

# Fag TTK4120

## Industriell datastyring

### Øving 1, PLS

#### **Obligatorisk øving**

Labveiledning undervisningsuke 3-5

Godkjenning av lab skjer på salen av stud.ass på sal innen undervisningsuke 5

Presentasjon og evaluering i undervisningsuke 16

#### **Praktiske opplysninger:**

Øvingen utføres på sanntidssalen, G232-G234, i Gamle elektro. Reservasjon av tid gjøres på liste på G232. Øvingen går ut på å modellere og implementere styresystemet for en personheis. Alt av mekanikk og elektronikk er koblet opp på forhånd. Dere skal implementere styresystemet ved hjelp av en PLS (Programmerbar Logisk Styringsenhet). Øvingen går over fire dager.

#### **Dag 1: Innledning til PLS**

Her vil dere bli kjent med Siemens PLS'er. Innledningsvis skal dere følge et eksempel på PLS-programmering i en "Getting Started" manual. Manualen er så detaljert at dere ikke trenger stoff hentet fra forelesningene. Etterpå skal dere lage noen enkle programblokker i tilknytning til personheisen.

#### **Dag 2: Sekvensprogrammering**

Vi starter med en innføring i sekvensprogrammering for PLS ved å følge et eksempel i en manual. Deretter skal dere implementere et svært forenklet styresystem for en personheis.

#### **Dag 3 og 4: Fullstendig implementeringen av styresystemet for personheisen**

Nå skal dere modellere og implementere det endelige styresystemet for en personheis. Styresystemet er mere omfattende enn det ser ut til ved første øyekast. Det vil derfor være en fordel om dere har et utkast til løsning før dere kommer på labben 3. dag.

Godkjenning blir gjort av stud.ass. på salen i veiledningstimene.

Presentasjon og evaluering vil bli gjennomført i uke 15 samtidig med den store labben i C-programmering av mikrokontrollere. Prosjektene skal presenteres (demonstreres på laboratoriet), og denne presentasjonen vil være tellende på den endelige karakteren. Det er viktig at dere tenker gjennom hvordan dere kan få fram hva dere har fått til og hva dere har lært på best mulig måte, da tiden til selve presentasjonen kan bli knapp. Det vil også være lurt å lage tilstrekkelig med dokumentasjon, slik at både dere og vit.ass. raskt kan sette seg inn i prosjektet. Husk at det går mange uker fra dere avslutter PLS-oppgaven til dere skal presentere den. For PLS-oppgaven sin del er det kun den fullstendige implementasjonen av personheisen som skal presenteres. Dere skal altså ikke presentere "kokebokoppskriftene" dere gjorde i manualene og de innledende oppgavene fra de to første ukene. Men det dere lærte de to første ukene er selvsagt viktig.

# INNHOOLD

<b>1. Kort om PLS .....</b>	<b>3</b>
<b>2. Personheis - funksjonell beskrivelse .....</b>	<b>4</b>
<b>3. Hardware .....</b>	<b>5</b>
3.1 Motordriften .....	5
3.2 Etasjedetektorene .....	6
3.3 Betjeningstavlene .....	6
<b>4. Interface mot PLS .....</b>	<b>7</b>
4.1 Digital IO .....	7
4.2 Analog IO .....	8
4.3 Simulert IO .....	8
<b>5. Test av utstyr .....</b>	<b>8</b>
<b>Oppgaver dag 1 .....</b>	<b>9</b>
<b>Oppgaver dag 2 .....</b>	<b>11</b>
<b>Oppgaver dag 3 og 4 .....</b>	<b>11</b>

## 1. Kort om PLS

PLS (Programmerbar Logisk Styringsenhet) er et utbredt verktøy for å løse automatiseringsoppgaver i industrien. En PLS består av en prosessor, digitale og/eller analoge innganger og utganger og eventuelle kommunikasjonsmoduler.

Utgangene er gitt av en logisk funksjon av inngangene. Den logiske funksjonen bestemmes av den styreoppgaven som skal løses. Inngangene styres gjerne av sensorer av forskjellige slag, ofte kalt signalgivere. Eksempler på signalgivere er fotoceller, endebrytere og detektorer. Utgangene styrer et sett av pådragsorganer. Eksempler på pådragsorganer er motorer, ventiler, lamper og releer.

I denne laboppgaven benyttes en SIMATIC S7-300 PLS, med en 313 prosessor samt analoge og digitale I/O-enheter. Kommunikasjonen mellom datamaskinen og PLS går over serielinja via et MPI-grensesnitt (**M**ultipoint **I**nterface). I tillegg til den fysiske PLS'en kan vi også simulere PLS'en på datamaskinen. Dette kan være nyttig under programutvikling.

SIMATIC STEP 7 (S7) er et Windows program for styring av Siemens SIMATIC S7-300 og S7-400 PLS'er. Grunnbjelken i programpakken er SIMATIC Manager, hvor en holder rede på prosjektene som lages og starter opp de tilhørende S7 verktøyene.

SIMATIC STEP 7 støtter alle de 5 programmeringsspråkene i den internasjonale standarden IEC 1131-3, Programmable Controllers. De 5 språkene er:

- Strukturert tekst (PASCAL liknende språk)
- Instruksjonslister (assemblerliknende språk)
- Funksjonsblokkdiagram
- Stigediagram (relesymboler)
- Sekvensielle funksjonsdiagram

De to første er tekstbaserte språk mens de tre siste er basert på grafiske symboler. Vi skal i hovedsak benytte de tre siste språkene i denne oppgaven.

## 2. Personheis - funksjonell beskrivelse

Heisen skal betjene fire etasjer. Inne i heisen finnes det et panel med fire knapper hvor en kan velge hvilken etasje man skal til. Bak hver knapp ligger det et lyssignal som settes når knappen er trykket. Lyset i knappen skal slukkes når heisen har nådd denne etasjen. På samme panel er det en stopp-knapp som skal nullstille alle etasjebestillinger og stoppe heisen.

Heisen inneholder også en obstruksjonsføler (berøringsføler). Når denne gir utslag skal heisen stoppe og døren åpnes hvis heisen er ved en etasje.

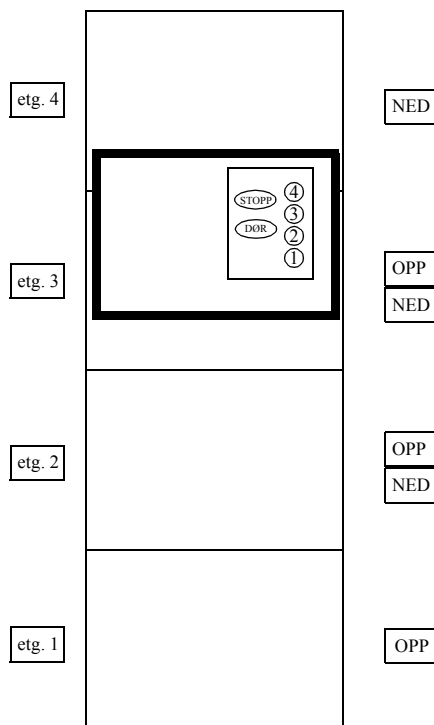
Ved hver etasje finnes en etasjedetektor. Når heisen har nådd ønsket etasje skal døra åpnes og holdes åpen i minst 3 sekunder. En samling på fire lamper (i hver etasje) viser hvilken etasje heisen er (eller sist var) i.

I hver etasje er det to trykknapper merket **OPP** og **NED**. Første etasje har bare **OPP**-knapp og fjerde etasje har bare **NED**-knapp. Når en av disse knappene er trykket skal heisen gå til vedkommende etasje. Hvis heisen allerede har et oppdrag, skal den kunne stoppe og ta med passasjerer som skal i samme retning. **OPP**- og **NED**-knappene har innebygd lys. Lyset skal tennes når knappen trykkes og slukkes når heisen tar med passasjerer i ønsket retning fra vedkommende etasje.

Pådrag til heisen kan gis ved disse tre aksjonene:

- Kjør opp
- Kjør ned
- Stopp

Figuren nedenfor viser en modell av heisen.



**Figur 2.1** Modell av heisen

### 3. Hardware

#### 3.1 Motordriften

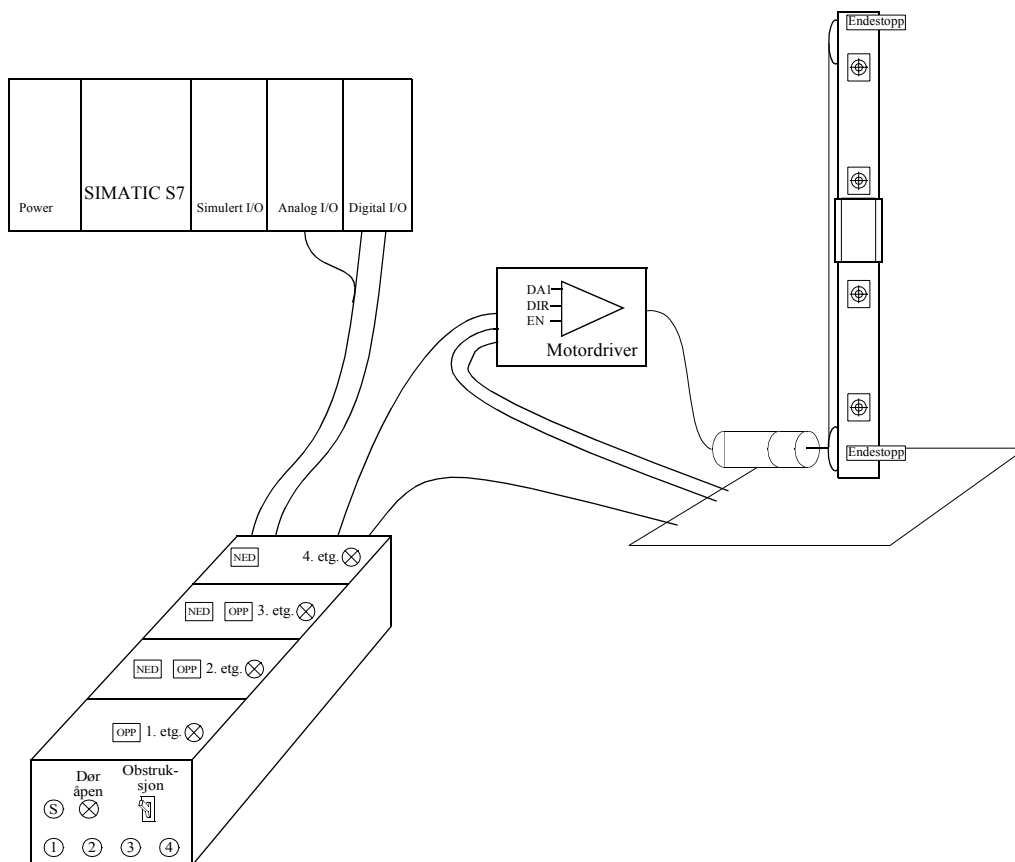
Heisen blir drevet med snortrekk av en likestrømsmotor. Motoren styres via en motordriver (merket med “modulboks” på labben). Motordriveren har innebygd strømforsyning. Ved hjelp av en kraftforsterker blir det analoge inngangssignalet til motordriveren omsatt til et pådrag til motoren. For å sette retningen på motoren, brukes et separat retningssignal.

Heisen er utstyrt med en øvre og en nedre endestoppbryter. Dersom heisen kommer utenfor operasjonsområdet vil endestoppbryterne sørge for at enablesignalet til motordriveren blir resatt og motoren stopper. For å kunne kjøre motoren etter at heisen har utløst en endestoppbryter, må vi skyve heisen tilbake igjen.

Enablesignalet med endestoppbryterne er kun en beskyttelsesmekanisme i hardware og skal ikke inngå i selve styresystemet.


Et gir med utveksling 1:100 er koplet på motoren. Momentet på motoren blir dermed så stort at den kan trekke heisen uten regulering (tilbakekopling). Når pådraget til motoren settes lik null vil tregheten i giret medfører at heisen blir stående i ro.

Sjekk at motoren er koplet til utgangen fra motordriveren (rød og blå ledning skal koples til klemmer merket M+ og M-).



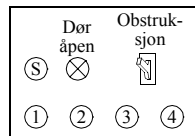
**Figur 3.1** Heismodell med betjeningstavler, motordriver og PLS

### 3.2 Etasjedetektorene

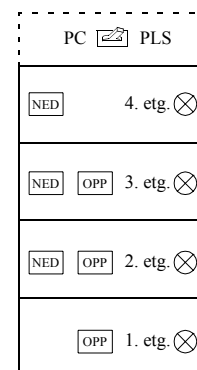
Hver etasje er utstyrt med en etasjedetektor med symbol  i figuren. Disse gir utslag ca. 0.5 cm før heisen kommer til en etasje og varer ca. 0.5 cm etter heisen har forlatt etasjen.

### 3.3 Betjeningstavlene

Av praktiske årsaker har vi valgt å legge betjeningstavlene for heisen avskilt fra den mekaniske modellen. Tavlene er montert på en boks som inneholder elektronikk for nivåtilpasning av de elektriske signalene. Boksen har tilkoplingsmuligheter for både PLS (24 V) og PC (5 V). Ønsket system (PC eller PLS) velges med en bryter på betjeningstavla. I denne oppgaven skal bryteren stå i PLS modus hele tiden.



Betjeningstavle i heis



Betjeningstavle for signaler fra etasjene

Knappene som har med bestilling av heisen (OPP, NED, 1, 2, 3, 4, og STOPP) vil gi en logisk ener så lenge de holdes inne. Når de slippes vil signalet igjen bli logisk null. Alle disse knappene har innebygd lys for å kunne kvittere at en knapp er trykket. Det finnes ingen kopling mellom bryterfunksjonen og lyssignalet i elektronikken så lyset må settes av software.

I heisen finnes det en obstuksjonsføler som gir utslag dersom noen kommer for nær heisdøra (på innsida) eller går ut av heisen når døra er åpen. I vår heis er dette realisert som en bryter som kan stå i en av to posisjoner. Ved siden av denne bryteren står en lampe som viser om heisdøra er åpen.

En samling på fire lysdioder skal vise hvilken etasje heisen befinner seg i (eller sist befant seg i). For å spare utgangssignaler er signalene kodet på følgende måte:

	etg. indikator 2	etg. indikator 1
etasje 1	0	0
etasje 2	0	1
etasje 3	1	0
etasje4	1	1

## 4. Interface mot PLS

### 4.1 Digital IO

De digitale signalene til og fra heisen er samlet i to 25-pins D-kontakter på boksen til betjenings-tavlene. Fra D-kontaktene er det laget en fast kabel til de digitale inn- og utgangene på PLS'en. Dette medfører at hvert signal får en bestemt adresse i PLS'en. Adressetabellen er som følger:

Adresse (PLS)	Signal heismodell
I 8.0	Obstruksjon
I 8.1	Stopp-knapp
I 8.2	H1 - Bestillingsknapp nr. 1 inne i heisen
I 8.3	H2 - Bestillingsknapp nr. 2 inne i heisen
I 8.4	H3 - Bestillingsknapp nr. 3 inne i heisen
I 8.5	H4 - Bestillingsknapp nr. 4 inne i heisen
I 8.6	Opp-knapp i etasje 1
I 8.7	Opp-knapp i etasje 2
I 9.0	Ned-knapp i etasje 2
I 9.1	Opp-knapp i etasje 3
I 9.2	Ned-knapp i etasje 3
I 9.3	Ned-knapp i etasje 4
I 9.4	Føler 1. etasje
I 9.5	Føler 2. etasje
I 9.6	Føler 3. etasje
I 9.7	Føler 4. etasje
Q 8.0	DIR - Retning på motor (Opp=0, Ned=1)
Q 8.1	Lys i stopp-knapp
Q 8.2	Lys i bestillingsknapp nr. 1
Q 8.3	Lys i bestillingsknapp nr. 2
Q 8.4	Lys i bestillingsknapp nr. 3
Q 8.5	Lys i bestillingsknapp nr. 4
Q 8.6	Lys i opp-knapp i etasje 1
Q 8.7	Lys i opp-knapp i etasje 2
Q 9.0	Lys i ned-knapp i etasje 2
Q 9.1	Lys i opp-knapp i etasje 3
Q 9.2	Lys i ned-knapp i etasje 3
Q 9.3	Lys i ned-knapp i etasje 4
Q 9.4	Lys i indikator for åpen dør
Q 9.5	Ubrukt
Q 9.6	Etasjeindikator bit 1
Q 9.7	Etasjeindikator bit 2

## 4.2 Analog IO

Det analoge pådragssignalet til motoren får følgende adresse på PLS'en:

Adresse (PLS)	Signal heismodell
PQW 272	Pådrag til motor

W angir at pådraget er av type **word**.

Vi ønsker at pådragssignal skal ligge i området 0 - 5 V. Noen aktuelle verdier vil da være:

Heksadesimal verdi	Spenning
W#16#0000	0 V
W#16#3DFF	2.5 V
W#16#7EFF	5.0 V

En enkel måte å sette ut det analoge signalet er å bruke MOVE-blokken.

## 4.3 Simulert IO

I tillegg til de fysiske digitale og analoge I/O-linjene har vi også noen "simulerte" digitale I/O-linjer. Disse ligger i modul nr. 4 i PLS'en og har adressene I0.0-I0.7 og Q0.0-0.7.

## 5. Test av utstyr

Er dere usikker på om utstyret er rett koblet opp eller fungerer kan dere kjøre testprogrammet xheis som ligger på E:\Siemens\Step7\S7proj. Testprogrammet lastes ned på vanlig måte fra SIMATIC Manager og virker slik: Når en av knappene på heispanelet trykkes, tennes lyset i knappen. Når "opp-knapp" i første etasje holdes inn går heisen oppover. Tilsvarende går heisen nedover når "ned-knapp" i fjerde etasje holdes inn.



## Oppgaver dag 1

Vi skal i starten konsentrere oss om programmering i stigediagram og/eller funksjonsblokkdiagram.

### Oppgave 1a

På hver labplass ligger det en manual med en detaljert veiledning for å komme i gang med programmering på PLS'er fra Siemens. Gå igjennom og utfør de 6 første kapitlene i manualen "Working with STEP 7 V5.1, Getting Started".

Dere trenger ikke å gå igjennom avsnittene som omhandler statementlist (STL) (4.3, 5.3 og 5.7). Dessuten er programmering i stigediagram (LAD) og programmering i funksjonsblokkdiagram (FBD) så like at det egentlig er nok å sette seg inn i en av delene, dvs. velg enten avsnittene 4.2, 5.2 og 5.6 eller avsnittene 4.4, 5.4 og 5.8. Men prøv gjerne å se programmet deres i alle tre språkene via menyen **View->LAD**, **View->FBD** eller **View ->STL**

Etter å ha utført de 6 første kapitlene i manualen har dere nå et program som er klart til å lastes ned.

### Oppgave 1b

En vanlig situasjon i industrien er at vi ikke alltid kan eller ønsker å teste ut styresystemet direkte på prosessen med en gang. Økonomi, sikkerhet, miljø og at utstyret ennå ikke er bygd ennå kan være gode grunner for å gjøre de første testene i en simulator.

Programmet dere skrev i pkt. a) er skrevet for en annen PLS enn den fysiske PLS'en vi har på labben. Vi vil derfor laste programmet ned i en simulert PLS.

- Start S7-PLCSIM fra SIMATIC Manager: **Options->Simulate Modules**
- I S7-PLCSIM oppretter dere de enhetene som trengs (bl.a. I/O): **Insert->Input Variable** og **Insert->Output Variable**. Dere må huske å sette inn aktuell adresse i de enhetene dere oppretter.

Se kapittel 3 i manualen "Testing Your S7 Programs with S7-PLCSIM" hvis dere trenger nærmere forklaring på bruk av simulatoren.

Den simulerte PLS'en vil i all hovedsak oppføre seg lik en virkelig PLS. Utfør kapittel 7 i "Working with STEP 7 V5.1, Getting Started" og last ned programmet deres i den simulerte PLS'en. Så lenge S7-PLCSIM er åpnet vil nedlasting automatisk gå til simulatoren. Hvis dere lukker S7-PLCSIM vil derimot nedlastingen gå til den fysiske PLS'en.

### Oppgave 1c

Opprett et nytt prosjekt knyttet til den fysiske personheismodellen som står på labben. Typebetegnelsene på modulene står på PLS'en. Vær oppmerksom på at slot 3 er reservert for kommunikasjon og skal stå åpen i systemet vårt. Den simulerte I/O-modulen i slot 4 står ikke i lista. Sett istedet inn en digital I/O-enhet med 8 inn- og 8 utganger (SM 323).

Skriv deretter et enkelt program som kjører heisen manuelt opp eller ned. Som inngangssignal brukes knappene på heispanelet: Når minst en av "Jeg vil ned knappene" trykkes skal heisen kjøres nedover. Tilsvarende skal heisen kjøre oppover når minst en av "Jeg vil opp knappene" trykkes. Når ingen av opp- eller ned-knappene holdes nede skal heisen stå i ro.

## Oppgave 1d

Gjør følgende utvidelse av programmet i oppgave 1c: Registrer hvilken etasje heisen sist passerte og sett ut lys i tilsvarende lysdiode på betjeningstavla for heisen. Lysdioden skal lyse helt til heisen når en ny etasje.

### Tips:

Etterhvert vil programmet dere lager bli forholdsvis stort. For å lette oversikten kan det lønne seg å gruppere programdelene. Lag for eksempel registreringen av sist passerte etasje som en funksjon (FCn) som dere kaller i OB1. (OB1 er en organisasjonsblokk som kjøres syklisk i PLS'en og er den eneste organisasjonsblokken vi trenger i hele denne lab-oppgaven. Funksjonsblokker (FB'er) og funksjoner (FC'er) ligger på nivået under organisasjonsblokkene i programhierarkiet. FB'er og FC'er må bli kalt fra en organisasjonsblokk for at koden skal bli utført. FB'er har en datablokk (DBn) knyttet til seg mens FC'er ikke har det. Hvis man ikke trenger å lagre unna data er det derfor best og enklest å bruke en FC (funksjon), noe som er tilfelle i oppgave 1d.)

## Oppgave 1e

Gjør følgende utvidelse av programmet i oppgave 1d: Når noen utløser obstruksjonsføleren i døra skal døra åpnes og holdes åpen i 3 sekunder etter at signalet fra obstruksjonsføleren har gått lavt. Deretter skal døra lukkes.

### Tips:

Se hjelpemeny eller manualer for å finne beskrivelse av timer instruksjonene. Kapittel 8 i manualen "Working with STEP 7 V5.1, Getting Started" kan også gi noen tips.

### Lagring:

Det kan lønne seg å ta vare på programmet i de foregående punktene da deler av det kan brukes senere i oppgaven. Trenger dere å ta med dere prosjektet kan dere enten kopiere alle filene STEP 7 lager eller dere kan lagre prosjektet komprimert. I sistnevnte tilfelle brukes menyvalget **File->Archive**. For å kunne jobbe med prosjektet senere må dere pakke det ut med menyvalget **File->Retrieve**.

## Oppgave 1f

Implementer en svært forenklet algoritme for personheisen:

- Stopp heisen og åpne døra i 2 sekunder i hver etasje.
- Etter 2 sekunder kjører heisen videre i samme retning til en ny etasje nås.
- Når heisen har nådd toppetasjeetasjen settes retningen til nedover.
- Når heisen har nådd bunnetasjen settes retningen til oppover.
- Bestillingsknappene skal ikke ha noen virkning i denne forenklete implementeringen.
- Ta gjerne med registreringen av siste passerte etasje som dere gjorde i oppgave 1d.
- (Hvis dere får lite tid kan dere forenkle algoritmen enda mere. La for eksempel heisen kun kjøre mellom to etasjer.)

## Oppgave 1g (Frivillig)

Når stoppknappen holdes inne skal heisen stoppe og lyset i stoppknappen blinke.

## Oppgaver dag 2

Temaet for dag 2 av oppgaven er sekvensprogrammering. Her skal vi bruke verktøyet “S7 GRAPH”

### Oppgave 2a

Gå igjennom og utfør kapittel 3 i manualen “S7 Graph V5.1 for S7-300/400 Programming Sequential Control Systems”. Prøv gjerne å laste ned dette eksempelet både til fysisk og simulert PLS.

Før dere laster ned programmet til den fysiske PLS'en må dere gjøre en liten endring i oppsettet til funksjonsblokken som inneholder sekvensprogrammet: I “GRAPH”-editoren velger dere **Options->Block Settings->Compile/Save->Executability->Full code**. Hele koden for sekvensen dere har programmert samt en del systemspesifikke ting blir da lagt inn i funksjonsblokken (FBn) og den tilhørende datablokken (DBn). Hvis systemet allerede har opprettet en blokk som heter FC72, må dere slette den fra prosjektet deres. FC72 inneholder en del generelle ting for administrering av sekvenser. Dersom vi kun har en sekvensblokk vil vi spare minneplass hvis vi genererer “full kode” i den aktuelle funksjonsblokka. PLS'ene vi har på labben har så begrenset minnekapasitet at vi kun har plass til 1 sekvensblokk og denne må være generert som “full kode”.

### Oppgave 2b

Implementer den svært forenklede algoritmen for personheisen som er beskrevet i oppgave 1f, men nå ved bruk av sekvensprogrammering (S7 GRAPH).

## Oppgaver dag 3 og 4

### Oppgave 3

Lag et PLS-program for heisstyringen som oppfyller den funksjonelle beskrivelsen gitt i kapittel 2.

**Obs:** Oppgaven er mer omfattende enn den kan se ut for.

#### Noen tips:

- Det kan lønne seg å dele opp programmet i flere funksjoner.
- Tenk igjennom hvilke funksjonaliteter dere vil realisere som sekvensdiagram og hva som er mest hensiktsmessig å realisere som rene logiske relasjoner.
- Det er ikke mulig å lagre en blokk med feil. Trenger dere allikevel å lagre en halvferdig blokk med feil må den lagres som “kildefil” isteden.
- Oppgaven er forholdsvis omfattende. Trenger dere å gjøre begrensninger i funksjonaliteten til heisstyringen, beskriv dette i besvarelsen. Synes dere derimot at oppgaven var for liten kan dere legge til funksjonalitet.