



Revisjonshistorie

År	Forfatter
2016	Konstanze Kölle
2020	Kolbjørn Austreng
2021	Kiet Tuan Hoang
2022	Kiet Tuan Hoang
2025	Terje Haugland Jacobsson

I Praktisk rundt filene

I denne laben får dere ikke utlevert noen .c eller .h-filer. Dere skal arbeide i Sysmac Studio, som dere finner på Windows-partisjonen på Sanntids-PCene. Dere får heller ingen utdelte .smc2-filer, som er filformatet som Sysmac støtter, men dere skal lage et nytt slikt program fra start. Hvordan dette gjøres vil vi vise dere gjennom denne lab-teksten.

II Introduksjon - Praktisk rundt laben

PLS (programmerbar logisk styringsenhet) er et utbredt verktøy for å løse automatiseringsoppgaver i industrien. En PLS er en forholdsvis enkel innretning, som består av en prosessor, digitale- og/eller analoge innganger og utganger, og eventuelt også en eller flere kommunikasjonsgrensesnitt.

I praksis vil man koble en rekke sensorer (fotoceller, endebrytere, nivåfølere, osv.) til PLSSens innganger, og en rekke pådragsorganer (motorer, ventilører, lamper, releer, osv.) til PLSSens utganger. På selve PLSSen implementerer man et sett med logiske funksjoner, slik at man kan sette utgangene basert på hva som inngangene registrerer. Disse logiske funksjonene kan være rent kombinatoriske funksjoner, sekvensielle funksjoner, eller en blanding av de to.

I denne laben skal vi bruke en PLS fra produsenten Omron kalt NX102-9000, samt tilleggsmodulene NX-PF0730, NX-ID5442, NX-DA2605 og NX-OD5256.

Det som er så kjekt med PLSSer er det at de er utrolig robuste. Dersom man trenger å implementere et automatisk styresystem for et industrielt anlegg, som skal stå i

20 år uten omstart, da er en PLS et godt egnet verktøy. Hadde man prøvd å styre det samme anlegget med en Windows-maskin, ville anlegget krevd en omstart i uka, hackere hadde vært inne allerede ved lanseringsdatoen, og om et par år må man kjøpe en ny lisens.

Vurdering

PLS-laben gir dere en introduksjon til programmering av Omron NX102-9000 i Sysmac Studio, som er programvaren som brukes for å strukturere og overføre programvare til Omrons PLSer. Programmeringsmetoden som skal brukes er såkalte stigediagram, som er et grafisk programmeringsspråk. Det er viktig å merke seg at programmet for å kjøre PLSen, altså Sysmac, er installert på Windows, og ikke på Ubuntu, for datamaskinene i Sanntidssalen.

Den første delen av laboppgaven er strukturert som en guide for å få dere kjapt inn i programmering av Omron NX102-9000 gjennom Sysmac. Deretter skal dere programmere en noe forenklet versjon av heislogikken fra heislaben som dere skal begynne med i neste uke. PLS-laben er obligatorisk, og dere skal kun bruke én labøkt på dette.

III Introduksjon - Heisen på Sanntidslaben

I figur 1 ser dere PLSen brukt på Sanntidssalen. Dette er en modulbasert PLS, som betyr at PLSen funksjonaliteter kan utvides ved behov, ved å legge til ekstra moduler. Det er faktisk bare den venstre delen av "klossen", altså den markert med Omron NX1 og NX102-9000 som er selve PLSen. Den første klossen til høyre etter dette er strømforsyning, og de andre er moduler som kan gjøre Input/Output.

I tillegg til dette finner dere vår kjære, vakre heismodell på pulten deres (se figur 2):

III .1 Heismodellen

Denne heismodellen består av en sjakt og en bevegelig heisstol. Det er denne vi skal få til å bevege seg i labopplegget. Over øverste etasje, og under nederste etasje er det montert endestoppbrytere, som vil kutte motorpådraget dersom heisen kjører utenfor sitt lovlige område. Dette er for å beskytte heisens motor mot skade. Om heisen skulle treffe en av endestoppene, må heisstolen manuelt skyves bort fra bryterne før man kan be motoren om et nytt pådrag. Dette er en beskyttelsemekanisme i hardware og skal ikke inngå i selve styresystemet som skal bli utviklet i denne laben.

III .2 Motorstyringsboks

Heisens pådrag kommer fra en motorstyringenhet, den store svarte boksen som står ved siden av heismodellen i figur 2. Motorstyringsboksen er utstyrt med innebygd

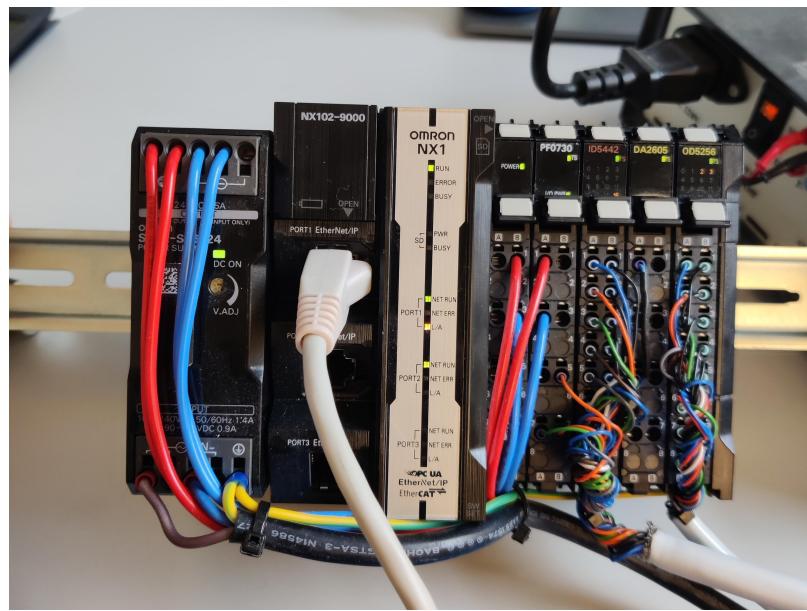


Figure 1: PLSen på Sanntidslaben.

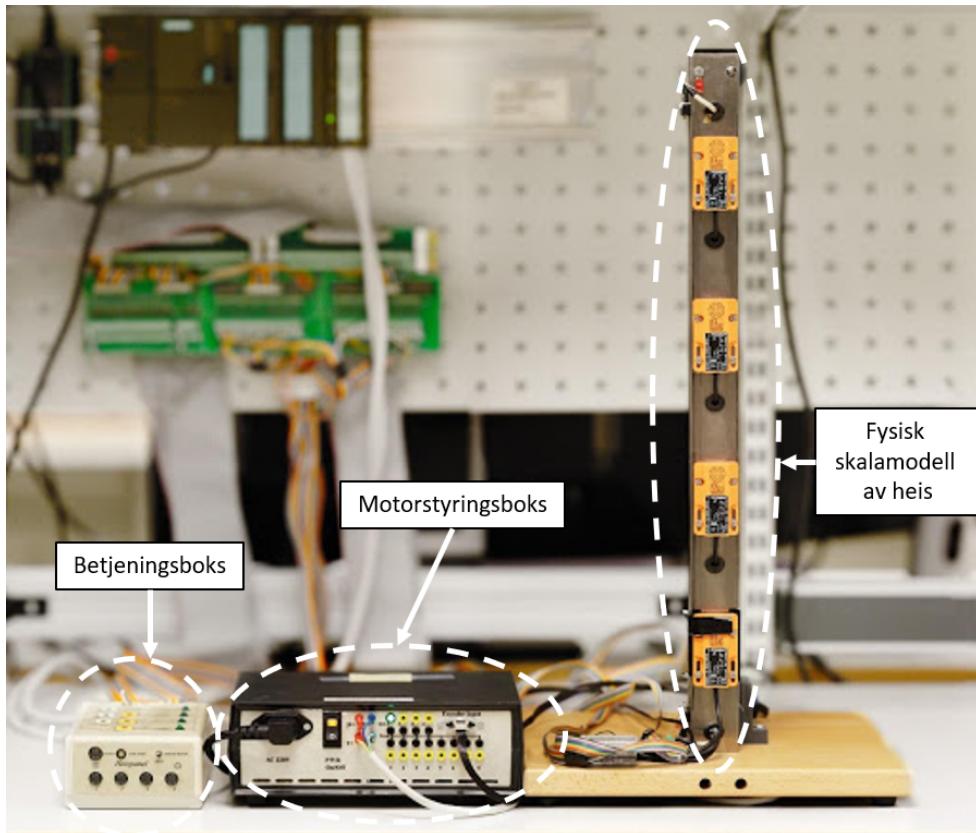


Figure 2: Heis-modellen på Sanntidslaben.

strømforsyning og kraftforsterker. Veien motoren skal gå settes ved et ekstra retningsbit i styringsboksens grensesnitt. Alt dette gjøres via funksjonskall i styringsprogrammet. Før dere gjør noe annet, skrur dere denne på, og passer på at den røde

ledningen fra heisen er koblet til M+, og den blå ledningen til M-.

III .3 Betjeningsboks

Til slutt har dere en "betjeningsboks". Øverst på betjeningsboksen finnes en bryter som velger om datamaskinen eller PLSen skal styre heismodellen. Denne skal stå i "PLS" gjennom hele laben.

Dersom man ser nærmere på betjeningsboksen, kan man se at den består av et etasjepanel, og et heispanel (se figur 3):

Etasjepanelet finnes på oversiden av betjeningsboksen fra figur 2. Dette panelet blir brukt for å simulere bestillingsknappene for opp- og nedretning fra hver etasje. Hver av knappene er utstyrt med lys som skal indikere om en bestilling er mottatt eller ei. Etasjepanelet har også ett lys for hver etasje for å indikere hvilken etasje heisen befinner seg i. Det finnes ingen direkte kopling mellom knappene og lyssignalet i elektronikken så lyset må settes av styringssystemet.

Heispanelet derimot, finner man på kortsiden av betjeningsboksen og representerer de knappene man forventer å finne inne i heisrommet til en vanlig heis. Her har man bestillingsknapper for hver etasje, samt en stoppknapp for nødstans. Alle knappene er utstyrt med lys som kan settes via styreprogrammet. I tillegg til knappene er panelet utstyrt med etasjeindikatorlys som kan settes via styreprogrammet og et lys, markert med "Dør Åpen", som indikerer om heisdøren er åpen. Heispanelet har også en obstruksjonsbryter, som kan brukes for å simulere at en person blokkerer døren når den er åpen.

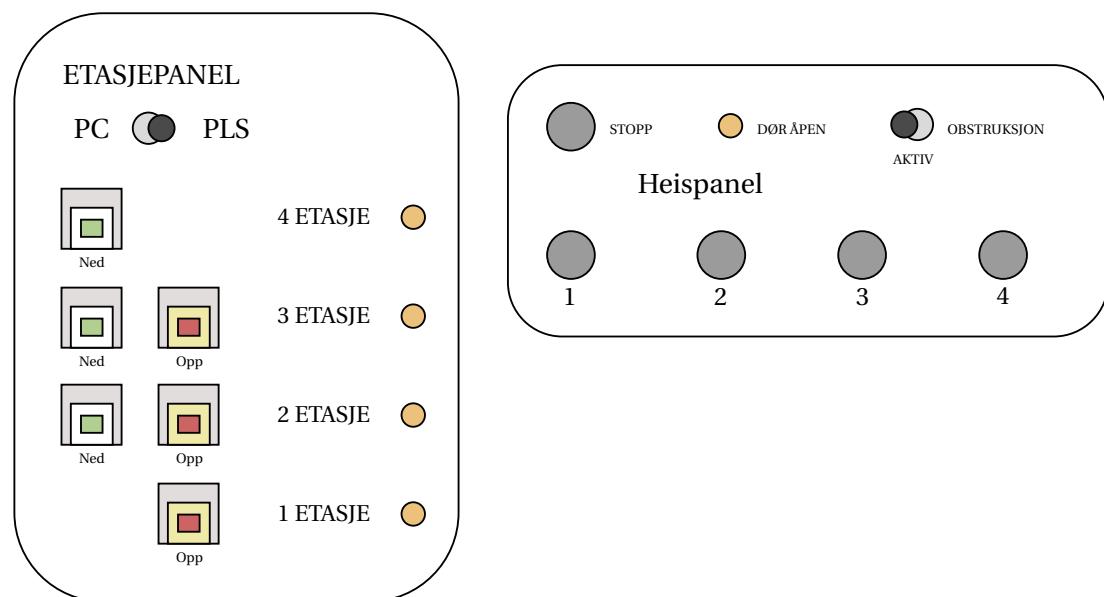


Figure 3: Etasje- og Heispanel i Sanntidslaben

IV Introduksjon - Programmering av PLS

Måten man programmerer PLSer på er standarisert og følger en industristandard kalt IEC 61131-3. Denne standarden definerer flere forskjellige måter man kan programmere/fortelle en PLS hva den skal gjøre. Fire av de vanligste av disse er:

- Stigediagram (LD), eller "Ladder diagram": er et grafisk programmeringsspråk, hvor man lager linjer som representerer programflyt, hvor linjene kan ha brytere som representerer kontrollstrukturer.
- Funksjonsblokdiagram (FBD): er også et grafisk programmeringsspråk, men her trekker man koblinger mellom utgangen på blokker inn i inngangene til andre blokker.
- Strukturert tekst (ST): er et programmeringsspråk som faktisk bruker tekst - som navnet tilsier. Det ligner på et språk som heter "PASCAL".
- Sekvensielle funksjonsdiagram (SFC) eller "Sequential function chart": er nok et grafisk programmeringsspråk, men har i tillegg noen elementer for å kunne blande sekvensielle operasjoner med parallelle operasjoner.

For å programmere i Sysmac skal dere bruke den utdelte manualen, ved navn "Sysmac Studio Version 1 Operation Manual", samt kurskompendiet, som begge to finnes i organisasjonen til TTK4235 på GitHub. Ta en titt i disse ressursene før dere eventuelt ber om hjelp, ettersom alt man trenger å vite står her. I programmet Sysmac, som skal være installert på alle Sanntids-PCene sine Windows-partisjoner, har man blant annet muligheten til følgende:

- Sette opp og organisere prosjekter
- Konfigurere hardware
- Programmere blokker
- Laste ned program til PLSen
- Debugge programmet

Av de forskjellige programmeringstypene skal vi kun benytte oss av stigediagrammer i denne laben.

IV .1 Innføring i bruken av Sysmac

Når dere skal programmere NX102-9000 er det første man gjør å åpne Sysmac Studio fra Windows-partisjonen på PCene på Sanntidssalen. Deretter må man sjekke at man er tilkoplet PLSen. Dette gjør man ved å trykke på "Connect to Device" fra hovedmenyen. Her velger man alternativet "Direct Connection Via Ethernet", og trykker slik at boksen "Transfer from Device" er tom, og deretter "Connect".

Nå skal man være tilkoplet PLSen, hvis du ikke er det enda, trykk på den gule trekanteren som heter "Online" i verktøylinja. Nå kan det hende at dere må enten skrive inn eller velge brukernavn og passord. I begge tilfellene skriver dere inn "TTK4235" som

brukernavn, og "Sanntid15" som passord. Før vi går videre skal vi slette minnet som er på PLSen, slik at dere får det man kaller for en "fresh start" på engelsk. Trykk derfor på det runde hjulet i verktøylinja som heter "Synchronize". Her velger dere alternativet "Transfer To Controller", og deretter "Yes". Nå har vi overført et blankt program til PLSen.

Hvis vi så trykker på "Configurations and Setup" i "Multiview Explorer" på venstre side av brukergrensesnittet, deretter "CPU/Expansion Racks", og så "CPU Rack". Da dukker det opp et bilde av PLSen vår uten såkalte "expansions" til høyre for "Multiview Explorer". Høyreklikk på denne og velg alternativet "Compare and Merge with Actual Unit Configuration", og deretter "Apply Actual Unit Configuration".

Nå skal fabrikkinstillingene fra PLSen være matet inn i Sysmac, slik at vi kan begynne å programmere. Dette innebærer blant annet at Sysmac nå kjenner til det fysiske PLS-oppsettet vårt, noe som gjør det enkelt å tilordne inngangene og utgangene til PLSen med de tilsvarende til heismodellen, noe som er selve kjernen i PLS-programmering. Vær bevisst på det at vi må kople oss av PLSen for å kunne programmere, og dette gjøres ved å trykke på den gule trekanten med strek over i verktøylinja, som heter "Offline". Ikke kople deg av helt enda.

IV .1.1 Tilordning av I/O

Hvis vi så trykker på "Configurations and Setup" i "Multiview Explorer" på venstre side av brukergrensesnittet, deretter "I/O Map", og så "NX Bus Master", så ser vi det at vi nå har importert hardware-oppsettet til PLSen fra dens minne, slik at det fysiske oppsettet vårt speiles i Sysmac. Vi kan derfor se de fire enhetene som er koplet til PLSen, altså NX-PF0730, NX-ID5442, NX-DA2605, samt NX-OD5256. Hvis vi nå kopler oss fra PLSen ved å trykke på "Offline", og deretter høyreklikker på "NX Bus Master", og så velger "Create Device Variable", så vil vi definere variabler basert på inn- og utgangene våre. Dette vil være nyttig senere når vi skal programmere. Hvis vi klikker på de fire NX-modulene under NX Bus Master vil vi få opp kartet over I/O-portene våre.

De digitale inngangene og utgangene til PLSen er koplet til heismodellen på følgende måte:

Port	Beskrivelse
Input Bit 00	Obstruksjon
Input Bit 01	Stopp-knapp
Input Bit 02	H1 - Bestillingsknapp nr. 1 inne i heisen
Input Bit 03	H1 - Bestillingsknapp nr. 2 inne i heisen
Input Bit 04	H1 - Bestillingsknapp nr. 3 inne i heisen
Input Bit 05	H1 - Bestillingsknapp nr. 4 inne i heisen
Input Bit 06	Opp-knapp i etasje 1
Input Bit 07	Opp-knapp i etasje 2
Input Bit 08	Ned-knapp i etasje 2
Input Bit 09	Opp-knapp i etasje 3
Input Bit 10	Ned-knapp i etasje 3
Input Bit 11	Ned-knapp i etasje 4
Input Bit 12	Føler 1. etasje
Input Bit 13	Føler 2. etasje
Input Bit 14	Føler 3. etasje
Input Bit 15	Føler 4. etasje
Output Bit 00	DIR - Retning på motor (Opp=0, Ned=1)
Output Bit 01	Lys i stopp-knapp
Output Bit 02	Lys i bestillingsknapp nr. 1
Output Bit 03	Lys i bestillingsknapp nr. 2
Output Bit 04	Lys i bestillingsknapp nr. 3
Output Bit 05	Lys i bestillingsknapp nr. 4
Output Bit 06	Lys i opp-knapp i etasje 1
Output Bit 07	Lys i opp-knapp i etasje 2
Output Bit 08	Lys i ned-knapp i etasje 2
Output Bit 09	Lys i opp-knapp i etasje 3
Output Bit 10	Lys i ned-knapp i etasje 3
Output Bit 11	Lys i ned-knapp i etasje 4
Output Bit 12	Lys i indikator for åpen dør
Output Bit 13	Ubrukt
Output Bit 14	Etasjeindikator bit 1
Output Bit 15	Etasjeindikator bit 2

Fra tabellen over, kan man se at man bare har to bit for å sette etasjeindikatorlysene. Dette er for spare utganger på PLSen. I vårt tilfelle, er lyset kodet slik:

Bit 2	Bit 1	Etasjeindikator lys
0	0	Etasje 1
0	1	Etasje 2
1	0	Etasje 3
1	1	Etasje 4

Sørg for at du er koplet fra PLSen, og gi passende navn til variablene i "I/O Map"

ved å endre på feltet "Variable". Eksempler på navn kan være "DI00", altså forkortelsen for "Digital Input 00", til "Input Bit 00". Her hører "NX-ID5442" til de digitale inngangene, og "NX-OD5256" til de digitale utgangene. Vi har også to analoge utganger fra "NX-DA2605", hvorav vi kun bryr oss om den første, altså "Ch1 Analog Output Value". Denne styrer pådragssignalet til heismotoren, og er mellom 0 og 5V, alt ettersom hvilken av de heksadesimale eller desimale verdiene man gir utgangen fra den følgende tabellen:

Heksadesimal verdi	Desimal verdi	Spenningsverdi
W#16#0000	0	0.0 V
W#16#3DFF	15871	2.5 V
W#16#7EFF	32511	5.0 V

Kople til PLSen ved å trykke på "Online", og trykk så på "Synchronize" og deretter "Transfer To Controller" for å synkronisere det vi har gjort i Sysmac med PLSen. Trykk "Yes" i de påfølgende dialogboksene. Sørg nå for at du er i "I/O Map". Dersom du nå trykker på knappene/bryterne på heisboksen, vil du se at de tilhørende inngangene går fra "False" til "True" under feltet "Value". I tillegg, sorg for at heisen er i nederste posisjon, slik at "Input Bit 12" er "True". Sjekk også at "Output Bit 00" er "False". Prøv nå å skrive 15871 til "Value"-feltet til "Ch1 Analog Output Value", altså den første porten til NX-DA2605. Du vil nå kunne se at heisen beveger seg oppover. Skriv 0 til dette feltet slik at motoren stopper, før du går videre.

IV.1.2 Utvikling og flashing av software

Nå skal vi introdusere hvordan man kan programmere PLSen. På venstre hånd av Sysmac, i "Multiview Explorer", klikker vi derfor på "Programming", så "POUs", "Programs", "Program0", og så til slutt "Section0". Her får vi opp programmet vårt, slik det skal være flashet hos PLSen. Akkurat nå skal diagrammet være helt tomt, dersom dere har fulgt guiden til nå. Vi minner på at vi kun kan endre på diagrammet når vi er offline, altså ikke koplet til PLSen. Kople dere derfor av PLSen nå.

Vi skal nå lage et kjempeenkelt program, som skal gi dere litt intuisjon for hvordan man gjør stigeprogrammering i Sysmac. Dette skal deretter flashes til PLSen. Før dere gjør oppgavene i denne laben anbefaler vi dere å lese "Section 4 Programming" i databladet, da særlig "4-5 Programming Ladder Diagrams". Skjermbildet som dere skal se i Sysmac nå er vist i figur 4.

Dere ser nå et stort hvitt felt med det vi i stigeprogrammering kaller for et trinn (eller "rung" på engelsk) helt øverst til venstre. Det er i slike trinn man gjør programmeringen. Ved å høyreklikke på det får man opp en del valg, som for eksempel å lage flere trinn (dersom man høyreklikker helt til venstre på trinnet), eller som å sette inn elementer i trinnet (høyre del). Helt til høyre, ved siden av programmeringsfeltet, har vi "Toolbox". Her har du alle funksjoner du kan trenge å bruke til stigeprogrammering. Vi skal her og nå lage et kjempeenkelt program som kun skal sette lyset i indikatoren for åpen dør, dersom obstruksjonsbryteren er aktiv.



Figure 4: Programmeringsområdet vårt med tomt stigetrinn.

Det første vi gjør er å velge den inngangen vi ønsker. Vi høyreklikker derfor på trinnet, og velger "Insert Input", og skriver inn "DI00", dersom dette er navnet vi ga til inngangen for obstruksjonsbryteren. Deretter trykker vi på trinnet igjen og velger "Function", og skriver deretter inn "MOVE" for å velge riktig "Function". Det denne gjør er at når feltet "EN" går høy, settes det som står på "In" på "Out". Derfor skriver vi "1" på "In", og "DO12" på "Out", ettersom det er dette vi har kalt utgangen for lysindikatoren for åpen dør. Nå har programmert slik at lyset skrur seg på når vi skrur på bryteren, og vi må programmere slik at det skrur seg av når obstruksjonsbryteren ikke lenger er aktiv. Lag derfor et nytt trinn ved å høyreklikke på den venstre delen av det første trinnet vårt, og deretter "Insert Rung Below". Nå kan du lage akkurat samme struktur som tidligere. Deretter høyreklikker du på den nye inputen og velger "Invert", og så velger "0" i stedet for "1" på "In". Nå er det enkle programmet vårt ferdig, og det bør se ut som i figur 5.

Så trykker vi på "Project", og deretter "Build Controller" i menylinja helt øverst, for å sjekke at programmet vårt ikke har noen feil. Dersom dette viser seg å stemme, kan vi kople oss til PLSen med "Online", og deretter flashe programmet vårt til PLSen ved å trykke på "Synchronize", og deretter "Transfer to Controller" i den påfølgende dialogboksen. Nå må vi sjekke at PLSen er i "RUN Mode". Dette er et symbol til høyre for "Online" på verktøylinja. Hvis denne lar seg trykke på, gjør dette, og hvis den allerede er grå, så er vi i riktig modus. Hvis vi nå trykker på obstruksjonsbryteren, så vil lyset som indikerer at døren er åpen lyse.



Figure 5: Programmeringsområdet vårt med stigediagram.

Nå har dere fått en liten innføring i Sysmac, og i oppgavene som kommer skal dere få lage et litt mer komplisert program, basert på god bruk av de utdelte databladene. Hvis dere har problemer underveis, anbefaler vi dere å sjekke appendiksen. Dersom dere ikke klarer å løse problemet med dette, sjekk det utdelte databladet eller kurskompendiet. Dersom dette ikke fungerer, spør en studass. Dersom dette ikke fungerer, ta en gåtur rundt Hovedbygget for å klarne tankene. Prøv deretter på nytt.

1 Oppgave 1 (100%) - Heisprogrammering med stigediagram

Å jobbe med tilpassede datasystemer er ofte en øvelse i å lese datablad, og å jobbe med PLS er på ingen måte noe unntak. En av grunnene til dette er det at tilpassede datasystemer er svært spesialiserte systemer, og i motsetning til generell programering i en IDE finnes det derfor ofte kun informasjon om disse datasystemene i dokumentasjonen som blir gitt av leverandøren.

I denne laboppgaven skal dere derfor bruke de utdelte databladene til å implementere grunnleggende heislogikk, og dette er derfor også en slags introduksjon til den neste laboppgaven i TTK4235, nemlig heislaben. Denne oppgaven inneholder to deler:

heisspesifikasjon og datablad. Tanken er at dere skal implementere heisspesifikasjonen i Sysmac og overføre den til vår NX102-9000. Til hjelp så har dere walkthroughen vi gikk gjennom i starten av dette lab-dokumentet, samt den utdelte dokumentasjonen.

Denne oppgaven skal gi dere grunnleggende kunnskap om programmering av PLSen med stigediagrammer. Før dere spør læringsassisterter, sjekk ut appendiksen, dokumentasjonen eller kurskompendiet for å se om dere finner svar på spørsmålene deres der.

1 .1 Dokumentasjon

Dere har fått utdelt to pdf'er med dokumentasjon, nemlig "Sysmac Studio Version 1 Operation Manual.pdf" og "Sysmac Kurskompendie.pdf". Førstnevnte er den fullstendige dokumentasjonen til Sysmac, og kan lastes ned fra Omron sin nettside [HER](#), men også i TTK4235 sin GitHub-organisasjon, [HER](#). Den andre er en slags introduksjon til programmet, og finnes også i GitHub-organisasjonen vår. Dersom dere lurer på noe veldig spesifikt ville jeg konsultert dokumentasjonen, mens kurskompendiet kan gi litt praktisk info om hvordan man faktisk bruker programmet. I denne laben forventes det bruk av dokumentasjonen, ettersom det er dette man får tilgang til dersom man skulle jobbet med NX102-9000 ute i industrien.

Tidligere i denne laben gikk vi gjennom hvordan man raskt kommer i gang med bruk av Sysmac, og dere trenger nok derfor ikke å gå så nøye gjennom de tre første kapitlene av dokumentasjonen. Kapittel 3 i dokumentasjonen, altså "Section 3 System Design", kan derimot være kjekk å ta en titt på dersom du har trøbbel med selve Sysmac-programmet.

Kapittel 4, "Programming", er det dere vil få mest bruk for, og da særlig delkapittel 4-5 "Programming Ladder Diagrams". Her vil dere finne det dere trenger for å programmere stigediagrammer.

Ut over dette vil kapittel 5, 6 og 7 kunne være nyttige for henholdsvis oppsett av PLSen, tilkopling til PLSen, samt debugging, dersom dette skulle trengs.

1 .2 Heisens funksjonalitet

Heisfunksjonaliteten skal implementeres gjennom stigeprogrammering, som består i å bygge stigediagram av stigetrinn slik vi gjorde tidligere i denne labteksten. I programmeringsområdet i Sysmac vil man finne byggeblokkene under "Toolbox", som man finner på høyre side av brukergrensesnittet. Her har man en søkefunksjon det kan lønne å benytte seg av. To viktige underkategorier i "Toolbox" er "Timer" og "Ladder Tools", hvor særlig "Rung", "Input" og "Output" er sentrale. Ellers refererer vi

til kapittel 4.5 i dokumentasjonen. Eller anbefaler vi å eksperimentere med diverse "Function" og "Function Block" i stigediagrammene deres. Disse dukker altså opp når man høyreklikker på et stigetrinn i programmerings vinduet. Her har man muligheten til å søke etter kodeord for ting man kan trenge, og det kan være lurt å teste ut kodeord som "logic", "timer", "MOVE", osv. Det er mulig å lese om de ulike byggeklossene direkte i Sysmac, også kan man også trykke på "Help" i menylinja helt øverst, eller se i den utdelte dokumentasjonen.

1 .2.1 Programmeringsoppgaven

Implementer følgende forenklede algoritme for heisoppførsel:

- Heisen starter i en etasje, og kjører deretter i heissjakten.
- Heisen stopper og åpner døra i 3 sekunder i hver etasje, og kjører deretter videre i samme retning til en ny etasje nås.
- Når heisen har nådd toppetasjen, settes retningen til nedover.
- Når heisen har nådd bunnetasjen settes retningen til oppover.
- Bestillingsknappene skal ikke ha noen virkning i denne forenklede implementeringen.
- Når en etasje nås skal det tilhørende etasjelyset lyse helt til en ny etasje nås.

A Appendiks - PC- og PLS-bryteren på heisboksen

Denne skal alltid stå på "PLS" under hele labjobbingen, ellers får ikke PLSen kontakt med heis-hardwaren.

B Appendiks - Brukernavn og passord

Dersom dere blir spurtt om brukernavn og passord i løpet av laben, så er dette henholdsvis "TTK4235" og "Sanntid15".

C Appendiks - Direkte tilkopling til PLS

Når vi skal kople til PLSen, så skal dette gjøres via en ethernet-kabel som går fra SanntidsPCen til PLSen utgang på framsiden markert med "PORT1 EtherNet/IP". Sørg for at dette er gjort. Deretter skal man kunne kople seg til PLSen gjennom å åpne opp Sysmac Studio, og deretter velge "Connect to Device" fra hovedmenyen. Her velger du "Direct Connection via Ethernet", og deretter "Connect".

Dersom dette ikke skulle fungere, sjekk at PLSen er koplet til SanntidsPCen med ethernet-kabel. Dersom dette heller ikke skulle fungere, prøv å bytte der kabelen er koplet til PLSen fra Port 1 til Port 2 eller omvendt.

D Appendiks - Programmering, bygging og flashing

Det er viktig å huske på det at vi må være offline for å både kunne programmere og endre på variabelnavn i Sysmac. Når dette er gjort, kan vi trykke på "Project" og så "Build Controller" for å bygge prosjektet vårt. Når vi nå ønsker å overføre programmet vårt til PLSen (altså å "flashe"), er vi nødt til å kople oss på ved å trykke på "Online", og deretter trykke på det runde hjulet "Synchronize", for så å velge "Transfer to Controller".

Det er verdt å være obs på det at vi må ha et stigetrinn, eller "rung", som ikke er tomt for at man ikke skal få feilmeldinger når man trykker "Project" og så "Build Controller".

E Appendiks - Å kjøre programmet på PLSen

Når programmet vårt er ferdig flashet til PLSen har vi lyst til å kjøre det. Da kopler man seg rett og slett på PLSen gjennom å være "Online", og så trykke på knappen "RUN Mode" som er rett ved siden av "Online"-knappen. Hvis denne knappen allerede er grå, betyr det at du allerede er i dette moduset.

F Appendiks - Hvor programmerer man?

På venstre hånd i Sysmac under "Multiview Explorer" trykker du på "Programming", deretter på "POUs", deretter på "Programs", deretter på "Program0", og så "Section0". I det store hvite feltet får du da opp en såkalt "rung", som er de to vertikale endepunklene med en horisontal linje mellom. Programmet vårt skal bestå av disse, og du programmerer her.

G Appendiks - Hvor finner vi I/O-oppsettet vårt?

På venstre hånd i Sysmac under "Multiview Explorer" trykker du på "Configuration and Setup", og deretter på "I/O Map". Dersom du har gjort synkroniseringen riktig vil du finne det faktiske PLS-oppsettet vårt med tilhørende digitale og analoge I/O-porter her.