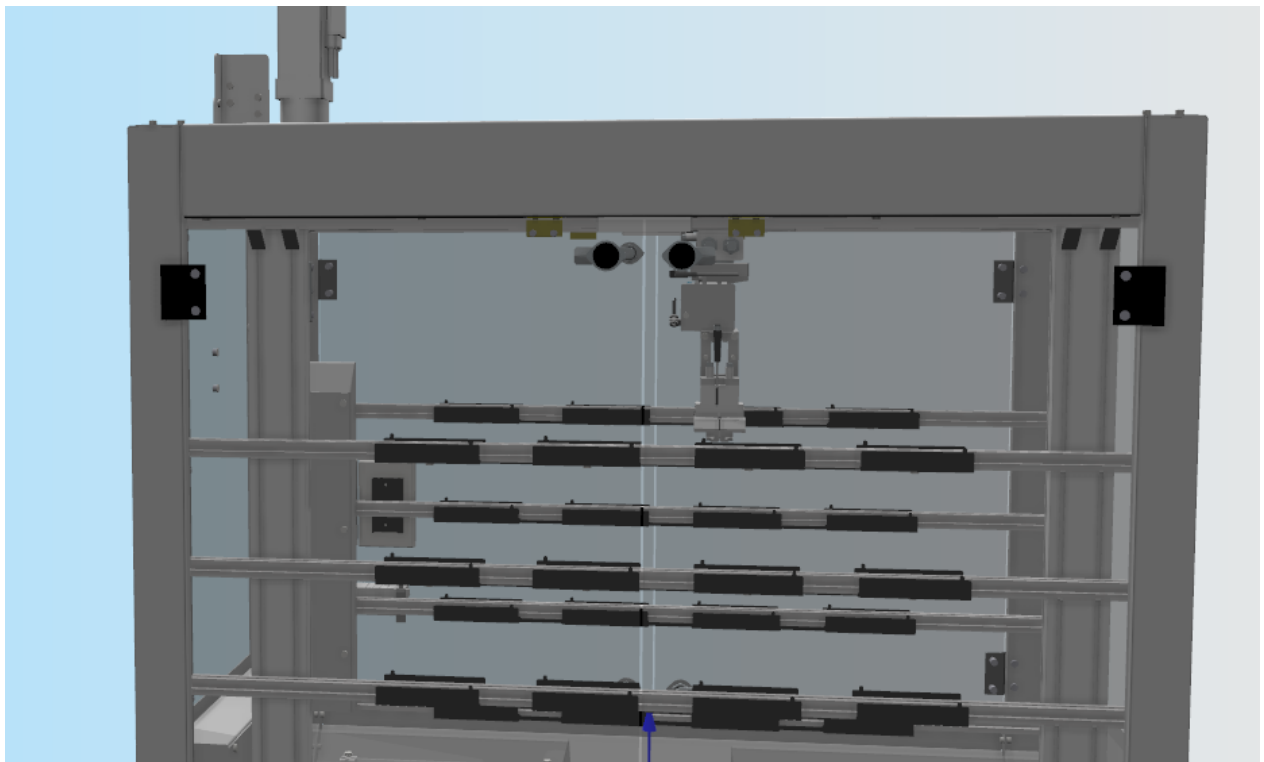


Figur 29: Flytskjema for ASRS32

seg på.



Figur 30: I/O definert for servokontrolleren til x og z akse



Figur 31: Robot i posisjon hvor den ikke flytter seg lenger langs x akse

45		XActPos_X	Real	%ID2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	34816.0
46		XRefBusy_Z	Bool	%I4.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE
47		XRefBusy_X	Bool	%I4.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE

Figur 32: Verdi av koordinat til robot i x akse posisjon fra Figur 31 vist i tag-table.

Dette problemet forble uløst, og vi klarte ikke å identifisere årsaken eller lokalisere hvor problemet oppsto. Imidlertid har vi ulike hypoteser angående mulige årsaker. En av mistankene våre er at en real-verdi i CIROS ikke blir tolket på riktig måte i TIA-portalen. En annen grunn kan være at CIROS ikke simulerer servostyringen og enkoder på en måte TIA-portalen forventer, og dermed blir offset-verdien feil. Dette påvirker «XActPos_X» og

resulterer i at roboten låser seg. Det er også mulig at problemet skyldes en feil i CIROS eller en manglende kommunikasjonsprotokoll, men dette kunne vi ikke bekrefte. I etterkant oppdaget vi også flere utganger som er relatert til bevegelse i x og z-aksene, men disse hadde tyske navn. Det er derfor mulig at vi har brukt feil utganger for bevegelsen av roboten. Dette fikk vi ikke anledning til å teste ytterligere.

Under testingen møtte vi også på et problem med å synkronisere bevegelsen av aksene. Det var ikke mulig å bevege seg først langs x-aksen og deretter langs z-aksen, eller å bevege seg i begge aksene samtidig. I Festos kode til den interne PLS-en ble aksesykronisering håndtert ved hjelp av ventefunksjoner. De er derimot ikke mulig å bruke i TIA-Portalen. Dette støtter antagelsen om at CIROS ikke simulerer en servokontroller, men bare definerer et objekt kalt "servokontroller" med tilhørende I/O. Vi prøvde å løse aksesykroniseringsproblemet ved å bruke tidtakere, men dette fungerte ikke. Vi vurderte også å definere et teknologisk objekt i TIA-Portalen for å styre bevegelsen langs aksene, men konkluderte med at mål-PLS-en måtte være en type PLS vi ikke kan simulere. Dermed fant vi heller ingen løsning på dette problemet. Om det må konkluderes en feilårsak mener vi det er størst sannsynlighet for at CIROS ikke har denne funksjonaliteten.

5 Testing

Målet med testing av lab-oppgavene var å avdekke feil i tekst eller kode. Ved en av samlingene til studentene fikk de fra robotteknologi og digital produksjon mulighet til å teste lab-oppgavene [15]. På denne måten testet vi vanskelighetsgraden, og fant eventuelle feil. Lab 1 ble testet med oss til stede, mens lab 2 og 3 gjennomførte studentene alene. De sendte oss tilbakemeldinger i etterkant ved å fylle ut spørreundersøkelser vi lagde før testen.

5.1 Test av lab 1

» vil studentene ha begrenset teoretisk bakgrunn, og det er derfor viktig å tilby en intuitiv introduksjon til programmene. På denne måten kan en gradvis økning i vanskelighetsgraden skapes, og det vil danne en sammenhengende progresjon. Lab 1 vil fungere som en veiledning i å koble sammen programmene, og navigere i TIA-Portalen. Lab 2 vil gi studentene en enkel programmeringsoppgave for å forberede dem på Lab 3. Dette vil være den mest utfordrende lab-oppgaven, og vil utfordre studentene i prosessen med å lage et PLS-program.

Lab 1 - CP Lab Linear (CP-L-LINEAR) ble gjennomført med oss til stede. På denne måten kunne vi hjelpe studentene som møtte på problemer. Lab-øvelsen startet ved at studenten

fulgte nedlastningsveiledning. Det viste seg at det manglet instruks for Virtuelt Privat Nettverk-tilkoblingen (VPN). For lab 1 lagde vi oppgaver i både SCL og funksjonsblokkdiagram, derfor fikk studentene lov til å velge hvilken versjon de ville benytte seg av selv. Her valgte alle å bruke funksjonsblokkdiagram versjonen. Det var et veldig variert kunnskapsnivå innen programmering i klassen. For enkelte gikk lab-oppgaven helt problemfritt, mens for andre ble det en krevende lab. De fleste gjennomførte lab-oppgaven på rundt 1 time, noe som oppdragsgiver og vi var fornøyd med. Det ble bekreftet at brukerveiledningen for å koble sammen TIA-portalen og CIROS var feilfri. Det var derimot noen små utfordringer med lab 1. Det største problemet var for mye informasjon i oppgaven, som gjorde at studentene ble utålmodige og hoppet over viktige steg. Det andre problemet var at det ene bildet på funksjonsblokkdiagram-versjonen var feil. Vi fikk følgende tilbakemeldinger på lab-oppgave 1:



The screenshot shows a survey titled "Spørreundersøkelse Lab" with the following questions and answers:

- Lab 1
- Hvor lang tid brukte du på Lab 1?
2t
- Hvordan var vanskelighetsgraden?
1 av 6
- Var noen av spørsmålene uklare? Hvilke og hva var uklart?
Programmeringsoppgaven var ikke spesielt lett å forstå hva som egentlig skulle gjøres.
- Har du noen punkt som kan forbedres?
(Empty text box)

Figur 33: Første tilbakemelding lab 1. Kilde: Skjermdump

Tilbakemeldingen fra studenten viser at det var utfordringer med å forstå hva som skulle gjøres i programmeringsoppgaven. Studenten opplevde lab-oppgaven som enkel.

Hei.

etter å fullført Lab 1 oppleves den som svært tungt lest og som et grov utkast.
Det oppgave beskrivelse stemmer ikke med selve programmet deres, eks oppgave 3.

dokumentet burde være strukturert på en annen måte og dere bør se på hvordan dere formulerer dere slik at det er lettere å tolke det dere er ute etter.

mesteparten av tiden går med på å lese oppgavene deres flere ganger og tolke hva dere er ute etter.

Figur 34: Andre tilbakemelding lab 1. Kilde: Skjermdump

Tilbakemeldingen fra studenten viser vansker med å forstå oppgaven, og hen henviser til feil i oppgavebeskrivelse. Studenten kommenterer ikke vanskelighetsgrad.

Spørreundersøkelse Lab

Lab 1
Hvor lang tid brukte du på Lab 1?

Ca. 1 time 20 min

Hvordan var vanskelighetsgraden?

Har programmert en del fra før så jeg synes de var lett

Var noen av spørsmålene uklare? Hvilke og hva var uklart?

Jeg ble ikke forvirret

Har du noen punkt som kan forbedres?

Opgaven 2.6, her var det litt vanskelig å se hvilke tagger som endret verdi under simuleringen da simuleringen gikk ganske kjapt.

Figur 35: Tredje tilbakemelding lab 1. Kilde: Skjermdump

Tilbakemeldingen fra studenten viser at oppgaven var enkel og at alle spørsmålene var klare. Studenten kommenterer et punkt av oppgaven som litt vanskelig.

5.2 Test av lab 2

Lab 2 - Press (CP-AM-PRESS) gjennomførte studentene uten oss til stede, så vi benytter tilbakemeldingene vi fikk som referanse. Vi opplevde å få positive tilbakemeldinger fra

studentene. I utgangspunktet tenkte vi at de ikke hadde fått mye opplæring om tidtakere. Her hadde de mer kunnskap enn vi trodde, så vanskelighetsgraden var lavere enn vi hadde sett for oss. Noen av oppgavene var litt uklare, men vi tenker at denne lab-oppgaven kan være gunstig å bruke for lærere som skal introdusere studentene for tidtakere. Vi mottok følgende tilbakemelding på lab-oppgave 2:



The image shows a survey titled "Spørreundersøkelse Lab" with the following content:

Lab 2
Hvor lang tid brukte du på Lab 2?

Ca. 1 time

Hvordan var vanskelighetsgraden?

Lett. Bruk av timere var ikke nytt for oss.

Var noen av spørsmålene uklare? Hvilke og hva var uklart?

Har du noen punkt som kan forbedres?

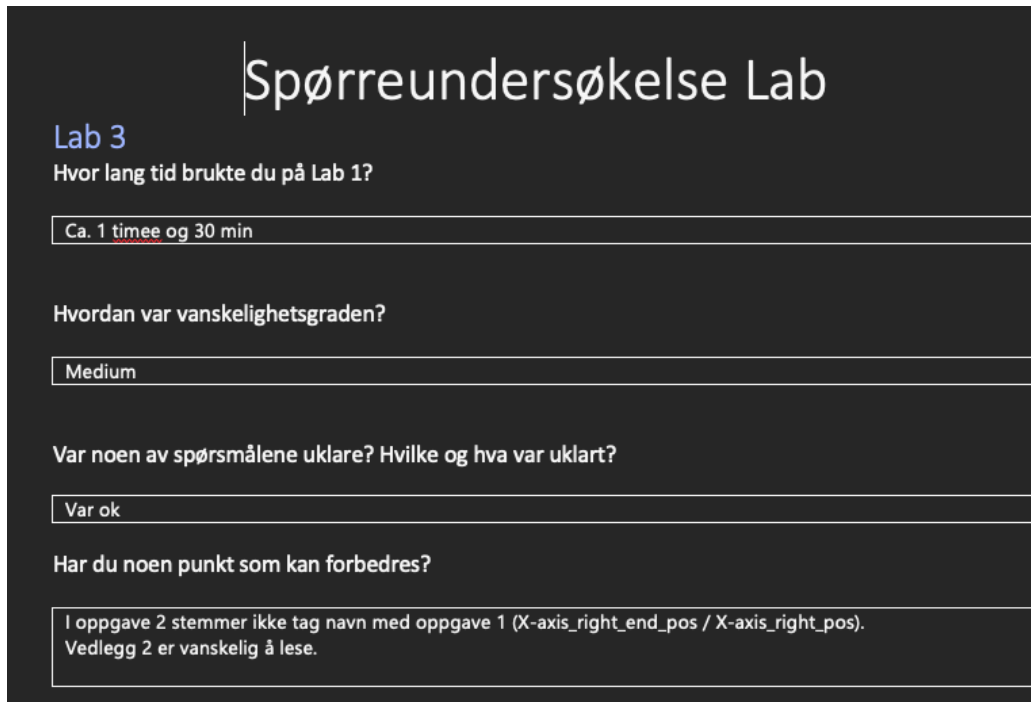
Kanskje oppgave 1.3 kan være forvirrende om man ikke snur og vender på modellen for å se forskjellen på vognene.

Figur 36: Tilbakemelding fra lab 2. Kilde: Skjermdump

Tilbakemeldingen fra studenten viser at lab-oppgaven var enkel, og at dette var noe studenten kjente til fra før. Studenten oppgir et punkt til forbedring.

5.3 Test av lab 3

Lab 3 - iDrilling (CP-AM-iDRILL) gjennomførte studenten alene. Vi har derfor benyttet studentens tilbakemeldinger som referanse. Vi mottok følgende tilbakemelding på lab-oppgave 3: Figur 37



The screenshot shows a survey form with a dark background and white text. The title is 'Spørreundersøkelse Lab'. Below it, 'Lab 3' is written in blue. The first question is 'Hvor lang tid brukte du på Lab 1?' with the answer 'Ca. 1 timee og 30 min'. The second question is 'Hvordan var vanskelighetsgraden?' with the answer 'Medium'. The third question is 'Var noen av spørsmålene uklare? Hvilke og hva var uklart?' with the answer 'Var ok'. The fourth question is 'Har du noen punkt som kan forbedres?' with the answer 'I oppgave 2 stemmer ikke tag navn med oppgave 1 (X-axis_right_end_pos / X-axis_right_pos). Vedlegg 2 er vanskelig å lese.'

Figur 37: Tilbakemelding på lab 3. Kilde: Skjermdump

Tilbakemeldingen fra studenten viser feil i teksten, noe som førte til utfordringer med sammenkoblingen mellom TIA-Portalen og CIROS. Det var også et av flytdiagrammene som var vanskelig å lese. Det er viktig å presisere at vi kun mottok en tilbakemelding på denne lab-oppgaven. Årsaken til dette er at studentene som gjennomførte lab-oppgaven studerer på deltid, og har dermed begrenset tid på skolen.

5.4 Revisjon av lab-oppgaver og nedlastningsguide

Etter tilbakemelding fra studentene ble vi enige om å revidere lab-oppgavene og nedlastningsguiden. Det var ingen feil på programsiden av lab-oppgavene, men det ble gjort utbedringer av feil på tekst og bilder. I samtale med oppdragsgiver bestemte vi oss også for å lage en ny versjon av lab 1 med mindre tekst og forklaring. Vi gav studentene mer konkrete oppgaver, med mindre informasjon og veiledning. Dette for å utfordre studentene i noe større grad, samt å unngå forvirring. Lab 2 ble utbedret slik at studentene lettere skulle forstå oppgaven. Tilbakemeldingen på lab 3 gjorde at vi rettet feilene som skapte problemer i sammenkoblingen mellom CIROS og TIA-Portalen. Det ble endret på en skrivefeil, samt laget et nytt flytdiagram som var større og enklere å lese.

6 Diskusjon

6.1 Teknisk grensesnitt

Som skrevet om i «Valgt løsnin» ble det besluttet å bruke det tekniske grensesnittet PLCSIM Advanced 2.0. Imidlertid har det vist seg at denne løsningen lider av ustabilitet og begrensninger, noe som gjør det nødvendig å vurdere alternative løsninger. For å sikre at studentene får et brukergrensesnitt som fungerer pålitelig og effektivt, anbefales det å teste ut flere alternativer. Selv om oppdragsgiver tidligere har benyttet seg av versjon 2.0, bør det vurderes om oppgraderingen til versjon 3.0 ville vært en bedre løsning for studentene og deres læringsutbytte. En slik vurdering bør gjøres i samråd med oppdragsgiver.

Andre løsninger som ble diskutert, inkluderer fysisk PLS-kommunikasjon via EasyPort og EzOPC eller OPC UA, samt CODESYS og EZOPC. Disse alternativene kan være mer avanserte for studentene å lære, og det er en kostbar investering for fagskolen å anskaffe nok fysiske PLS-er for å bruke disse i sammenkoblingen mellom PLS og CIROS via EasyPort og EzOPC. Videre diskusjon og vurdering fra oppdragsgiver vil være nødvendig for å sikre at den beste løsningen blir valgt for studentene og fagskolen.

6.2 Lab-oppgaver

Det kan diskuteres hvor godt lab-oppgavene har truffet målet som ble satt. Lab 1 kunne vært laget som en standard lab fra første stund, og ikke etter revisjon. Tanken var at denne skulle fungere både som en lab-oppgave og en forelesning. Det viste seg å være problematisk at lab 1 inneholdt så mye informasjon. Dette bidro til at studentene synes det ble for mye å lese før de startet på selve lab-oppgaven. I ettertid ser vi at det hadde vært mer hensiktsmessig for studentene å få noe av informasjonen i forkant. Det kan derfor tenkes at en forelesning i forkant av lab 1 er gunstig for å unngå for mye informasjon i lab-oppgaven.

I etterkant av lab 2 ser vi at vanskelighetsgraden på oppgavene var for lave. I samtalene med oppdragsgiver fikk vi informasjon om at nivået på studentene var ganske lavt, men her hadde de mer kunnskap enn vi trodde. Lab 2 kunne derfor inneholdt mer utfordrende oppgaver, og mindre fokus på tidtakere. Eventuelt kan oppgaven brukes tidligere i sammenheng med introduksjon til tidtakere. Lab 4 klarte vi ikke å fullføre i tide. Under utviklingen ble det tydelig at modellen var for kompleks i oppbygningen, og at det ville ta for lang tid fullføre den. Vi brukte mye tid på utviklingen, men vi kunne gjerne avsluttet arbeidet tidligere. På den måten hadde vi kanskje hatt tid til å modellere en annen komponent for lab 4. Ettersom lab 4 ikke ble ferdigutviklet endte lab 3 som den vanskeligste lab-oppgaven vi leverte. Dersom vi hadde visst dette i forkant, hadde vi økt nivået på lab-oppgave 3 for å gi bedre læringsutbytte.

7 Konklusjon og videre arbeid

I dette prosjektet utviklet vi et grensesnitt mellom CIROS og TIA-Portalen for bruk til studenter ved Fagskolen på Vestlandet. Videre har vi utviklet flere lab-oppgaver som benytter dette grensesnittet. Dette gjorde vi slik at studentene kan teste ut egenutviklede PLS programmer på en trygg måte, før oppgavene testes på fysisk fabrikk system. Ved fremtidig arbeid anbefales det å vedlikeholde grensesnittet og lab-oppgavene med mer stabile versjoner av PLCSIM Advanced fra Siemens.

7.1 Teknisk grensesnitt

Fagskolen ønsker å verifisere student produserte PLS programmer som benyttes på de fysiske fabrikkmodellene. I prosjektet har vi evaluert flere alternative metoder for dette grensesnittet. Basert på allerede tilgjengelige programvare lisenser ved Fagskolen, ble det utviklet et teknisk grensesnitt mellom CIROS og TIA-Portalen ved hjelp av S7 PLCSIM Advanced v2.0. Denne løsningen er tilstrekkelig stabil for å gjennomføre lab-oppgaver.

7.2 Lab-oppgaver

I prosjektet har vi produsert lab-oppgaver med økende vanskelighetsgrad. Årsaken til dette var at studentene skulle få best mulig læringsutbytte. Vi har anvendt noen av CIROS-modellene for den digitale versjonen av den fysiske fabrikken. Og koblet disse opp mot TIA-Portalen via PLCSIM. En test av lab-oppgavene ble gjennomført med en testgruppe av studenter fra fagskolen. Resultatene av denne testen avdekket noen feil i veiledningene og lab-oppgavene, men oppnådde på generell basis et godt resultat. På denne måten fikk vi bekreftet at lab-oppgavene oppfylte kravene satt av oppdragsgiver.

Det ble ikke mulig å teste de forskjellige lab-oppgavene mot de tilhørende fysiske modulene, grunnet tekniske feil og nedetid på fabrikken. I tillegg har vi laget nedlastning og brukerveiledninger til studentene og faglærer for å oppnå sammenkoblingen. Nedlastning –og brukerveiledninger oppfyller kravene satt til oss, og har fått godkjenning fra oppdragsgiver. De var detaljerte og har blitt testet med gode resultater.

7.3 Videre arbeid

Gjennom prosjektet ble det notert at PLCSIM Advanced V2.0 var noe ustabil i sammenkoblingen av CIROS og TIA-portalen. Dette fikk vi bekreftet av en ingeniør fra Festo som anbefalte å prøve V3 istedenfor V2. Derfor vil vi anbefale som fremtidig arbeid å undersøke de andre sammenkoblingsmetodene diskutert i kapittel 3, og finne den som best

passer formålet. Det kan gjøres videre arbeid på lab 4 og modellen benyttet, ved å utrede de tekniske feilene å finne en løsning på disse. Vi mener modellens vanskelighetsgrad er for høy for studentene ved fagskolen, og vurderer derfor at den ikke hadde vært hensiktsmessig å benytte i undervisning.

Referanser

- [1] Festo Didactic SE, «CP Lab Linear (CP-L-LINEAR) - Festo Didactic InfoPortal», *CP Lab Linear (CP-L-LINEAR)*. <https://ip.festo-didactic.com/InfoPortal/CPFactoryLab/hardware/base/details.php?model=CP-L-LINEAR&lang=en> (åpnet 30. mars 2023).
- [2] Festo Didactic SE, «iDrilling (CP-AM-iDRILL) - Festo Didactic InfoPortal», *iDrilling (CP-AM-iDRILL)*. <https://ip.festo-didactic.com/InfoPortal/CPFactoryLab/hardware/application/details.php?model=CP-AM-iDRILL&lang=en> (åpnet 30. mars 2023).
- [3] Festo Didactic SE, «Press (CP-AM-PRESS) - Festo Didactic InfoPortal», *Press (CP-AM-PRESS)*. <https://ip.festo-didactic.com/infoportal/CPFactoryLab/hardware/application/details.php?model=CP-AM-PRESS&lang=en> (åpnet 12. april 2023).
- [4] «Fagskulen Vestland | Fagskule og vidareutdanning». <https://www.fagskulen.no/> (åpnet 30. mars 2023).
- [5] K. Martinsen, «den fjerde industrielle revolusjon», *Store norske leksikon*. 27. januar 2023. Åpnet: 19. april 2023. [Online]. Tilgjengelig på: https://snl.no/den_fjerde_industrielle_revolusjon
- [6] C. P. credit "Christoph R. at AllAboutLean.com", *English: Illustration of Industry 4.0, showing the four «industrial revolutions» with a brief English description. See also File:Industry 4.0 NoText.png for a version without text.* 2015. Åpnet: 19. april 2023. [Online]. Tilgjengelig på: https://commons.wikimedia.org/wiki/File:Industry_4.0.png
- [7] «Business divisions | Festo NO». https://www.festo.com/no/en/e/about-festo/company/business-divisions-id_3703/ (åpnet 19. april 2023).
- [8] «Siemens Quarterly Profit Surge Comes With Cautious Outlook», *Bloomberg.com*, 7. november 2019. Åpnet: 20. mai 2023. [Online]. Tilgjengelig på: <https://www.bloomberg.com/news/articles/2019-11-07/siemens-expects-2020-decline-in-some-businesses-amid-car-slump>
- [9] «What is a digital twin? | IBM», *What is a digital twin?* <https://www.ibm.com/topics/what-is-a-digital-twin> (åpnet 19. april 2023).
- [10] SumitAwinash, *English: This is the digital twin of Oil rig*. 2020. Åpnet: 19. april 2023. [Online]. Tilgjengelig på: https://commons.wikimedia.org/wiki/File:Oil_rig_Jan_23.jpg
- [11] «Virtual simulation | Festo NO». https://www.festo.com/no/en/e/technical-education/digital-learning/virtual-simulation-and-modelling-id_31275/ (åpnet 11. april 2023).
- [12] Festo Didactic, «CIROS 7, universal 3D simulation system CIROS», *ciros-7 universal 3d simulation-system*. https://www.festo.com/se/en/p/ciros-7-universal-3d-simulation-system-id_PROD_DID_8140772/?page=0 (åpnet 11. april 2023).
- [13] «CODESYS», *Wikipedia*. 17. mars 2023. Åpnet: 12. april 2023. [Online]. Tilgjengelig på: <https://en.wikipedia.org/w/index.php?title=CODESYS&oldid=1145172568>
- [14] «Hva er OPC og OPC UA?», *Novotek Norge*. <https://www.novotek.no/insights/hva-er-opc-og-opc-ua/> (åpnet 12. april 2023).
- [15] Fagskulen på Vestlandet, «Studieplan for fagskoleutdanning i Robotteknologi og digital produksjon i industrien». Fagskulen på Vestlandet, 15. juli 2020. Åpnet: 28. mars 2023. [Online]. Tilgjengelig på: <https://www.fagskulen.no/siteassets/studietilbod/studieplan/studieplan-robotteknologi-og-digital-produksjon---revidert-20.08.20.pdf>
- [16] «CP System Combination FL 2». Festo Didactic SE, 2019.
- [17] «EzOPC. Get the software safely and easily.», *Software Informer*, 8. april 2023. <https://ezopc.software.informer.com/5.5/> (åpnet 25. april 2023).
- [18] «CODESYS Control Win SL». <https://store.codesys.com/en/codesys-control-win-sl-bundle.html> (åpnet 27. april 2023).
- [19] «Festo Learning Experience», *Festo LX*. <https://lx.festo.com/en> (åpnet 25. april 2023).
- [20] «fagskolens-egenart---forum-for-fagskoler-01-03-2016.pdf». Åpnet: 19. april 2023. [Online]. Tilgjengelig på: <https://www.abelia.no/contentassets/e2f22f88c2b34ed899a20eb61fba1de1/fagskolens-egenart---forum-for-fagskoler-01-03-2016.pdf>

- [21] Festo Didactic SE, «CP Factory High-bay storage for pallets (CP-F-ASRS32-P) - Festo Didactic InfoPortal», *CP Factory High-bay storage for pallets (CP-F-ASRS32-P)*. <https://ip.festo-didactic.com/InfoPortal/CPFactoryLab/hardware/station/details.php?model=CP-F-ASRS32-P&lang=en> (åpnet 30. mars 2023).
- [22] «S7-PLCSIM Advanced». <https://mall.industry.siemens.com/mall/en/WW/Catalog/Products/10317038#Overview> (åpnet 25. april 2023).
- [23] G. Industrie-Automatisierungssysteme, «Siemens Aktiengesellschaft».
- [24] «Switching from robot systems to Cartesian handling systems». <https://www.linearmotiontips.com/switching-robot-systems-cartesian-handling-systems/> (åpnet 26. april 2023).
- [25] Editor, «Switching from robot systems to Cartesian handling systems», *Linear Motion Tips*, 7. november 2014. <https://www.linearmotiontips.com/switching-robot-systems-cartesian-handling-systems/> (åpnet 26. april 2023).
- [26] «D15000100121836.pdf». Åpnet: 26. april 2023. [Online]. Tilgjengelig på: <https://www.festo.com/media/pim/836/D15000100121836.PDF>
- [27] «CP Factory High-bay storage for pallets (CP-F-ASRS32-P) - Festo Didactic InfoPortal datablad». <https://ip.festo-didactic.com/InfoPortal/CPFactoryLab/hardware/station/datasheet.php?model=CP-F-ASRS32-P&lang=en> (åpnet 1. mai 2023).

Appendiks A Lister

A.1 Forkortelser og ordforklaringer

TIA-Portal	Totally Intergrated Automation Portal
PLS	Programmerbar Logisk Styring
MES	Manufacturing Execution System
Festo LX	Festo Learning Experience
SCL	Structured Control Language
I/O	Input/Output
CODESYS	Controller development system
IDE	integreert utviklings miljø
OPC	Open Platform Communications
OPC UA	Open Platform Communications Unified Platform
EzOPC	Easy OPC
BUS	Binary Unit System
PLCSIM Advanced	Programmable Logic Controller SIMulation Advanced
CPU	Central Processing Unit
COSIMIR	COmputer SIMulation and Robotics
CIROS	Computer-Integrated Robot System
Soft-PLS	Software programmable logic controller
PASCAL	Programming Language for Algorithmic and Symbolic Computation

A.2 Figurer og tabeller

Figur 1: Logo Fagskolen på Vestlandet.....	9
Figur 2: Den fjerde industrielle revolusjonen. Kilde:[6]	9
Figur 3: Eksempel på MPS-fabrikk i CIROS	10
Figur 4: Digital tvilling av olje rigg. Kilde:[10]	10
Figur 5: Illustrasjon oppsett av fabrikken. Kilde: [16]	14
Figur 6: Bilde av EasyPort enhet, Kilde: [16].....	14
Figur 7: Blokkdiagram av sammenkobling med fysisk PLS	15
Figur 8: Skjerm bilde av EzOPC oversikt tab. Kilde: [17].....	15
Figur 9: Eksempel brukergrensesnitt CODESYS Control Windows. Kilde: [18].....	16
Figur 10: Eksempel grensesnitt til TIA i CIROS	17
Figur 11: Festo LX logo. Kilde: [19].....	18
Figur 12: Feilmelding møtt på i installasjon, skjermdump av Windows PC	19
Figur 13: Filene etter problem er fikset, skjermdump av Windows pc	19
Figur 14: Eksempel på feilmelding etter ustabil PLCSIM	20

Figur 15: Eksempel på koblingen fikset	20
Figur 16: Eksempel på funksjonsblokkdiagram.....	21
Figur 17: Eksempel på hvordan man legger til iDrilling modulen til transportbånd modulen.	22
Figur 18: CP-L-Linear i CIROS.....	23
Figur 19: Programkode for styring av transportbånd i TIA-Portalen.....	23
Figur 20: Implementering av funksjonsblokk i Main.....	24
Figur 21: Transportbånd med pneumatisk presse i CIROS.....	24
Figur 22: Programkode for styring av pneumatisk presse.....	25
Figur 23: iDrilling modulen i sammenkobling med transportbånd modulen.....	26
Figur 24: Festo lineær gantry robot lignende til drillen. Kilde: [24].....	26
Figur 25: Illustrasjon av hvordan Festo sine roboter bruker enkle systemer for å lage avanserte systemer. Laget selv i diagrams.net, oversatt versjon av kilde: [25].....	27
Figur 26: Bilde av CECC-LK fra datablad. Kilde: [26]	27
Figur 27: Highbay storage modell i CIROS	28
Figur 28: kartesisk robot med markerte akser.....	28
Figur 29: Flytskjema for ASRS32.....	29
Figur 30: I/O definert for servokontrolleren til x og z akse.....	30
Figur 31: Robot i posisjon hvor den ikke flytter seg lenger langs x akse.....	30
Figur 32: Verdi av koordinat til robot i x akse posisjon fra Figur 31 vist i tag-table.....	30
Figur 33: Første tilbakemelding lab 1. Kilde: Skjermdump	32
Figur 34: Andre tilbakemelding lab 1. Kilde: Skjermdump	33
Figur 35: Tredje tilbakemelding lab 1. Kilde: Skjermdump.....	33
Figur 36: Tilbakemelding fra lab 2. Kilde: Skjermdump	34
Figur 37: Tilbakemelding på lab 3. Kilde: Skjermdump.....	35
Figur 38: Skjermdump av funksjonsblokkdiagrammet i SCL fra lab 1	85
Figur 39: Skjermdump av Main i SCL i lab 1.....	86
Figur 40: Funksjonsblokk CP-L-Conveyor	87
Figur 41: Funksjonsblokk for CP-AM-PRESS	87
Figur 42: Implementering av funksjonsblokker i "Main"	88
Figur 43: Skjermdump av menyen av programblokkene	88
Figur 44: Skjermdump av menyen i TIA-portalen, tag tabellen.....	88
Figur 45: Skjermdump av tag-tabell for lab 3.....	89
Figur 46: Skjermdump av variablene i funksjonsblokken i lab 3	90
Figur 47: Skjermdump av første del av funksjonsblokk koden i lab 3.....	91
Figur 48: Skjermdump av andre del av funksjonsblokk kode i lab 3.....	91
Figur 49: Skjermdump av Main i lab 3.....	92

Appendiks B Prosjektledelse og styring

B.1 Prosjektorganisasjon

For å skape en mer kvalitetssikret arbeidsfordeling, valgte vi å fordele forskjellige titler på oss i gruppen. Dette betydde ikke at vi kun arbeidet med det feltet vi ble fordelt, vi samarbeidet fortsatt med de samme feltene, men vi fikk hvert vårt ansvarsfelt. Fordeling var følgende:

Johannes Eidesvik – overordnet ansvarlig, han fikk ansvaret for at prosjektet fulgte en god plan, og at vi klarte å følge denne planen på en tilstrekkelig måte.

Josef Heimset – kommunikasjonsansvarlig, han fikk ansvaret for å kommunisere spesielt med oppdragsgiver og veileder, for å skape en god kommunikasjon. Dette var i hovedgrunn for å ivareta en god struktur på kommunikasjonen, for å unngå forvirring og gjentakelse av beskjeder.

Daniel Klepsvik – Tekniskansvarlig, han fikk ansvaret for å sette seg dypt inne i det tekniske aspektet av prosjektet, for å passe på at vi leverer et godt produkt som ikke inneholder teknologiske mangler.

B.2 Prosjektform

Grunnet prosjektets kronologiske rekkefølge velger vi å bruke prosjektledelsesmetodikken kalt fossefall-metoden. Vi ser for oss at prosjektet vil gå gjennom 5 steg som følger prosjektmodellen.

1. Kravspesifikasjon: Hvilket behov har oppdragsgiver og hvilke ressurser krever prosjektet. Dette blir dekket i forstudiefasen av prosjektet.
2. Design: Lage en plan og utkast for lab-oppgavene. Lager så modellene og programmene vi må ha i CIROS og TIA-portalen for gjennomføring av oppgavene.
3. Utførelse: Setter i gang arbeidet med å lage modellene og oppgavene.
4. Verifikasjon: I dette stadiet vil vi teste ut oppgavene på fagskolestudenter og revidere oppgavene etter behov og tilbakemeldinger.
5. Implementering: Overlever alle dokumenter, modeller, programmer og brukerveiledninger til oppdragsgiver og avslutt prosjektet ved levering av bachelorrapport og endelig presentasjon.

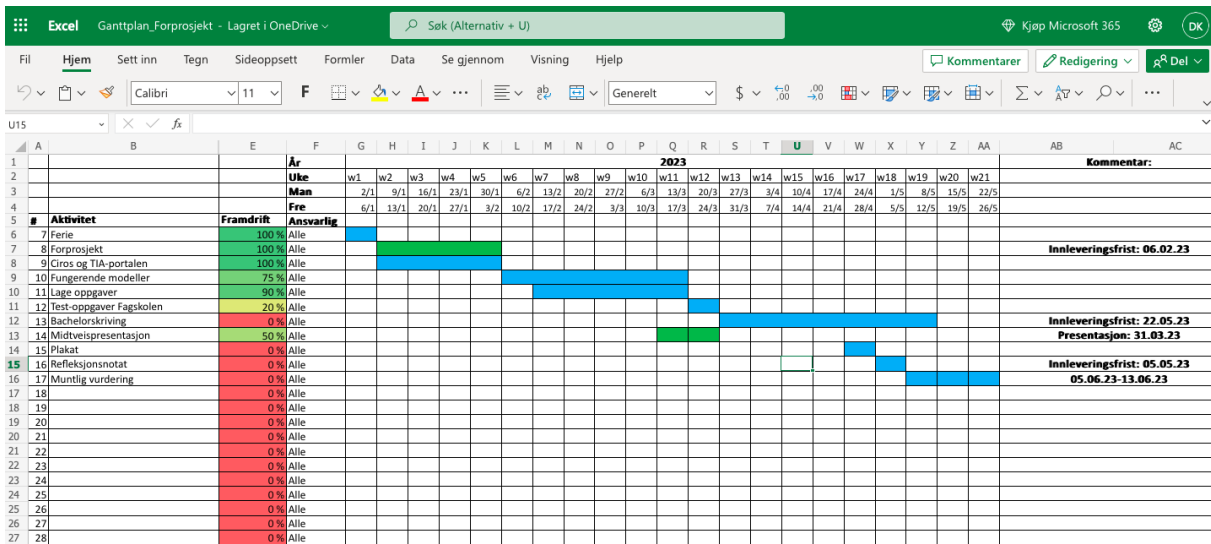
Dette var planen når vi startet, men vi har måtte gå litt frem og tilbake, men vil si vi har holdt oss ganske trofast til metoden.

B.3 Fremdriftsplan

Vi startet prosjektet med å lage en Gantt-plan for å planlegge hvordan vi kunne få arbeidet på en strategisk og smart måte. I figuren under kan dere se hvordan vår originale plan var:



Vi fulgte planen veldig godt i starten, men ettersom vi startet å sitte mer fast på enkelte punkter, endte vi opp med å måtte revidere Gantt planen vår til følgende:



Dette ble vår endelige gantt plan som vi fulgte til vi var ferdig.

Appendiks C Brukerdokumentasjon

Et viktig punkt i bachelor oppgaven var å lage brukerveiledninger og nedlastningsveiledninger, følgende vil komme i dette appendikset.

C.1 Nedlastningsguide CIROS

CIROS nedlastning

Du vil først trenge en VPN du bruker mens du laster ned CIROS, dette er slik at du kan kjøre CIROS hvor som helst. Du vil trenge en authenticator for å logge inn på nedlastningssiden til VPN'en. Her anbefaler jeg å laste ned «Microsoft authenticator» på telefonen, og logge inn med iskule kontoen din.

1. Gå til siden: <https://ssl.sfj.no/dana/home/index.cgi>
2. Her skal du logge på med kun brukernavn, ikke inkludert email.
3. Bruk authenticator til å skanne QR koden, og skriv inn mottatt kode i skrivefeltet. 3.

Trykk på download

Pulse Secure Application Launch

ssl.sfj.no/dana/home/psalwait.cgi?app=pulse&b=%2Fdana%2Fhome%2Findex.cgi&c=%2Fdana%2F...

Vestland fylkeskommune

Searching for the application launcher...

55

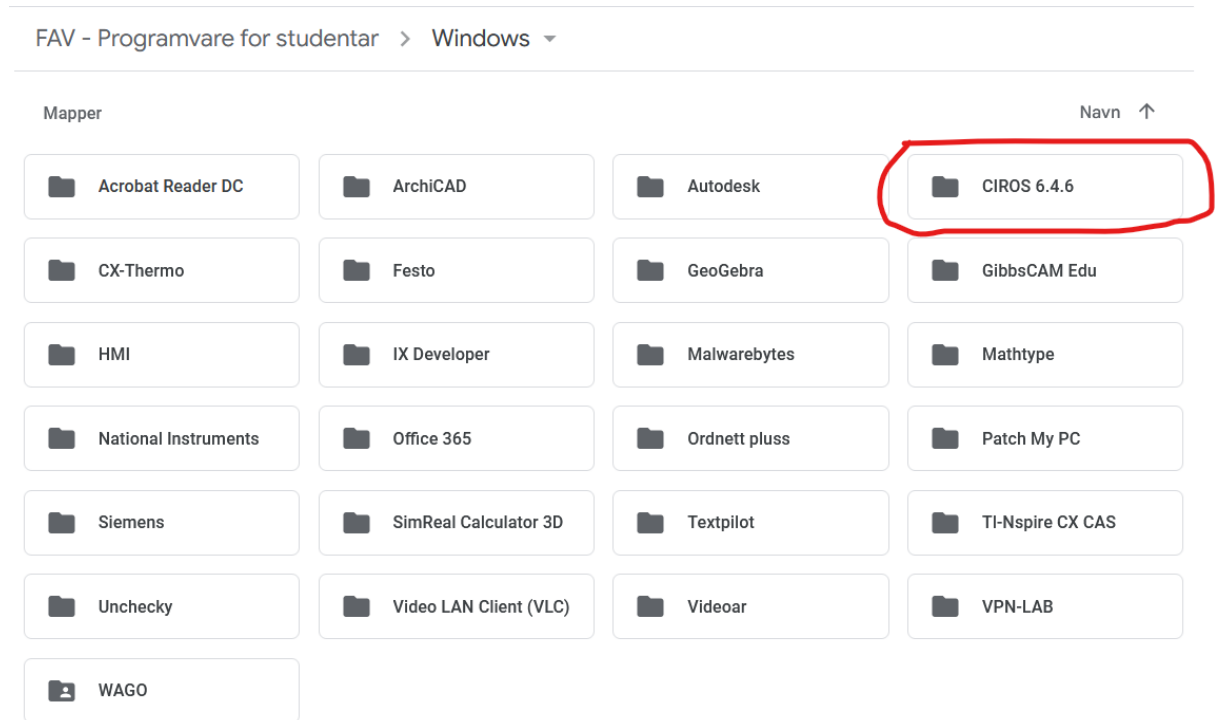
If you know that the application launcher is not installed, skip the wait and download now

Download

If you do not want to proceed, please click [here](#) to go back.

4. Følg nedlastnings instruksene.

5. Gå på google drive, og last ned CIROS 6.4.6



6. Nedlastet filer:

> Dokumenter > CIROS ▾ ↻

Navn	Endringsdato	Type	Størrelse
CIROS 6.4.6-20230118T090916Z-002	18.01.2023 10:12	Komprimert (zipp...	761 042 kB
Setup_Ciros_x64-001	18.01.2023 10:14	Program	3 083 691 ...

7. Pakk ut zippet mappe.
8. Koble deg til VPN
9. Åpne mappen og kjør «Start».

» Dokumenter » CIROS » CIROS 6.4.6-20230118T092036Z-001 » CIROS 6.4.6

Navn	Endringsdato	Type	Størrelse
➔ Addons	18.01.2023 10:25	Filmappe	
➔ CIROS	18.01.2023 10:25	Filmappe	
➔ Documentation	18.01.2023 10:25	Filmappe	
➔ Tools	18.01.2023 10:25	Filmappe	
📄 autorun	18.01.2023 10:25	Informasjon om in...	1 kB
📄 festo	18.01.2023 10:25	ICO-fil	1 kB
📄 Net_Inst_CM	18.01.2023 10:25	Firefox PDF Docu...	1 217 kB
📄 NetzInst_CM	18.01.2023 10:25	Firefox PDF Docu...	1 221 kB
📄 Start	18.01.2023 10:25	Program	857 kB
📄 Start	18.01.2023 10:25	Konfigurasjonsinns...	25 kB

10. Da får du opp følgende vindu:

CIROS® 6.4 - Copyright © 2019 Festo Didactic SE

FESTO

CIROS® 6.4

The whole virtual learning environment of CIROS® on one DVD.

- ➔ **CIROS® Applications**
Click here to install the CIROS applications.
- ➔ **Additional Applications**
Click here to install MES4 and Python.
- ➔ **Online Service**
Links to CIROS® 32-bit version, Adobe Reader, Festo Didactic and CIROS®

Language
Select your language here

EN - English

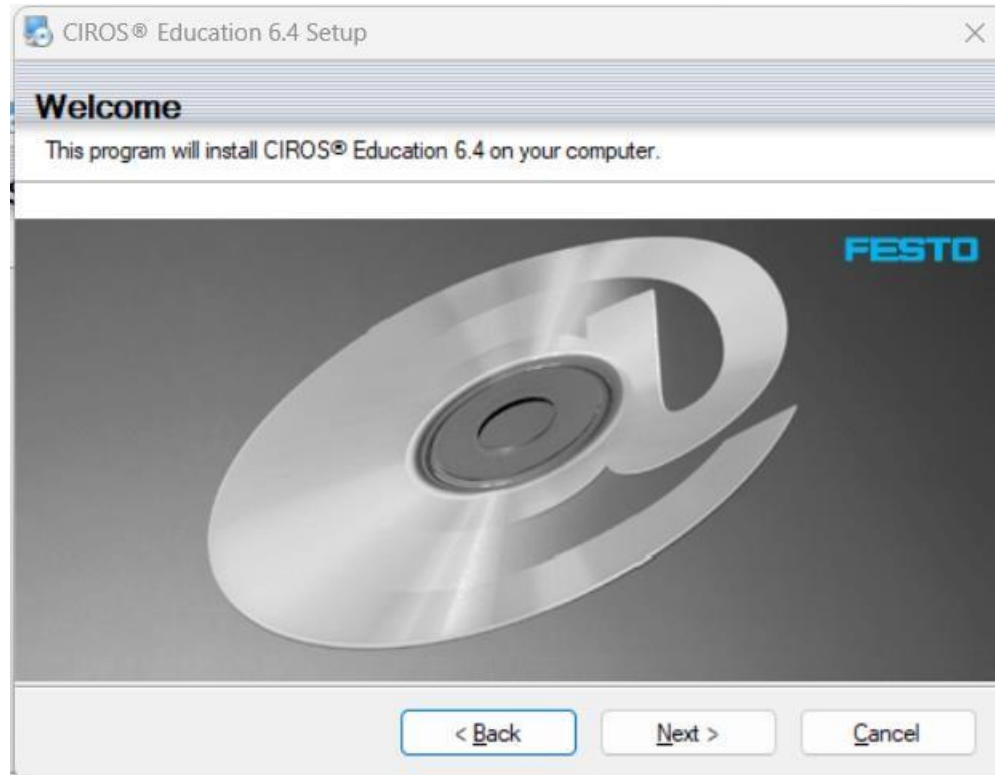
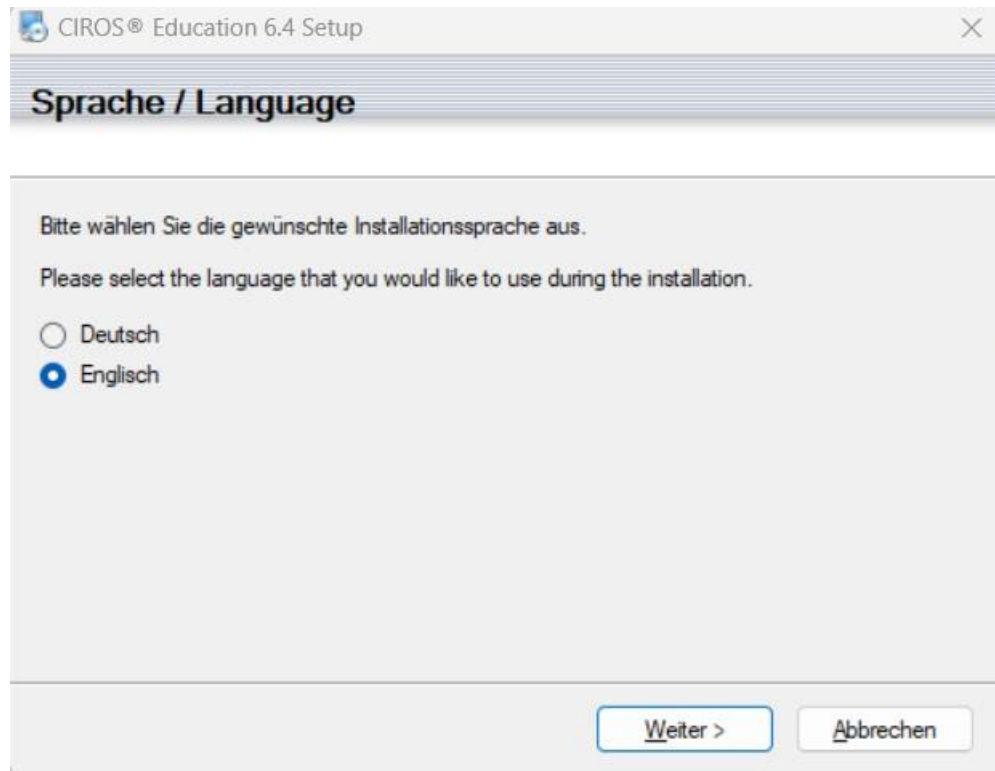
License Agreement
Make sure to read the attached license agreement carefully

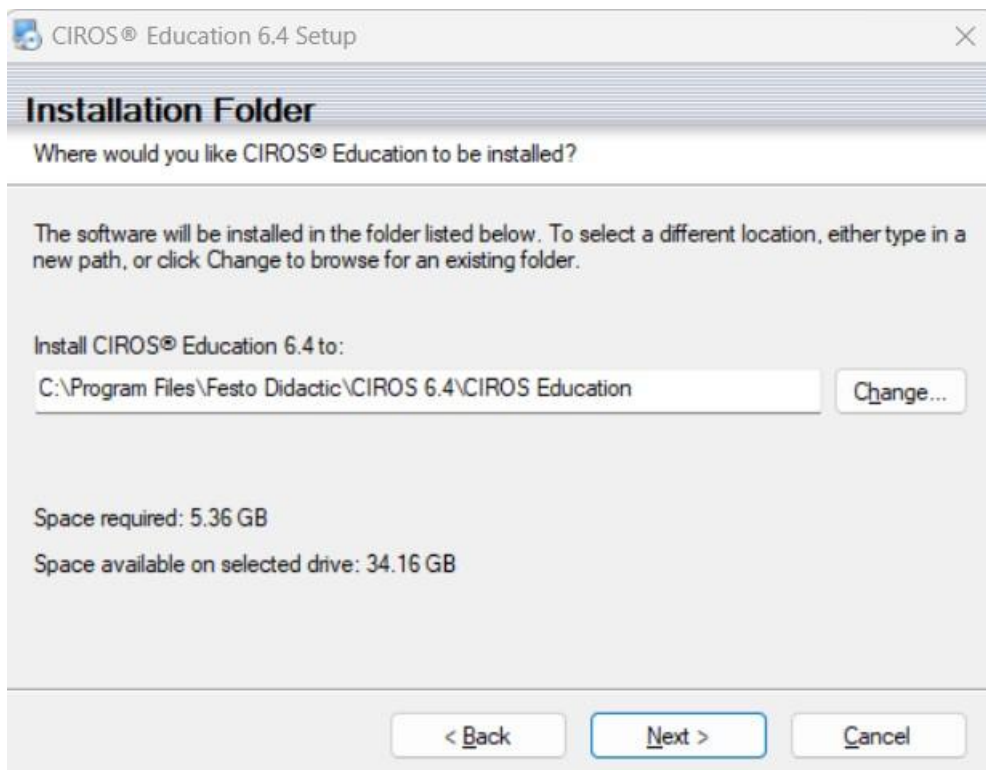
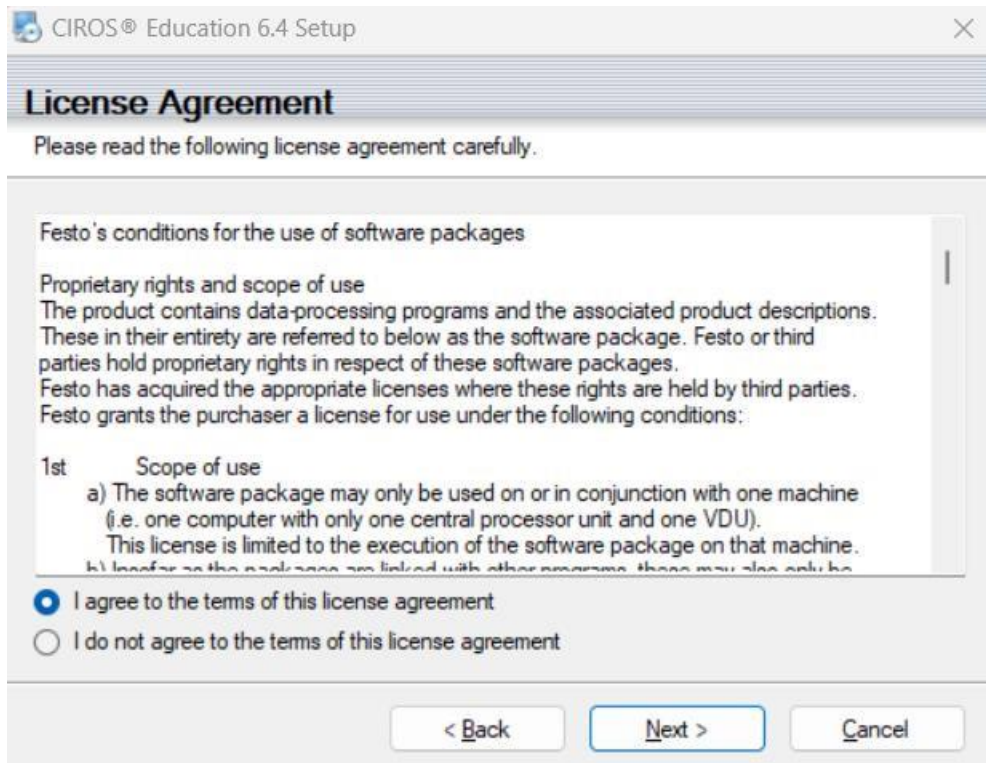
Show

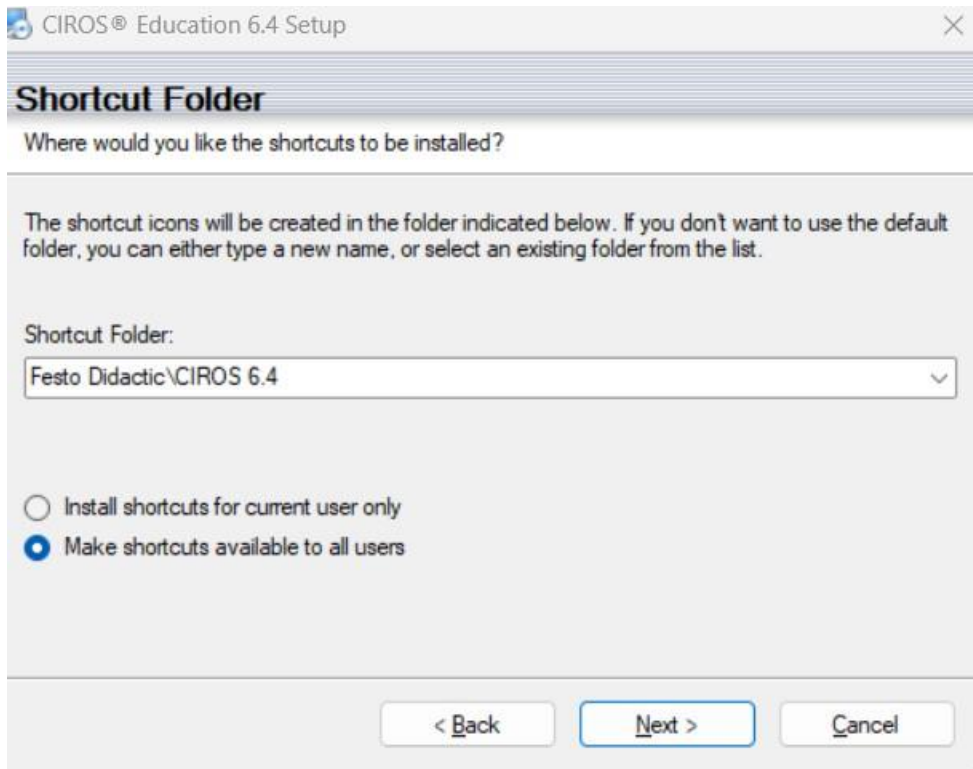
Exit

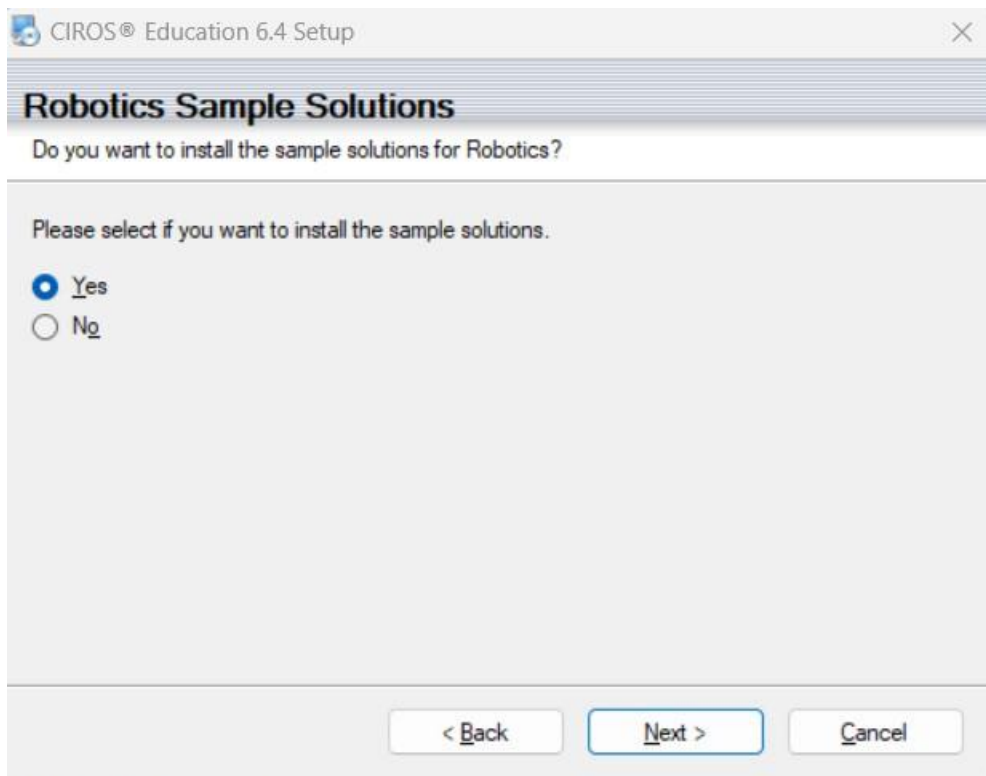
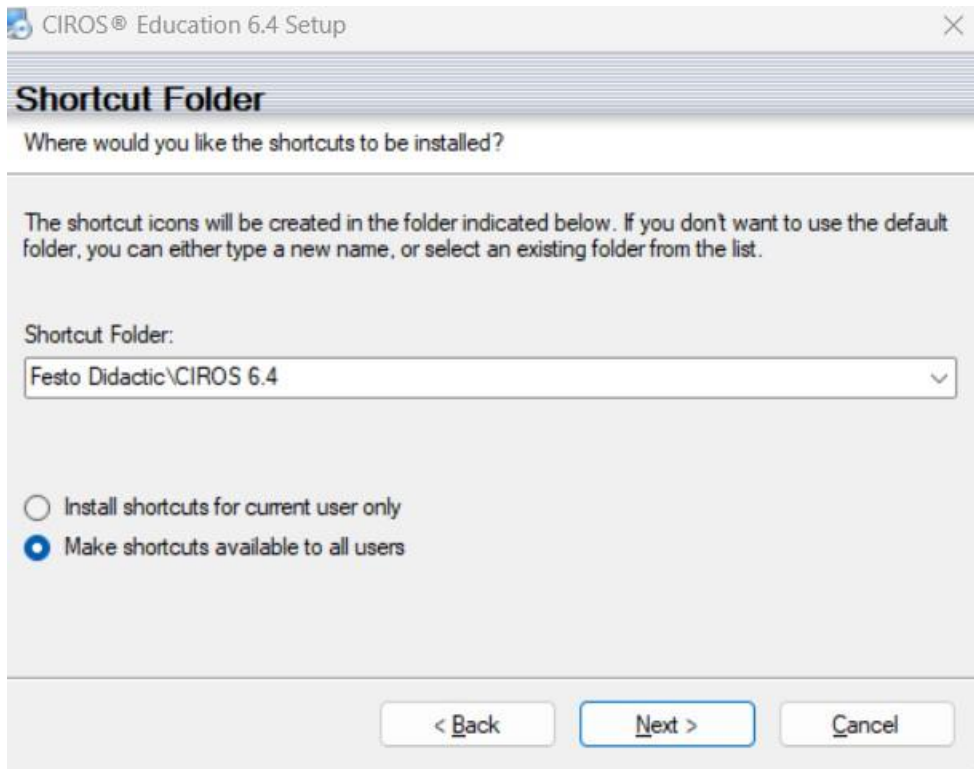
11. Trykke på CIROS Applications.

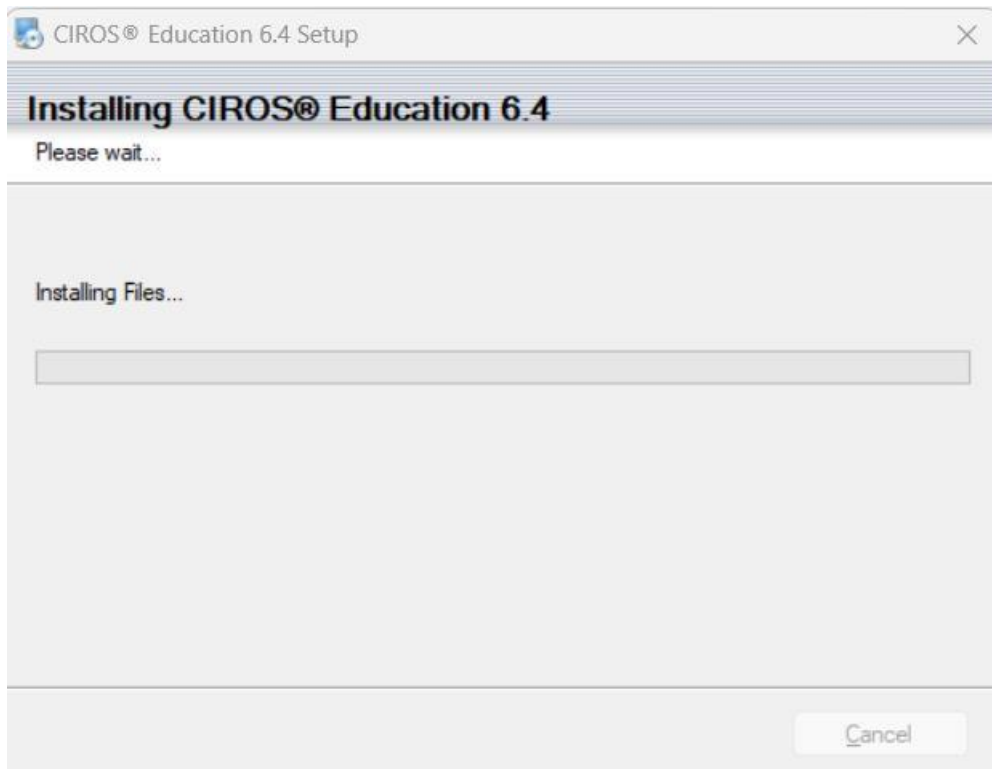
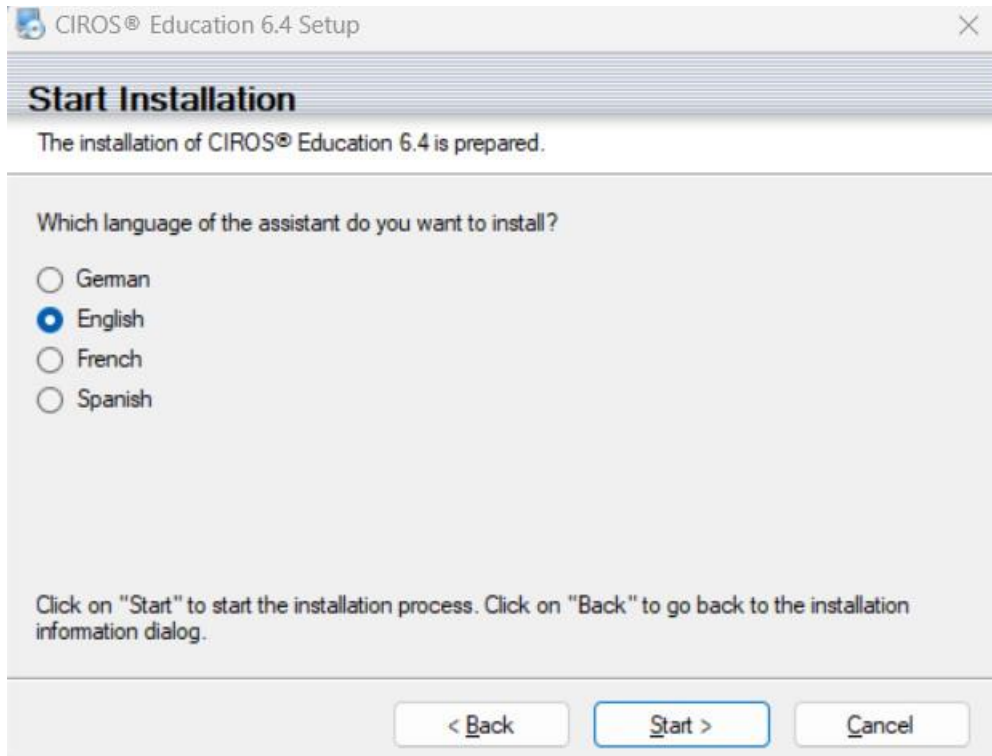
12. Velg CIROS Education





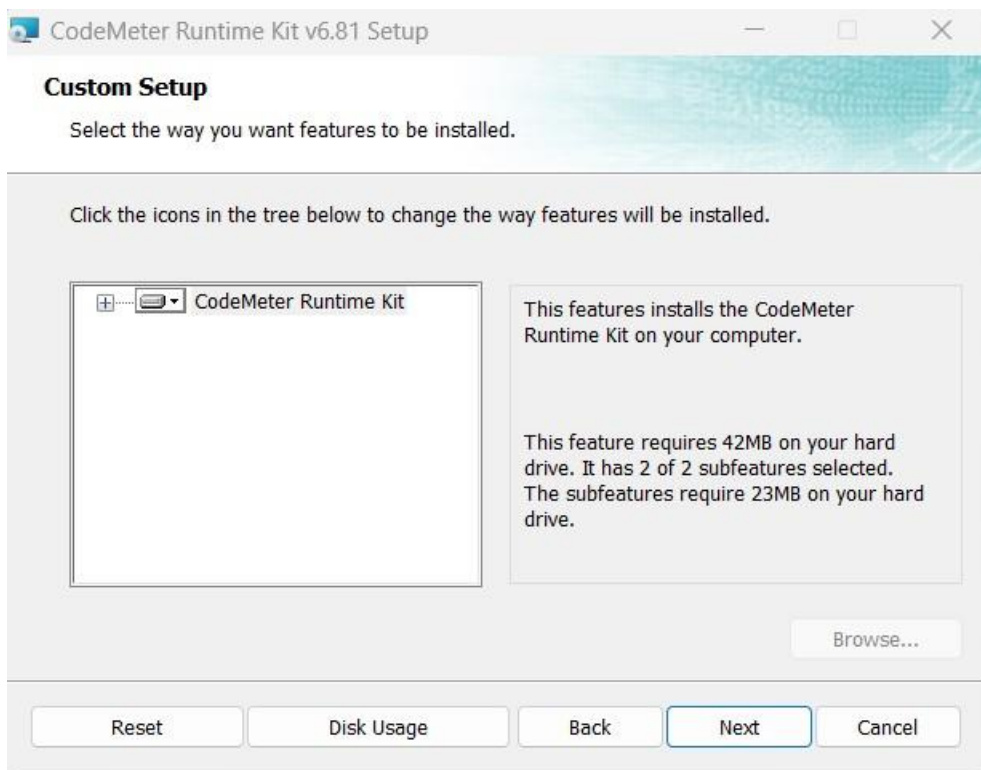
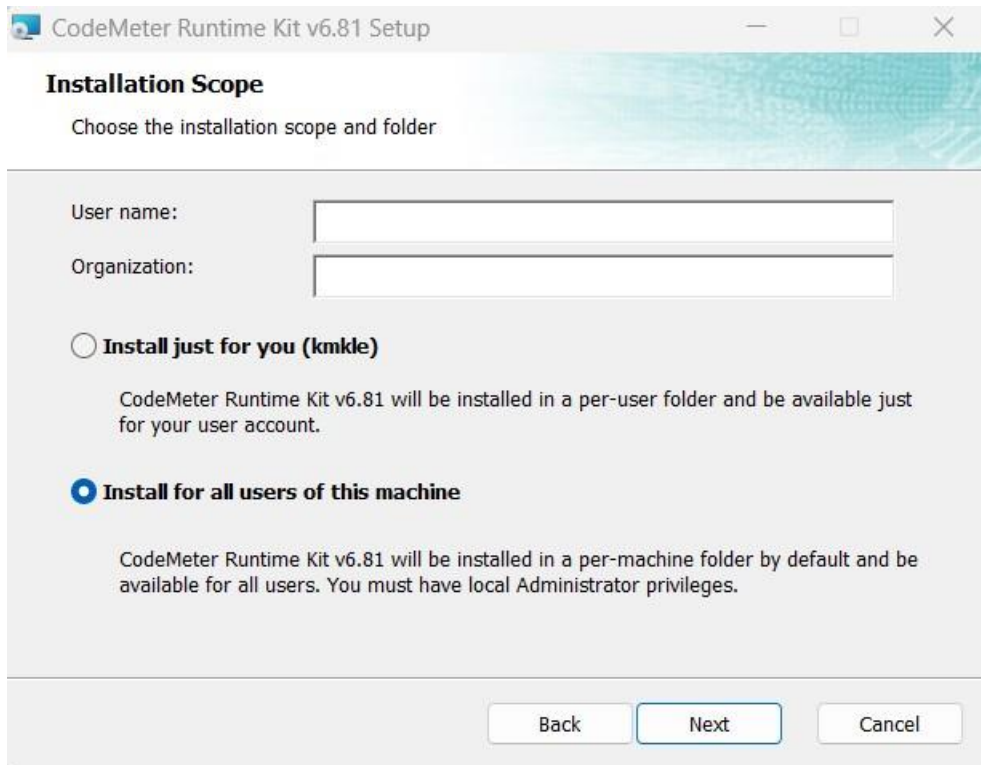


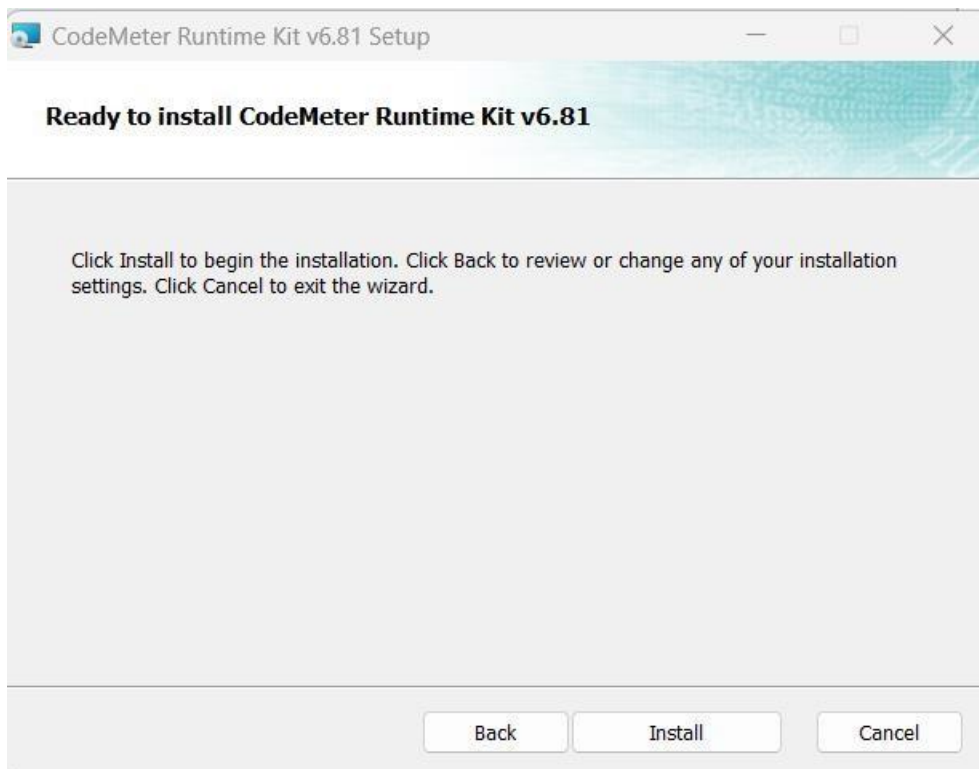




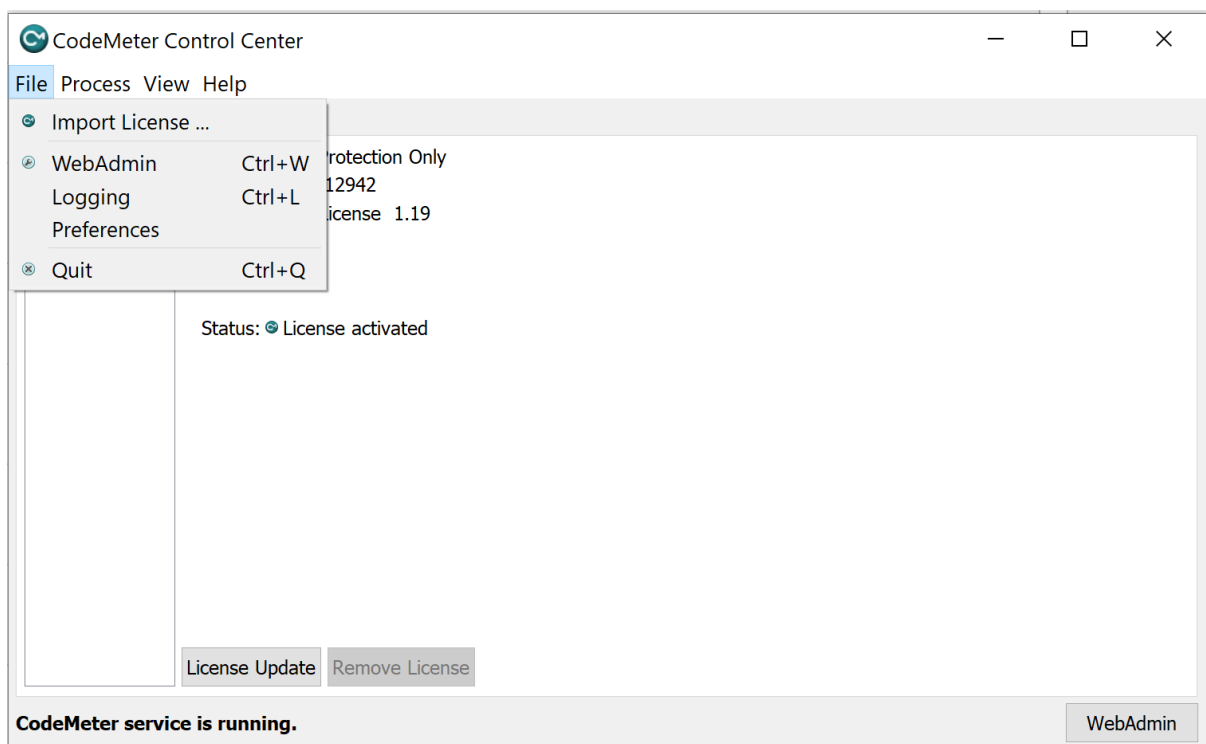


13. Da vil følgende vindu poppe opp:

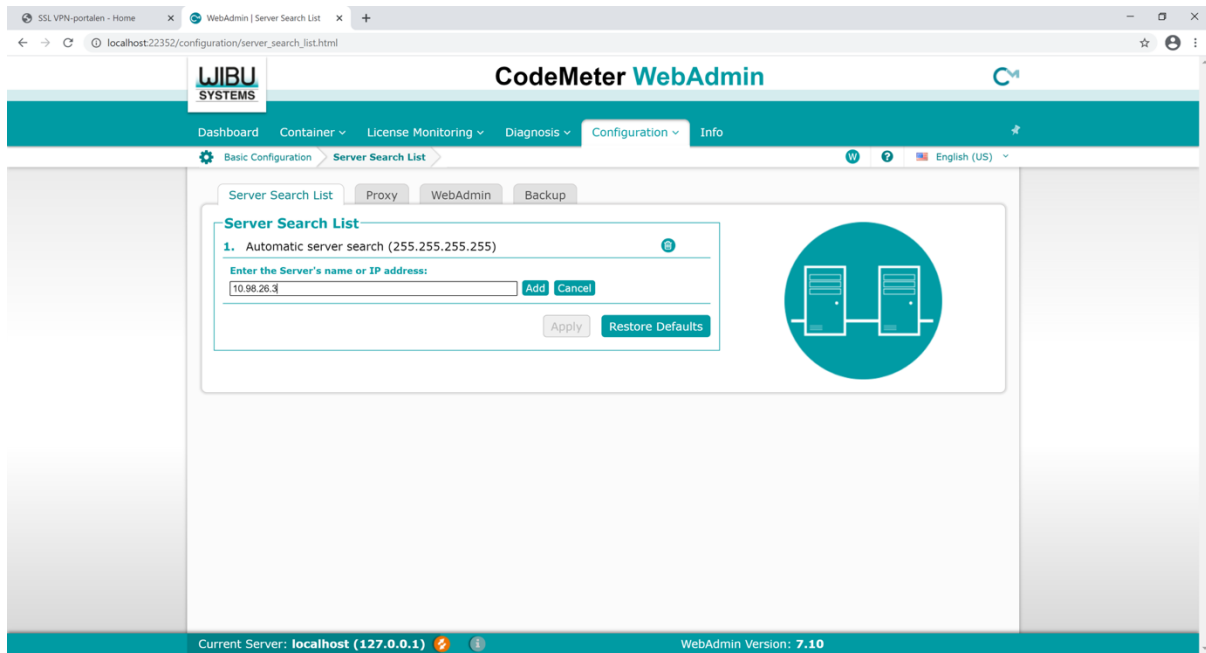




14. Åpne CodeMeter:



15. Skriv inn følgende IP-adresse: 10.x.xxx.xx (kan ikke gi IP-adressen her)



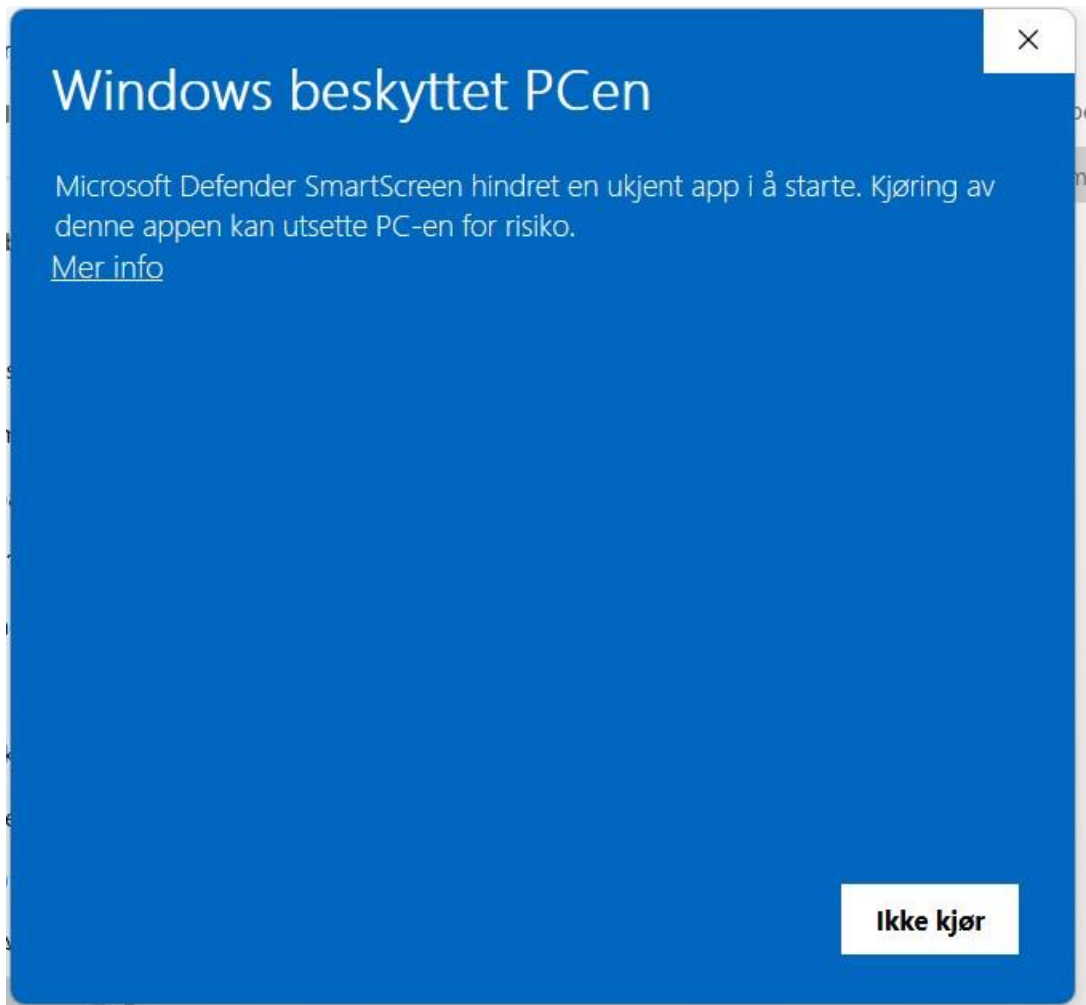
16. Følgende feilmelding kan dukke opp!



17. Da må du flytte Setup_Ciros_x64 inn i CIROS mappen og passe på den har riktig navn:

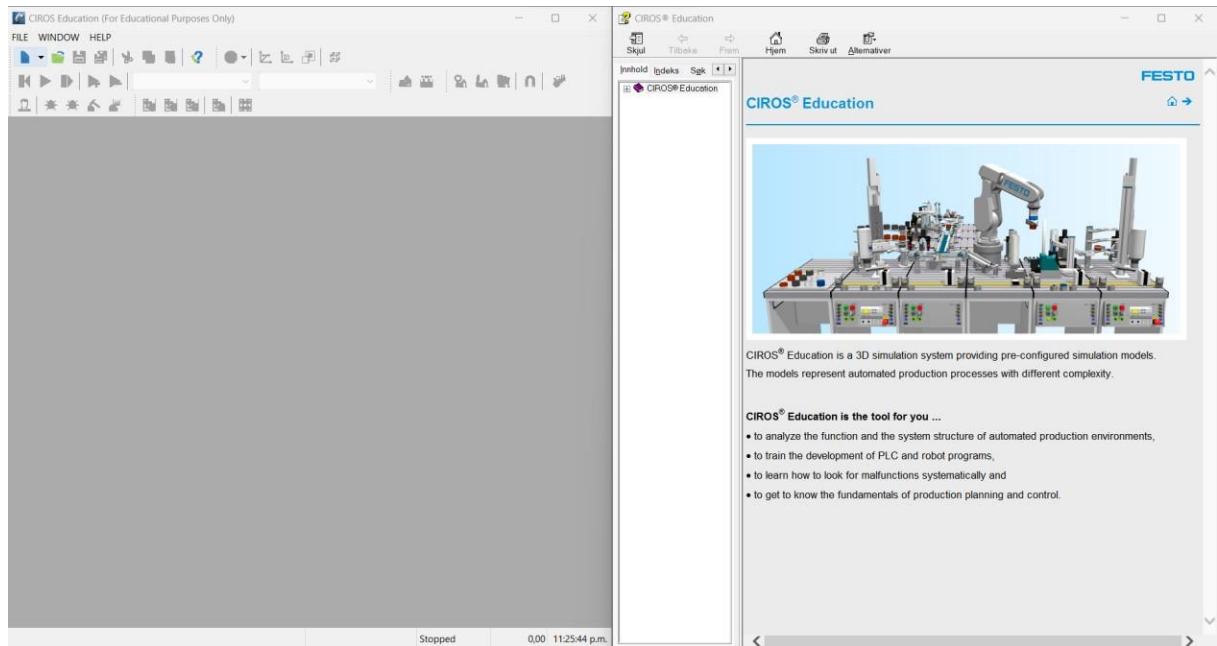


18. Får du opp denne feilmeldingen:



Trykker du bare på «Mer info», og «kjør likevel» knappen som dukker opp.

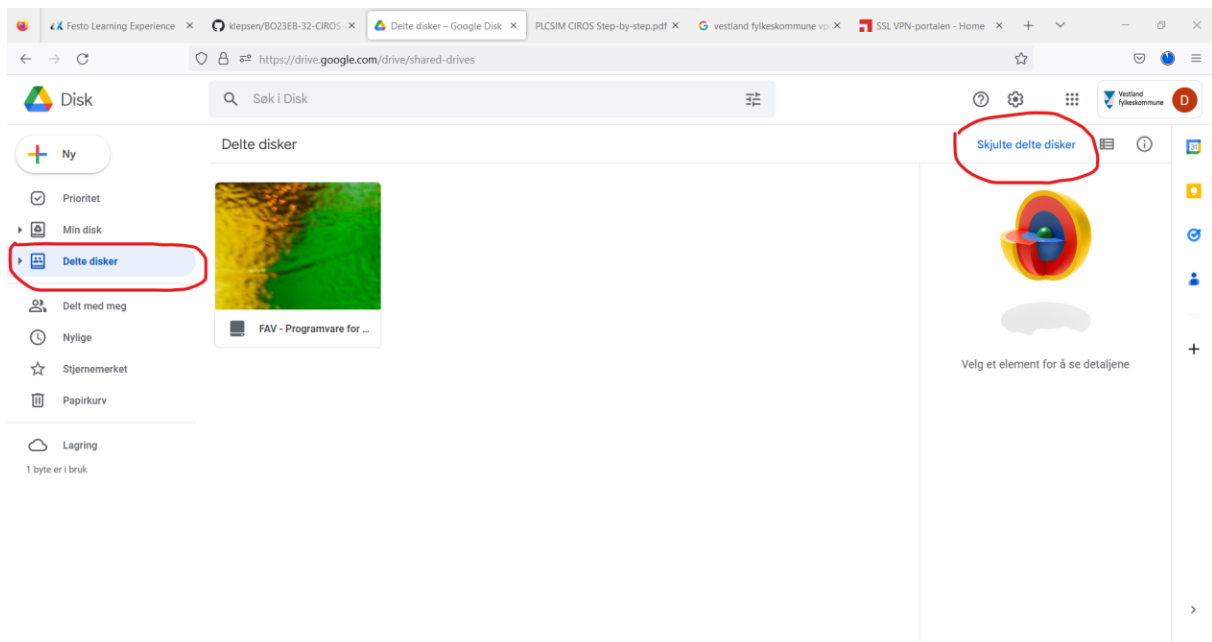
19. Da bør Ciros være klar:



C.2 Nedlastningsguide TIA-portalen

PLS nedlastning

1. Logg inn på Google Drive med iskule brukeren din
2. Hvis det ikke er noe delte filer på drive'en din, trykk på «vis skjulte filer»



Figur 1 Google drive siden

3. Trykk innpå FAV, så windows.
4. Der finner du en mappe som heter SIEMENS, den kan du markere, og laste ned. Hvis du ikke ønsker SIEMENS LOGO som er en mini PLS simulator, trenger du kun disse:

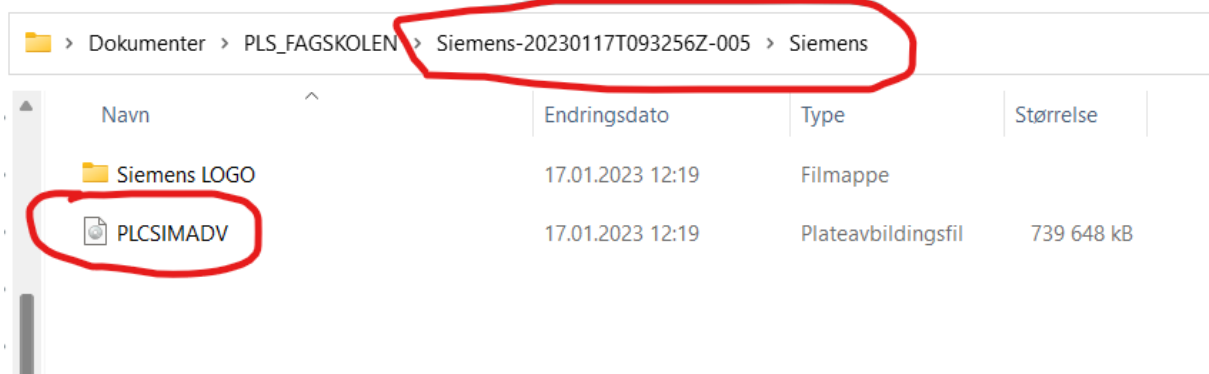
Navn	Endringsdato	Type	Størrelse
Siemens-20230117T093256Z-002	17.01.2023 12:18	Filmappe	
Siemens-20230117T093256Z-005	17.01.2023 12:19	Filmappe	
TIAPORTAL STEP7ProWinCCAdv2-001	17.01.2023 12:01	Plateavbildingsfil	6 853 632 ...
TIAPORTAL STEP7ProWinCCAdv-003	17.01.2023 12:08	Plateavbildingsfil	7 451 136 ...

Figur 2 Filer du trenger hvis du ikke ønsker LOGO

5. Videre trykker du innpå TIAPORTAL STEP7ProWinCCAdv-003, åpner filer, og trykker på «Start» filen. Den vil da restarte pc'en din, dette kan ofte medføre at du får opp en feilmelding, men da er det bare å gjøre det pånytt.
6. Når du får åpnet installer, må du passe på å krysse av på engelsk språk!
7. Videre gjør du samme stegene for de andre mappene, ikke TIAPORTAL STEP7ProWinCCAdv2-001.

Navn	Endringsdato	Type	Størrelse
Siemens LOGO	17.01.2023 12:18	Filmappe	
PLCSIMADV.MDS	17.01.2023 12:18	MDS-fil	5 kB
S7-PLCSIM V15_1	17.01.2023 12:18	Plateavbildingsfil	1 693 622 ...
TIAPORTAL STEP7ProWinCCAdv.mds	17.01.2023 12:18	MDS-fil	9 kB
TIAPORTAL STEP7ProWinCCAdv2.mds	17.01.2023 12:18	MDS-fil	9 kB

Figur 3 For å laste ned fra fil 002

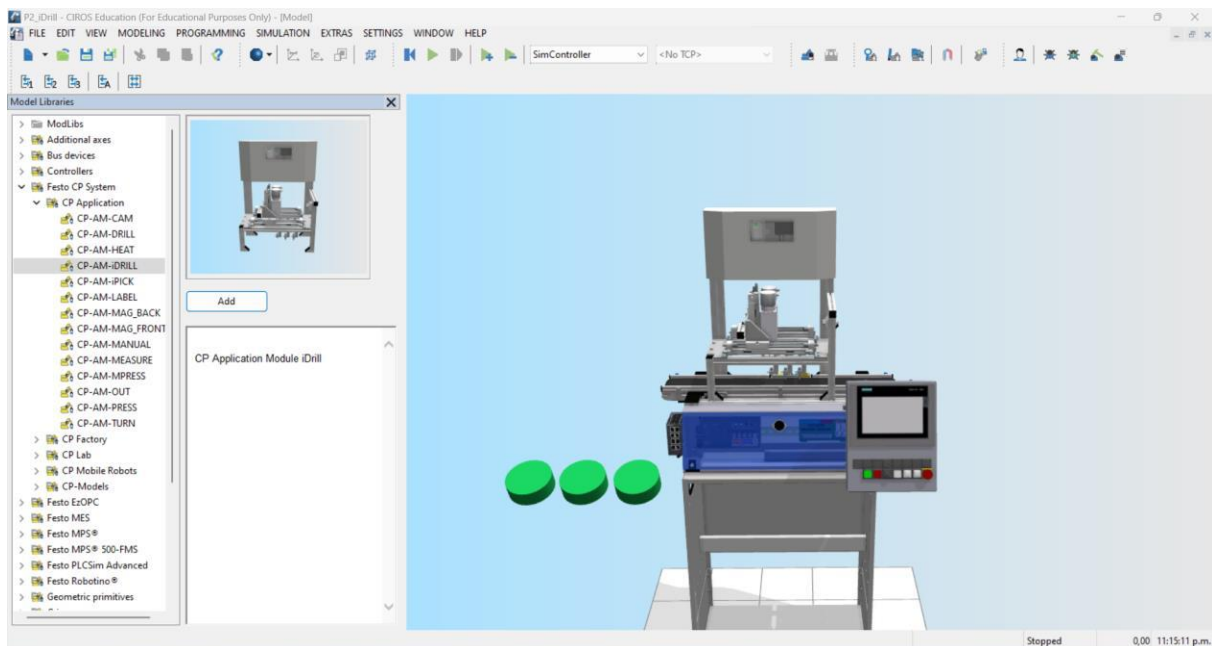


Figur 4 For å laste ned fra mappe 005

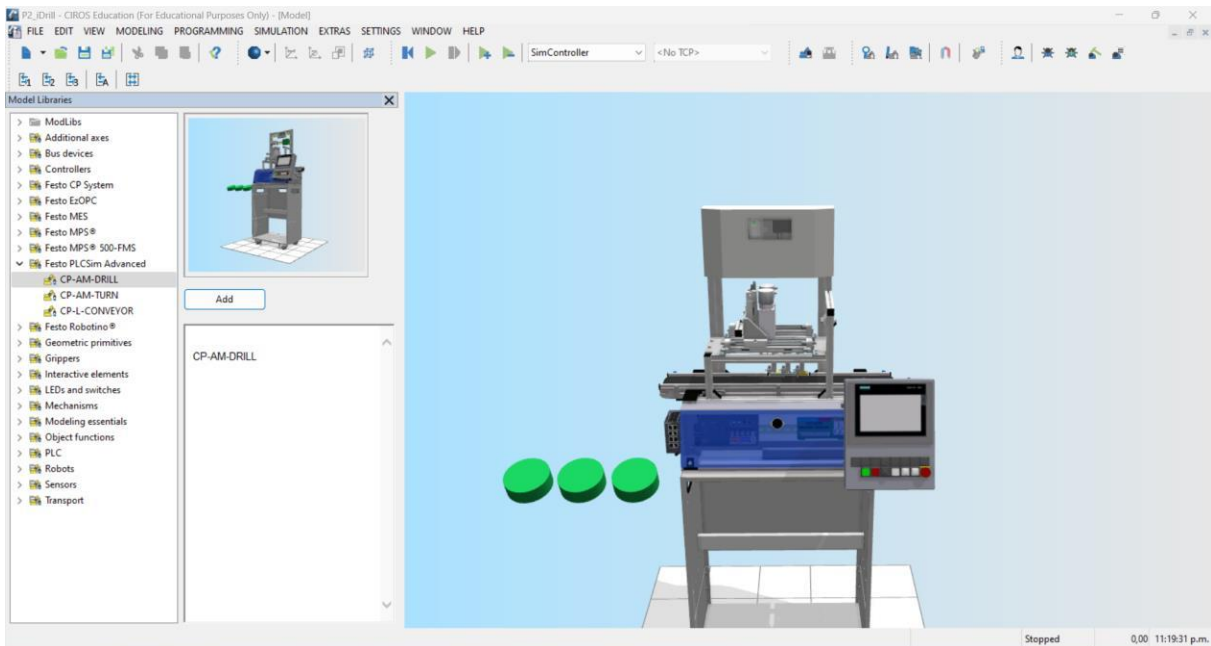
C.3 Guide for å legge til moduler i CIROS

Legg til iDrilling modul

Det første du må gjøre for å legge til iDrilling modulen, er å åpne «Model explorer». Her finner du frem til iDrill under «Festo CP System» og «CP application».

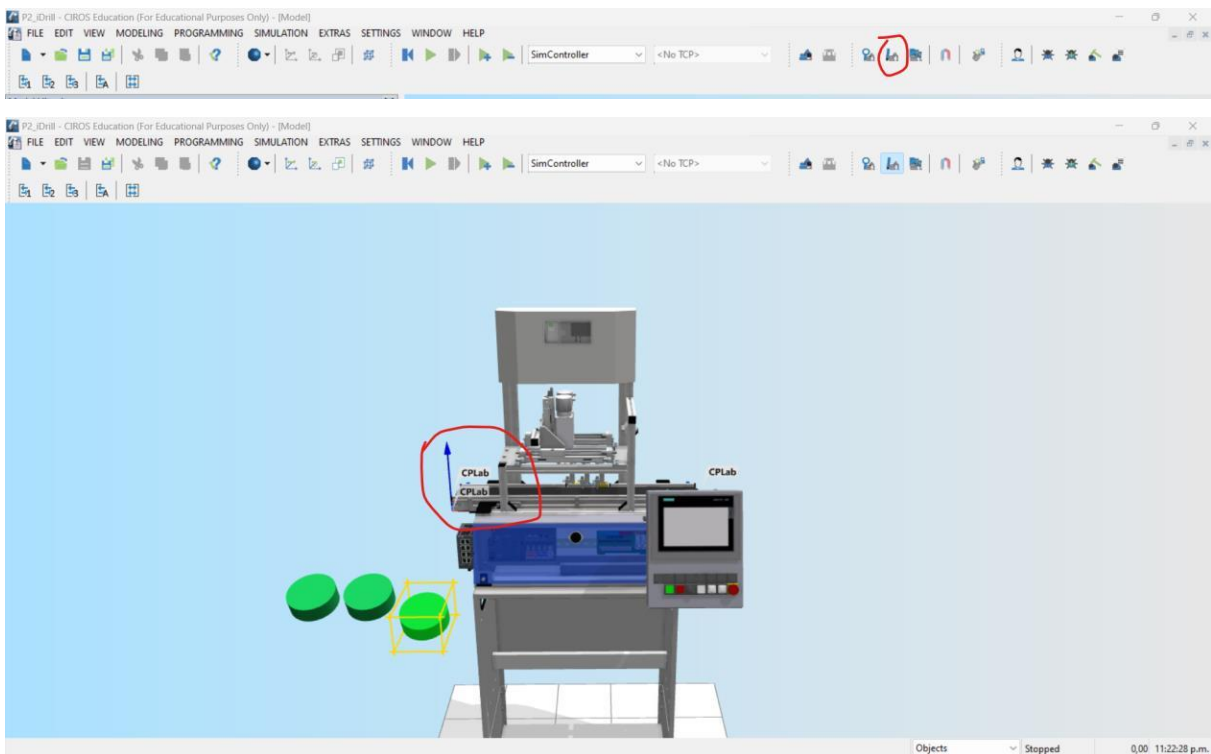


Videre trenger vi en source som kan generere arbeidstykker som inneholder mobildeksel, som skal borres i. Dette er litt tungvindt. Vi må inn og velge «Festo PLCSim Advanced» så legge til «CP-AM-DRILL»



Det du gjør nå, er å slette den originale sourcen vi har på transportbåndet. Så går du i «edit mode»

Så drar du på plass alle tre sourcene fra CP-AM-DRILL. Da må du passe på at CPLab merkene er oppå hverandre:



Dette må sjekkes både forfra og ovenfor.

Da må du innpå nettsiden

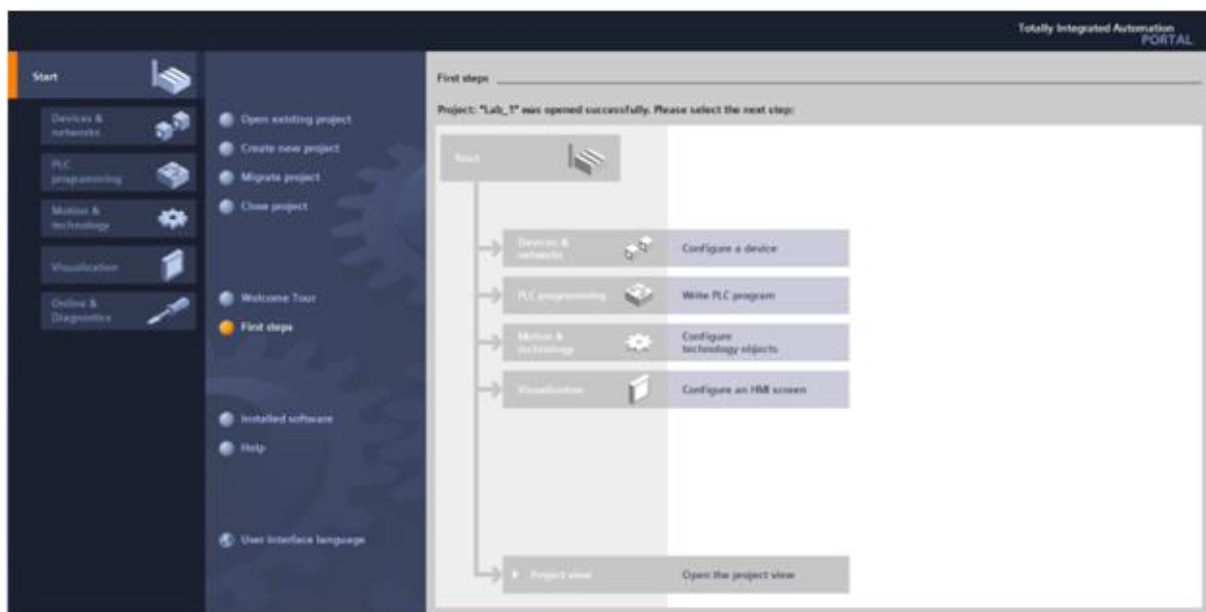
<https://ip.festo-didactic.com/InfoPortal/CPFactoryLab/hardware/application/datasheet.php?model=CP-AM-iDRILL&lang=en>

Og legge inngangene til i tag table i PLS prosjektet ditt. Når dette er gjort, drar du de inn i CIROS sin I/O som gjort tidligere.

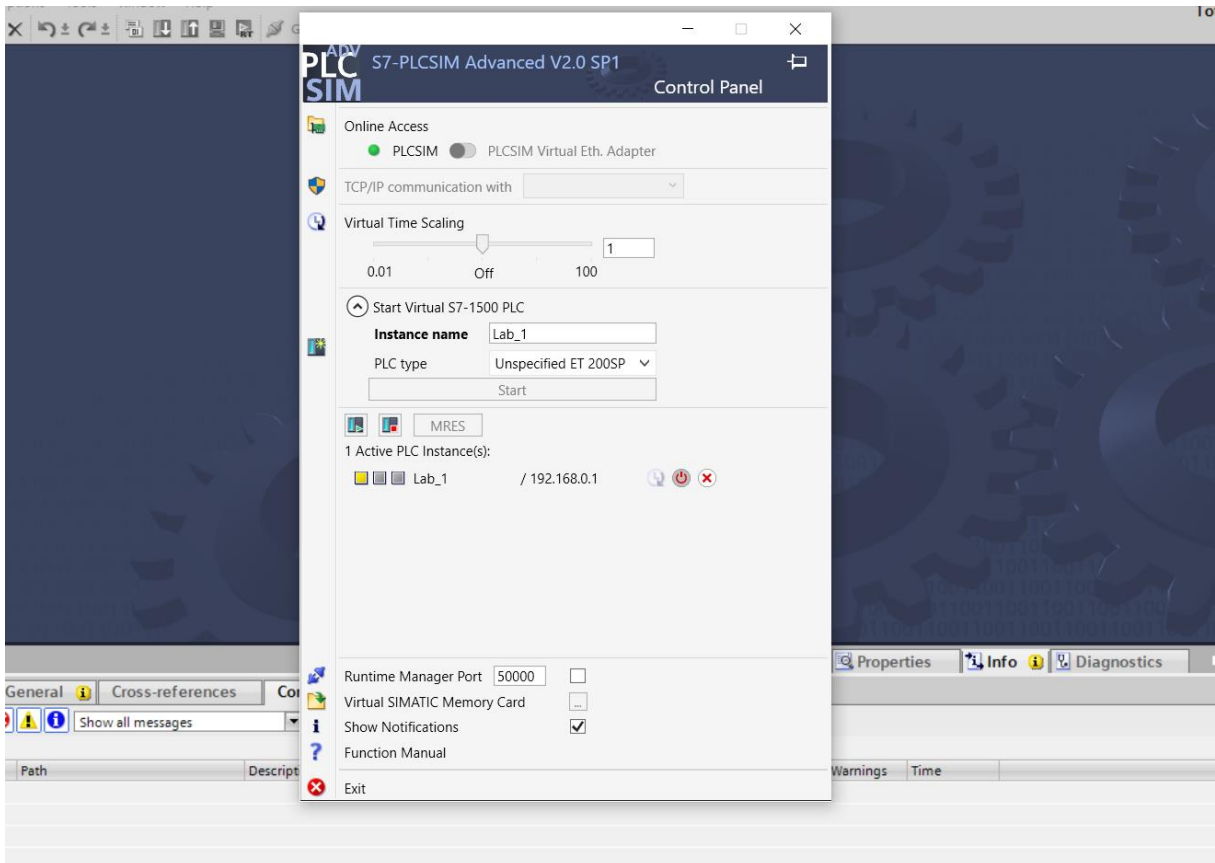
C.4 Guide for å koble sammen TIA-portalen og CIROS

Oppstart

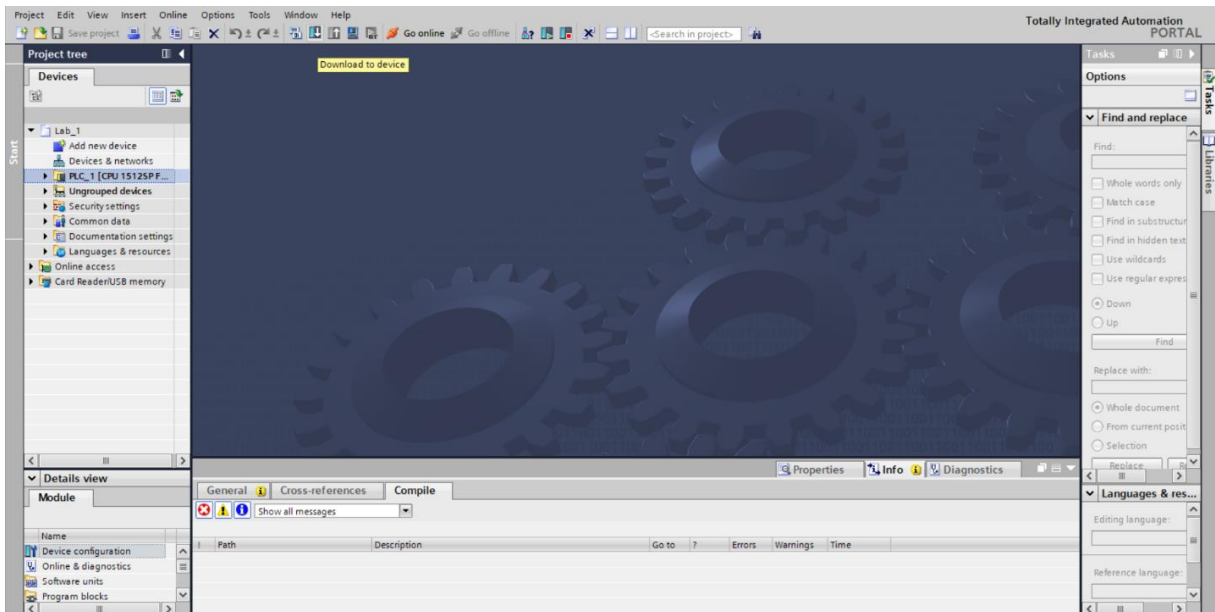
1. Pakk ut zip-filen «Lab_1».
2. Åpne filen «Lab_1» i TIA-Portalen og trykk på «Project view».



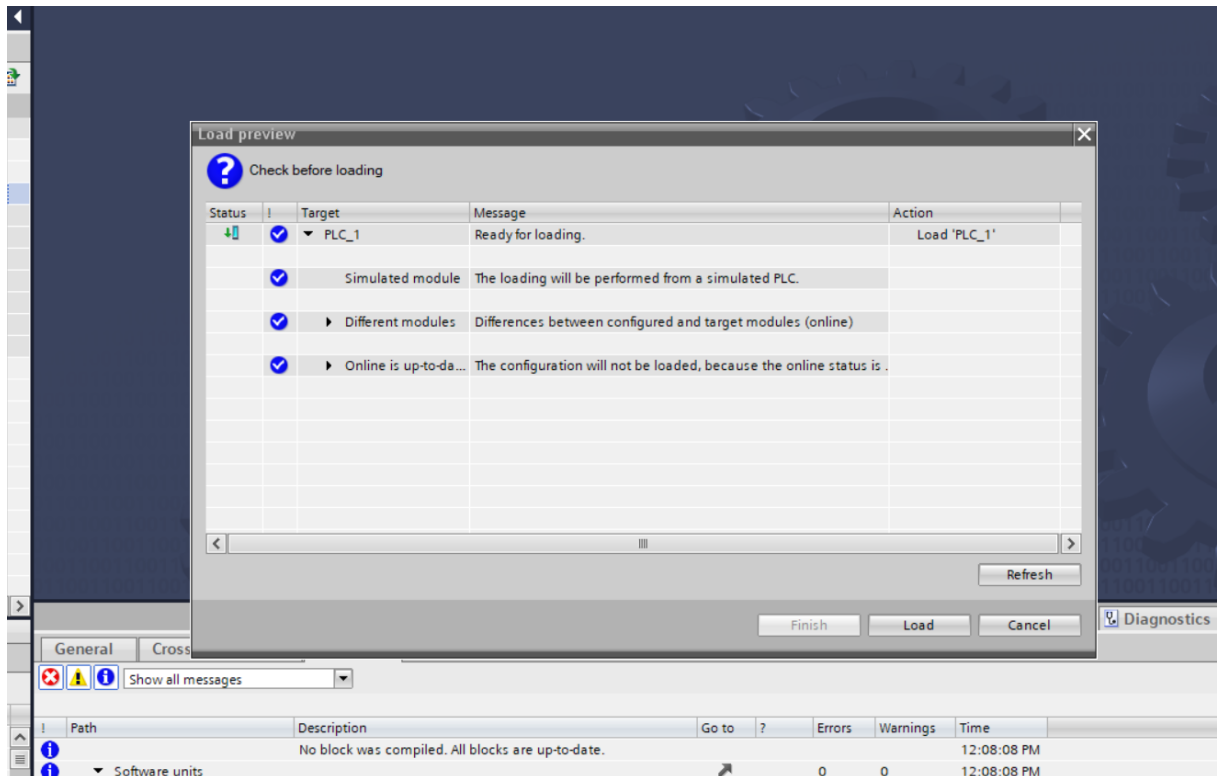
3. Åpne PLCSIM og start en instans med navnet «Lab_1» og PLC-type «Unspecified ET 200SP». Denne instansen må skrus av og på en gang før den godtar nedlastingen fra TIA-Portalen, dette gjør du ved å trykke på «POWER» to ganger.



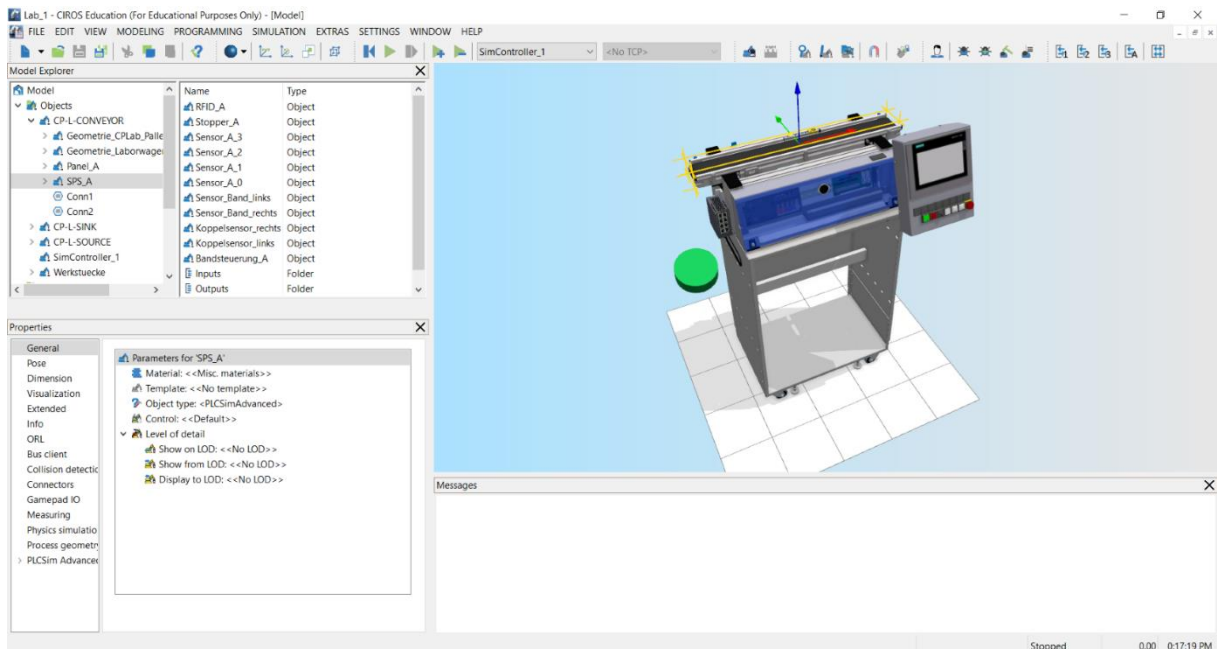
4. Gå tilbake i TIA-Portalen og velg CPU-en din, deretter trykker du på «Download to device».



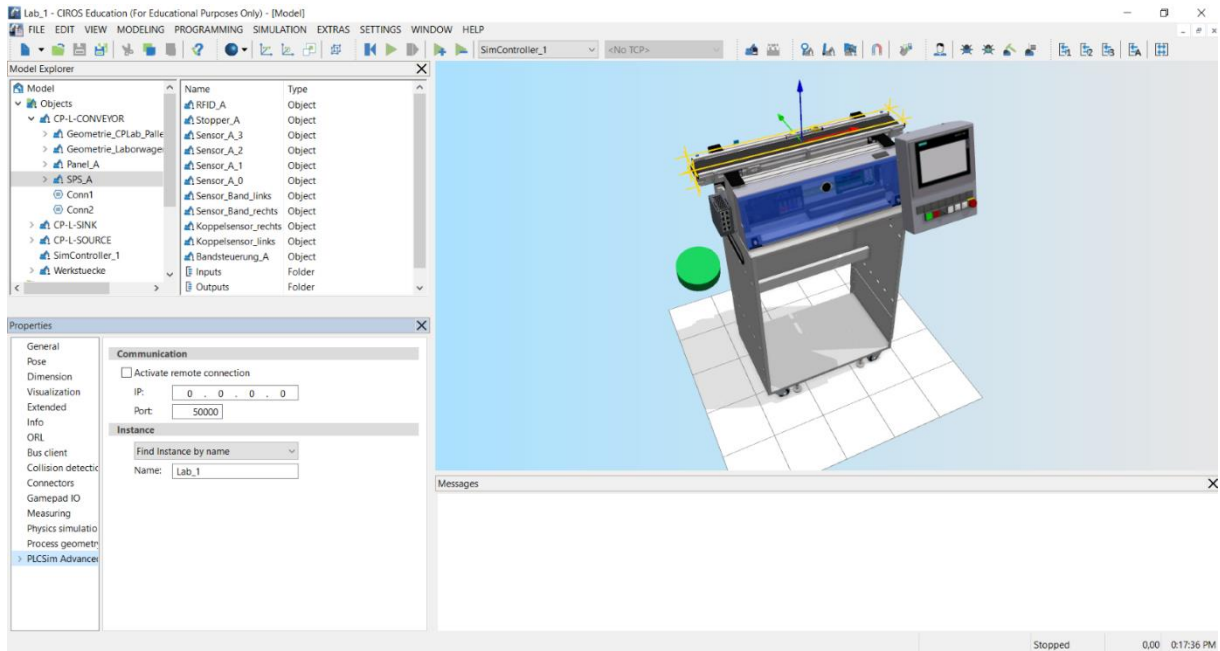
5. Etter du har trykt «Download to device» vil dette vinduet komme, her trykker du først «Load» og deretter «Finish».



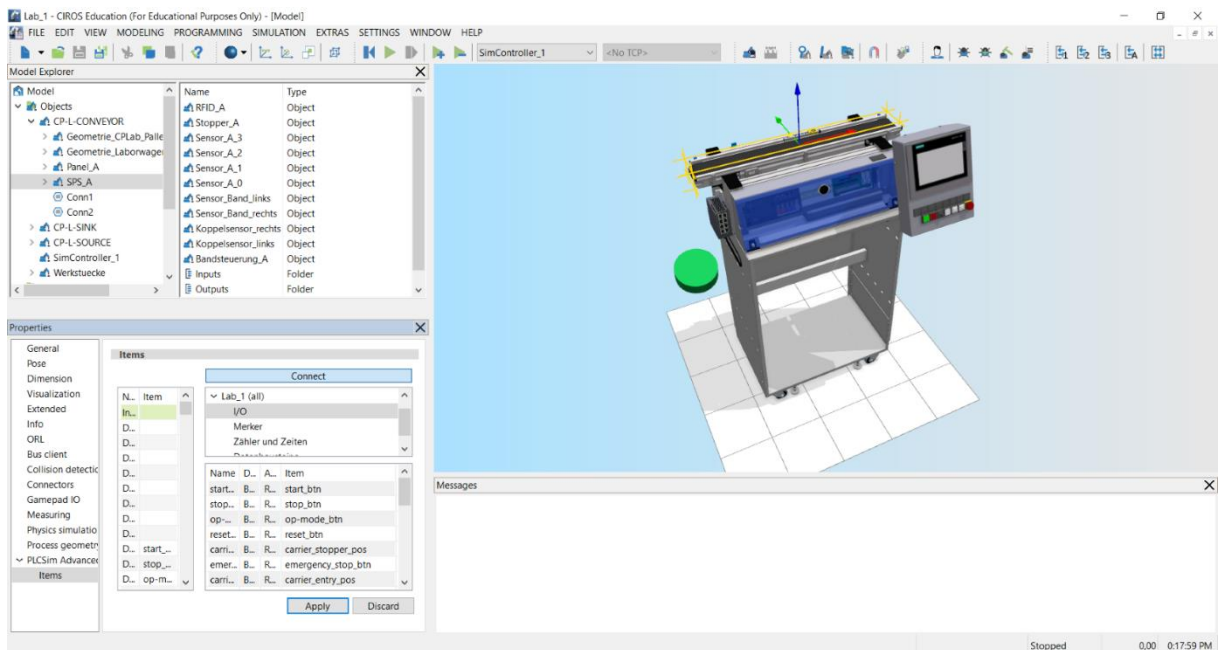
6. Åpne opp CIROS-fila «Lab_1». Når du åpner CIROS vil det se slik ut. Under Model Explorer vil du være i CP-L-CONVEYOR-menyen. Trykk på SPS_A og deretter PLCSim Advanced under Properties.



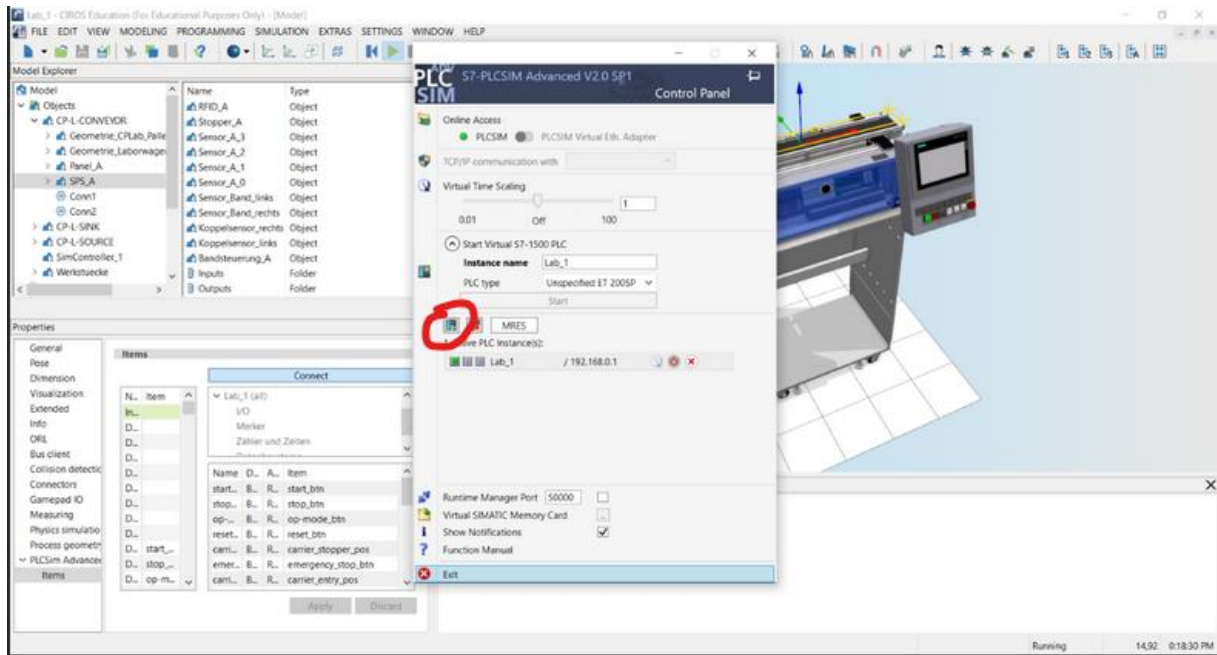
7. Det er viktig at det instansnavnet som står her er det samme som er i PLCSim eller så får du ikke koblet sammen CIROS og TIA. Etter du har passet på å få riktig navn på begge trykker du på den lille pilen til venstre for PLCSim Advanced.



8. Trykk på «Items» og deretter på «Connect». Trykk på pila ved siden av «Lab_1(all)» deretter på I/O og til slutt på «apply».



9. Nå er du klar til å starte simuleringen. Finn fram PLCSim igjen og highlight instansen din, deretter trykker du på Play-knappen(markert). Til slutt trykker du «RUN» i CIROS.



Appendiks D Lab oppgaver

D.1 Lab 1

Laboratorium 1

Conveyor-Belt

Innledning

Lab-oppgavene er laget slik at dere skal bli introdusert til PLS-programmering i TIA-Portalen. Lab-oppgavene vil ha flere forskjellige typer oppgaver der noen av de vil være i funksjonsblokkdiagram, og noen vil være i SCL. Grunnen til at vi bruker CIROS og TIA-Portalen sammen er at det vil gjøre det mulig for dere å teste programmene deres uten å sette fysisk utstyr i fare. Koblingen mellom programmene kan til tider være ustabil og med veldig trege PC-er kan oppsettet være vanskelig å kjøre.

CIROS

I denne labben bruker vi en enkel modell i CIROS som kun inneholder 3 elementer: CP-L-CONVEYOR, CPL-SINK og CP-L-SOURCE. CP-L-SOURCE vil generere et arbeidsstykke som følger CP-L-CONVEYOR og til slutt blir slettet når det ankommer CP-L-SINK. Når arbeidsstykket er kommet halvveis over transportbåndet vil det stoppe slik at en applikasjonsmodul kan få gjøre sitt arbeid før det går videre. I denne labben ser vi kun på transportbåndet, og derfor håndterer vi dette stoppet på en annen måte som vi vil se på nærmere.

TIA-Portalen

Det er laget et lite program i TIA-Portalen som vi vil bruke i denne labben. Programmet gir transportbåndet litt grunnleggende funksjonalitet, men utover det så er det dere som skal programmere resten. Programmet gir transportbåndet to forskjellige modus: manuell og automatisk. I manuell modus vil båndet kun ha to funksjoner. Startknappen vil kjøre båndet framover og stoppknappen vil stoppe båndet. I automatisk modus vil båndet bevege seg fremover når et nytt arbeidsstykke blir generert, og det vil stoppe når arbeidsstykket har nådd enden av båndet. I begge moduser vil den store grønne knappen til venstre for transportbåndet generere et nytt arbeidsstykke.

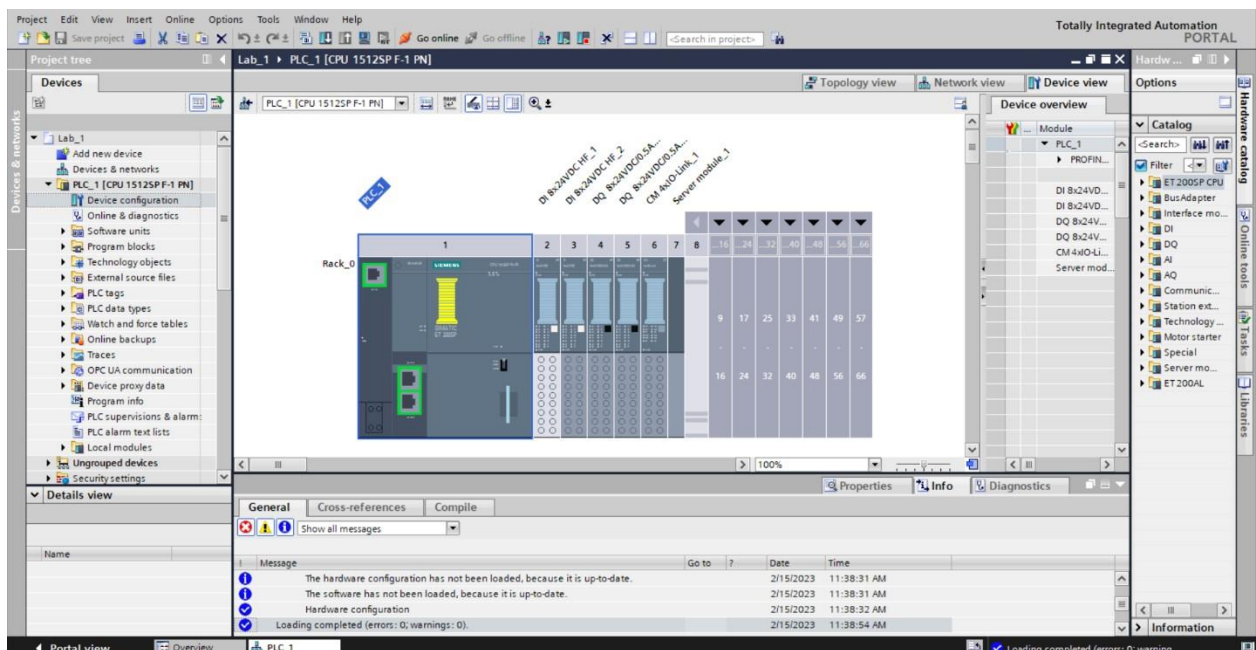
Oppgave 1:

Filen «Introduksjon_Lab.pdf» forklarer hvordan du starter CIROS og TIA-Portalen sammen for første gang. Gå gjennom denne filen og fullfør alle trinnene. Det er viktig at dere følger denne prosessen nøye.

Oppgave 2:

I den andre oppgaven skal du lære hvordan du navigerer i TIA-Portalen og hvilke element som er viktigst for oss i starten. Du skal til slutt observere hvilke verdier som endrer seg når simuleringen kjører. Det første vi skal gjøre er å se på modellen av de fysiske PLS-ene vi har i dette systemet. Trykk på pilen ved siden av CPU-en din og deretter på «Device configuration».

Du skal være komnt til dette vinduet:



Her kan du trykke på alle de forskjellige modulene og deretter på «Properties» for å se på egenskapene deres.

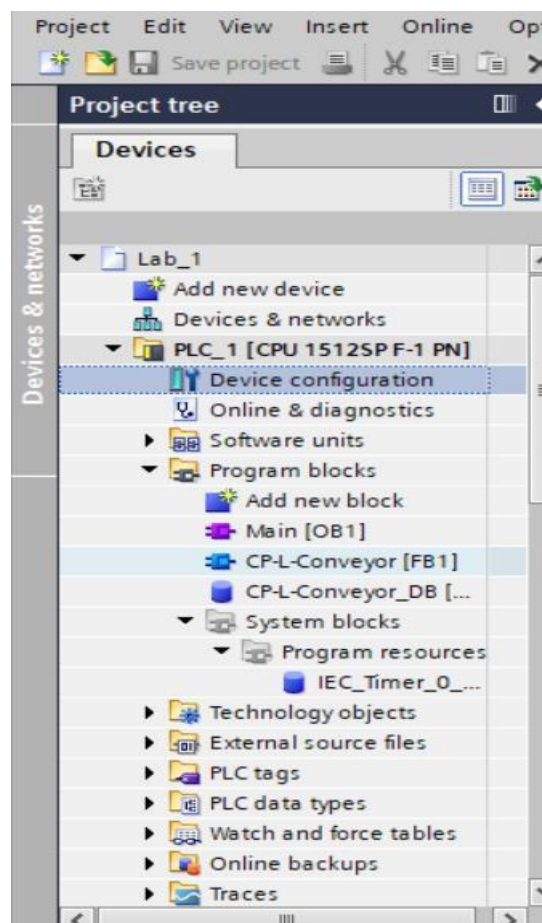
Oppgave 2.0: Under «Startup» i CPU-en vår, hva er «Configuration time»?

Om du har flere forskjellige CPU-er eller HMI paneler kan du trykke på «Topology view» og se hvordan alle de forskjellige komponentene er koblet sammen. Til høyre på siden har du en katalog med forskjellige moduler som kan legges til om nødvendig.

Oppgave 2.1: I denne menyen, hvor mange forskjellige typer «Interface modules» er der?

Neste meny vi skal se på er «Program blocks». Her vil du se alle programblokkene som brukes i programmet. PLS-en har en hovedprogramblokk («Main [OB1]») som blir kjørt syklisk hele tiden, og mengden tid hver av rundene bruker er spesifisert under «Properties» til CPU.

Oppgave 2.2: Hva er «Maximum cycle time» og «Minimum cycle time» på CPU-en vår?



Når du åpner programblokken «Main [OB1]» vil du se et lite program som styrer transportbåndet. Vanligvis så programmerer vi ikke direkte i Main men siden programmet er

såpass lite så går det helt fint. Vanligvis bruker du hovedsakelig Main til å kalle på forskjellige blokker laget på forhånd som vi setter i sammen til et helt program ved å kalle på de i Main.

Oppgave 2.3: Hvilke «Inputs» finnes det i «Main» nå?

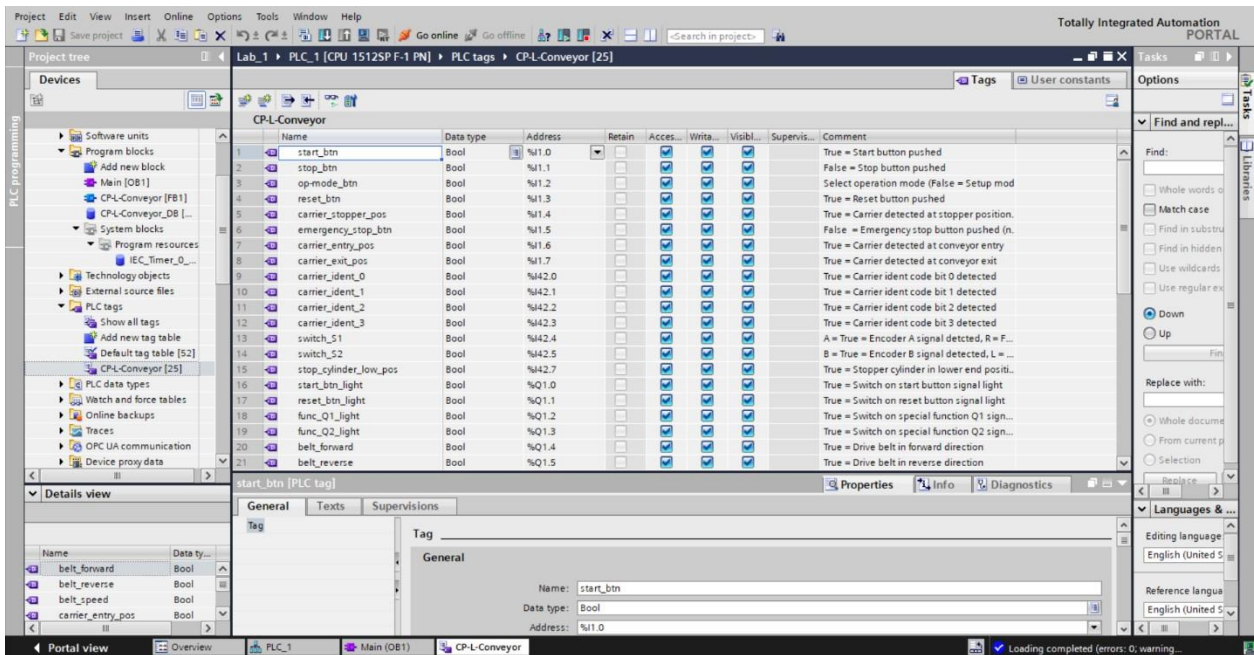
Ved å lage blokker gjør du det også veldig enkelt å bruke samme kode om igjen uten å måtte lage koden flere ganger. Når du kaller på en funksjonsblokk i Main blir det laget en instansdatabase (DB) som utfører funksjonen til blokken. Dette vil si at du kan kalle på den samme blokken flere ganger og for hver bruk får du en korresponderende DB.

Under «System blocks» og «Program resources» har vi DB-er som er laget fordi vi har kallet på funksjoner som kommer med TIA-Portalen. TIA-Portalen har mange forskjellige funksjonaliteter direkte i programmet som vi enkelt kan implementere. I dette programmet har vi kalt på en spesiell type klokke som heter «Timer-ON-Delay» (TON).

Oppgave 2.4: Hvilken andre timere i katalogen?

Det neste vi skal se på er PLC-tags. Åpne menyen for «PLC tags» og dobbeltklikk på CP-L-CONVEYOR.

Skjermen din skal se slik ut:



Taggene er de vi så i CIROS under trinn 7 i introduksjonen av labben. Disse tagene defineres med et navn, en datatype og en tilhørende IO adresse til PLS-en.

Oppgave 2.5: Hva er adressen til «Switch_S1»?

Det er disse adressene vi bruker for å kommunisere med CIROS. Ved å trykke på knappen «Monitor all» (brillene med et play-symbol) så vil du kunne se hvilken verdi taggene har i simuleringen.

Oppgave 2.6: (I CIROS) Vri på bryteren under HMI-en, bryteren er på plass nr.3 fra venstre. Denne bryteren velger hvilken modus båndet er i, ved å vri på denne setter vi nå bandet i automatisk modus. Når du nå generer et arbeidsstykke vil bandet kjøre, noter navnet på alle tagger som endrer verdi under simuleringen.

Navn:	Adresse

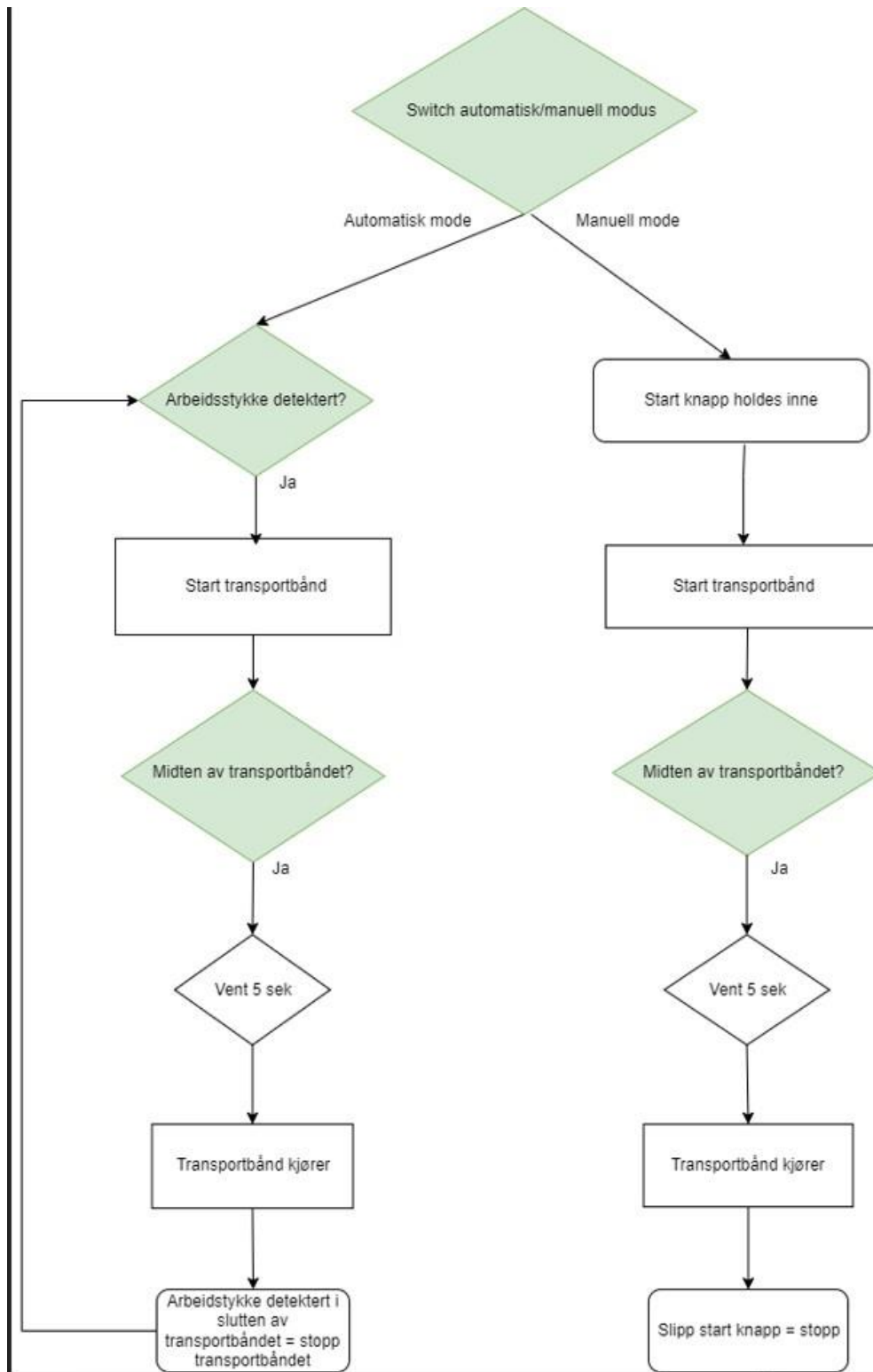
--	--

Oppgave 2.7: Hvilke 2 adresser samsvarer med kjøreretning av båndet? Hvilke 2 adresser detekterer når et arbeidsstykke er på starten eller slutten av båndet? Hvilke av disse adressene er input eller outputadresser? Fyll inn tabell under

	Adresse:	Input eller output:	Bånd eller arbeidsstykke
1			
2			
3			
4			

Oppgave 3:

I innledningen til denne labben forklarte vi funksjonaliteten til transportbåndet, og i oppgave 2 kjørte vi bandet i automatisk modus. I denne oppgaven skal du programmere funksjonaliteten bandet har i manuell modus. Flytgrafene nedenfor forklarer funksjonaliteten.



D.2

Lab 2

Laboratorium 2

Conveyor-Belt with Pneumatic Press Module

Innledning

I denne laboratorieøvelsen skal vi se på samme transportbåndet vi så på i øvelse 1, men denne gangen har vi også satt inn en pneumatisk presse. Du skal programmere noen større funksjonaliteter i programmet og denne gangen vil du også bli introdusert til tidtakere. Pressen vi ser på denne gangen har en ganske enkel funksjonalitet med bare 2 utganger og 3 innganger. Ved oppstart følger du samme introduksjonsdokumentet som forrige gang med unntak av at du åpner filene med navn «Lab_2».

Vedlagt under finner du et flytdiagram som forklarer funksjonaliteten til programmet.

Oppgave 1

Denne gangen har vi flere funksjonsblokker, en for pressen og en for transportbåndet. I begge blokkene vil du finne en timer som styrer noe.

Oppgave 1.0: Hva styrer tidtakerne?

I listen over «Basic instructions» finner dere tidtakerne. Tidtakerne har noen forskjeller som er viktig å huske på.

Oppgave 1.1: Hvilke tidtakere bruker vi i programmet? Hva er forskjellen på dem?

--

Nå skal du kjøre simuleringen. Merk deg at denne gangen har vi 3 forskjellige knapper vi kan trykke på for å generere et objekt.

Oppgave 1.2: Prøv de 3 forskjellige knappene for å generere et objekt. Hva skjer når du trykker på dem?

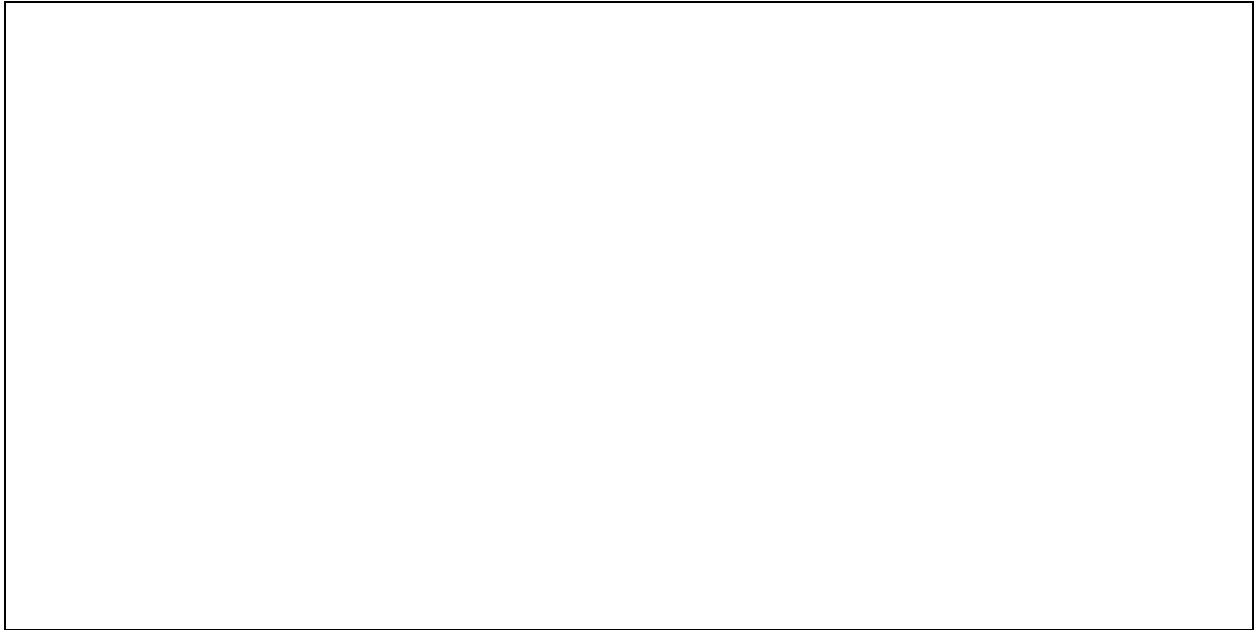
Knapp 1 (Venstre)	
Knapp 2 (Midt)	
Knapp 3 (Høyre)	

Oppgave 1.3: Finn grunnen til at en av knappene ikke fungerer på samme måte som de to andre og forklar.

Oppgave 1.4: Se på tilstanden til nødstopp-knappen og stoppknappen. Er det noen spesiell grunn til at de er programmert på denne måten?

Oppgave 2

Vi bruker denne pressen til å presse dekslene til en mobil sammen. Se på programmeringen vi har gjort så langt å se om det er et signal som passer bedre til å håndtere aktiveringen av pressen. Husk at PLC-tags er kommenterte. Forklar hvilket signal som passer best og hvorfor du mener den nåværende metoden er bedre eller dårligere.

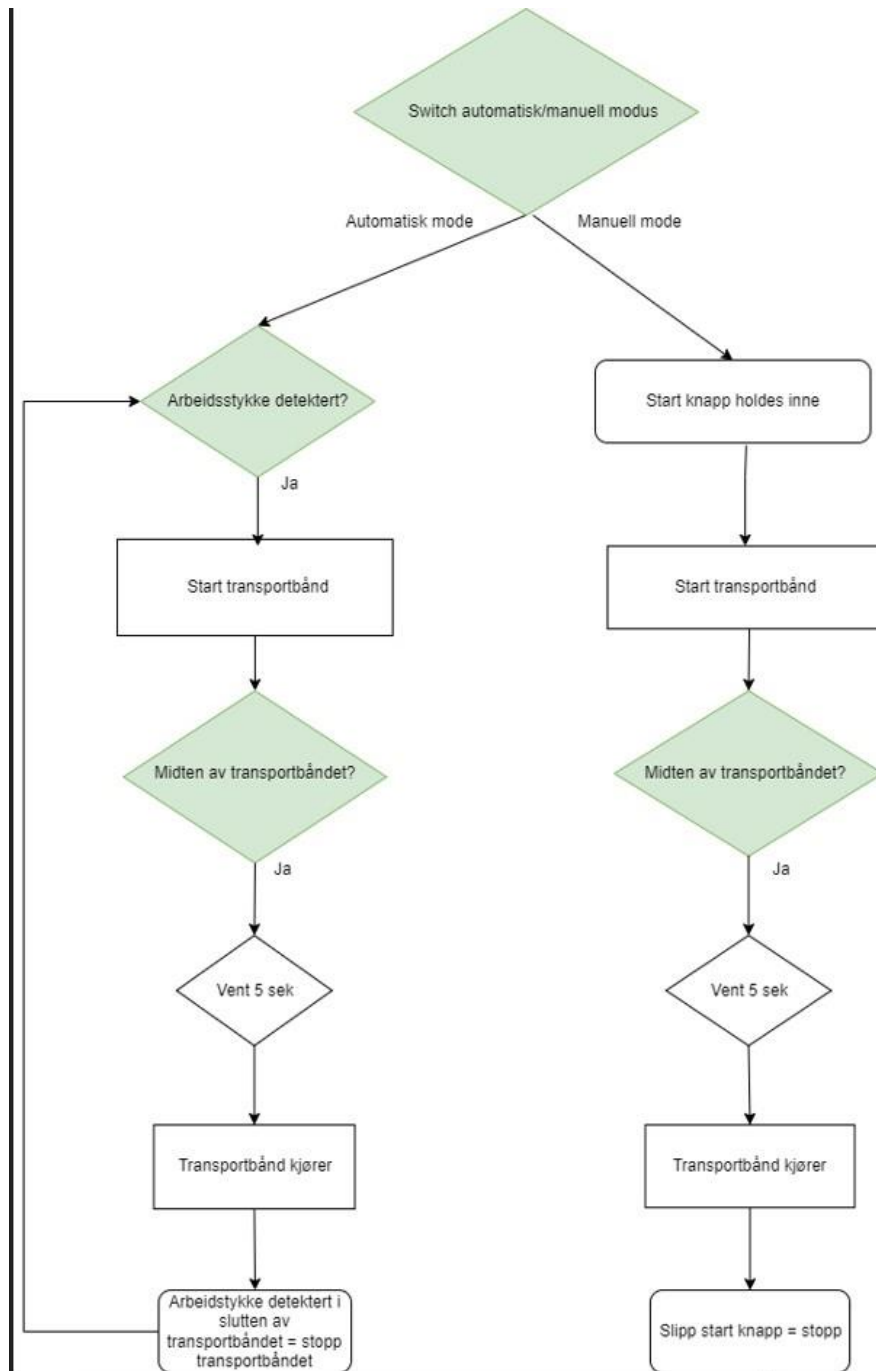


Oppgave 3

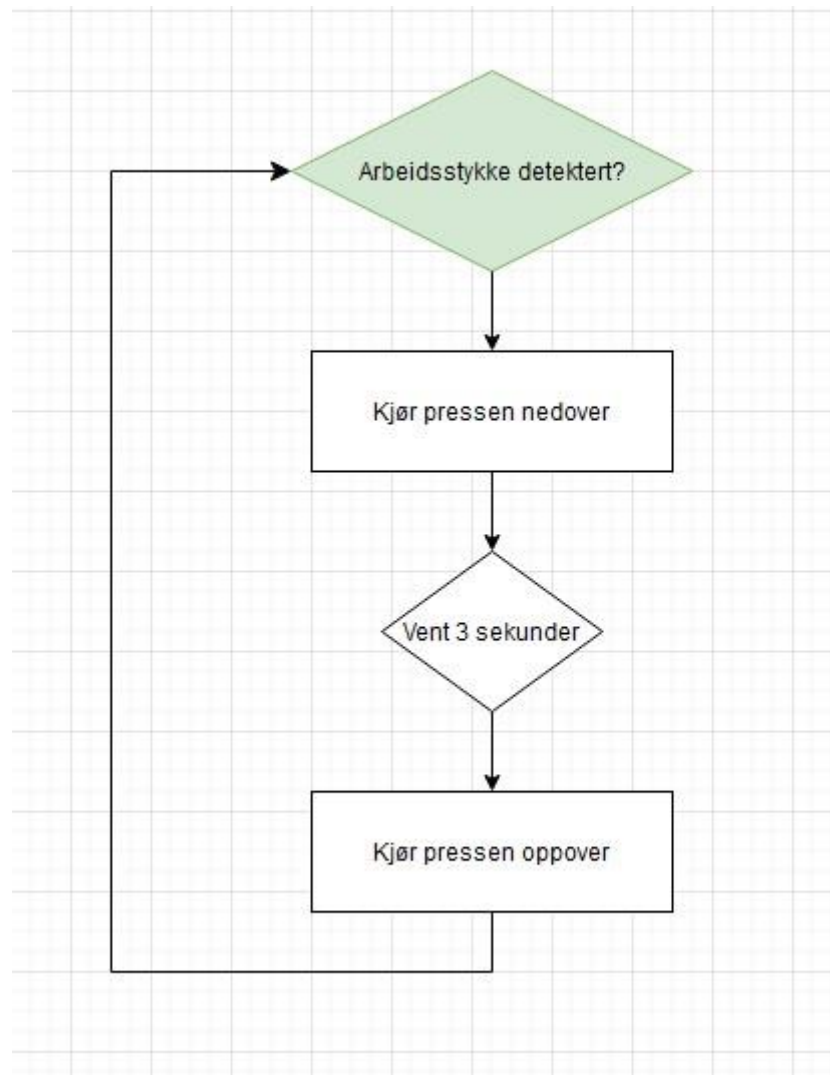
Modifiser programmet slik at pressen kun aktiveres med den venstre knappen. Programmet skal fungere identisk på alle andre måter.

Vedlegg

CP-L-CONVEYOR



CP-AM-PRESS



D.3 Lab 3

Laboratorium 3

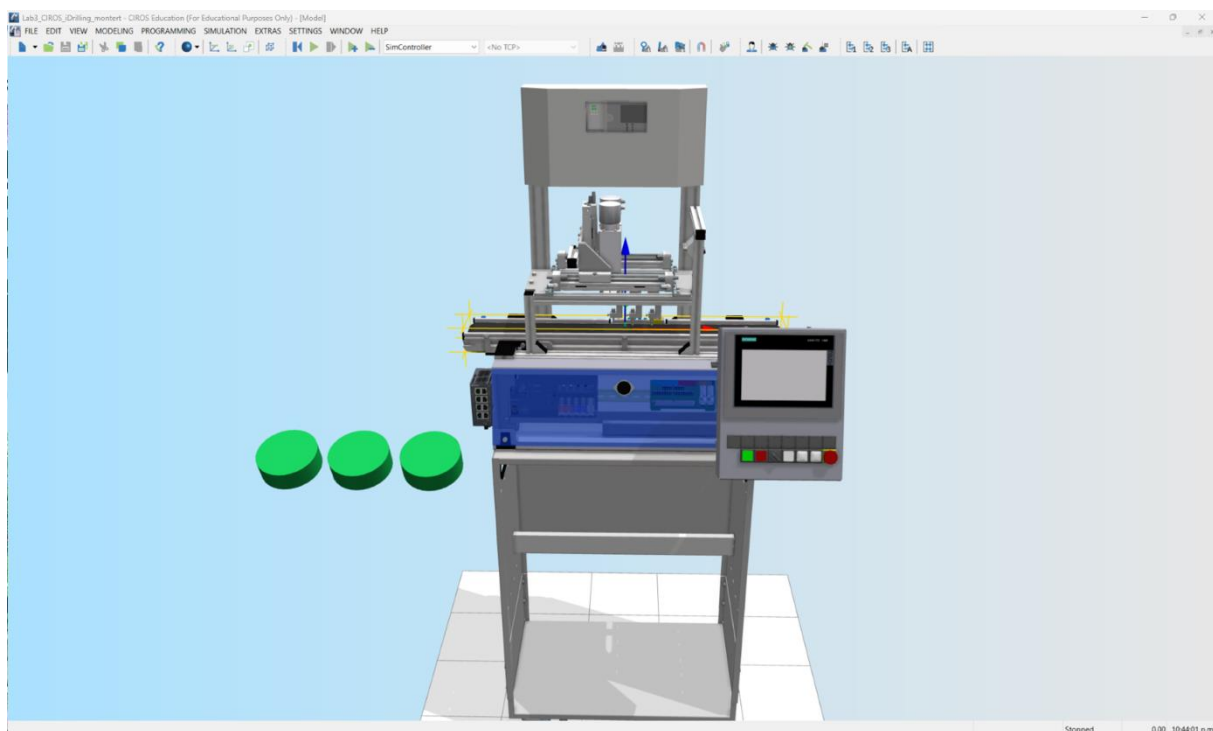
iDrilling applikasjons modul

Innledning

I denne oppgaven skal du bygge på transportbåndmodulen du ble introdusert til i Lab 1. Din oppgave blir å programmere funksjonaliteten til applikasjonsmodulen iDrilling. Du har fått tildelt en «.zip» fil som inneholder en CIROS modell med et samlebånd og applikasjonsmodulen, og en TIA-Portal prosjekt mappe. Pakk ut filene å start opp CIROS og TIA prosjektet for å sette i gang med oppgavene. TIA prosjektet er et løsningsforslag til Lab 1, du kan enten bruke denne eller bruke din egen fra når du løste denne.

Følgende fil av transportbåndet skal fungere følgende: «**Se vedlegg 1**»

Video demonstrasjon av applikasjonen som skal legges til (vi skal kun borre to hull):

[iDrilling applikasjons modul video](#)

Oppgave 1:

Applikasjonsmodulen iDrilling har en egen PLS av typen Festo CECC-LK som har nye input og output adresser (merk deg beskrivelsene av de forskjellige adressene). Denne PLS-en kommuniserer med master PLS-en som sitter i transportbånd modulen via en Profibus, dette er den samme PLS-en som styrer transportbåndet.

Din oppgave i TIA-portalen er å definere disse nye adressene i tag-tabellen, disse finner du i tabellen under. Siden vi har laget modellen, må dere navngi adressene nøyaktig det samme som oss. For mer informasjon om adressene, se i databladet: [Datablad iDrilling](#).

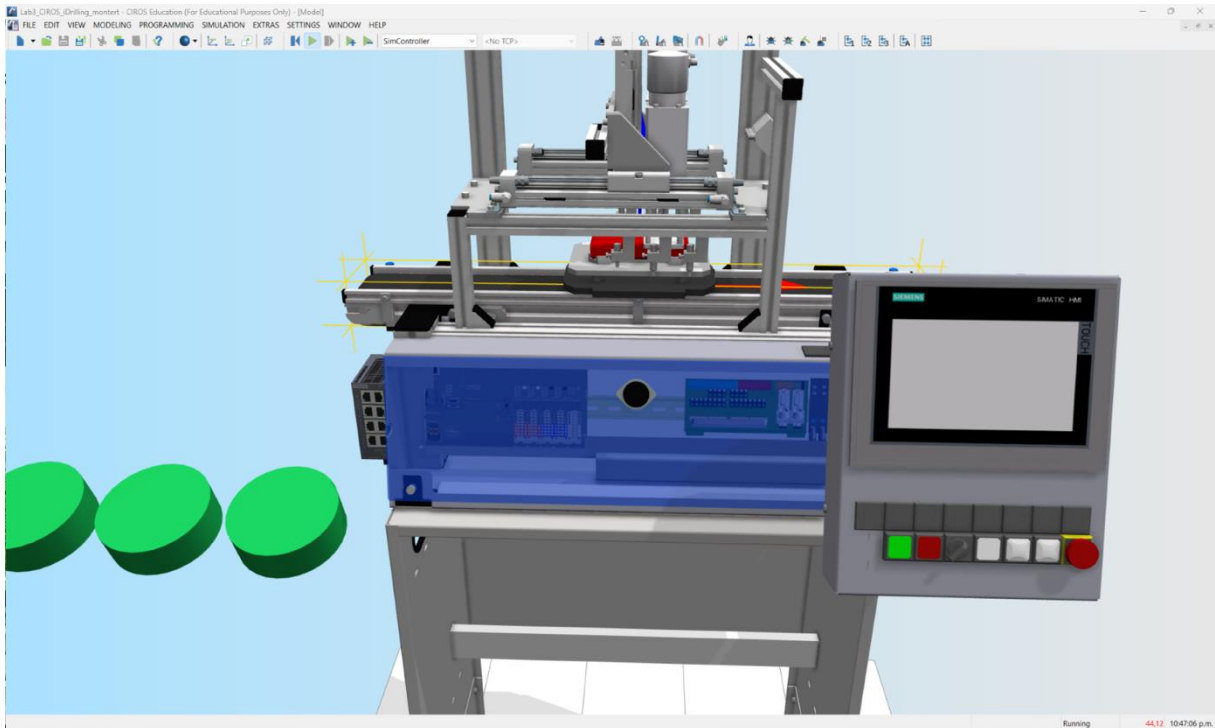
Besvarelse: Legg til følgende tabell inn i tag tabell, med samme navn og adresse. Deretter se i databladet og fyll inn kommentar feltet, med informasjon om hva den enkelte taggen gjør.

Navn:	Datatype:	Adresse:	Kommentar:
X-axis_left_end_pos	Bool	%I0.0	
X-axis_right_pos	Bool	%I0.1	
front_cover_correct	Bool	%I0.2	
front_cover_present	Bool	%I0.3	
Z-axis_upper_pos	Bool	%I0.4	
Z_axis_lower_pos	Bool	%I0.5	
back_cover_present	Bool	%I0.7	
move_X-axis_left	Bool	%Q0.0	

move_X-axis_right	Bool	%Q0.1	
switch_drill_1	Bool	%Q0.2	
switch_drill_2	Bool	%Q0.3	
move_Z-axis_up	Bool	%Q0.4	
move_Z-axis_down	Bool	%Q0.5	
open_Z-axis_end_pos_lock	Bool	%Q0.6	

Oppgave 2:

Nå skal du programmere funksjonen til applikasjonsmodulen som vist i sekvens diagrammet. Merk deg at du blir å ha behov for en timer til å løse dette, les deg opp på IEC timer og velg ut en som passer. Merk at vi skal kun borre hull bak på arbeidsstykket.



a. Sett drillen i start posisjon, det vil si at følgende tags skal få følgende verdier:

Tag	Verdi
X-axis_right_pos	True
Z-axis_upper_pos	True
switch_drill_1	False
switch_drill_2	False

open_Z-	True
axis_end_pos_lock	

Besvarelse: Ta en screenshot av tag tabell i online modus, med riktige verdier.

- b. Få drillen til å borre i 5 sekunder når arbeidsstykket er detektert i midten av transportbåndet, så sendes arbeidsstykket videre og drillen settes i startposisjon.

Besvarelse: Ta en video med telefonen av prosessen.

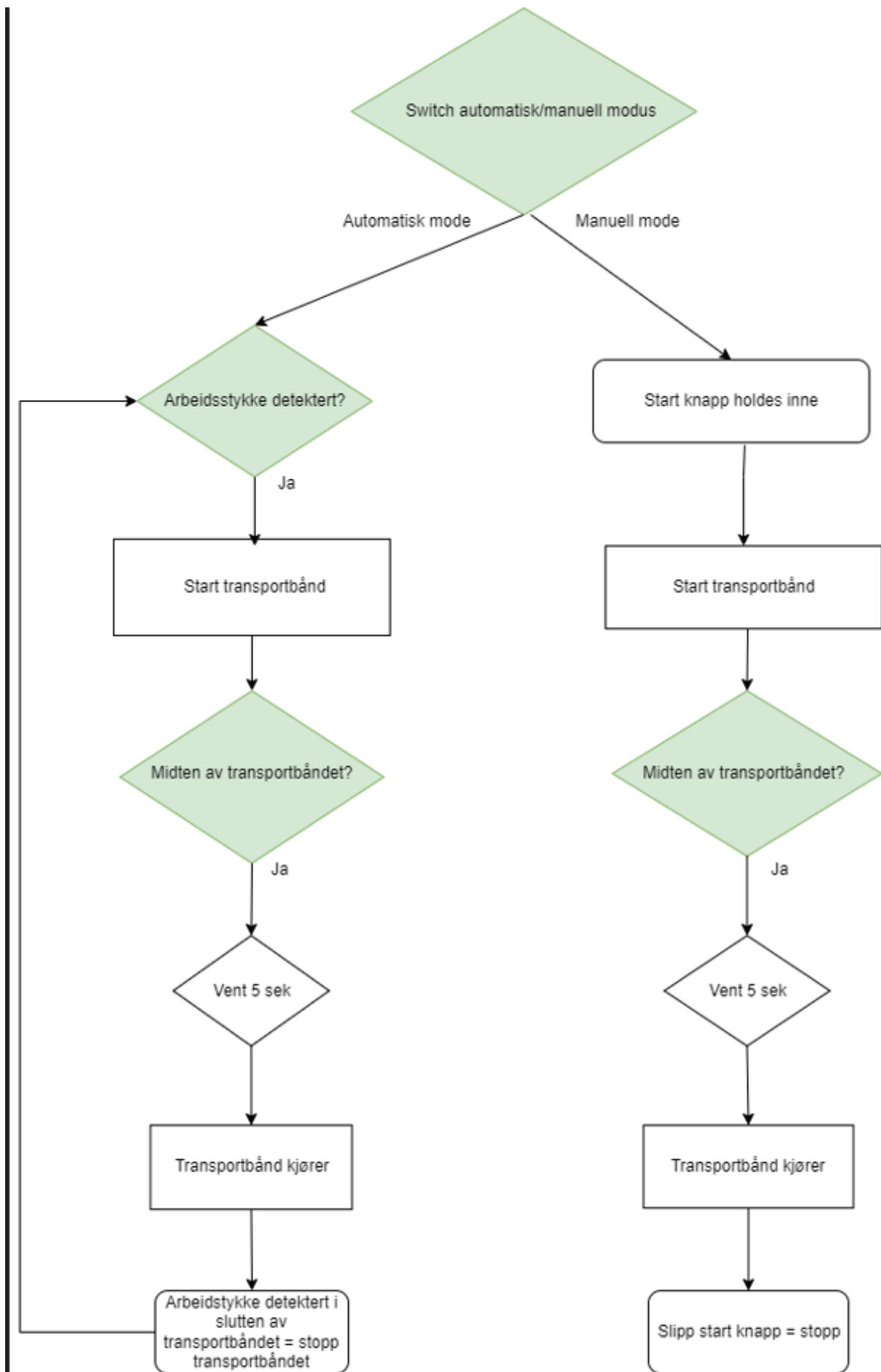
- c. Lag en sikkerhets sjekk, om at bakdeksel må være på plass for å drille i arbeidsstykket, om det ikke er på plass, skal det sendes videre uten å drilles.

Besvarelse: Test med et arbeidsstykke uten bakdeksel

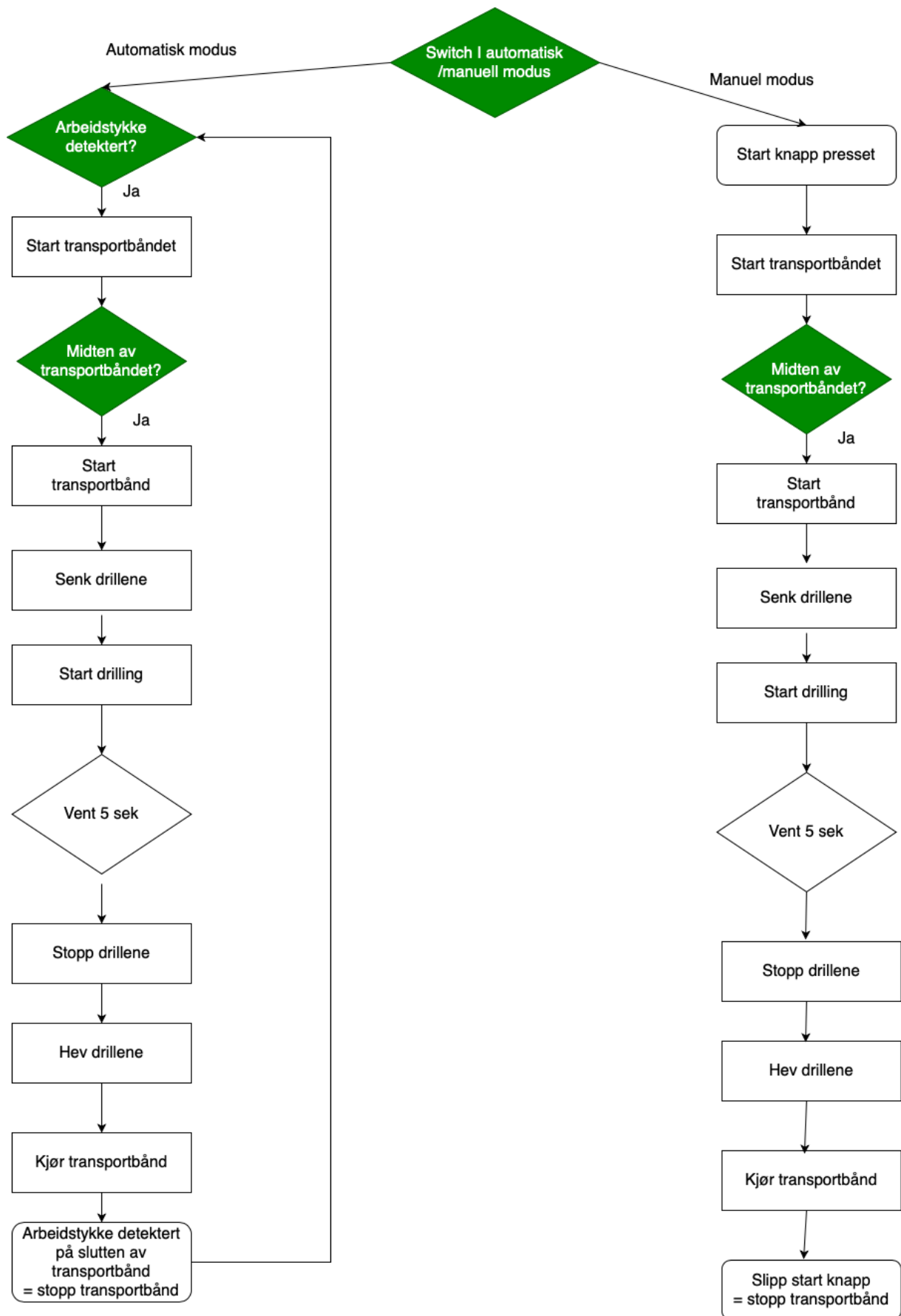
Sekvens diagram: **Vedlegg 2**

[Vedlegg](#)

Vedlegg 1:



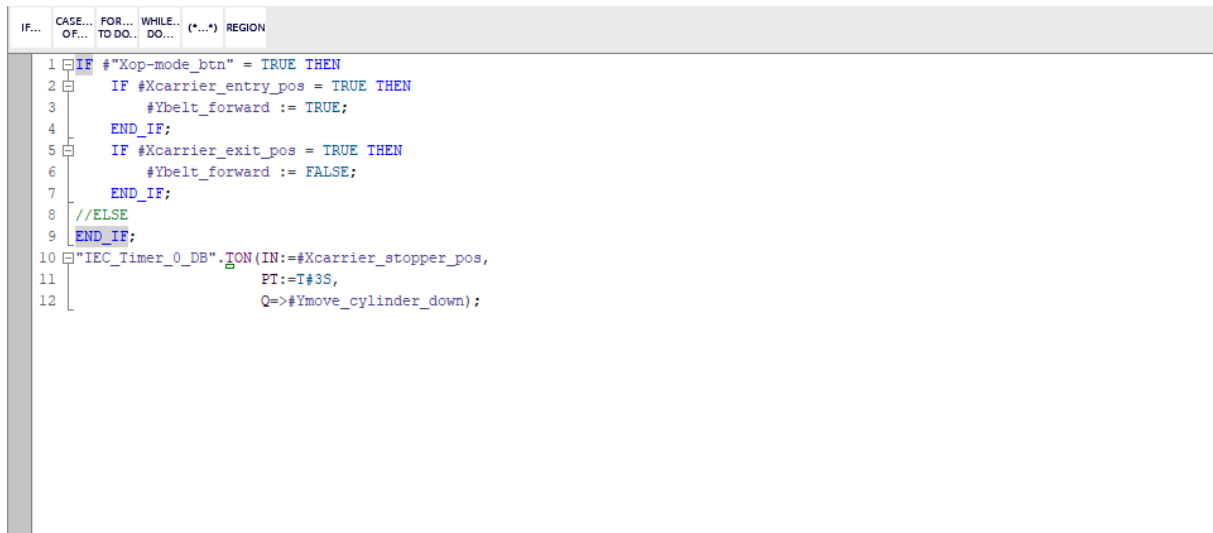
Vedlegg 2:



Appendiks E Programkode

Her vil programkodene brukt i lab øvelsene bli forklart. Dette er for å skape forståelse i hvordan programmene skal fungere.

E.1 Lab 1



```
IF... CASE... FOR... WHILE... (*...*) REGION
OF... TO DO... DO...
1 IF #Xop-mode_btn = TRUE THEN
2   IF #Xcarrier_entry_pos = TRUE THEN
3     #Ybelt_forward := TRUE;
4   END_IF;
5   IF #Xcarrier_exit_pos = TRUE THEN
6     #Ybelt_forward := FALSE;
7   END_IF;
8 //ELSE
9 END_IF;
10 IEC_Timer_0_DB.TON(IN:=#Xcarrier_stopper_pos,
11 PT:=T#3S,
12 Q=>#Ymove_cylinder_down);
```

Figur 38: Skjermdump av funksjonsblokkdiagrammet i SCL fra lab 1

Dette er kode fra funksjonsblokken. Her vil «Xop-mode_btn» være en boolsk inngang som er en bryter på panelet i CIROS modellen.

Hvis den står i «TRUE» modus, vil den fungere som i automatisk modus.

Da vil transportbåndet starte, hvis det detekteres et arbeidsstykke av sensoren i starten av transportbåndet kalt «Xcarrier_entry_pos».

Hvis det arbeidsstykket detekteres av sensoren i slutten av transportbåndet kalt «Xcarrier_exit_pos», vil transportbåndet stanse.

IF...	CASE... OF...	FOR... TO DO...	WHILE... DO...	(*...*)	REGION
1					"CP-L-Conveyor_DB" (Xstartbtn:="start_btn",
2					Xstopbtn:="stop_btn",
3					Xop-mode_btn:="op-mode_btn",
4					Xreset_btn:="reset_btn",
5					Xcarrier_stopper_pos:="carrier_stopper_pos",
6					Xemergency_stop_btn:="emergency_stop_btn",
7					Xcarrier_entry_pos:="carrier_entry_pos",
8					Xcarrier_exit_pos:="carrier_exit_pos",
9					Xcarrier_ident_0:="carrier_ident_0",
10					Xcarrier_ident_1:="carrier_ident_1",
11					Xcarrier_ident_2:="carrier_ident_2",
12					Xcarrier_ident_3:="carrier_ident_3",
13					Xswitch_S1:="switch_S1",
14					Xswitch_S2:="switch_S2",
15					Xstop_cylinder_low_pos:="stop_cylinder_low_pos",
16					Ystart_btn_light=>"start_btn_light",
17					Yreset_btn_light=>"reset_btn_light",
18					Yfunc_Q1_light=>"func_Q1_light",
19					Yfunc_Q2_light=>"func_Q2_light",
20					Ybelt_forward=>"belt_forward",
21					Ybelt_reverse=>"belt_reverse",
22					Ybelt_speed=>"belt_speed",
23					Ymove_cylinder_down=>"move_cylinder_down",
24					Ycoupling_signal_left=>"coupling_signal_left",
25					Ycoupling_signal_right=>"coupling_signal_right");
26					

Figur 39: Skjermdump av Main i SCL i lab 1

Her er koden fra Main, her kobles funksjonsblokken sammen med tag-tabell, slik at riktige I/O blir koblet sammen med variablene i funksjonsblokkdiagrammet.

E.2 Lab 2

Lab 2 er en videreutvikling av lab 1 og inneholder samme funksjonsblokk for transportbåndet.

Her ser du også løsningen for den siste oppgaven i lab 1.

Lab 2 er en videreutvikling av lab 1 og inneholder samme funksjonsblokk for transportbandet.

Her ser du også løsningen for den siste oppgaven i lab 1.

Lab 2 er en videreutvikling av lab 1 og inneholder samme funksjonsblokk for transportbandet.

Her ser du også løsningen for den siste oppgaven i lab 1.

Blokken begynner ved å deklare to statiske bool-verdier som styrer bevegelsene til pressen. Deretter bruker vi en if-setning for å resette pressen til original posisjon når sensoren midt på bandet ikke er aktiv. Til slutt bruker vi en puls som varer i 1500ms for å kjøre pressen nedover. Denne ligger i en if-else setning som resetter den etter bruk.

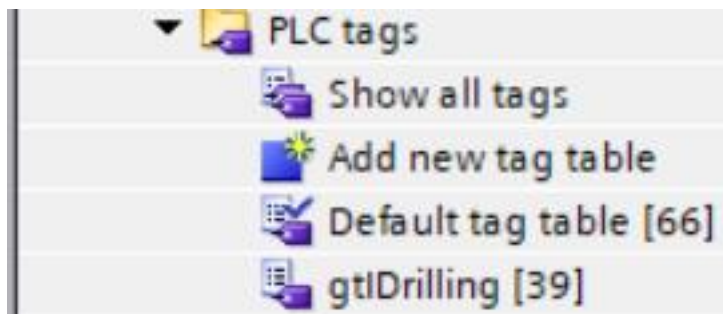
Blokkene implementeres til slutt i «Main».

1	//Implementing FB1		
2	"CP-L-Conveyor_DB" (Xstartbtn:="start_btn",	"CP-L-Con...	%DB2
3	Xstopbtn:="stop_btn",	"stop_btn"	%I1.1
4	"Xop-mode_btn":="op-mode_btn",	"op-mode_...	%I1.2
5	Xreset_btn:="reset_btn",	"reset_btn"	%I1.3
6	Xcarrier_stopper_pos:="carrier_stopper_pos",	"carrier_...	%I1.4
7	Xemergency_stop_btn:="emergency_stop_btn",	"emergenc...	%I1.5
8	Xcarrier_entry_pos:="carrier_entry_pos",	"carrier_...	%I1.6
9	Xcarrier_exit_pos:="carrier_exit_pos",	"carrier_...	%I1.7
10	Xcarrier_ident_0:="carrier_ident_0",	"carrier_...	%I42.0
11	Xcarrier_ident_1:="carrier_ident_1",	"carrier_...	%I42.1
12	Xcarrier_ident_2:="carrier_ident_2",	"carrier_...	%I42.2
13	Xcarrier_ident_3:="carrier_ident_3",	"carrier_...	%I42.3
14	Xswitch_S1:="switch_S1",	"switch_S1"	%I42.4
15	Xswitch_S2:="switch_S2",	"switch_S2"	%I42.5
16	Xstop_cylinder_low_pos:="stop_cylinder_low_pos",	"stop_cyl...	%I42.7
17	Ystart_btn_light:="start_btn_light",	"start_bt...	%Q1.0
18	Yreset_btn_light:="reset_btn_light",	"reset_bt...	%Q1.1
19	Yfunc_Q1_light:="func_Q1_light",	"func_Q1_...	%Q1.2
20	Yfunc_Q2_light:="func_Q2_light",	"func_Q2_...	%Q1.3
21	Ybelt_forward:="belt_forward",	"belt_for...	%Q1.4
22	Ybelt_reverse:="belt_reverse",	"belt_rev...	%Q1.5
23	Ybelt_speed:="belt_speed",	"belt_spee...	%Q1.6
24	Ymove_cylinder_down:="move_cylinder_down",	"move_cyl...	%Q1.7
25	Ycoupling_signal_left:="coupling_signal_left",	"coupling...	%Q42.4
26	Ycoupling_signal_right:="coupling_signal_right");	"coupling...	%Q42.5
27	//Implementing FB2		
28	"CP-AM-PRESS_DB" (Xcylinder_upper_pos:="cylinder_upper_pos",	"CP-AM-PR...	%DB3
29	Xcylinder_press_pos:="cylinder_press_pos",	"cylinder...	%I0.1
30	Xcarrier_stppoer_pos:="carrier_stopper_pos",	"carrier_...	%I1.4
31	Xback_cover_present:="back_cover_present",	"back_cov...	%I0.7
32	Ycylinder_up:="cylinder_up",	"cylinder...	%Q0.0

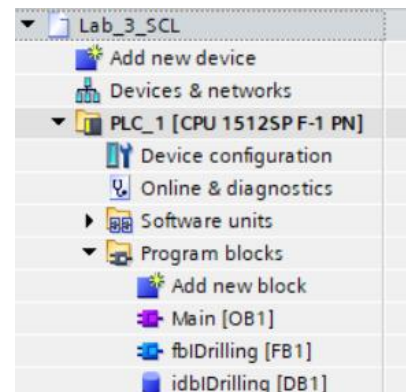
Figur 42: Implementering av funksjonsblokker i "Main"

E.3 Lab 3

Her er koden fra løsningsforslaget til lab 3 i SCL.



Figur 44: Skjermdump av menyen i TIA-portalen, tag tabellen



Figur 43: Skjermdump av menyen av programblokkene

Lab_3_SCL ▶ PLC_1 [CPU 1512SP F-1 PN] ▶ PLC tags ▶ gtlDrilling [39]										
gtlDrilling										
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Supervis...	Comment	
1	start_btn	Bool	%I1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Start button pushed	
2	stop_btn	Bool	%I1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		False = Stop button pushed	
3	op-mode_btn	Bool	%I1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Select operation mode (False = Setup mod	
4	reset_btn	Bool	%I1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Reset button pushed	
5	carrier_stopper_pos	Bool	%I1.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Carrier detected at stopper position.	
6	emergency_stop_btn	Bool	%I1.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		False = Emergency stop button pushed (n.	
7	carrier_entry_pos	Bool	%I1.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Carrier detected at conveyor entry	
8	carrier_exit_pos	Bool	%I1.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Carrier detected at conveyor exit	
9	carrier_ident_0	Bool	%I42.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Carrier ident code bit 0 detected	
10	carrier_ident_1	Bool	%I42.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Carrier ident code bit 1 detected	
11	carrier_ident_2	Bool	%I42.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Carrier ident code bit 2 detected	
12	carrier_ident_3	Bool	%I42.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Carrier ident code bit 3 detected	
13	switch_S1	Bool	%I42.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		A = True = Encoder A signal detected, R = F...	
14	switch_S2	Bool	%I42.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		B = True = Encoder B signal detected, L = ...	
15	stop_cylinder_low_pos	Bool	%I42.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Stopper cylinder in lower end positi..	
16	start_btn_light	Bool	%Q1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Switch on start button signal light	
17	reset_btn_light	Bool	%Q1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Switch on reset button signal light	
18	func_Q1_light	Bool	%Q1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Switch on special function Q1 sign...	
19	func_Q2_light	Bool	%Q1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Switch on special function Q2 sign...	
20	belt_forward	Bool	%Q1.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Drive belt in forward direction	
21	belt_reverse	Bool	%Q1.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Drive belt in reverse direction	
22	belt_speed	Bool	%Q1.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Select belt speed (False = normal, True = sl	
23	move_cylinder_down	Bool	%Q1.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Move stopper cylinder down	
24	coupling_signal_left	Bool	%Q42.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Send coupling signal left	
25	coupling_signal_right	Bool	%Q42.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Send coupling signal right	
26	X-axis_left_end_pos	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = drill in X-axis left position	
27	X-axis_right_pos	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = drill in X-axis right position	
28	front_cover_correct	Bool	%I0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = front cover correct after inspection	
29	front_cover_present	Bool	%I0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = front cover present	
30	Z-axis_upper_pos	Bool	%I0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = drill in Z-axis upper position	
31	Z_axis_lower_pos	Bool	%I0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = drill in Z-axis lower position	
32	back_cover_present	Bool	%I0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Back cover present on workpiece	
33	move_X-axis_left	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = drill will move to X-axis left positi	
34	move_X-axis_right	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = drill will move to X-axis right positi...	
35	switch_drill_1	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = drill 1 will start drilling	
36	switch_drill_2	Bool	%Q0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = drill 2 will start drilling	
37	move_Z-axis_up	Bool	%Q0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = drill will move to Z-axis upper posit...	
38	move_Z-axis_down	Bool	%Q0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = drill will move to Z-axis lower posit...	
39	open_Z-axis_end_pos_lock	Bool	%Q0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = drill is allowed to move up or down	
40	<Add new>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			

Figur 45: Skjermdump av tag-tabell for lab 3

Her kan dere se den fulle tag tabellen. Den inneholder alle I/O-ene med en kort kommentar på hver tag som forklarer dens tilstander.

Lab_3_SCL ▶ PLC_1 [CPU 1512SP F-1 PN] ▶ Program blocks ▶ fblDrilling [FB1]

fblDrilling

	Name	Data type	Default value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	▼ Input							
2	Xstartbtn	Bool	false	Non-ret...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	Xstopbtn	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	Xreset_btn	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	Xcarrier_stopper_pos	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	Xcarrier_entry_pos	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	Xcarrier_exit_pos	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	XX-axis_left	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	XX-axis_right	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	XZ-axis_up	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	XZ-axis_down	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	Xop_mode_switch	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	Xback_cover_present	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	▼ Output							
15	Ystart_btn_light	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16	Ybelt_forward	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	Yreset_btn	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18	Ymove_X_axis_left	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19	Ymove_X_axis_right	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
20	Ydrill_1_on	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
21	Ydrill_2_on	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
22	Ymove_Z_axis_down	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
23	Ymove_Z_axis_up	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
24	Yopen_Z_axis_lock	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
25	Ymove_cylinder_down	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
26	▼ InOut							
27	<Add new>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
28	▼ Static							
29	stTimeNow	Time	T#0ms	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
30	sbTimerFinished	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
31	▼ Temp							
32	<Add new>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
33	▼ Constant							
34	<Add new>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figur 46: Skjermdump av variablene i funksjonsblokken i lab 3

Her kan dere se alle variablene som har blitt lagt til i funksjonsblokken til lab 3. Disse brukes som i stedet for I/O-ene for å skape et sikrere program.

```

Lab_3_SCL ▶ PLC_1 [CPU 1512SP-F1 PN] ▶ Program blocks ▶ fbiDrilling [FB1]
Block interface
IF... CASE... FOR... WHILE... (*...*) REGION
1 (*
2 -----
3 Conveyer belt:
4 *)
5
6 //Get's the conveyer to drive in automatic mode
7 IF #Xop_mode_switch = TRUE THEN //When true, initialize automatic mode
8   IF #Xcarrier_entry_pos = TRUE THEN //Checks if workpiece is present on belt
9     #Ybelt_forward := TRUE; //Starts moving the belt
10  END_IF;
11  IF #Xcarrier_exit_pos = TRUE THEN //If the workpiece is at the end of the belt
12    #Ybelt_forward := FALSE; //Stops the belt from moving
13  END_IF;
14 ELSE //If the switch is in manual mode
15   IF #Xstartbtn = TRUE THEN //If the start button is pressed
16     #Ystart_btn_light := TRUE; //Starts the start button light
17     #Ybelt_forward := TRUE; //Starts the belt moving forward
18   END_IF;
19   IF #Xstopbtn = FALSE THEN //If the stop button is pressed
20     #Ystart_btn_light := FALSE; //Turns off the start button light
21     #Ybelt_forward := FALSE; //Stops the belt moving
22   END_IF;
23 END_IF;
24
25 //Timer to stop the workpiece at the middle of the belt
26 IF #Xcarrier_stopper_pos THEN //Checks if workpiece is in middle of belt
27   T:=T#S5; //Stops for 5 seconds
28   Q:=#sbTimerFinished; //Boolean output for timer, true when finished
29   ET:=#stTimeNow; //Static timer to show time during timing

```

Figur 47: Skjermdump av første del av funksjonsblokk koden i lab 3

Her kan dere se del 1 av koden i funksjonsblokken. Det er denne koden som vil bli utlevert som løsningsforslag til Lab 1. Denne koden skal fungere med at inngang «Xop_mode_switch» skal fungere som en bryter for å sette modellen i automatisk eller manuell modus. Hvis den er i automatisk modus skal den detektere arbeidsstykket i begynnelsen av transportbåndet, starte å gå, og skru seg av når den er detektert i slutten av transportbåndet. Når arbeidsstykket er detektert på midten av båndet ved hjelp av sensor «Xcarrier_stopper_pos» skal den stoppe transportbåndet i 5 sekunder, for å så kjøre videre.

```

32 (*
33 -----
34 iDrilling modul:
35 *)
36
37 //Resets timer when not triggered
38 IF NOT #Xcarrier_stopper_pos THEN //Checks if there is not a workpiece in the middle of the belt
39   RESET_TIMER("Timer"); //Resets the timer
40   #Ymove_cylinder_down := FALSE; //Moves the cylinder down, so it ready for the next
41 ;
42 END_IF;
43
44 //Puts the drill back at it's initial position
45 #Yopen_z_axis_lock := TRUE; //Opens the lock, so the drill can move in Z-direction
46 #Ymove_z_axis_up := TRUE; //Moves the drill up in Z-direction
47 #Ymove_x_axis_down := FALSE; //This is false so the two outputs don't contradicts eachother
48 #Ymove_x_axis_right := TRUE; //Moves the drill to right in X-direction
49 #Ymove_x_axis_left := FALSE; //Turns the X-axis left to false so it doesn't contradict the right command
50 #Ydrill_1_on := FALSE; //Turns drill1 off
51 #Ydrill_2_on := FALSE; //Turns drill2 off
52 //#Yopen_z_axis_lock := FALSE; //Closes the lock again
53
54 //Drills the workpiece for the set time in "Timer" ("Timer".PT)
55 IF (NOT #sbTimerFinished) AND #Xcarrier_stopper_pos AND #Xback_cover_present THEN //If the timer is not finished, and there is a workpiece at
56   #Ydrill_1_on := TRUE; //Turns drill1 on
57   #Ydrill_2_on := TRUE; //Turns drill2 on
58   #Yopen_z_axis_lock := TRUE; //Opens the lock, so the drill can move in Z-direction
59   #Ymove_x_axis_right := TRUE; //Moves the drill to right in the X-axis
60   #Ymove_x_axis_left := FALSE; //Turns the X-axis left to false so it doesn't contradict the right command
61   #Ymove_z_axis_up := FALSE; //Turns the Z-axis up to false so it doesn't contradict the down command
62   #Ymove_z_axis_down := TRUE; //Puts the drill down in Z-axis, so it can start drilling
63 ;
64 END_IF;
65
66 //Sends the workpiece forward after it's been drilled
67 IF #sbTimerFinished AND #Xcarrier_stopper_pos THEN //If the timer is finished and the workpiece is still at the middle of the belt
68   #Ymove_cylinder_down := TRUE; //Moves the cylinder down, so the belt can send it forward
69 ;
70 END_IF;
71
72 IF #Xcarrier_stopper_pos AND (NOT #Xback_cover_present) THEN //If the back cover is not present, we cannot drill
73   #Ymove_cylinder_down := TRUE; //So we send it forward
74 ;
75 END_IF;
76

```

Figur 48: Skjermdump av andre del av funksjonsblokk kode i lab 3

