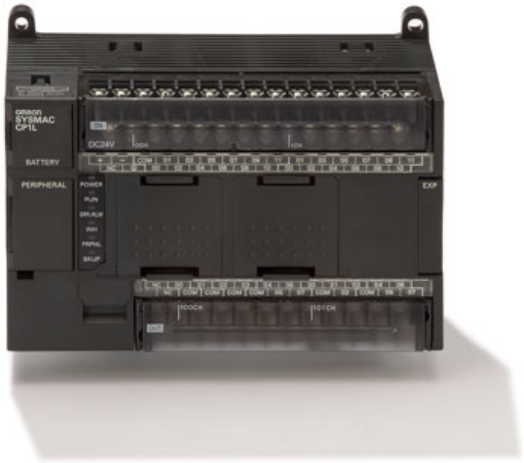
GRUNNKURS
I
PLS



Postadresse:
Postboks 109, Bryn
N-0611 Oslo

Besøksadresse:
Brynsalléen 4,
0611 OSLO

Telefon: 22 65 75 00
Fax: 22 65 83 00
Ordrefax: 22 65 83 01

Bankgiro 7262.05.0578
Foretaksregisteret:
NO 852 345 632 MVA

E-mail: omron.norway@eu.omron.com

Postadresse:
Postboks 109, Bryn
N-0611 Oslo

Besøksadresse:
Brynsalléen 4,
0611 OSLO

Telefon: 22 65 75 00
Fax: 22 65 83 00
Ordrefax: 22 65 83 01

Bankgiro 7262.05.0578
Foretaksregisteret:
NO 852 345 632 MVA

E-mail: omron.norway@eu.omron.com

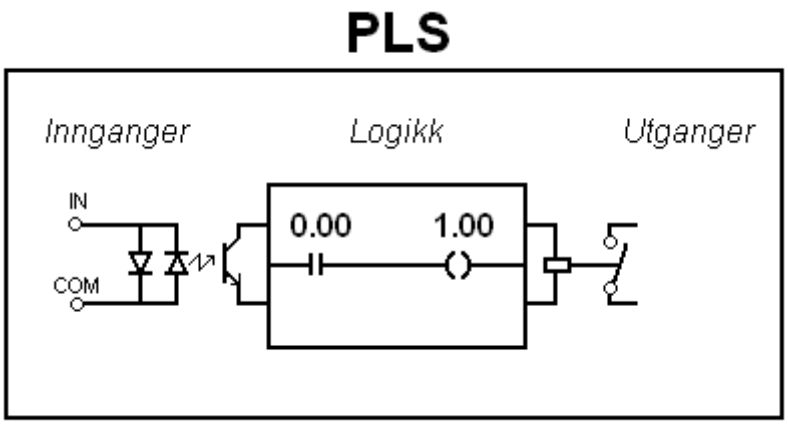
1 Litt om PLS

1.1 PLS, Hva er det?

PLS er en forkortelse for Programmerbar Logisk Styring.

En PLS er et styresystem som er bygget opp av digital elektronikk.

Skjematisk framstilling av en PLS:



1.2 Hvor bruker vi PLS?

Vi bruker PLS til styring av en eller flere funksjoner.

En PLS er en "universalboks" som ikke er produsert for å utføre bare en bestemt oppgave. PLS'en tilkobles de innganger som har betydning for styringen og de utganger som skal styres. Programvaren vi lager tilpasses til den jobben vi ønsker å utføre.

I prinsippet kan vi si at en PLS kan overvåke alt vi gir den informasjon om.

Eksempler på bruksområder:

- Maskinstyring
- Temperaturregulering
- Portstyring
- Vannverk
- Overvåking
- Sammensatte prosesser

NB! Etterhvert som PLSene blir kraftigere og billigere blir det stadig flere bruksområder. Dette stiller stadig større krav til de som skal drive og vedlikeholde PLS-anlegg.

1.3 Hvorfor bruker vi PLS ?

Det første kravet (og ofte det viktigste) er at det må være billigere å bruke PLS enn vanlige relé. tidsrelé etc.

Sammenlignet med en vanlig reléstyring er det mye enklere å gjøre endringer i en PLS-styring. Her kan vi endre på programvaren i stedet for å koble om på anlegget.

Ved hjelp av PC-programvare, kan det enkelt tas ut dokumentasjon av programmet.

Inn- og utgangene på en PLS er konstruert for å operere i industrimiljø, og tåler mye støy før anlegget blir påvirket av dette.

1.4 PLS typer

Omron har gjennom tidene hatt mange typer PLSer. Selve programmeringen og tenkemåten har alltid vært lik, men kapasitet og funksjonalitet har endret seg etter hvert. Dette kurset bruker en CP1L type PLS som utgangspunkt, men det skal være mulig å også programmere andre typer med samme underlaget.

Gjennom introduksjonen av CS1, CJ1, CP1H og CP1L PLSene markerte dette et markant skille, spesielt på minnestruktur, kapasitet og funksjonalitet. Minnet ble betraktelig større samt at organiseringen av de ulike områdene endret seg noe. Dette forklares senere i kompendiet. Antall instruksjoner/funksjoner ble også betydelig utvidet. Tidligere hadde en grovt sett ca 100 instruksjoner tilgjengelig. På CS1, CJ1, CP1H og CP1L er dette tallet ca. 400. Basert på disse endringene snakker vi om et generasjonsskifte innen PLS typene.

”Gamle” typer:
CPM1(A), SRM1, CPM2, CQM1(H), C200H(S)(α)

“Nye” typer:
CJ1, CS1, CP1H og CP1L

I tillegg finnes det en del enda eldre PLSer som ikke er nevnt her.

1.5 Programmeringsmetoder

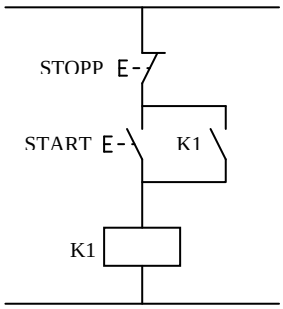
Programmet er en samling av instruksjoner som setter en PLS i stand til å utføre oppgaven den skal løse. Programmet vil da naturligvis variere med de forskjellige funksjonene PLSen skal ha.

For å programmere PLS'en bruker vi ulike programmeringsspråk. Disse følger internasjonale standarder vedtatt av International Electrotechnical Commission (IEC).

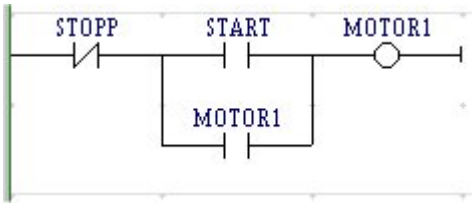
1.1.1 Ladder programmering

Med dagens programvare programmerer vi såkalt "ladder" programmering. Denne programmeringsformen stammer fra vanlig koblingsskjema. Ladder programmering ble utviklet i USA når store reléstyringer ble erstattet med mikroprosessorer. Når en programmerte disse prosessorene var det en fordel om "utseende" på programmet var like lett å lese som "gamle" relékoblingsskjema.

Styrestrømsskjema/reléskjema



Ladder program

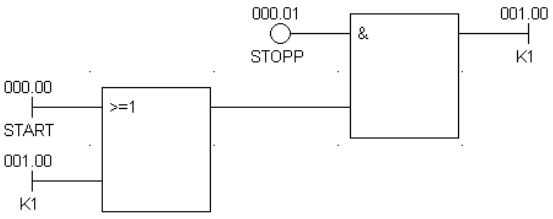


Av figurene over ser vi at likheten mellom koblingsskjema og tilsvarende ladderprogram er slående. Flyten i ladder går ovenfra og ned, mens den går fra venstre mot høyre for koblingsskjemaer. Symbolene har også endret utseendet, men likevel ikke så mye at omveltningen blir så stor når man skal starte å programmere en PLS.

NB!
Rent logisk blir overgangen fra reléskjema til ladder en smule feil i eksemplet ovenfor. For at dette skal bli korrekt, må stoppknappen kobles på samme måte som START og K1 inn til PLSen. Altså normalt åpen. En normalt åpen STOPP-knapp skal ikke forekomme av sikkerhetsmessige grunner. STOPP i Ladder skal derfor alltid programmeres som en ÅPEN kontakt.

1.1.2 Logikk programmering

I Omrons tidligere programmeringsverktøy, Syswin, hadde en muligheten til å programmere i såkalt logikkplan. Denne muligheten har bortfalt i CX-Programmer.



1.1.3 Instruksjonsliste programmering

LDNOT STOPP	LDNOT 0.01
LD START	LD 0.00
OR K1	OR 1.00
ANDLD	ANDLD
OUT K1	OUT 1.00

Den siste programmeringsformen er statement liste, eller instruksjonsliste. Dette er på denne formen programmet blir liggende på nede i PLSen. Bruker en en håndterminal vil programmet der se slik ut. Ligger ikke symbolnavnene i PLSen, vil bare adressen vises, mens i CX-Programmer vises kun symbolnavnet.

2 Minnestruktur

2.1 Ord, byte og bit

PLS'en består av flere forskjellige typer minneområder. Områdetyper og størrelser varierer med de forskjellige PLS familiene.

Minnemessig og instruksjonsmessig har en i Omron PLSene to plattformer. Ved introduksjonen av CS1, CJ1, CP1H og CP1L fikk en et skille når det gjelder kapasitet og muligheter på en PLS. Denne nye plattformen har mange flere instruksjoner og større minne en den gamle plattformen som inneholder alle andre PLS typer. Ved programmering av digitale/relé styringer vil disse forskjellene begrense seg til hvilken plass i minnet bit/reléer får. Utseende/strukturen på ladder programmet blir det samme. Det er når en kommer til analog behandling at forskjellen kan bli betydelig. "Kan" fordi de fleste gamle instruksjoner er beholdt men det er kommet mange nye instruksjoner.

Minneområdene i PLSen er organisert i kanaler (Ch), også kalt register eller ord (word).

Et ord består av 16 bit (2 byte), nr: 00 - 15 (Et bitnr. er to-sifret).
I en bit-adresse er de to sifrene til høyre for punktum bitnummer i ordet, og de resterende siffer til venstre er ordnummeret.

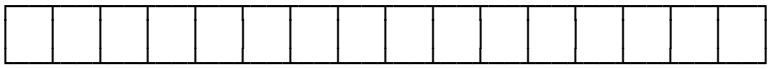
eks: Adresse 1.08 er ord nr.1 og bit nr.08.

Ett av disse minneområdene er CIO området (Core I/O area). En del av dette området brukes til digitale innganger og utganger. Måten inn- og utganger blir allokert på er ikke den samme for de forskjellige PLS familiene.

Skjematisk framstilling av et ord:

```
siffer:|    3      |    2      |    1      |    0      |

bit:    15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
```



Et ord kan også presenteres som 4 siffer, enten hexadesimalt eller i BCD kode (desimalt).

2.2 Minneområdene i Omron PLS

Som nevnt tidligere har Omron to generasjoner med PLSer når det gjelder bl.a. minnestruktur og størrelse. Tabellen under viser den siste generasjons minnestruktur. Størrelsene på de ulike områdene varierer med PLS typen. Det refereres til "Operation Manual" for aktuell PLS for detaljert spesifikasjon av hvert minneområde. For CP1L, CJ1 og CQM1H finnes det en detaljert oversikt i appendikset.

Minnestruktur
CIO (Core I/O Area)
W (Work Area)
CF (Condition Flag Area)
H (Holding Area)
A (Auxiliary Area)
TR (Temporary Area)
T (Timers Area)
C (Counters Area)
IR (Index Register Area)
DR (Data Register Area)
TK (Task Flag Area)
D (Data Memory Area)
E (Extended Data Memory Area)
UM (Program Memory) (Bare CJ1 og CS1)
FM (File Memory) (Bare CJ1 og CS1)

PLSer som har ny generasjons minnestruktur er; CS1, CJ1, CP1H og CP1L. Alle andre PLS typer følger gammel struktur. Se bakerst i grunnkurset for oversikt over gammel minnestruktur.

2.2.1 CIO området

CIO området er et minneområde brukt til inn-, utganger og internt programminne (det som er ledig). CIO området kan behandles som både ord og bit og brukes uten benevning i programmet. Deler av CIO området kan være reservert til spesielle I/O moduler. På CP1L PLS'en begynner inngangen 0.00 og utgangene på 100.00

2.2.2 W området

W området er kun til bruk av interne bit/ord under programmeringen.

2.2.3 CF området

CF området brukes til spesielle funksjoner. Disse bitene er «read only», og kan derfor ikke overskrives.

Mye brukte CF bit:

CF113	Alltid på
CF114	Alltid av
CF104	1 minutts klokkepuls
CF103	0,02 sekunds klokkepuls
CF100	0,1 sekunds klokkepuls
CF101	0,2 sekunds klokkepuls
CF102	1 sekunds klokkepuls
CF004	CY: Carry flagg, oppdateres av enkelte instruksjoner
CF005	GT: Større enn flagg, oppdateres av enkelte instruksjoner
CF006	EQ: Lik flagg, oppdateres av enkelte instruksjoner
CF007	LT: Mindre enn flagg, oppdateres av enkelte instruksjoner

2.2.4 H området

H området har samme funksjon som internminne i CIO området og W området, med den forskjell at det er batteribackup på H. H bit/ord vil bevare verdien ved strømbrudd.

2.2.5 A området

A området er et område for system informasjon og kontroll.
Mye brukt A bit: A200.11 På første scan etter oppstart

2.2.6 TR området

TR området brukes av programvaren og **ikke** av programmereren.

2.2.7 T/C området

T og C områdene er et område for timere og tellere. C området har batteribackup ved strømbrudd.

2.2.8 IR/DR området

IR og DR områdene kan brukes ved indirekte adressering. Ved bruk av * og @ foran D og E områdene får en indirekte adressering på disse på henholdsvis BCD og hexadesimal form uten å bruke IR/DR.

2.2.9 TK området

TK området er for statusvisning av hvilke tasker (program) som er i drift (run mode).

2.2.10 D/E området

D og E områdene er et dataområde beregnet på ord (words). Disse kan bare behandles som bit med spesielle instruksjoner. Det er vanlig å bruke området til databehandling og datalagring. Deler av D området kan være reservert til spesielle funksjoner og eller I/O moduler. D og E områdene har batteribackup ved strømbrudd. Deler av E området kan omgjøres til FM.

2.2.11 FM området

FM området får man tilgang til når man installerer et minne kort, eventuelt så kan deler av E området omgjøres til FM. FM området har MS-DOS struktur, og oppleves som en ”disk” der filer av ønsket karakter kan lagres. Typisk kan man lagre alle filer som hører til et prosjekt; alt fra PLS programmet til elektriske tegninger og koblingslister. Minnekortene fåes i størrelser fra 8Mb og oppover.

2.3 Modulers minneokkupering

Dette kompendiet brukes sammen med en CP1L som i likhet med storebroren CP1H har et fast antall I/O. Omrons modulære PLS, CJ1 og den rackbaserte PLS’en CS1 har et annet oppsett for minneallokering.
CJ1 og CS1 PLSene har 3 ulike kategorier av moduler. Disse **er Basic I/O, Special I/O** og **CPU Bus moduler**.

1.1.4 Basic I/O Units

Modulene adresseres fortløpende sett fra CPUen og utover. Hver modul okkuperer så mange ord den behøver. Neste modul starter fra neste ledige ord. Basic I/O modulene begynner adresseringen fra (CIO) 0000.00; som er ord 0, bit 0. Modulene har ingen oppsett utover det som måtte være i fronten eller under tilkoblingsblokken på modulen.

Eksempel på Basic I/O moduler: Digitale inn- og utgangsmoduler og enkel temperatursensor modul.

1.1.5 Special I/O Units

Modulene adresseres etter modulens MACH No. innstilling i front. Dette er to vribrytere som tilsammen kan gi et nummer mellom 0 og 95. Modulene har oppsett i dataområdet (D2**00**00 →) og driftsinformasjon i CIO området (2**00**0 →). Det reserveres et multiplum, avhengig av modultype, av 100 D ord og 10 CIO ord pr modul.

Eksempel på Special I/O moduler: Analoge inn- og utgangsmoduler, temperatur moduler, tellemoduler, posisjoneringsmoduler og CompoBus/S master kommunikasjonsmodul.

1.1.6 CPU Bus Units

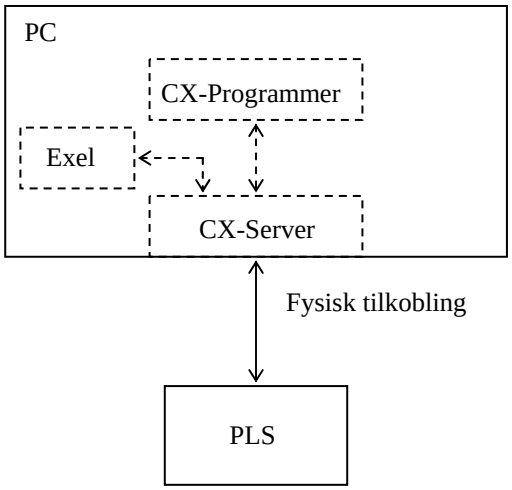
Modulene adresseres etter modulens UNIT No. innstilling i front. Dette er en vribryter som kan gi et nummer mellom #0 og F. Modulene har oppsett i dataområdet (D3**00**00 →) og driftsinformasjon i CIO området (1500 →). Det reserveres et multiplum, avhengig av modultype, av 100 D ord og 25 CIO ord pr modul.

Eksempel på CPU Bus moduler: Ethernet modul, seriell modul, Controller Link modul og DeviceNet modul.

3 CX-Programmer

3.1 CX-Programmer vs. CX-Server

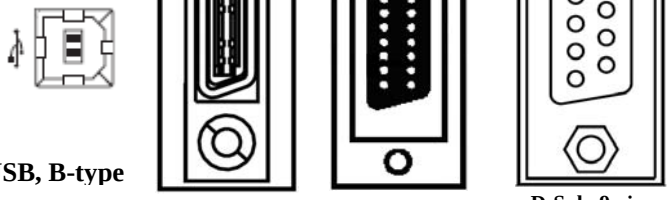
Dagens programvare for å programmere Omron sine PLSer heter CX-Programmer. For å kommunisere med en PLS fra PCen vi programmerer fra må vi også ha et program som heter CX-Server. Dette er et rent kommunikasjonsprogram som formidler alle henvendelser til PLSen. CX-Server medfølger CX-Programmer og disse to kan sees på som et program til vanlig. Det er mulig å få CX-Server til å kunne snakke med andre enn CX-Programmer. Dette hvis en samtidig vil ha kontakt med andre programmer som ikke er en del av Omrons programspekter. Eksempelvis hvis en vil overføre informasjon til et Excel ark for logging, trending, eller kun visning. Figuren under viser denne sammenhengen. Heltrukne linjer er maskinvare (hardware) mens stiplede linjer er programvare (software).



3.2 Kommunikasjonsinnstillinger

På de PLS typene som er nevt tidligere, finnes det fire ulike fysiske plugg-tilkoblinger. CP1L og CP1H har en USB B-type plugg, mens CJ1 og CS1 har to plugger i selve CPUen, den ene en 9 pins D-Sub og den andre en gammel eller ny "Peripheral" plugg. PLSene CPM2C, CQM1H, CJ1 og CS1 har den nye "peripheral" pluggen. De andre typene har den gamle pluggen.

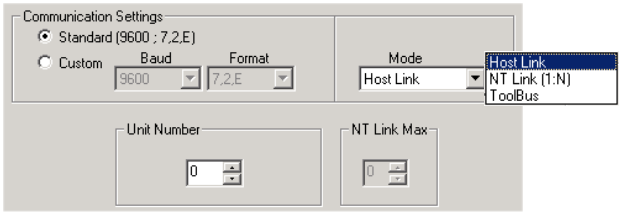
På CP1L og CP1H kan man plugge inn optionkort i front av PLS'en, og på denne måte utvide antall porter på PLS'en. På denne måten kan PLS'en snakke med annet utstyr. Oppsettet er likt som på CJ1-PLS'ene som blir beskrevet under:



USB, B-type | **Ny peripheral** | **Gammel peripheral** | **D-Sub, 9 pin**

CJ1-type PLS med ny *Peripheral Port* øverst og Host Link Port (D-Sub, 9 pin) port nederst.

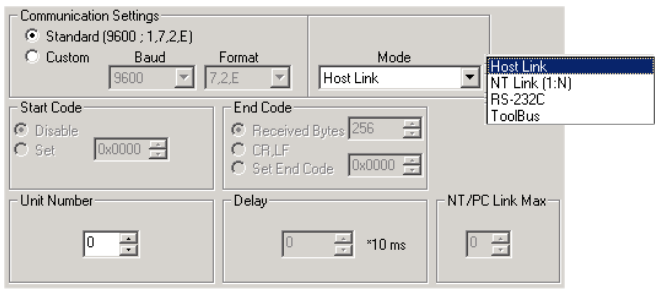
For CJ1 kan en velge hvilket språk (fra nå av kalt protokoll) som skal brukes på hver plugg (fra nå av kalt port). Internt i PLSen kan en velge blant flere ulike protokoller. Får å få kontakt med en port, må PCen (eller annet utstyr) ha innstilt samme protokoll og kommunikasjonsparametre som PLSen sin port. Aktuelle innstillinger gjøres under *Settings* i CX-Programmer. Til høyre er det vist innstillingene for *Peripheral Port* øverst og *Host Link Port* (D-Sub, 9 pin) nederst. *Mode* er det samme som protokoll type. Under et deksel over portene, er det montert 8 brytere. Bryter 4 og 5 brukes til å overstyre innstillingene internt i PLSen, da for henholdsvis *Peripheral Port* og *Host Link Port*.



Innstillinger for *Peripheral Port*

Når *Factory* er valgt, vil en kunne koble seg til med vilkårlig hastighet med *Toolbus* som protokoll. Med *Toolbus* valgt under *Settings*, vil *Format* innstillingene være fast 8,N,1.
NB! Host Link = Sysmac Way!

Bryter (Port)	OFF	ON
4 (Peripheral)	Factory	Settings
5 (Host Link)	Settings	Factory



Innstillinger for *Host Link Port* (D-Sub, 9 pin)

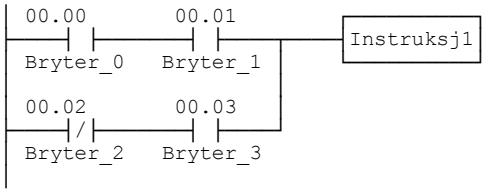
4 Grunnleggende instruksjoner

4.1 Bit status instruksjoner

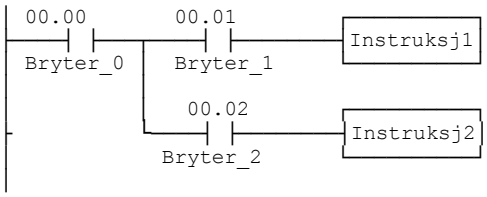
Et ladderdiagram består av en eller flere logiske betingelser som avsluttes i en eller flere instruksjoner. Når betingelsene er oppfylt, utføres instruksjonen(e).

Logiske betingelser:	AND	AND NOT	seriekobling	(og funksjon)
	OR	OR NOT	parallellkobling	(eller funksjon)

Eks. 1:



Eks. 2:



I eksempel 1 vil instruksjon 1 bli utført enten når bryter 0 og bryter 1 er operert eller når bryter 2 ikke er operert og bryter 3 er operert.
I eksempel 2 vil instruksjon 1 bli utført når bryter 0 og bryter 1 er operert. Instruksjon 2 vil bli operert når bryter 0 og bryter 2 er operert.

Instruksjoner:	OUTPUT	OUTPUT NOT	på/av bit (utganger)
	SET		set bit
	RESET		reset bit
	KEEP		set/reset vippe
	DIFU		flankedetektering
	TIMER		tidsrelé
	COUNTER		teller
	END		end

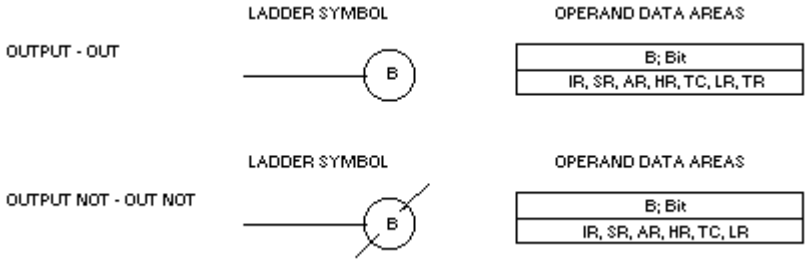
Programmet må alltid avsluttes med en END-instruksjon. Når en oppretter nye program i CX-Programmer, vil END automatisk legges til.

4.2 Bit kontroll instruksjoner

Disse instruksjonene kontrollerer statusen på enkeltbit (adresser) på og av på forskjellige måter. Et bit bør bare kontrolleres på/av en gang i en bit instruksjon (ikke SET - RESET instruksjonene).

For fullstendig beskrivelse av instruksjonene, les manualen eller Hjelp i PC-programmet.

4.2.1 OUTPUT - OUTPUT NOT

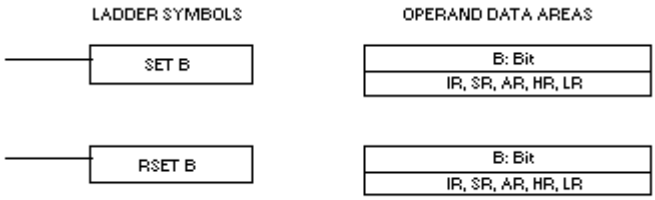


Beskrivelse: OUT og OUT NOT kontrollerer operand-bitet på og av etter status på betingelsen.

OUT kontrollerer bitet på hvis betingelsen er på og kontrollerer det av hvis betingelsen er av.

OUT NOT er inversfunksjonen til OUT.

4.2.2 SET - RESET

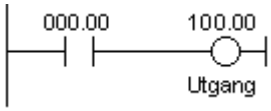


Beskrivelse: SET kontrollerer operand-bitet på når betingelsen er på, men påvirker ikke status på bitet når betingelsen er av.

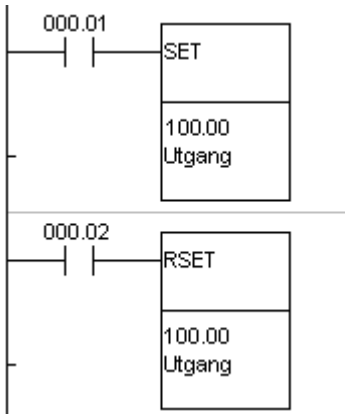
RSET kontrollerer operand-bitet av når betingelsen er på, men påvirker ikke status på bitet når betingelsen er av.

Eksempler: Eksemplene under viser forskjellen på OUT og SET/RSET. I det første eksempelet vil adresse 100.00 være satt på når 00.00 er på og av når 00.00 settes lav. I det andre eksempelet vil 10.00 settes på når 00.01 settes på. 100.00 blir stående på (uansett status på 00.01) til 00.02 settes på.

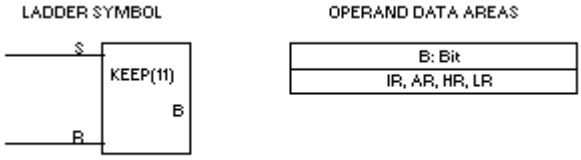
Eksempel 1:



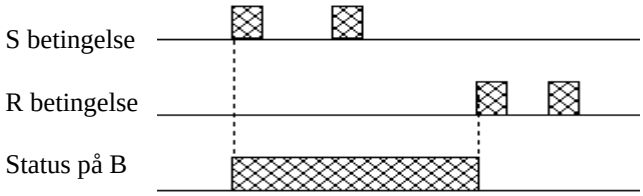
Eksempel 2:



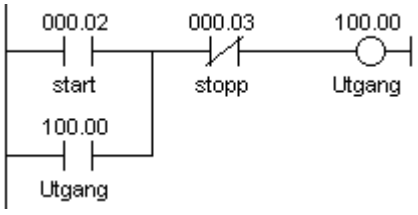
4.2.3 KEEP (11)

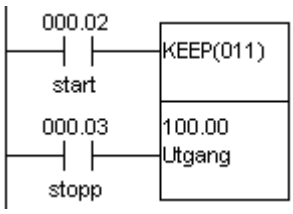


Beskrivelse: KEEP brukes til å sette verdien til et bit etter statusen på to betingelser S (set) og R (reset). Keep opererer som en set/reset vippe hvor resetbetingelsen er dominant.



Eksempler: Eksemplene under er laget på to forskjellige måter, men har nøyaktig samme funksjon.



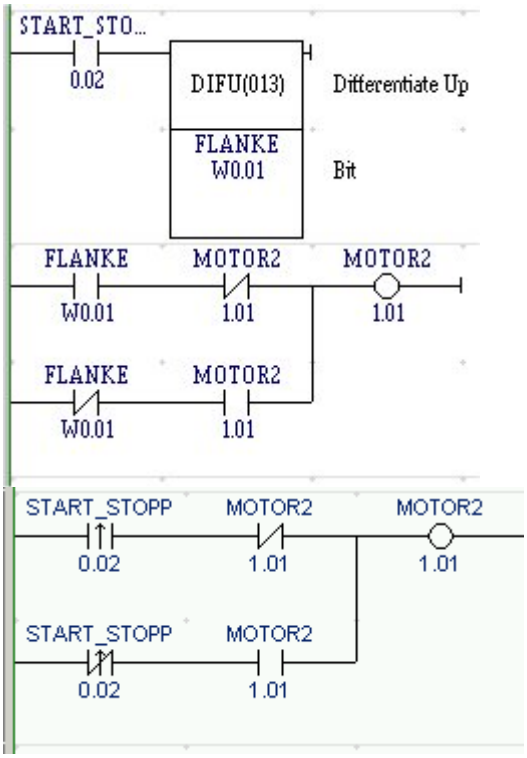


4.2.4 DIFFERENTIATE UP/DOWN - DIFU(13) / DIFD(14)

Beskrivelse: DIFU/DIFD instruksjonen blir brukt til flankedetektering. Når betingelsen går på/av aktiviseres operand-bitet i en programrunde (ett scan). Det kan brukes som betingelse på instruksjoner som bare skal utføres en gang på positiv/nagativ flanke, dvs. når betingelsen skifter fra av/på til på/av.

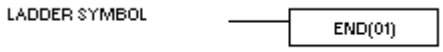
Eksempel: Det er koblet en impulsbryter til inngang 00.02. Inngangen skal fungere som en på/av bryter. For å lage en slik funksjon, må man detektere når bryteren aktiviseres, dvs. man trenger en flankedetektering. Hvis bryteren aktiviseres når utgangen er av, kontrolleres den på. Hvis bryteren aktiviseres når utgangen er på, kontrolleres den av.

Til høyre ser vi hvordan dette programmet blir i CS1, CJ1, CP1H og CP1L. Vi sparer W-adressen samtidig med at programmet blir lettere å lese.



4.3 Program kontroll instruksjoner

4.3.1 END (01)



Beskrivelse: END må være den siste instruksjonen i et program. Instruksjoner skrevet etter END vil ikke bli utført. Mangler END-instruksjonen, vil ikke programmet bli utført, og feilmeldingen «No End Inst» vil vises på PC eller Håndterminal.

4.3.1 JMP/JME – Program hopp

I enkelte tilfeller kan det være interessant å ikke utføre deler av et program. Et typisk eksempel på dette kan være hvis en ønsker å stoppe/fryse en timer. En kan da hoppe over utførelsen av den ønskede timeren, eller en kan samle timerene en ønsker å "fryse". Program som skal fryses omslynges av en hopp instruksjon (JMP) og en tilhørende slutt (JME). JMP og JME instruksjonene skal nummereres, og da slik at de pares. Antall mulige par avhenger av PLS type.

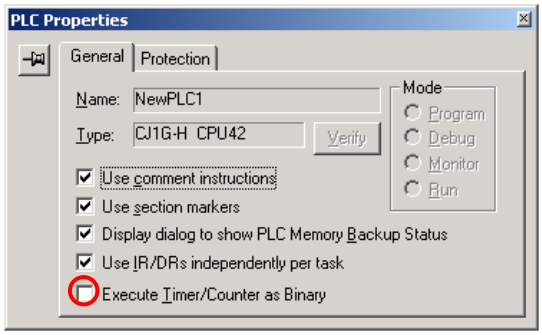
Det spesielle med JMP instruksjonen er at den har motsatt logisk virkning en hva som føles normalt. Når betingelsen foran JMP er aktiv/gyldig utføres programmet som vanlig (ingen hopp utføres), mens når betingelsen ikke er gyldig/aktiv utføres det et hopp fra JMP til JME.

For CS1/CJ1/CP1H/CP1L finnes det flere hopp instruksjoner. Disse er CJP og CJPN som begge skal pares med JME. CJPN har nesten identisk funksjonalitet som JMP, mens CJP har motsatt logisk virkning (muligens lettest å bruke/forstå).

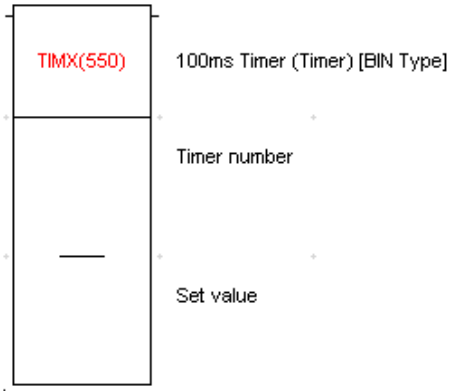
4.4 Tidsforsinkelse / Tellere

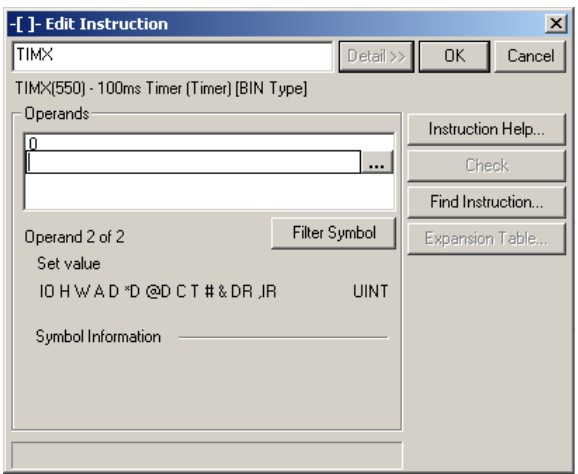
For CJ1 PLSene finnes det mange ulike varianter av både timere og tellere. I dette kompendiet er det kun tatt med de vanligste. For flere varianter refereres det til manualen eller hjelpen i CX-Programmer.

Vanligvis angis ønskede og virkelige verdier på BCD format. Skal disse verdiene kobles sammen med annen funksjonalitet i PLS programmet, vil det kunne være fornuftig å skifte over til desimal format på verdiene. Dette gjøres under egenskapene til PLSen (velg *Properties* i lista som kommer opp ved å trykke på høyre museknapp når en står over PLS ikonet i prosjektmenyen). Valget er vist med rød sirkel i vinduet til høyre.
Når desimale (*binary*) verdier er valgt, virker ikke BCD timere og tellere. Det er da et identisk sett med instruksjoner som har X som et tillegg i instruksjonsnavnet. Eksempelvis blir TIM til TIMX og CNT til CNTX.



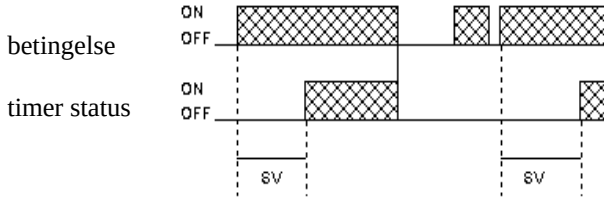
4.4.1



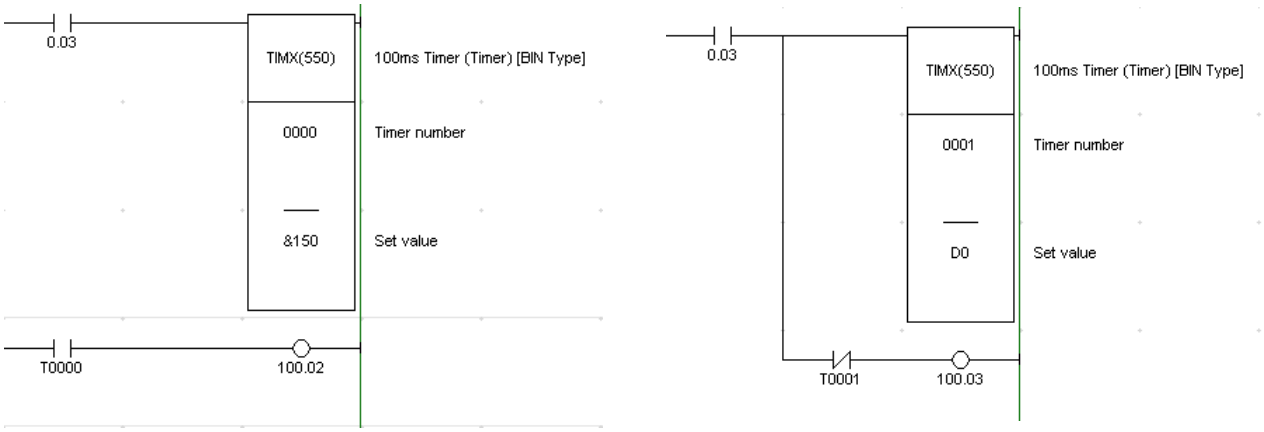


Beskrivelse: Timere beskrives i programmet med et nummer N og en forhåndsinnstilt set verdi, SV. Timeren aktiviseres når betingelsen er på og resettes når betingelsen går av. Når timeren er aktivisert teller timeren ned fra set-verdien (SV) mot null med enheten 0,1 sekund. Hvis betingelsen er på lenge nok og timeren når null, blir timeren satt. Timeren blir stående på til betingelsen går av.

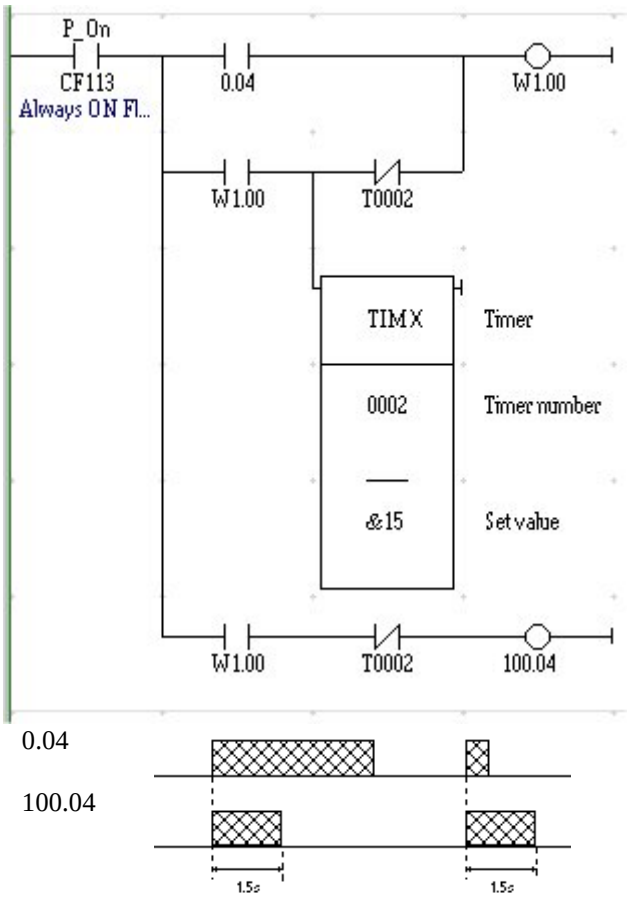
TIMX: Set verdien kan være fra &0 til &65535, dvs. til 6553,5 sekunder.
TIM: Set verdien kan være fra #0 til #9999, dvs. til 999,9 sekunder.



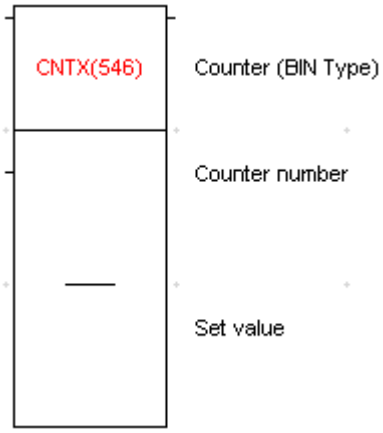
Eksempel 1: Her er vist to timere, TIM0 med konstant tid 15 sekunder og TIM1 med W register 100. TIM1 vil da ha set verdi lik verdien i register W100. Utgang 1.02 vil ha en forsinket ut funksjon. Den vil gå på 15 sekunder etter inngang 0.03. Utgang 1.03 vil gå på når inngang 0.03 går på og være på til TIM1 går på.

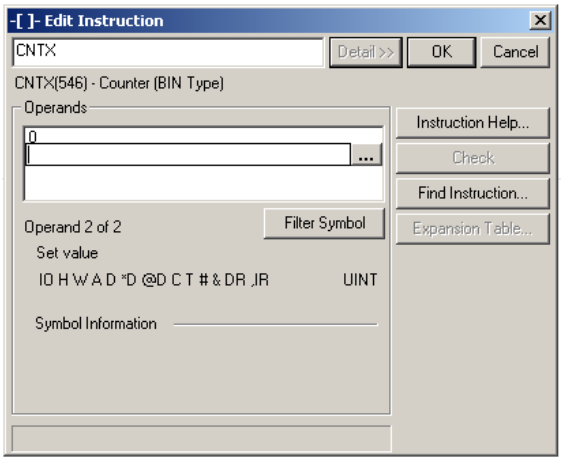


Eksempel 2: One-Shot. Denne programbiten gir en puls ut på en fast lengde, uansett lengden på inngangssignalet.



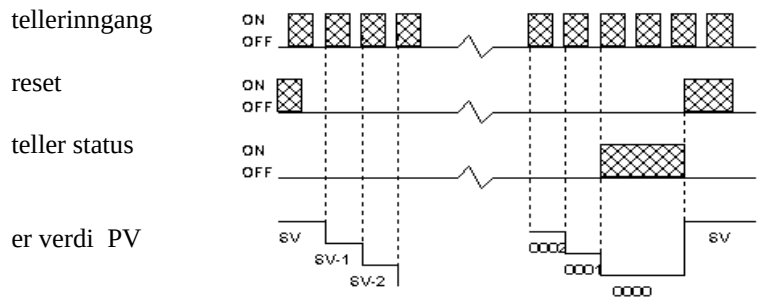
4.4.2 COUNTER – CNTX(CNT)



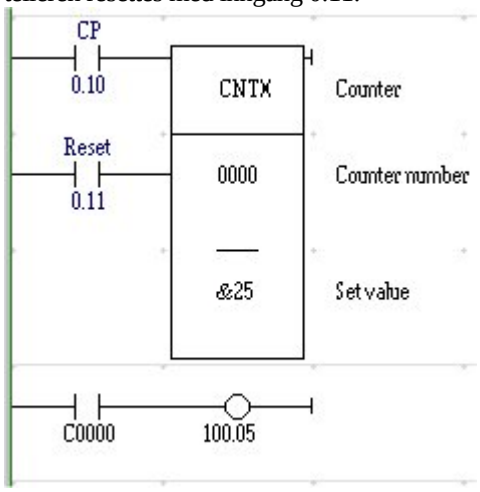


Beskrivelse: Tellere beskrives i programmet med et nummer N og en forhåndsinnstilt set verdi, SV. Telleren teller ned fra set verdien når tellerinngangen CP aktiviseres. Hver gang det er en positiv flanke på CP dekrementeres er verdien (PV) med en. Når PV er null, blir telleren satt. Telleren blir resatt når reset inngangen (R) aktiviseres. PV blir da satt til SV.

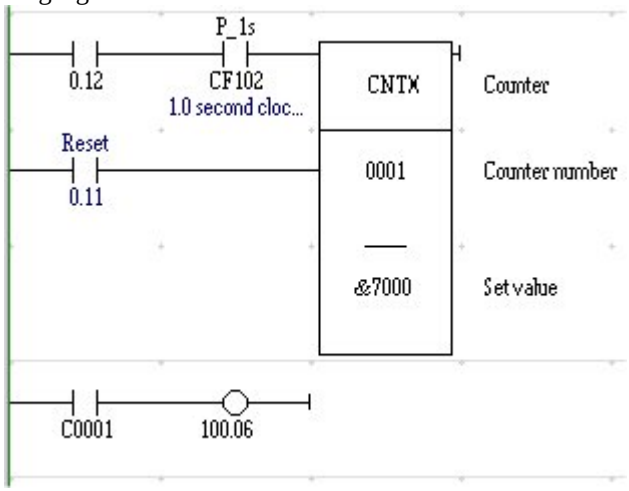
CNT: Setverdien kan være fra #0 til #9999.
CNTX: Setverdien kan være fra &0 til &65535.



Eksempel 1: For hver gang inngang 0.10 blir aktivisert, dekrementeres er verdien med en. Når 25 pulser er talt opp og inngang 00.02 ikke har vært aktivisert, går 1.05 på. Utgangen blir stående på til telleren resettes med inngang 0.11.



Eksempel 2: Teller som utvidet timer. Når inngang 0.12 er aktivert teller telleren ned med 1-sekunds klokkepuls. Etter 1 sekund x 7000, eller 116 minutter og 40 sekunder går utgang 1.06 på. Inngang 0.12 kontrollerer når telleren er aktiv.

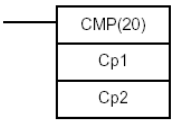


Det finnes også en reversibel (teller begge veier) teller, som heter CNTRX.

4.5 Sammenligninger

Det er flere instruksjoner for sammenligning av verdier. Når CS/CJ/CP1L/CP1H PLSene kom ble repertoaret betydelig utvidet, men den største forskjellen var innføring av instruksjoner uten bruk av statusbit, slik som var nødvendig tidligere.

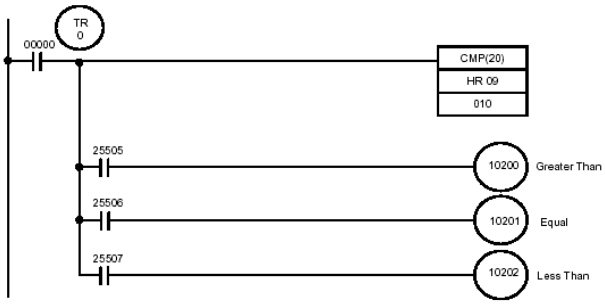
4.5.1 COMPARE - CMP



Cp1: First compare word
IR, SR, AR, DM, HR, TC, LR, #
Cp2: Second compare word
IR, SR, AR, DM, HR, TC, LR, #

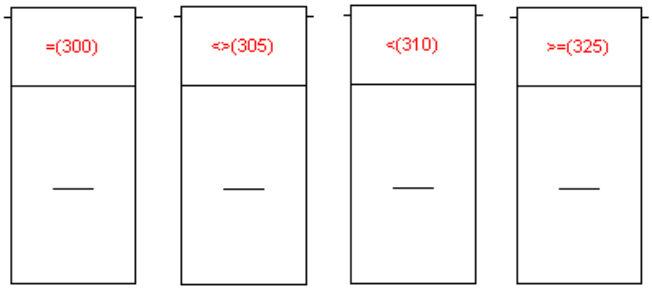
Flag	Address	C1 < C2	C1 = C2	C1 > C2
GR	25505	OFF	OFF	ON
EQ	25506	OFF	ON	OFF
LE	25507	ON	OFF	OFF

Compare brukes for å sammenlikne to verdier. Resultatet av sammenlikningen vil være 3 bit for henholdsvis >, = og <. Disse 3 bitene finner en i tabellen nedenfor (disse har skiftet adresse i CS/CJ/CP1L/CP1H). Vær oppmerksom på at alle COMPARE i programmet benytter de samme adressene. En bør derfor alltid lagre statusen på en eller flere av disse på hjelpebit i samme rung. Se eksempel nedenfor.



4.5.2 COMPARE – =, <>, <, >=, ...

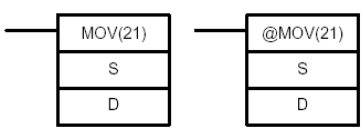
De nye sammenligningsinstruksjonene brukes som en vanlig betingelse/kontakt i PLS programmet. Det er også lettere å se når betingelsen er oppfylt, da hele boksen blir markert under monitorering.



4.6 Flytting / kopiering

Innenfor området data flytting/kopiering er det et uttall av instruksjoner. Her har vi kun tatt med den aller vanligste.

4.6.1 MOVE - MOV Kopiering

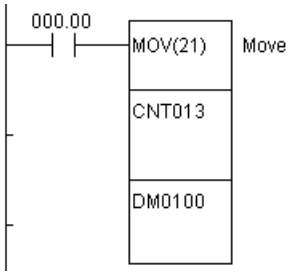


S: Source word
IR, SR, AR, DM, HR, TC, LR, #
D: Destination word
IR, SR, AR, DM, HR, LR

MOV kopierer data fra S til D. Som en ser til høyre kan S og D være for eksempel både IR, DM og innholdet i en Timer (S og D områdene i oversikten gjelder ikke for CS/CJ/CP1L/CP1H).

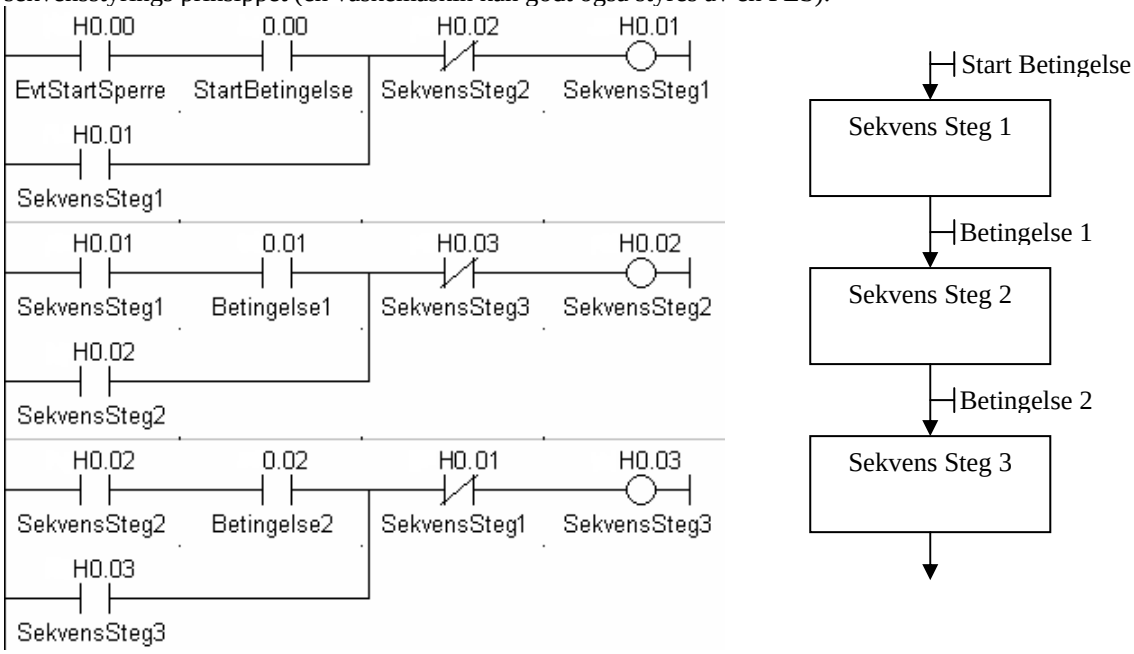
Legg også merke til @MOV. Det betyr at MOV bare blir utført idet betingelsen foran @MOV går fra 0 til 1. En aktivering på såkalt stigende flanke. Brukes !MOV (kun for CS/CJ/CP1L/CP1H) utføres instruksjonen på synkende flanke.

I eksempelet under vil telleverdien i CNT013 bli kopiert inn i DM0100 så lenge inngangen 000.00 er på. På denne måte kan en ta vare på verdier i gitte situasjoner for videre bearbeiding. Hadde vi benyttet @MOV ville vi registrert tellerverdien akkurat idet 000.00 gikk på. Selv om 000.00 er på lenger, vil ikke @MOV bli utført.



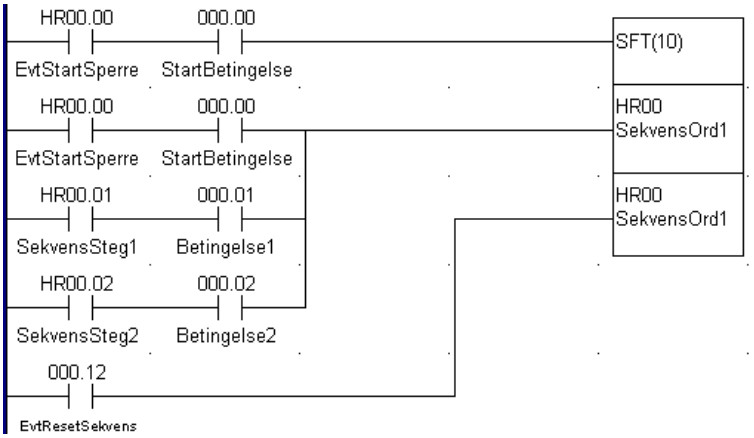
5 Sekvensprogrammering

I en sekvens styring ønsker en at noen hendelser skjer i en bestemt(e) rekkefølge (steg), der en venter på en tilbakemelding fra systemet fra sekvenssteg til sekvenssteg. Dette kan på en måte sammenlignes med virkemåten til en vaskemaskin, bare med den forskjellen at en vaskemaskin styres etter programstyrings prinsippet, ikke sekvensstyrings prinsippet (en vaskemaskin kan godt også styres av en PLS).



Når en skal lage en sekvensstyring kan/bør en bruke programmerings strukturen som vist over. En lager først et flyt/blokkskjema, slik at en har kontrollen over sammenhengene i systemet. Skjemaet bør være slik at personer som ikke er programmeringskyndige også kan lese og forstå tankegangen. TIPS: Flytskjemaet bør fullføres og godkjennes før en starter på selve programmeringen. Hver blokk/boks i flytskjemaet skal nå bli til en programstruktur som tilsvarer en (vilkårlig) holdekrets som vist i figuren over (flytskjemaet er en personlig avart av noe som kalles Grafset). Dette gir en enkel og grei oversikt, og gjør det lett å feilsøke (bruk først flytskjemaet for å finne ut hvor systemet avviker).

Her vil en bruke sekvensstegene videre til å styre av og på utganger, timere, tellere, … Betingelsen(e) vil bli oppfylt på grunnlag av at sekvenssteget den tilhører styrer noe som til slutt gir tilbakemelding.
Problemet med denne typen sekvensstyring i PLS, er at to sekvenssteg er aktive i et programscan. Dette skjer alltid i overgangen mellom to steg. Dette problemet kan fjernes ved å bruke et skiftregister til å holde orden på sekvensstegene. Dette er vist i figuren nedenfor. Ulempen med denne metoden er at den er mindre fleksibel hvis en har behov for å sette inn nye sekvenssteg inni mellom andre.



6 Installasjon og kabling

PLSene i SYSMAC-serien er meget driftssikre. Men for at de skal få utført sine funksjoner fullt ut, så er det noen forholdsregler en bør huske på under installasjonen:

Monteringsplassen og omgivelsesmiljøet
Unngå følgende:

- Steder der omgivelsestemperaturen er utenfor området 0 til 50°C. Dersom temperaturen ligger opp mot disse grensene, monter inn f.eks. varmekabel ved lave temperaturer og vifter for avkjøling ved høyere temperatur.
- Steder med hurtige temperaturendringer som medfører kondensering av vann.
- Steder med høyere relativ fuktighet enn 90%.
- Steder med korrosive- eller eksplosive gasser.
- Steder med ekstreme mengder med støv, salt eller jernstøv eller sponkonsentrasjon.
- Steder med kraftige vibrasjoner eller støt.
- Direkte sollys kan også medføre at en kommer ut over temperaturområdet.

Montering
Når en PLS blir montert i et skap eller i en maskin, ta hensyn til operasjonsdyktigheten og adkomsten ved en eventuell feilsøking/reparasjon eller utvidelse.

a) Ta hensyn til de nedenfornevnte punkter, slik at omgivelsestemperaturen ikke overskrider +50°C.

- Tilstrekkelig åpning rundt PLS'en for luftsirkulasjon.
- Unngå å montere PLS'en over varmeutviklende enheter.
- Monter PLSen horisontalt på en vertikal plate med fronten vendt fremover.

b) Dersom mulig, legg mest mulig av 220 volt utstyret i en avgrenset del av skapet eller i et eget skap. Dette p.g.a. sikkerheten for berøring og ellers for reduksjon av overslag til DC signalledere og generell støy.

c) Dersom 220/380 volts-kabler med store strømstyrker må legges i samme skap som PLS'en, så la det minst være 20 cm avstand mellom kabler og PLS-enhetene.

7 Igangkjøring av PLS anlegg

Ved prøve- og igangkjøring av PLS-anlegg bør en utvise stor varsomhet så ikke maskiner, utstyr og mennesker skades.

Etter at anlegget er montert, oppkoblet og påsatt styrespenning, legges programmet inn i PLS'en.

Første prøvekjøring bør så foregå etter disse retningslinjer:

1) **Viktig:** Velgerbryteren på programmeringsenheten skal stå i "**program mode**".

2) Prøv alle inngangsignaler. Husk også å prøve tilbakemeldinger fra kontaktorer, ventiler etc. ved hjelp av manuell betjening eller kortslutting av lukkekretser / frakobling av åpnekretser, slik at hovedspenning kan være av.

3) Prøv alle utgangssignaler. Dette kan gjøres lett med programmeringsenheten også når velgeren står i program mode.

   Kall fram aktuell utgang, og monitorer denne. Trykk så på "Force….On", og utgangen vil gå på. Se etter at det som PLS'en fysisk styrer blir aktivisert. observer også om eventuelle tilbakemeldinger til innganger går på/av. Husk å resette utgangssignaler ved å trykke "Force…Cancel" før neste utgang prøves.

4) Slå på hovedspenningen. Prøvekjør motorer, ventiler sylindrer etc. som i punkt 3. Se etter at motorer og sylindrer går i riktig retning når tiltenkt utgang aktiveres.

5) Når alt er tilsynelatende i orden, kan programmet forsiktig kjøres med velger i RUN/MONITOR. Med forsiktig menes at man på større eller kompliserte anlegg bare prøvekjører del for del av programmet. Dette kan gjøres ved å sette inn "END"-instruksjonen i passende del av programmet. Denne kan så flyttes nedover i programmet etter hvert.

6) Etter at programmet er prøvekjørt og korrigert, må dokumentasjonen rettes. Kjør utskrift av programmet, og lagre det på f.eks. en CD.

8 Eksempler

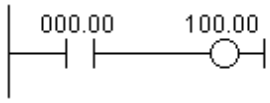
Adresseringen i CP1L er følgende:

Innganger: 000.00

Utganger: 100.00

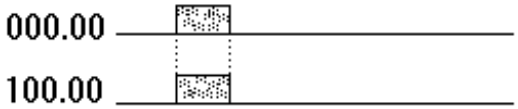
8.1 Eksempel 1a Innganger og utganger.

For å styre en utgang via en inngang lager man et skjema som ser ut som følger:



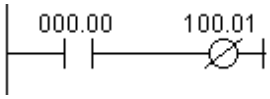
I dette skjema er 000.00 inngangen og 100.00 utgangen

Tidsdiagram for kretsen:



8.2 Eksempel 1b Innganger og utganger.

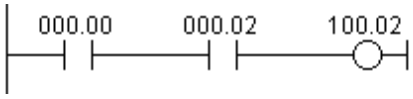
Når du vil ha en invertert funksjon av en utgang kan skjemaet se ut som følger:



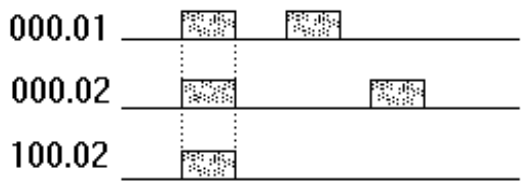
Tidsdiagram for kretsen:

8.3 Eksempel 1c Innganger og utganger i serie.

Når det er flere innganger som styrer en utgang kan disse være i *serie*

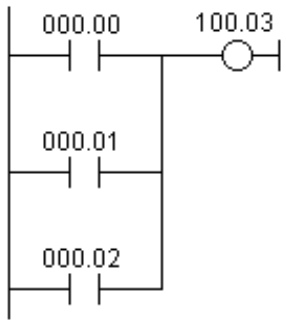


Tidsdiagram for kretsen:

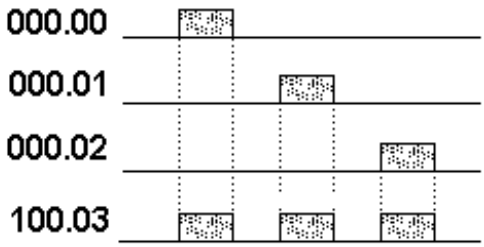


8.4 Eksempel 1d Innganger og utganger i parallell.

Parallell.

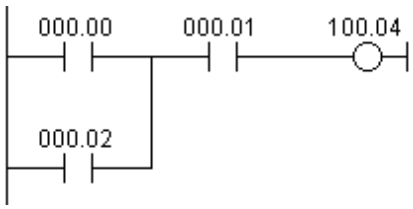


Tidsdiagram for kretsen:

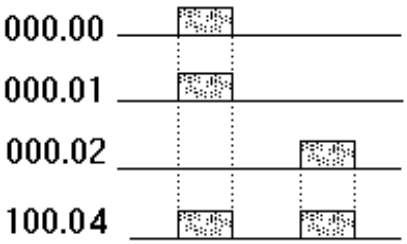


8.5 Eksempel 1e Innganger og utganger parallelt og i serie.

En kombinasjon av serielle og parallelle kontakter kan se ut som følger:



Tidsdiagram for kretsen:



Postadresse:
Postboks 109, Bryn
N-0611 Oslo

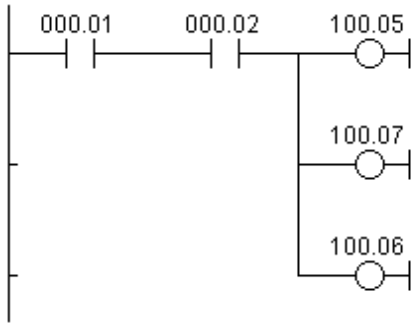
Besøksadresse:
Brynsalléen 4,
0611 OSLO

Telefon: 22 65 75 00
Fax: 22 65 83 00
Ordrefax: 22 65 83 01
E-mail: omron.norway@eu.omron.com

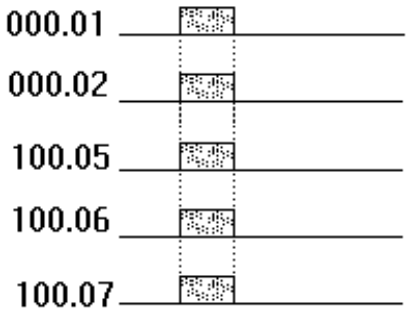
Bankgiro 7262.05.0578
Foretaksregisteret:
NO 852 345 632 MVA

8.6 Eksempel 1f Innganger og flere utganger.

Det kan naturligvis være flere utganger i en krets.



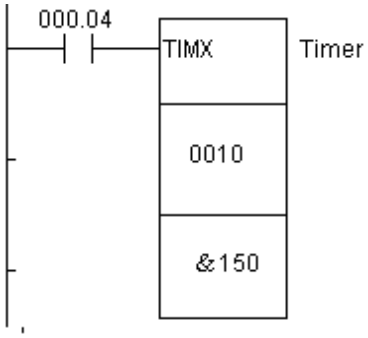
Tidsdiagram for kretsen:



8.7 Eksempel 2a Tidsfunksjoner

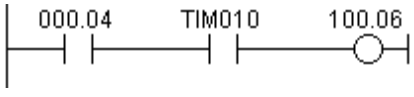
For å forlenge et signal kan du bruke en tidsfunksjon.
Det finnes ulike tidsfunksjoner med ulike oppløsninger på tidsbasen.

Grunntids funksjonen med tidsbasen 6553,5 sekunder er som følgende:

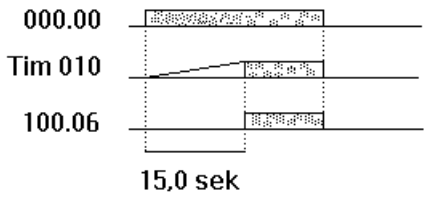


Den innstilte tiden er 15,0 sekunder

For å forlenge tiden lager du en krets og lar en kontakt få adressen fra tidsfunksjonen.
Tidsfunksjonens tid regnes fra innstilt verdi ned til null. Når tidsfunksjonen er null, aktiveres kontakten/ene med tidsfunksjonens adresse.

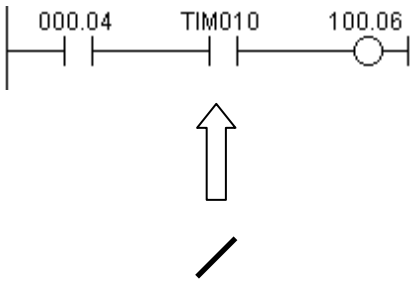


Tidsdiagram for funksjonene:

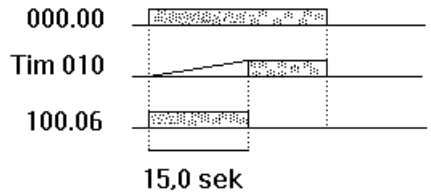


8.8 Eksempel 2b Tidsfunksjoner

Når du vil lage en forsinket utkopling legger du bare til ”not” funksjonen på kontakten.



Tidsdiagram for funksjonen:

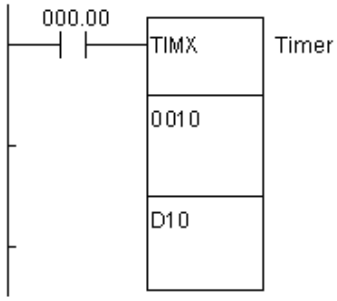


8.9 Eksempel 2c Endre tidsbasen under drift

Verdien for tidsfunksjonen kan skrives inn på to måter. Enten som en konstant verdi (& 0010) eller via et ord i et minneområde(DM 10).
Når du velger en konstant verdi skriver du in verdien direkte i tidsfunksjonen med tegnet & først.
Når du velger å hente verdier fra et ord i minneområde skriver du først minneordet og deretter skriver du inn verdien i det valgte ordet. Denne måten gjør det mulig å forandre tidsverdien under PLS-systemets drift. PLS-systemet MÅ være i ”monitor-mode”.

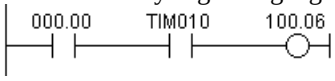
Eksempel:

RUNG # 1 Tidskrets

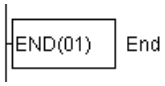


I DM 10 skriver du inn tidsverdien

RUNG # 2 Styring av utgangen



RUNG # 3 Slutt på programmet



Postadresse:
Postboks 109, Bryn
N-0611 Oslo

Besøksadresse:
Brynsalléen 4,
0611 OSLO

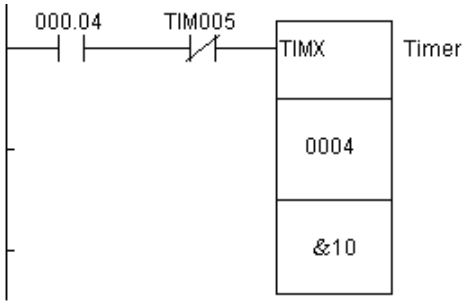
Telefon: 22 65 75 00
Fax: 22 65 83 00
Ordrefax: 22 65 83 01
E-mail: omron.norway@eu.omron.com

Bankgiro 7262.05.0578
Foretaksregisteret:
NO 852 345 632 MVA

8.10 Eksempel 2d Pause og gangtid.

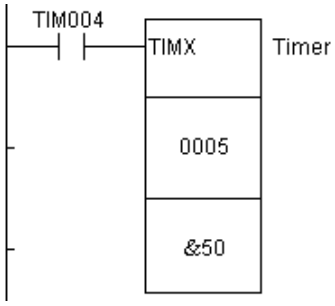
Når du vil lage en tidsstyring med *pause* og *gangtid* kan du bruke to tidsfunksjoner.

RUNG # 1



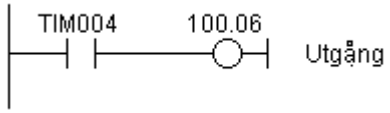
Pausetid

RUNG # 2

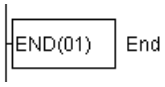


Gangtid

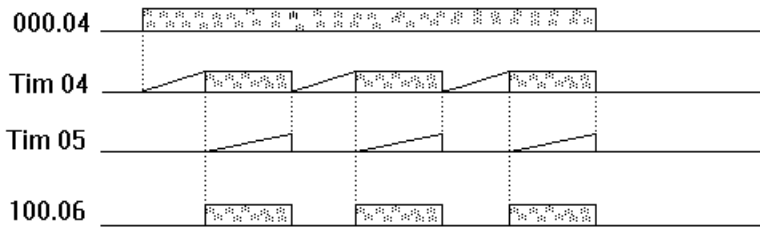
RUNG # 3



RUNG # 4



Tidsdiagram for programmet:

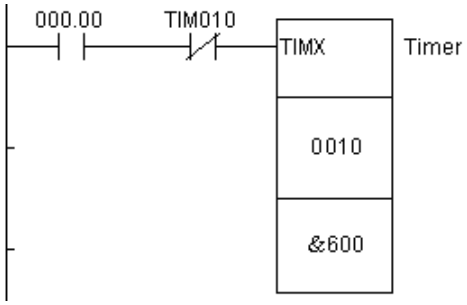


8.11 Eksempel 2e Lange tider

Tidsbasen i en grunntidsfunksjon er maksimalt 6553,5 sek (ca 18 timer). Dette holder ikke alltid, og da må man benytte seg av en teller som et minne der du f.eks. lagrer pulser som betyr 1 minutt. Du kan da få en tid som er 65535 minutter lang (ca 1092 timer -> 45dager).

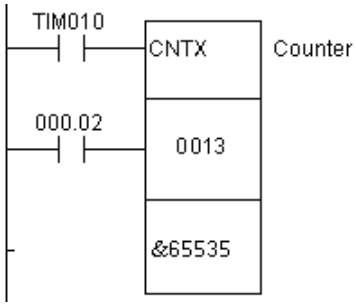
Eksempel:

RUNG # 1



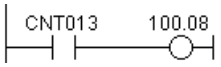
Verdien her er 60,0 sek dvs 1 minutt.

RUNG # 2



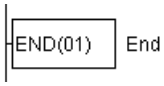
Verdien her tilsvarer 1092 timer.

RUNG # 3



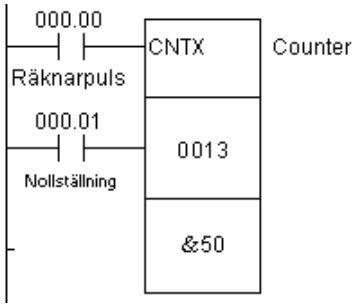
Utgangen blir aktivert når telleren har regnet ned til null.

RUNG # 4



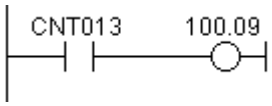
8.12 Eksempel 3a Tellefunksjoner.

Når du vil telle hvor mange ganger ting har blitt utført, kan du benytte deg av en tellefunksjon.
Grunnfunksjon med max regneverdi 65535 ser ut som følger:

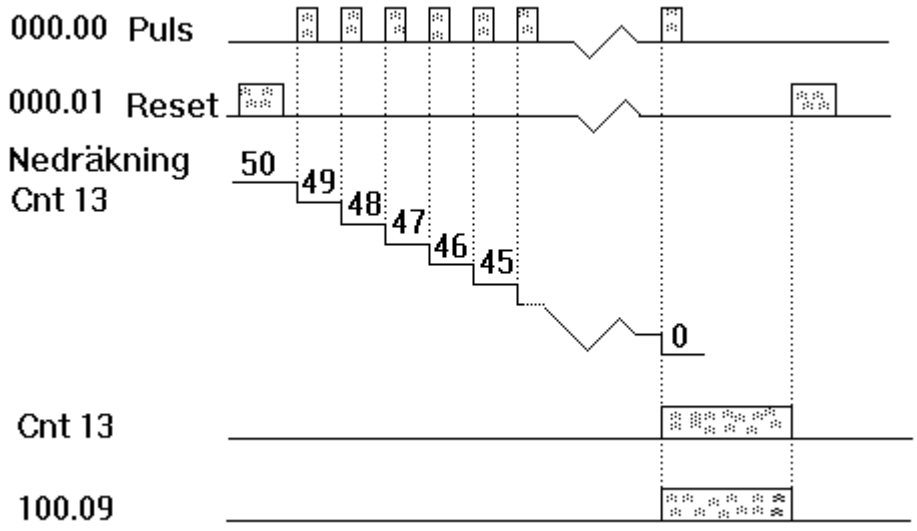


50 pulser

Når de innstilte pulsene er oppnådd aktiveres den kontakt som har adressen ”CNT 013” og utgang 100.09 aktiveres.



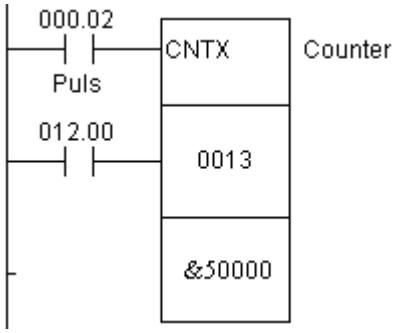
Tidsdiagram for kretsen:



8.13 Eksempel 3b Tellefunksjoner med store verdier.

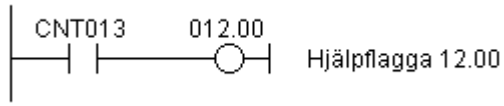
Når du vil regne flere pulser enn 65535 stk kan du koble sammen to eller flere tellere.
Teller nummer to kan da f.eks bety 50 000 tal.

RUNG # 1

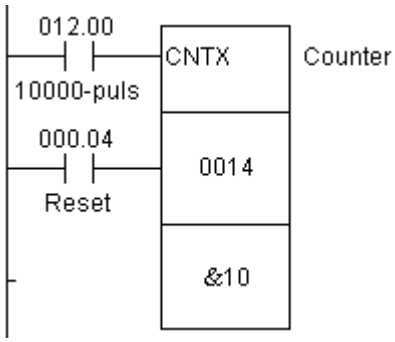


50000 pulser

RUNG # 2



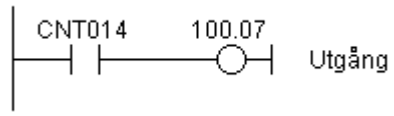
RUNG # 3



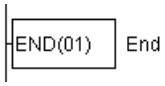
500 000 pulser

Resultatet når 500 000 pulser har blitt regnet er at utgang 100.07 blir aktiv.

RUNG # 4

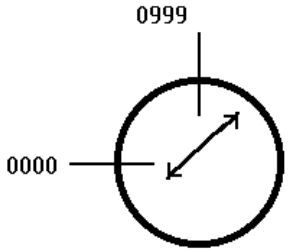


RUNG # 5



8.14 Eksempel 3c Reversibel teller (ringteller)

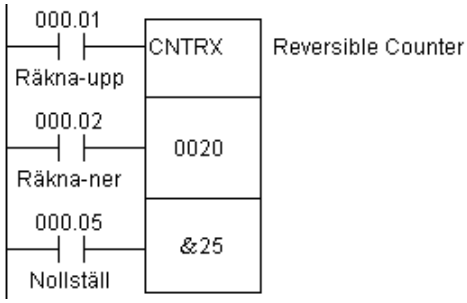
Det kan av og til være behov for en teller som kan telle både opp og ned med hjelp av ulike signaler.
Tellefunksjonen styres av to innsignaler og et resetsignal.
CNTXR kan til og med kalles en ringteller.



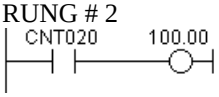
Når den innstilte verdien er oppnådd aktiveres utgangen. Hvis ytterligere pulser kommer starter telleren igjen og teller opp eller ned.

Eksempel.

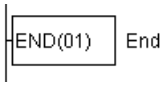
RUNG #1



RUNG # 2



RUNG # 3



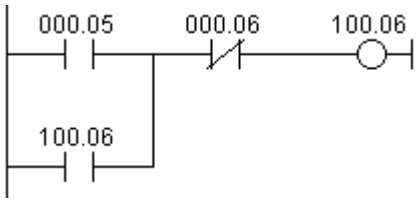
Signalet ”000.01” øker tellerens bufferminne med 1 og signalet ”000.02” minsker tellerens bufferminne med 1.
Når verdien overstiger med innstilt verdi aktiveres utgangen.
Telleren regner ikke under nullpunktet bortsett fra mellom 0 og innstilt verdi.

Signalene ”00.01” og ”00.02” gjenkjennes på den positive flanken av signalet og telleren venter deretter på et nytt positiv flankesignal.

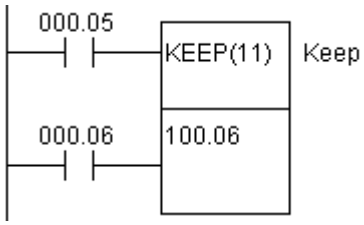
Utgangen 100.00 blir aktiv når tellerverdien overstiger innstilt verdi.

8.15 Eksempel 4a Holdekretser

I programmer kan det iblant behøves holdekretser av ulike slag.
Grunnkretsen kan se ut som følgende hvor holdekretsen lages ved hjelp av den ene utgangen.

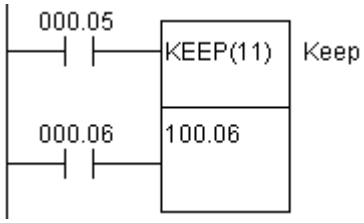


Dette kan forekomme ved hjelp av en ferdig funksjon i PLS-systemet.
Denne funksjonen heter ”KEEP” (11).

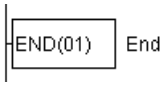


Eksempel:
Inngang 000.05 aktiverer utgangen 100.06 og inngangen 000.06 nullstiller utgang 100.06

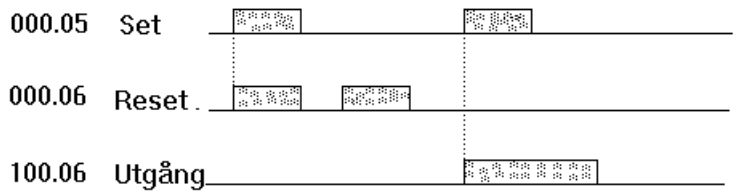
RUNG # 1



RUNG # 2



Tidsdiagram for kretsen:

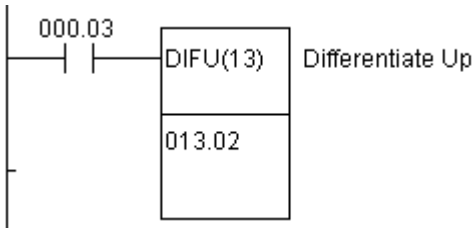


8.16 Eksempel 5a Pulsfunksjon.

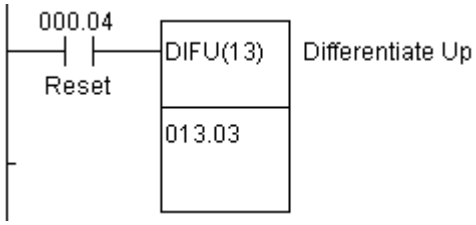
Av og til kan det være fint å kjenne av et signal som en puls selv om den ligger aktiv over lengre tid. Dette kan løses med funksjonene *DIFU(13) og DIFD(14).* Disse funksjonene kjenner av den positive eller negative flanken på et signal, og gir ifra seg en puls. Deretter er den null.

Eksempel: Vi kjenner av signalet 000.03 og lager pulsen 013.02 som aktiverer en KEEP funksjon.
Signalet 000.04 gir en puls via 13.03 som nuller KEEP funksjonen.

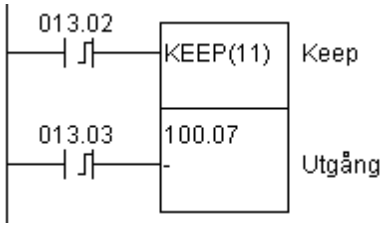
RUNG # 1



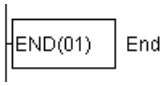
RUNG # 2



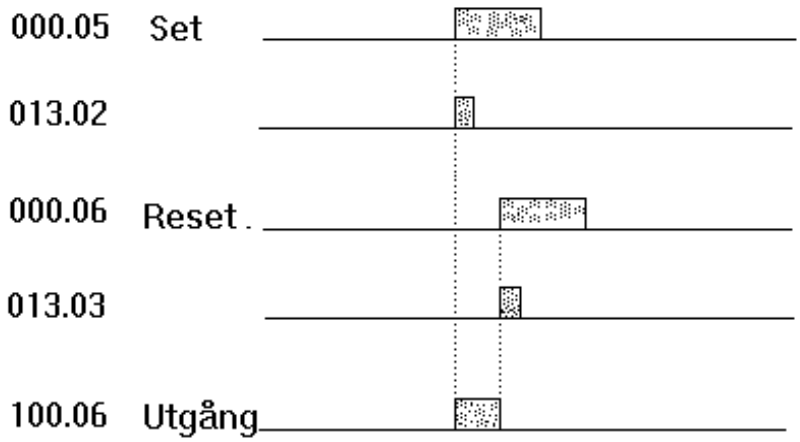
RUNG # 3



RUNG # 4



Tidsdiagram for kretsen:



Postadresse:
Postboks 109, Bryn
N-0611 Oslo

Besøksadresse:
Brynsalléen 4,
0611 OSLO

Telefon: 22 65 75 00
Fax: 22 65 83 00
Ordrefax: 22 65 83 01
E-mail: omron.norway@eu.omron.com

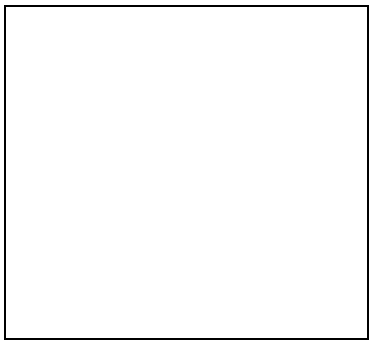
Bankgiro 7262.05.0578
Foretaksregisteret:
NO 852 345 632 MVA

8.17 Eksempel 6a Datakopiering

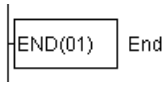
Av og til finns det et behov for å låse data og kopiere det til et annet sted i minneområde. Dette kan gjøres med funksjonen ”MOV”. PLS-systemet låser det som finnes i første dataminnet og kopierer over dette til det andre dataminnet.

Vi kopierer innholdet i første inngangsordet 000 til dataminne DM 000.

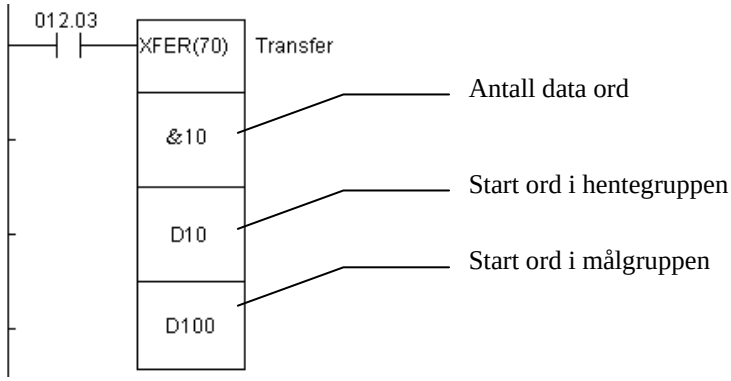
RUNG # 1



RUNG # 2



Hvis du vil kopiere en gruppe av dataminne kan du bruke funksjonen blocktransfer (XFER).
Denne funksjonen kopierer data i den definerte gruppen til destinasjons gruppen.

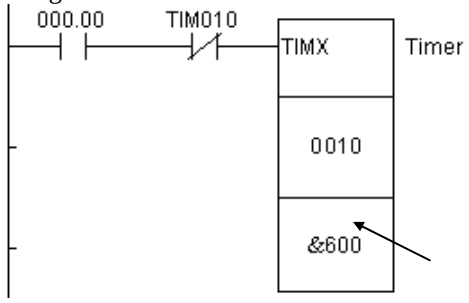


8.18 Eksempel 7a Driftsregning

Når du vil ha en kontroll på hvor lenge en maskin har vært i drift, kan du bruke følgende program hvor en tidsfunksjon og en teller brukes for å regne antall timer som maskinen har vært i drift.

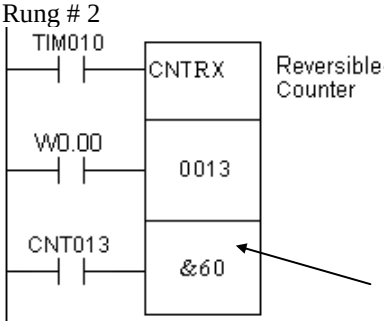
Eksempel:

Rung # 1



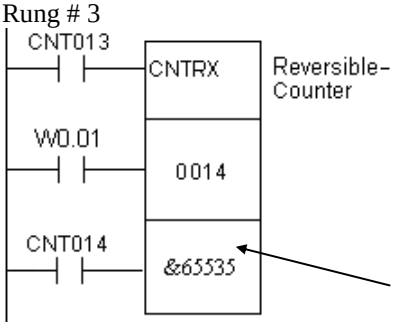
Verdien her tilsvarer 60 sekunder. Timeren setter altså T10 høy hvert minutt.

Rung # 2



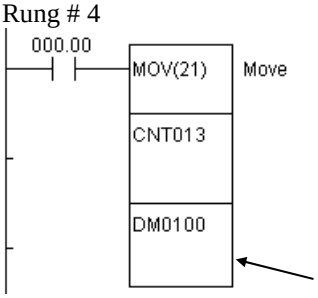
Verdien til teller 13 er antall minutter maskinen har kjørt. Når teller når når 60 (1 time) setter C13 høy.

Rung # 3



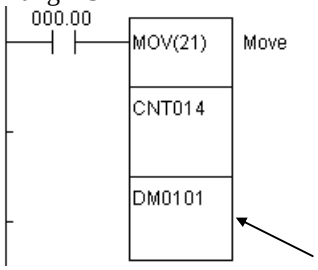
Verdien til teller 14 er antall timer maskinen har kjørt.

Rung # 4



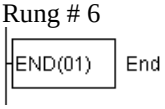
I dette minnet lagres antall minutter maskinen har kjørt.

Rung # 5



I dette minnet lagres antall timer maskinen har kjørt.

Rung # 6



Postadresse:
Postboks 109, Bryn
N-0611 Oslo

Besøksadresse:
Brynsalléen 4,
0611 OSLO

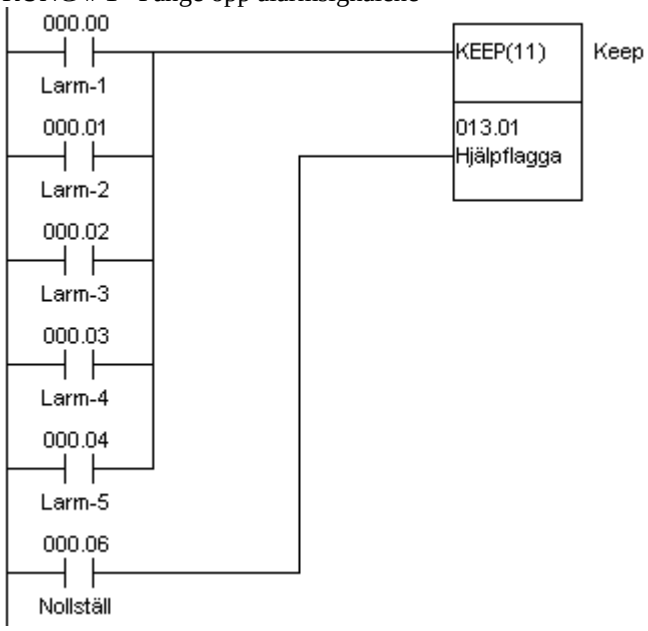
Telefon: 22 65 75 00
Fax: 22 65 83 00
Ordrefax: 22 65 83 01
E-mail: omron.norway@eu.omron.com

Bankgiro 7262.05.0578
Foretaksregisteret:
NO 852 345 632 MVA

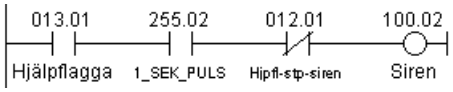
8.19 Eksempel 8a Alarmhåndtering

I mange prosjekter trenger man å fange opp alarmer og påkalle omgivelsenes oppmerksomhet.
Dette kan gjøres ved hjelp av noen funksjoner og signaler. Følgende program er et eksempel på dette.

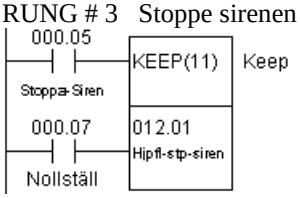
RUNG # 1 Fange opp alarmsignalene



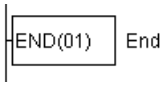
RUNG # 2 Starte sirenen



RUNG # 3 Stoppe sirenen



RUNG # 4 Slutt på programmet



Når noen av alarmene er aktive, eller sender en puls, fanger funksjonen ”Keep” opp signalet.
Denne funksjonen aktiverer deretter sirenen. For å stoppe sirenen aktiveres inngang 000.05.
Når alarmen nullstilles og det er en alarm igjen, aktiveres sirenen igjen.

9 Oppgaver

Her er noen oppgaver det går an å prøve seg på for å få litt trening. Løsningsforslag er vist i appendix E. Skriv I/O liste og ladderdiagram. Test ut programmet.

9.1 Oppgave 1

Start og stopp av motor. En vender bestemmer om en motor skal ha drift eller ei.

Materiell: 1 stk vender av/på.
1 stk lampe til indikering av drift.
1 stk kontaktor for motor.

9.2 Oppgave 2

Start og stopp av motor. To impulsbrytere (en grønn og en rød) brukes i en holdekrets for å starte (grønn) og stoppe (rød) en motor.

Materiell: 2 stk impulsbrytere, 1 start og 1 stopp.
1 stk lampe til indikering av drift.
1 stk kontaktor for motor.

9.3 Oppgave 3

Start og stopp av motor. Impuls start – impuls stopp funksjonalitet (hint: DIFU).

Materiell: 1 stk impulsbrytere, samme bryter gir start og stopp.
1 stk lampe til indikering av drift.
1 stk kontaktor for motor.

9.4 Oppgave 4

Start og stopp av motor samt indikering på motorveien utløst. Start/stopp som tidligere, men nå skal motorutgang også stoppe når motorvern blir utløst.

Materiell: 2 stk impulsbrytere, 1 start og 1 stopp.
1 stk lampe til indikering av drift.
1 stk lampe til indikering av motorvern utløst.
1 stk kontaktor for motor.

9.5 Oppgave 5

Start og stopp av to motorer. De skal ikke kunne gå samtidig. To separate motorer som er forriglet mot hverandre.

Materiell: 4 stk impulsbrytere, 2 start og 2 stopp.
2 stk lampe til indikering av drift.
2 stk kontaktor for motor.

9.6 Oppgave 6

Start og stopp av to motorer, M1 og M2. M1 skal starte med en gang startsignal blir gitt. M2 skal starte 10 sek. etter M1. Begge skal stoppe samtidig.

Materiell: 1 stk vender av/på.
2 stk lampe til indikering av drift.
2 stk kontaktor for motor.

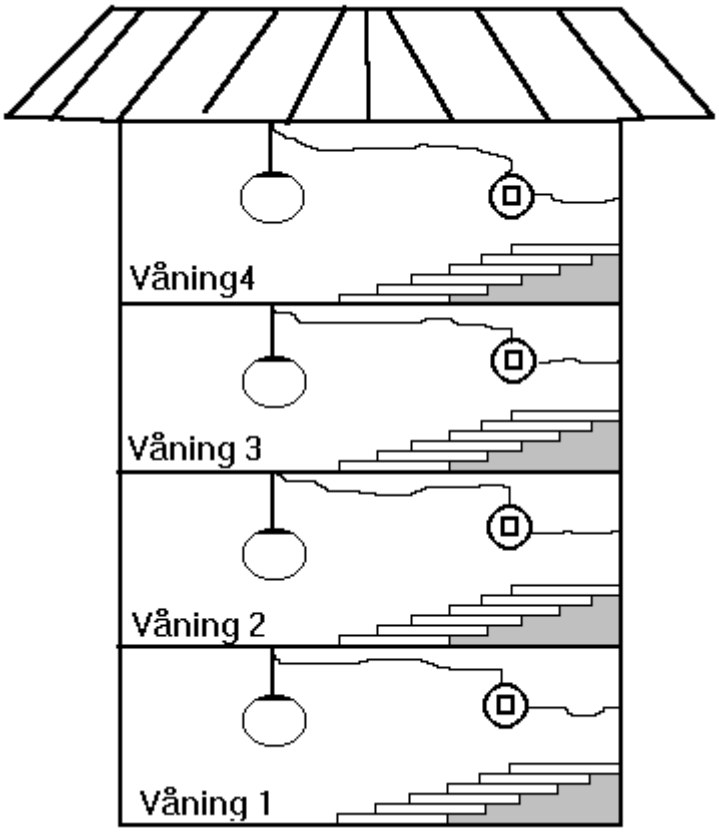
9.7 Oppgave 7

Start og stopp av to motorer, M1 og M2. M1. M1 og M2 skal gå annenhver gang. Hint: Lag en standard motorstyring og deretter bestem hvilken motor som skal startes.

Materiell: 2 stk impulsbrytere, 1 start og 1 stopp.
2 stk lampe til indikering av drift.
2 stk kontaktor for motor.

9.8 Oppgave 8 Trappebelysning

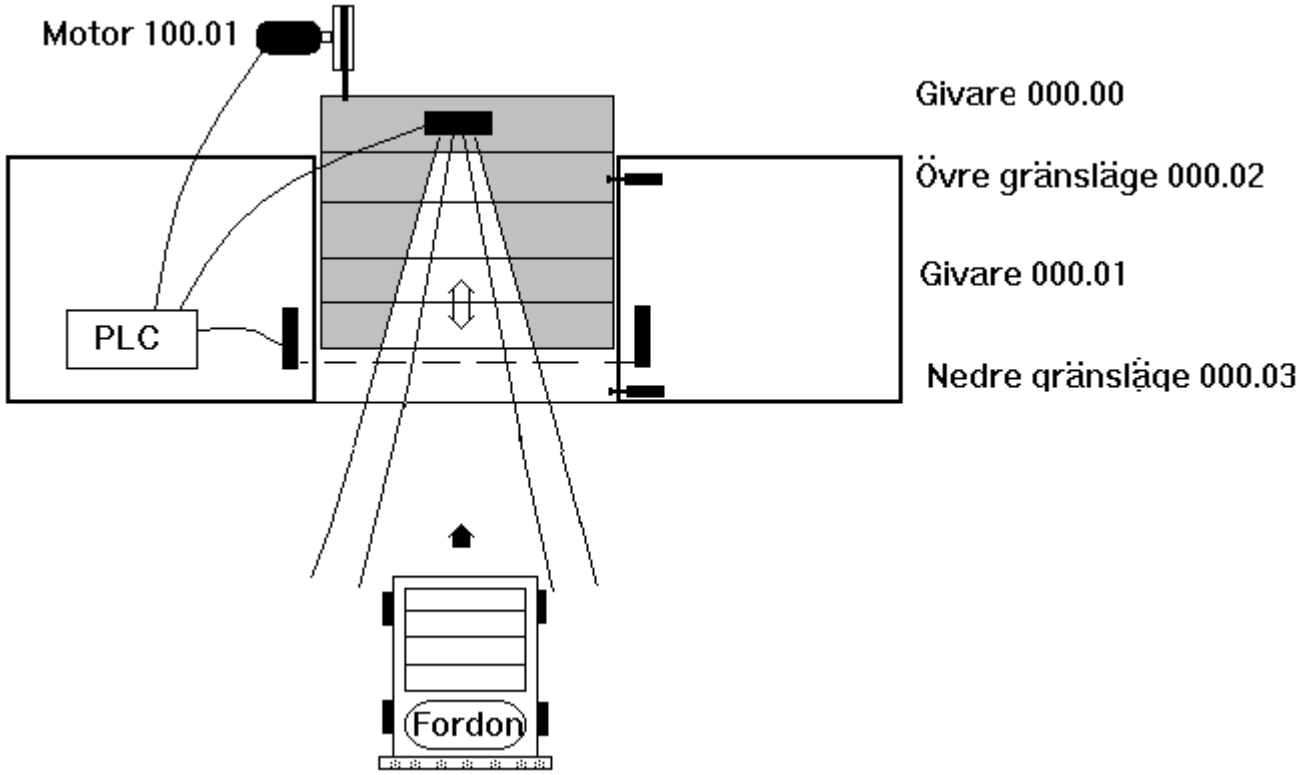
En trappebelysning i et hus skal automatiseres. Det skal være mulig og aktivere lyset fra hver etasje. Det er fire etasjer i huset. Lyset skal være tent i et halvt minutt og deretter slokke automatisk.



Innganger	*Adresser*
Bryter i etasje nr 1	000.01
Bryter i etasje nr 2	000.02
Bryter i etasje nr 3	000.03
Bryter i etasje nr 4	000.04
Utganger	
Belysning i etasje 1	100.01
Belysning i etasje 2	100.02
Belysning i etasje 3	100.03
Belysning i etasje 4	100.04
Tidsfunktion nr 1	Tid 30 sekunder

9.9 Oppgave 9 Automatisk portstyring

I dette programmet skal en port åpnes når et kjøretøy nærmer seg og stenges når kjøretøyet forlater porten. Til dette har du et PLS-system og et antall givere.



Beskrivelse

Systemet har fire enheter som sender signaler til PLS-systemet. En ultralyd detektor, en fotocelle og i porten finnes det to grenseområder. En motor åpner og stenger porten.
Ultralyd detektoren avgir ultralydbølger som, så snart det oppdager et kjøretøy i anslutning til porten, reflekteres tilbake til detektoren og registreres som signal til PLS-systemet.
Fotocellen består her av en sender og en mottaker. Senderen avgir en lysstråle som hele tiden registreres i mottageren. Hvis et kjøretøy bryter strålen registreres dette i mottakeren og den sender signaler til PLS-systemet.
Med hjelp av programmet i og de signaler som sendes til PLS-systemet styres porten til å åpne og stenge. Begge endebryterne leses av for å kunne stoppe porten i øvre eller nedre posisjon.

Følgende signaler finnes i eksemplet.

Inngang	*Adresse*
Ultralyd detektor	000.00
Fotocelle	000.01
Øvre grense	000.02
Nedre grense	000.03
Utganger	
Åpne dør	100.00
Stenge dør	100.01

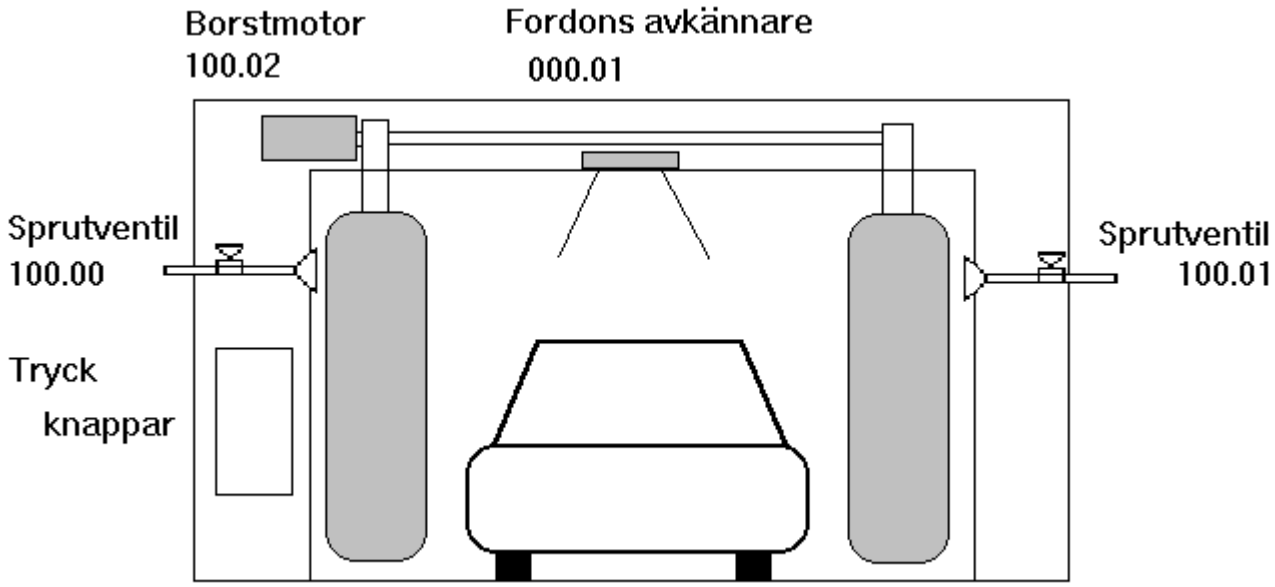
Funksjon

Når et kjøretøy kommer inntil porten avleses dette med ultralyd detektoren og PLS-systemet sender et signal til motoren om å åpne porten. Når porten har kommet til sin øvre grense stoppes utsignalet til portens motor.
Når kjøretøyet kjører inn og passerer fotocellen, brytes lysstrålen og et signal sendes til PLS-systemet. Når kjøretøyet passerer fotocellen, registreres lysstrålen igjen av mottakeren. Disse signalene taes hånd om i PLS-systemets program og starter stengningen av porten når kjøretøyet passerer fotocellen. Hvis lysstrålen fra fotocellen skulle brytes når porten holder på og stenges, stopper porten.

Postadresse: Postboks 109, Bryn N-0611 Oslo
Besøksadresse: Brynsalléen 4, 0611 OSLO
Telefon: 22 65 75 00 Fax: 22 65 83 00 Ordrefax: 22 65 83 01 E-mail: omron.norway@eu.omron.com
Bankgiro 7262.05.0578 Foretaksregisteret: NO 852 345 632 MVA

9.10 Oppgave 10 *Bilvask*

I dette programmet skal du automatisere en bilvask.



Beskrivelse

PLS-systemets innsignaler kommer fra trykknappene i manøvreringsboksen og fra kjøretøysregistrereren.
Ved hjelp av disse signaler åpnes en sprutventil og startes en motor for de roterende børstene.
Vasken forflytter seg etter hele kjøretøyet lengde under vasken. Når vasken pågår lyser en lampe.

Følgende signaler finnes i eksemplet.

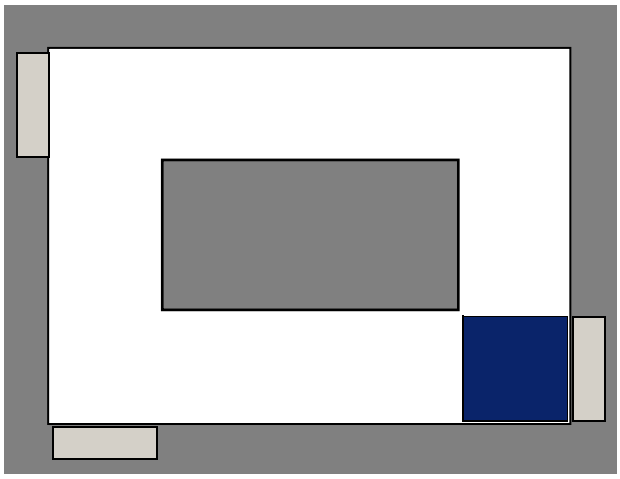
Innganger	*Adresser*
Start	000.00
Stopp	000.01
Kjøretøysregistrerer	000.02
Forerste ende	000.03
Bakre ende	000.04
Utganger	
Sprutventil-1	100.00
Sprutventil-2	100.01
Børstemotor	100.02
Forflytting av vask bakover	100.03
Forflytting av vask fremover	100.04
Indikering av pågående vask	100.05

Funksjon

Vasken settes igang når startknappen aktiveres. Børstene startes og sprutmunnstykkene åpnes av at kjøretøysregistrereren ” 000.01” sender et signal til PLS-systemet. Det er en liten tidsforsinkelse før start og ved vanningen (tim1 og tim2). Vasken fortsetter langsmed kjøretøyet så lenge som kjøretøyregistratoren er aktiv.
Når vasken kommer til bakre grense stopper børstene og vannspruten, og vasken vender til den andre siden etter en bestemt tid.
Sekvensen gjenntas bare en gang. Vasken avsluttes og alle funksjonene stenges av når den fremre grense igjen er aktivt. For å vaske en gang til kreves det at startsignalet ” 000.00 aktiveres.

9.11 Oppgave 11 *Evighetsmaskin...*

Den kvadratiske (mørke blå) klossen skal skyves med klokka rundt og rundt. 4 sylindre brukes til dette. Hver sylinder har endrebryter i begge endene. Mekanikken er stående, slik at klossen vil kunne falle hvis den ikke har noe som holder den oppe.
Klossen skal gå rundt og rundt helt til maskinen får beskjed om å stoppe, og klossen skal da stoppe der den startet.



Postadresse:
Postboks 109, Bryn
N-0611 Oslo

Besøksadresse:
Brynsalléen 4,
0611 OSLO

Telefon: 22 65 75 00
Fax: 22 65 83 00
Ordrefax: 22 65 83 01
E-mail: omron.norway@eu.omron.com

Bankgiro 7262.05.0578
Foretaksregisteret:
NO 852 345 632 MVA

10 Appendix

10.1 Appendix A: Minneoversikt CP1L

Data område		ord/kanaler	Bit	Beskrivelse
CIO område	I/O område	0000 til 0999	0000.00 til 0999.15	Tilordnet fysisk I/O Inn CIO 0: 0.00 til 0.11 Inn CIO 1: 1.00 til 1.11 Ut CIO 100: 100.00 til 100.07 Ut CIO 101: 101.00 til 101.03 Ekspansjon 1 2.00 til 2.11 (Inn) 3.00 til 3.11 (Inn) Ekspansjon 2 4.00 til 4.11 (Inn) 5.00 til 5.11 (Inn) Ekspansjon 3 6.00 til 6.11 (Inn) 7.00 til 7.11 (Inn) Ekspansjon 1 102.00 til 102.07 (Ut) 103.00 til 103.107 (Ut) Ekspansjon 2 104.00 til 104.07 (Ut) 105.00 til 105.07 (Ut) Ekspansjon 3 106.00 til 106.07 (Ut) 107.00 til 107.07 (Ut) Analogkort: Hvert kort allokerer et gitt antall ord (2 til 4), se manual W462 avsnitt 4-2-4 for mer informasjon
	1:1 Link Area	3000 til 3063		Brukes mellom Omron PLS'er Master/slave CP1L <-> CPM2
	Serial PLC Link Area	3100 til 3189		Data Link mellom CP1L og CP1H CPU'er
	Work Area	3800 til 6143		Ikke reservert – til fritt bruk
CF område			CF000 til CF011 CF100 til CF104 CF113 til CF114	Spesial bit
TR område		TR0	TR0.00 til TR0.15	Brukes av CX-P ved forgreininger i ladder
W område		W000 til W511	W000.00 til W511.15	Til fritt bruk (ingen batteribackup)
H område		H000 til H511	H000.00 til H511.15	Til fritt bruk (batteribackup)
A område	Read Only	A000 til A447	A000.00 til A447.15	For monitorering og kontroll av PLSen
	Read/Write	A448 til A959	A448.00 til A959.15	
T område		T0000 til T4095		Timere
C område		C0000 til C4095		Tellere
TK område			TK00 til TK31	Task flagg
IR område		IR00 til IR15		Brukes til indirekte CIO adressering
DR område		DR00 til DR15		
D område	Not Used	D00000 til D19999 D29600 til D29999 D31600 til D32767		Til fritt bruk (batteribackup)

10.2 Appendix B: Minneoversikt CJ1

Data område		**ord/kanaler**	**Bit**	**Funksjon**
CIO område	I/O område	0000 til 0999	0000.00 til 0999.15	Tilordnet I/O moduler
	Data Link Area	1000 til 1199	1000.00 til 1199.15	Tilordnet Controller Link
	Internal I/O Area	1200 til 1499 3800 til 6143	1200.00 til 1499.15 3800.00 til 6143.15	Ikke reservert – til fritt bruk Kan bli reservert senere
	CPU Bus Unit Area	1500 til 1899	1500.00 til 1899.15	Reservert til CPU Bus moduler
	Special Unit Area	2000 til 2959	2000.00 til 2959.15	Reservert til SIO moduler
	DeviceNet Area	3200 til 3799	3200.00 til 3799.15	Reservert til DeviceNet
	Not Used	1900 til 1999 2960 til 3199	1900.00 til 1999.15 2960.00 til 3199.15	Ikke reservert – til fritt bruk Kan bli reservert senere
CF område			CF000 til CF011 CF100 til CF104 CF113 til CF114	Spesial bit
TR område		TR0	TR0.00 til TR0.15	Brukes av CX-P ved forgreininger i ladder
W område		W000 til W511	W000.00 til W511.15	Til fritt bruk
H område		H000 til H511	H000.00 til H511.15	Til fritt bruk
A område	Read Only	A000 til A447	A000.00 til A447.15	For monitorering og kontroll av PLSen
	Read/Write	A448 til A959	A448.00 til A959.15	
T område		T0000 til T4095		Timere
C område		C0000 til C4095		Tellere
TK område			TK00 til TK31	Task flagg
IR område		IR00 til IR15		Brukes til indirekte CIO adressering
DR område		DR00 til DR15		
D område	Not Used	D00000 til D19999 D29600 til D29999 D31600 til D32767		Til fritt bruk
	Special Unit Area	D20000 til D29599		Reservert til SIO moduler
	CPU Bus Unit Area	D30000 til D31599		Reservert til CPU Bus moduler
E område		E00000 til E32767 E0_00000 til E2_32767		Til fritt bruk Kan brukes som FM område
TM – Trace minne		6	31	4000 ord til logging
FM – Fil minne		Avhenger av Minnekort størrelse		MS-DOS format

10.3 Appendix C: Minneoversikt CQM1H

Data område		**ord/kanaler**	**Bit**	**Funksjon**
IR område interne relé	inngangs område	IR 000 til IR 015	IR 000.00 til IR 015.15	Disse bitene kan allokeres til innganger
	utgangs område	IR 100 til IR 115	IR 100.00 til IR 115.15	Disse bitene kan allokeres til utganger
	arbeids område	IR 016 til IR 089 IR 116 til IR 189 IR 216 til IR 219 IR 224 til IR 229	IR 016.00 til IR 089.15 IR 116.00 til IR 189.15 IR 216.00 til IR 219.15 IR 224.00 til IR 229.15	Arbeids område brukt fritt i programmet
SR område spesial relé		SR 244 til SR 255	IR 244.00 til IR 255.07	bit med spesielle funksjoner. Eks. flagg og kontroll bit
TR område tilfeldige relé			TR 0 til TR 7	Brukt til midlertidig lagring i logikk.(listeform).
HR område holde relé		HR 00 til HR 99	HR 00.00 til HR 99.15	Disse adressene bevarer verdi ved strømbrudd
AR område ekstra relé		AR 00 til AR 27	AR 00.00 til AR 27.15	bit med spesielle funksjoner. Eks. flagg og kontroll bit
LR område link relé		LR 00 til LR 63	LR 00.00 til LR 63.15	Brukt til 1:1 data link til annen PLS
TC område Timer/Counter		TC 000 til TC 511 (Timer/ teller nummer)		Timere og tellere bruker samme område
DM område data register	«Read/write»	DM 0000 til DM 3071 DM 3072 til DM 6143		DM område kan ikke behandles i bit Kun for CQM1H-CPU51/61
	Feil log	DM 6569 til DM 6599		Brukes hvis feil-log funksjon er aktiv
	«Read only»	DM 6144 til DM 6568		Kan ikke bli skrevet til av programmet
	PLS setup	DM 6600 til DM 6655		parameter til PLS setup

I tillegg benyttes andre områder for spesialkortene som kan monteres i CPUen. Det finnes også et EM område på 6144 words i CQM1H-CPU61.

10.4 Appendix D: Installasjon av USB-driver for CP1L og CP1H

For å få kontakt med PLS'en første gang over USB må vi installere USB-driveren i Windows. Her er en rask gjennomgang:

- **1**. Plugg inn USB-kabelen i både PC og PLS og sjekk at PLS'en er påsatt spenning.
- **2**. Windows vil nå komme opp med en melding om at den har funnet ny maskinvare, OMRON PLC.
- **3**. Når "New Hardware Wizard" kommer opp velger du "Install from a list or a spesific location (Advanced)" og trykker "Next":

- **4**. Sjekk så at "Include this location in search" boksen er huket av, og pass på at [C:\Program Files\Omron\CX-Server\USB\Win2000_XP\Inf] er satt som mål. Trykk så "Next"
- **5**. Det kan nå hende Windows kommer opp med en melding om at driveren for OMRON PLC ikke er "Digitally signed", men her trykker du bare "continue anyway". Dette er bare en standardmelding om at Microsoft kjenner til driveren og derfor ikke vil gå god for den.
- **6**. Ferdig!

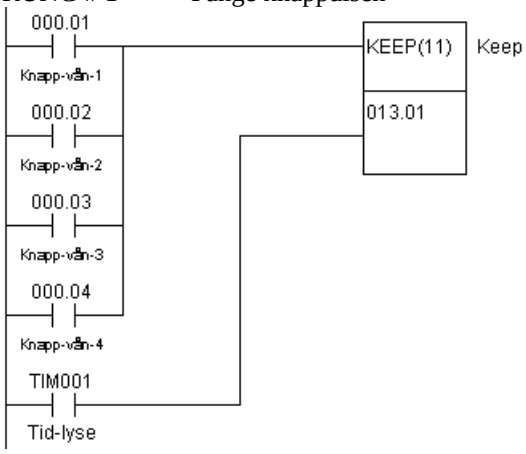
10.5 Appendix E: Løsningsforslag

10.5.1 Oppgave 1-7

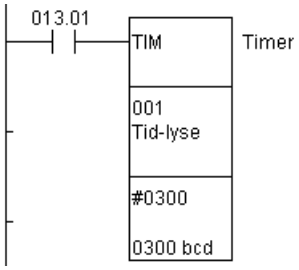
Se eget PLS program.

10.5.2 Oppgave 8

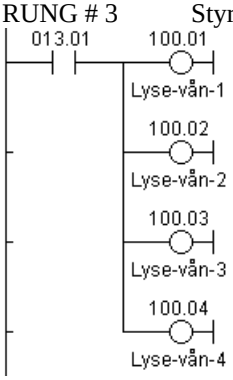
RUNG # 1 Fange knappulsen



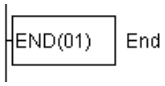
RUNG # 2 Tiden for tent lampe



RUNG # 3 Styring av belysningen

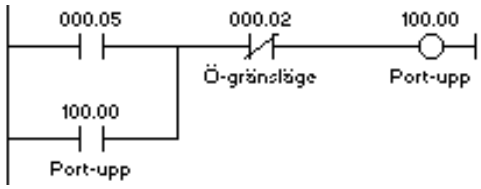


RUNG # 4 Slutt på programmet



10.5.3 Oppgave 9

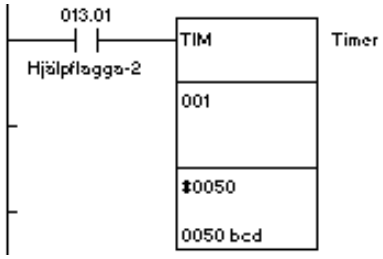
RUNG # 1



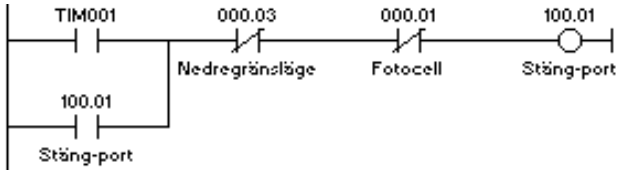
RUNG # 2

RUNG # 3

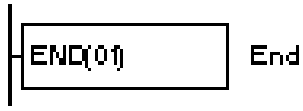
RUNG # 4



RUNG # 5

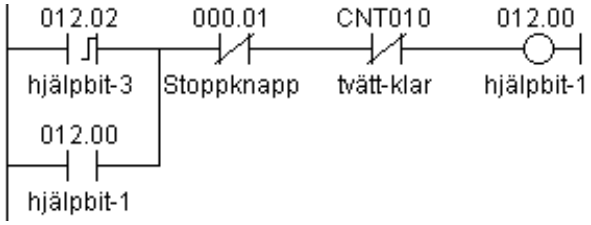


RUNG # 6



10.5.4 Oppgave 10

RUNG # 1



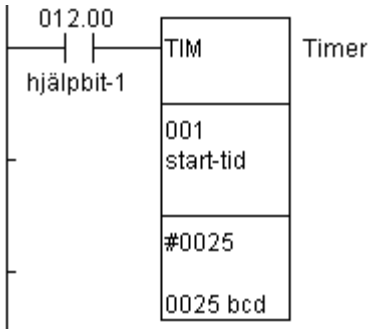
RUNG # 2

Postadresse:
Postboks 109, Bryn
N-0611 Oslo

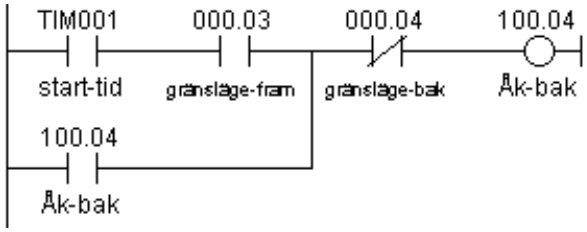
Besøksadresse:
Brynsalléen 4,
0611 OSLO

Telefon: 22 65 75 00
Fax: 22 65 83 00
Ordrefax: 22 65 83 01
E-mail: omron.norway@eu.omron.com

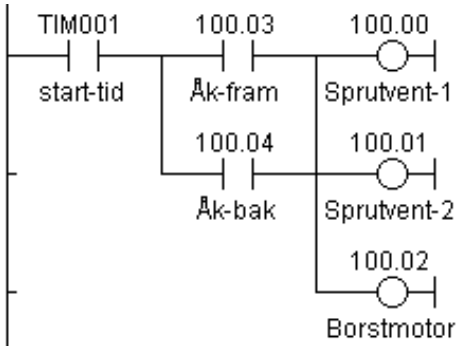
Bankgiro 7262.05.0578
Foretaksregisteret:
NO 852 345 632 MVA



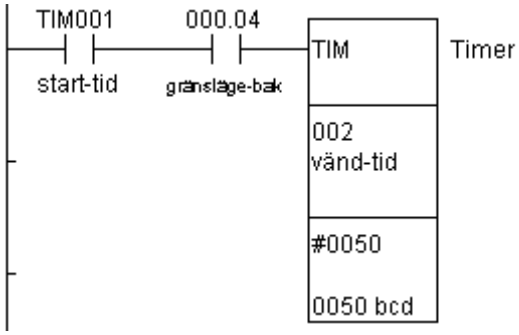
RUNG # 3



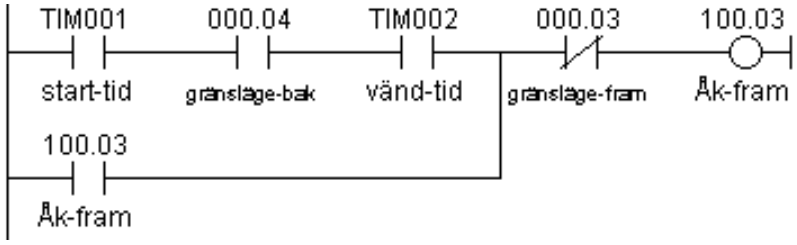
RUNG # 4



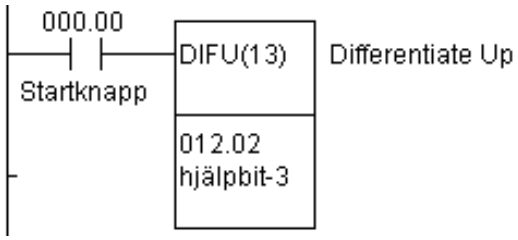
RUNG # 5



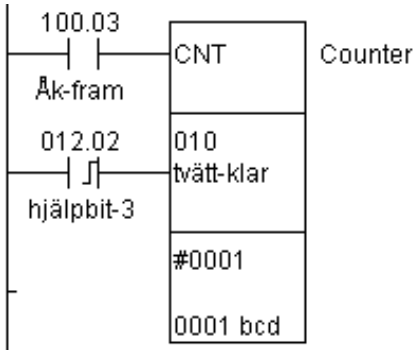
RUNG # 6



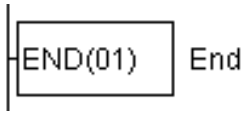
RUNG # 7



RUNG # 8



RUNG # 9



Postadresse:
Postboks 109, Bryn
N-0611 Oslo

Besøksadresse:
Brynsalléen 4,
0611 OSLO

Telefon: 22 65 75 00
Fax: 22 65 83 00
Ordrefax: 22 65 83 01
E-mail: omron.norway@eu.omron.com

Bankgiro 7262.05.0578
Foretaksregisteret:
NO 852 345 632 MVA

10.5.5 Oppgave 11

Sylinder 1 skubber klossen fra høyre mot venstre nede. Sylinder 2 løfter/senker klossen på høyre side. Sylinder 3 løfter/senker klossen på venstre side. Sylinder 4 skubber klossen fra venstre mot høyre oppe.