

PLC Programmering i CoDeSys



Grundläggande beskrivning

System: ABB, CPU ECO PM5xx

Program: Automation Builder 2.0

CoDeSys: 2.3.9.49

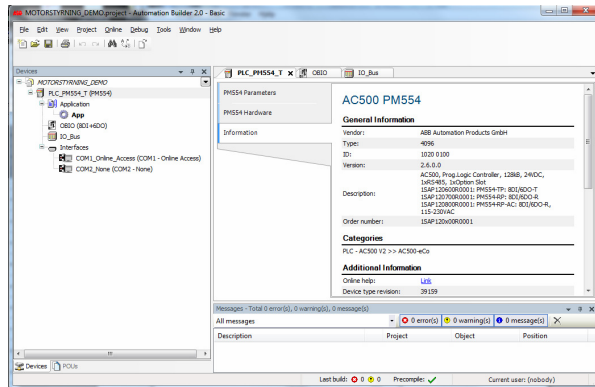
Följande beskrivs

- Skapa projekt
- Hårdvarukonfigurering
- Portinställning
- Skapa POU
- Deklarera variabler (lokala och globala)
- Programmering i IL, LD, FBD, ST och CFC
- Kompilering
- Nedladdning
- Boot project

Automation Builder (AB)

Programmet AB konfigurerar man sina enheter som PLC, paneler etc. Programmet är gratis (Basic) och kan laddas ned från ABB hemsida,
<http://new.abb.com/plc/automationbuilder/platform/software>

Vid installation av AB ingår programmet för CoDeSys.

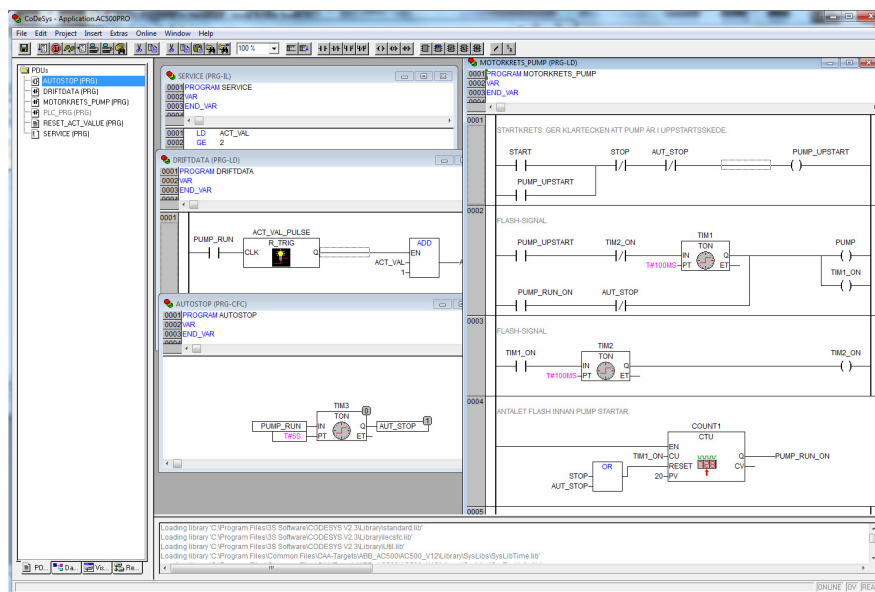


CoDeSys (CDS)

CoDeSys är ett program för att programmera PLC, HMI, nätverksenheter etc. Programmet använder standardiserat programmeringsspråk. Det innebär att man programmerar på samma sätt oberoende av fabrikat och tillverkare av produkterna. CoDeSys går att programmera i de flesta fabrikat som

- ABB
- Siemens
- Beckhoff
- Mitsubishi
- Omron
- Wago

m.fl.



Adressering

Adressering av variabler sker alltid via det deklarerade namnet (name). De adresser som man deklarerar med absoluta adresser är ingångar och utgångar. Det gör man i AB under fliken OBIO Mapping. Alla andra variabler deklarerar med namn och type. Det innebär att man inte ger någon specifik adress utan systemet håller reda på adresserna. Det som avgör hur många variabler en CPU kan deklarera beror på dess minne.

CPU

I detta exempel används en CPU PM554-T

Den har 8DI och 6DO

Minne 128kB

Kommunikationen sker seriell.



Exempel på projekt

Så här ska de färdiga blocken i projektet se ut. Vi kommer att gå igenom hur man skapar projektet i följande exempel. Följande begrepp tillämpas i texten:

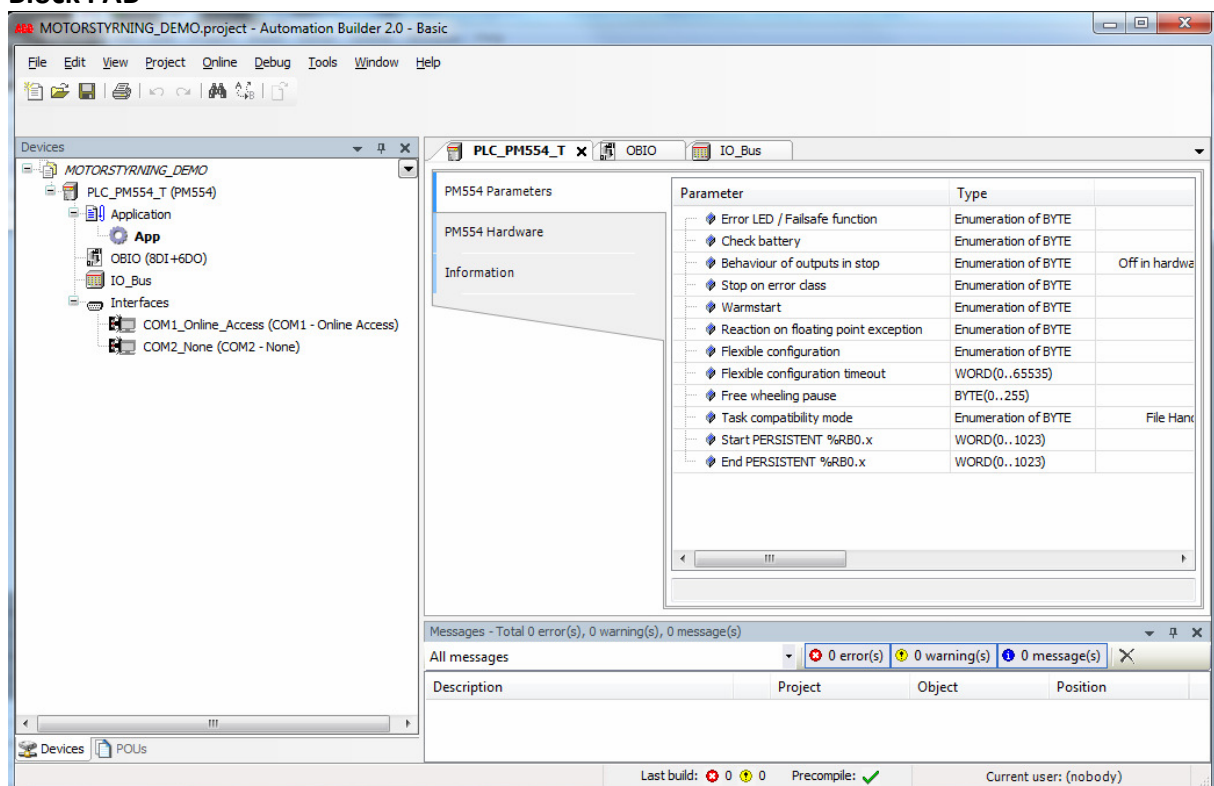
AB = Automation Builder

CDS = CoDeSys

Funktionsbeskrivning

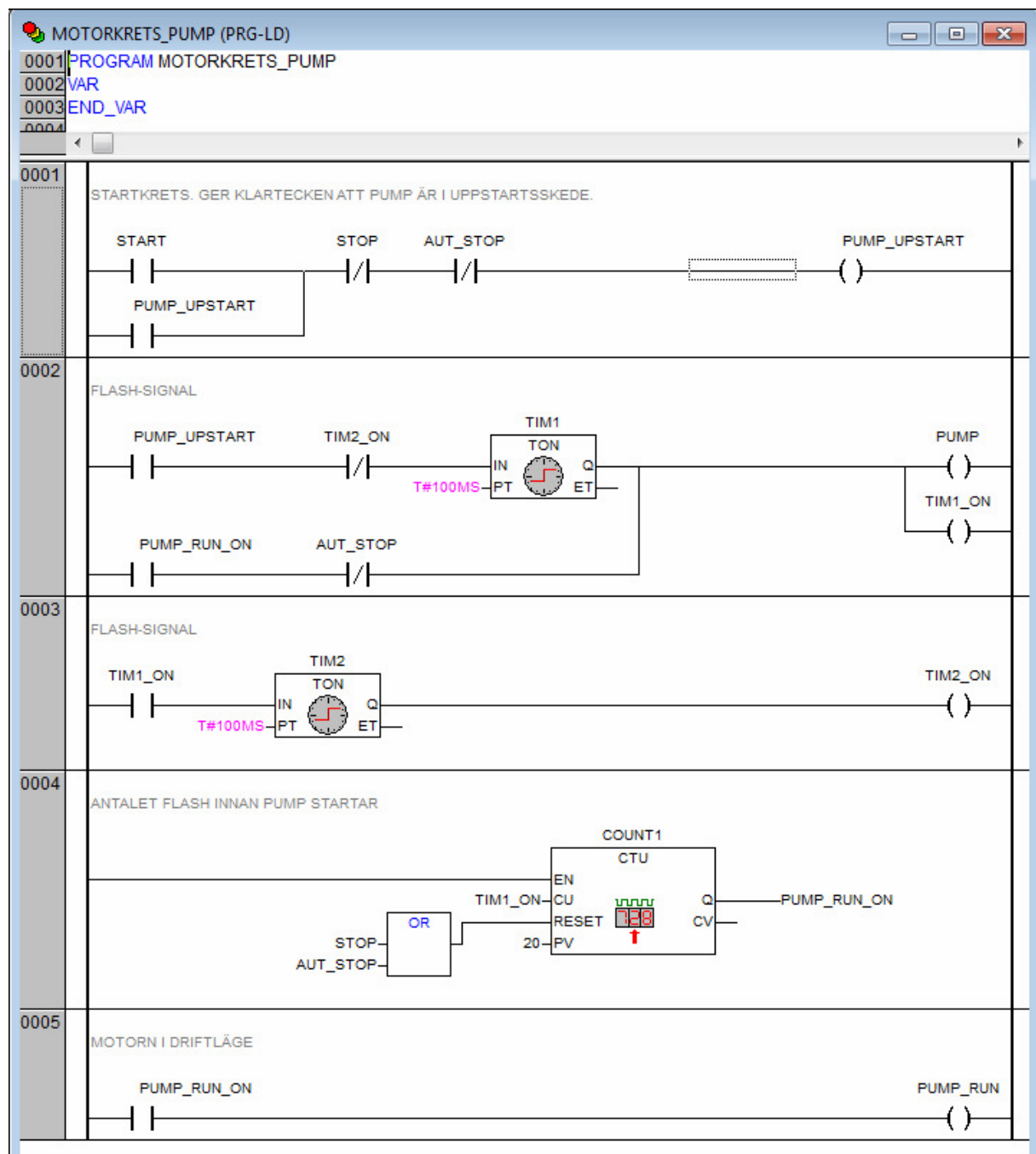
En motor till en pump ska startas. När man ger startsignal ska en signal med blinkande ljus signalera att motorn snart startas. Efter ett antal blinkningar startar motorn. Antalet starter som motorn har startat ska registreras. Efter ett visst antal (>3) starter ska en ljussignal tändas som signalerar att det är dag för service av motorn. Man ska kunna återställa räknaren som registrerat antalet starter först då antalet starter för service är uppnått.

Block i AB

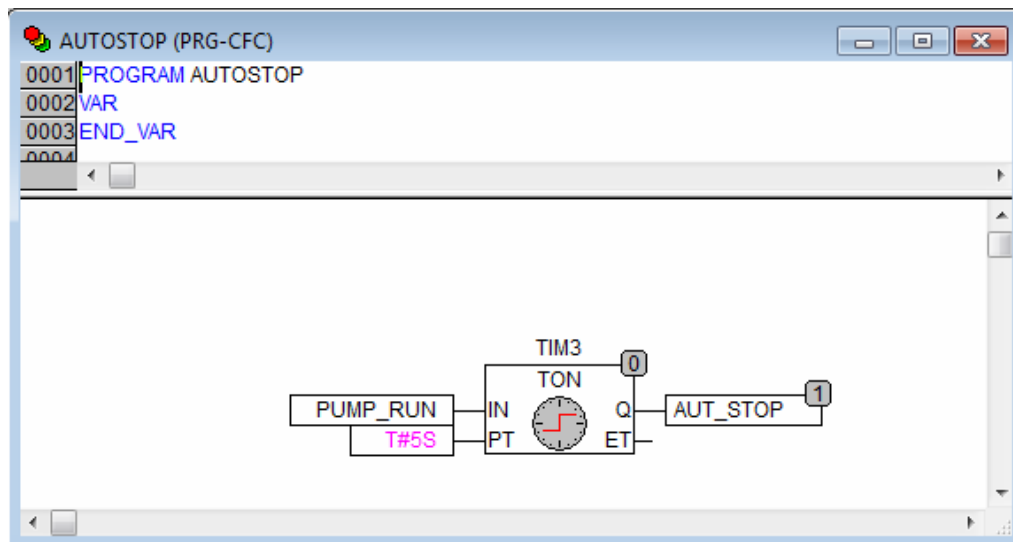


Projektblock

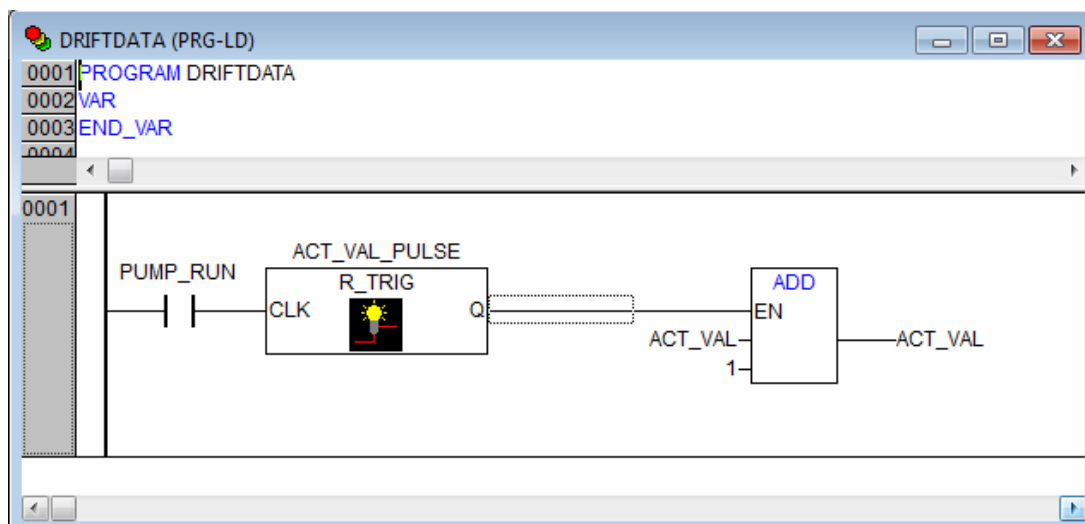
Block i CDS



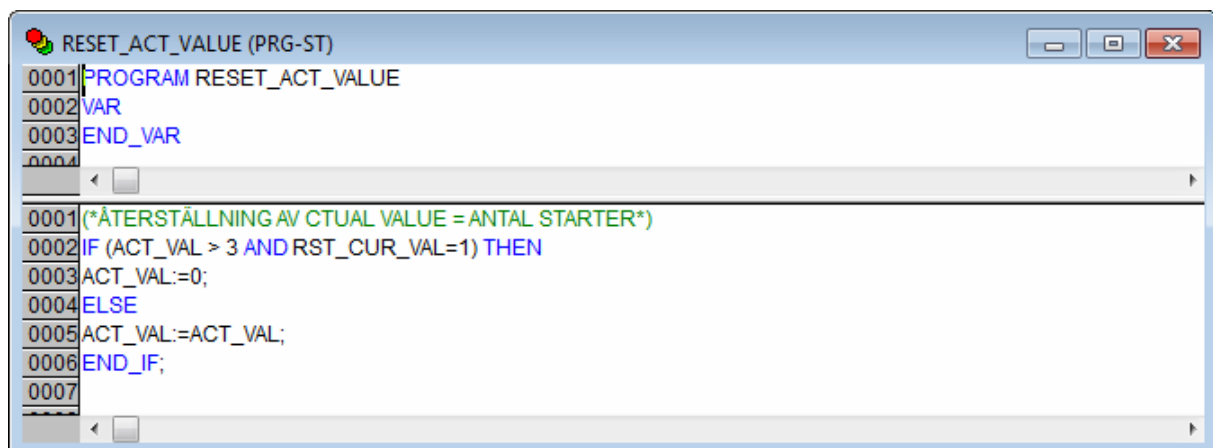
MOTORKRETS_PUMP (LD)



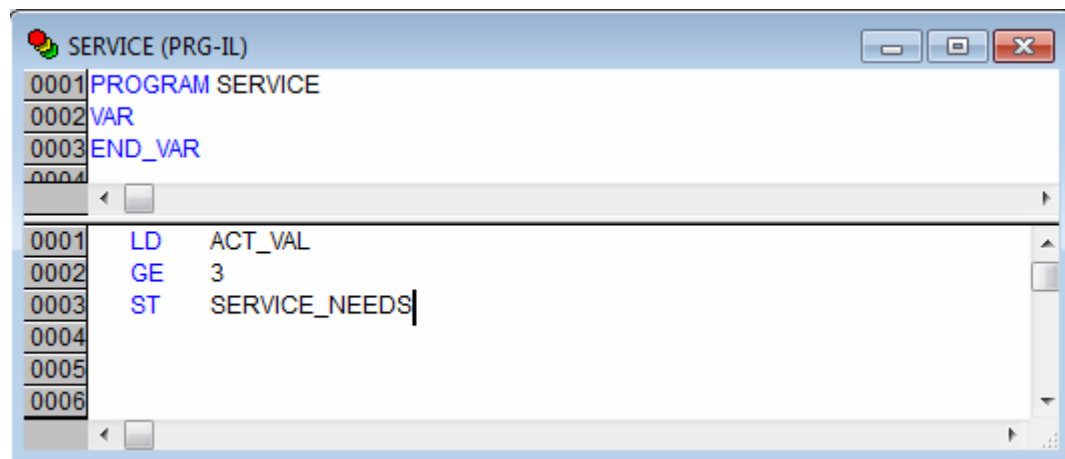
AUTOSTOP (CFC)



DRIFTDATA (LD)



RESET_ACT_VALUE (ST)



The screenshot shows a software window titled "SERVICE (PRG-IL)" with a standard Windows interface (minimize, maximize, close buttons). The window contains a ladder logic (LAD) editor. On the left, there is a vertical address bar with addresses 0001 through 0006. The main editing area shows the following code:

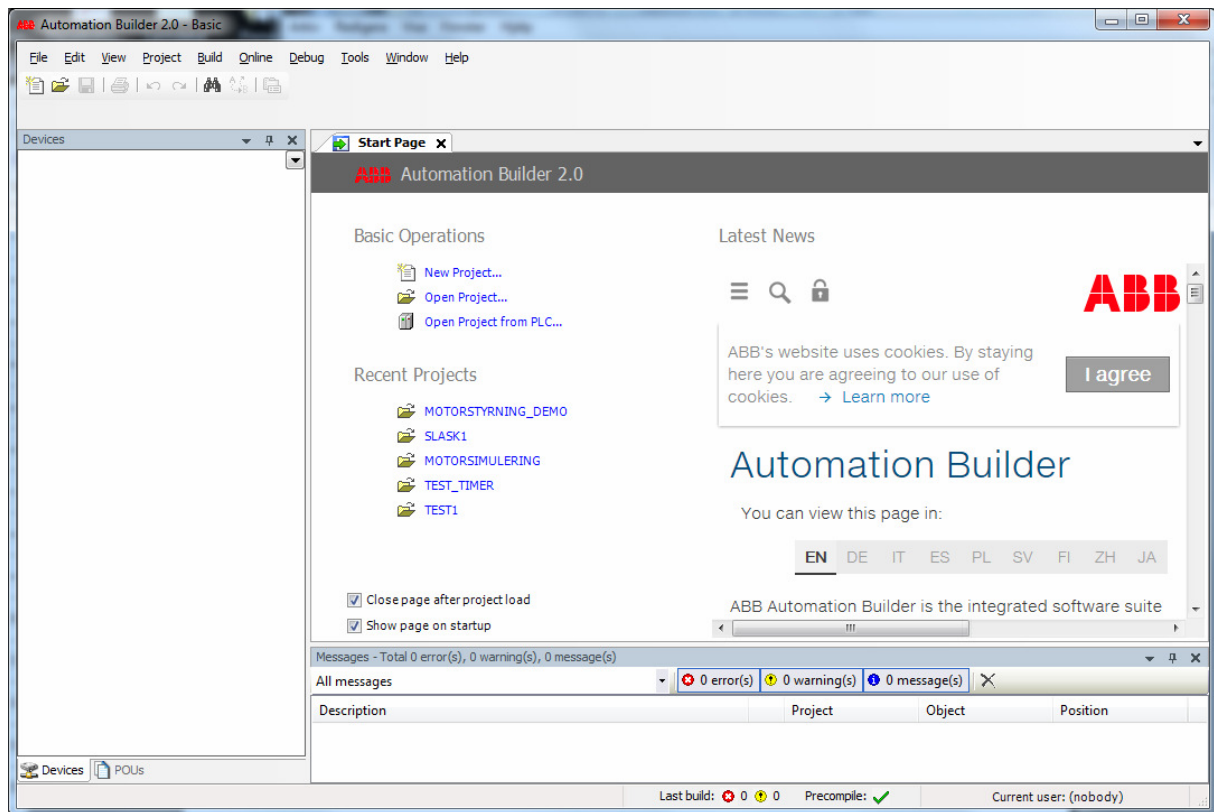
```
0001 PROGRAM SERVICE
0002 VAR
0003 END_VAR
0004
0001 LD ACT_VAL
0002 GE 3
0003 ST SERVICE_NEEDS
0004
0005
0006
```

There are horizontal scroll bars at the top and bottom of the code area, and a vertical scroll bar on the right side.

SERVICE (IL)

Konfigurering av systemet

Starta programmet AB



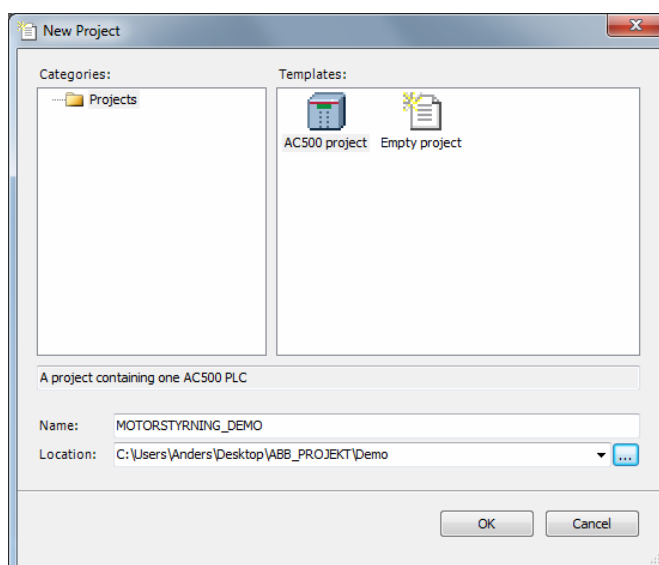
Klicka på

- New Project

Döp projektet till

- MOTORSTYRNING_DEMO

Välj location (var projektet ska sparas)



Klicka på

- OK

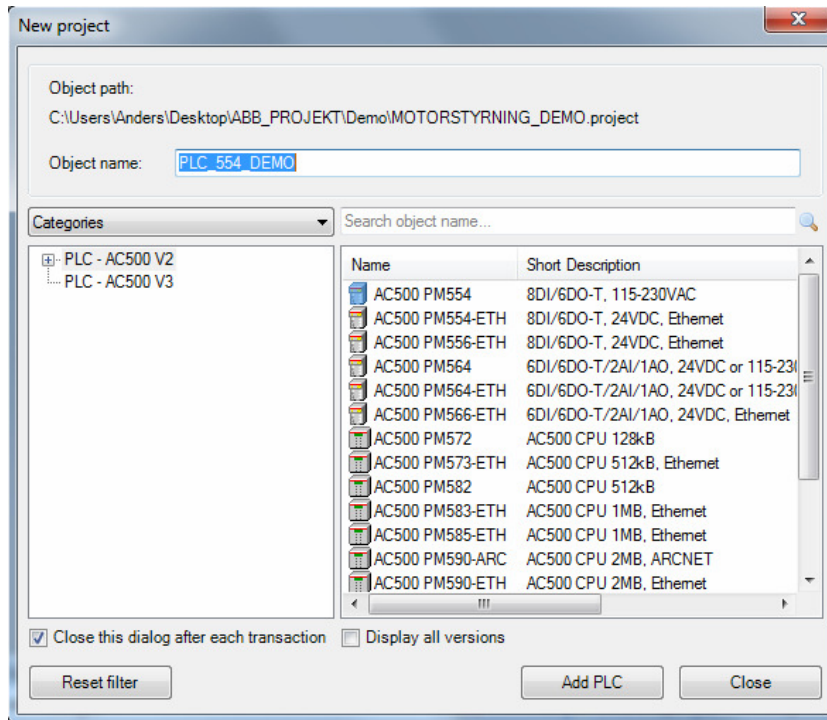
Klicka på

- PLC-AC500 V2

Och välj

- AC 500 PM554

Man kan även ändra namnet

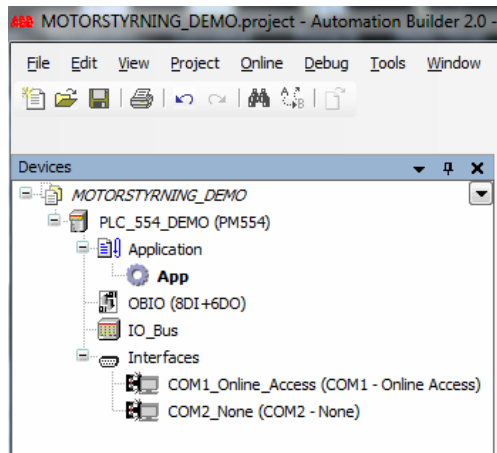


Klicka på

- Add PLC

Ställa in kommunikationen i Win

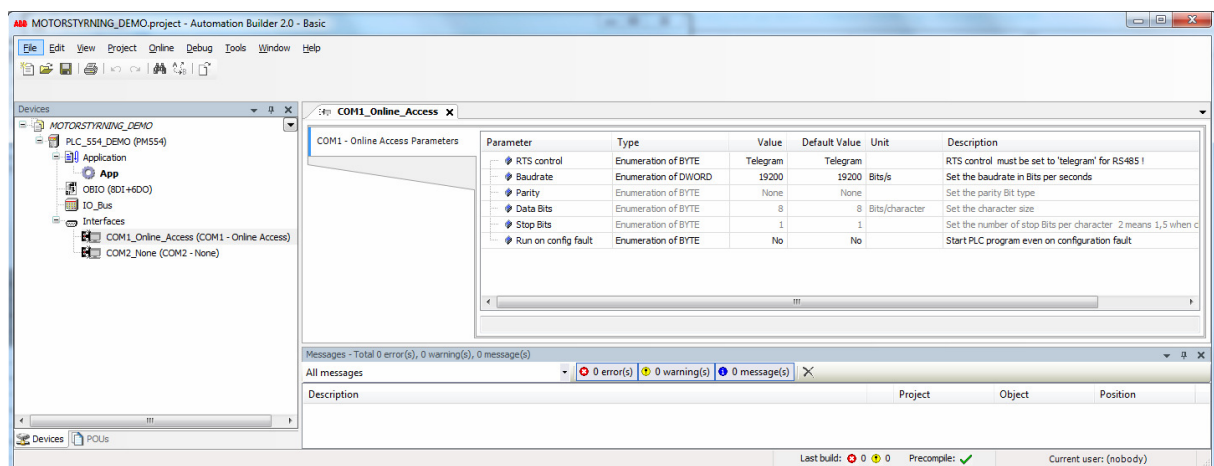
För att kommunikationen ska fungera måste man kontrollera inställningarna i Portar i Win.



I AB är det port COM1 som är vald.

Dubbelklicka på

- COM1



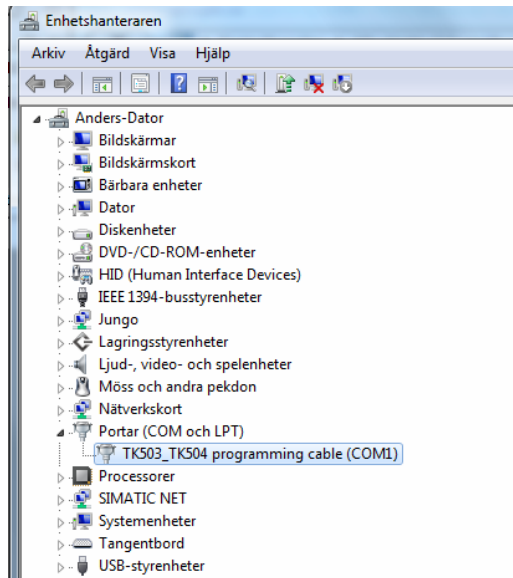
Kontrollera vilka värden som är inställt.

Kommunikationskabel TK503

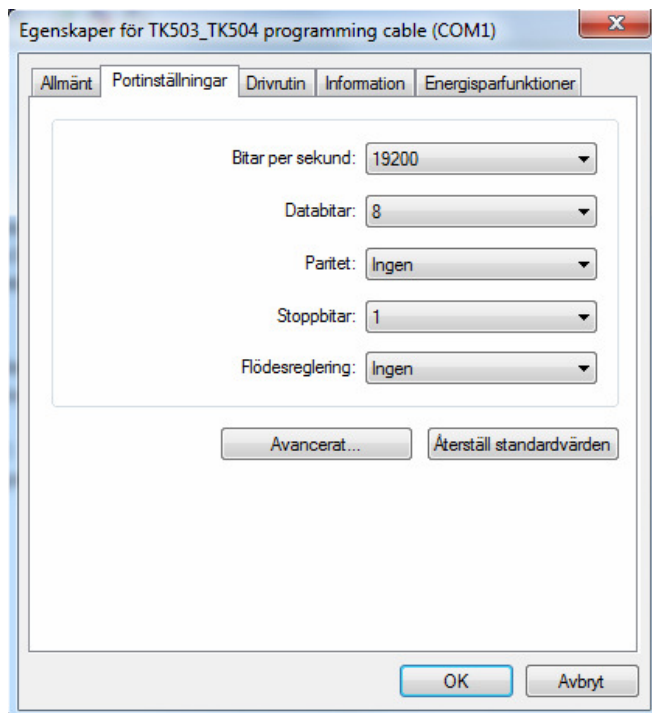
Vi använder oss av en kommunikationskabel typ: TK503
Den har en USB anslutning och en RS232 i andra ändan.



Öppna Enhetshanteraren i Windows för att kontrollera att den överensstämmer med AB inställningar. Detta förutsätter dock att kabeln är ansluten.



Dubbelklicka på
- TK503



För att kontrollera inställningarna på porten som hastighet och att den överensstämmer med inställningarna i AB

Klicka på

- Portinställningar

Ändra vid behov.

Om man behöver ändra port, klicka på

- Advanced

Här kan man ställa in önskad COM-port.

Klicka på

- OK

Deklarera in- och utgångar

De fysiska in- och utgångarna deklarerar i AB

Dubbelklicka på

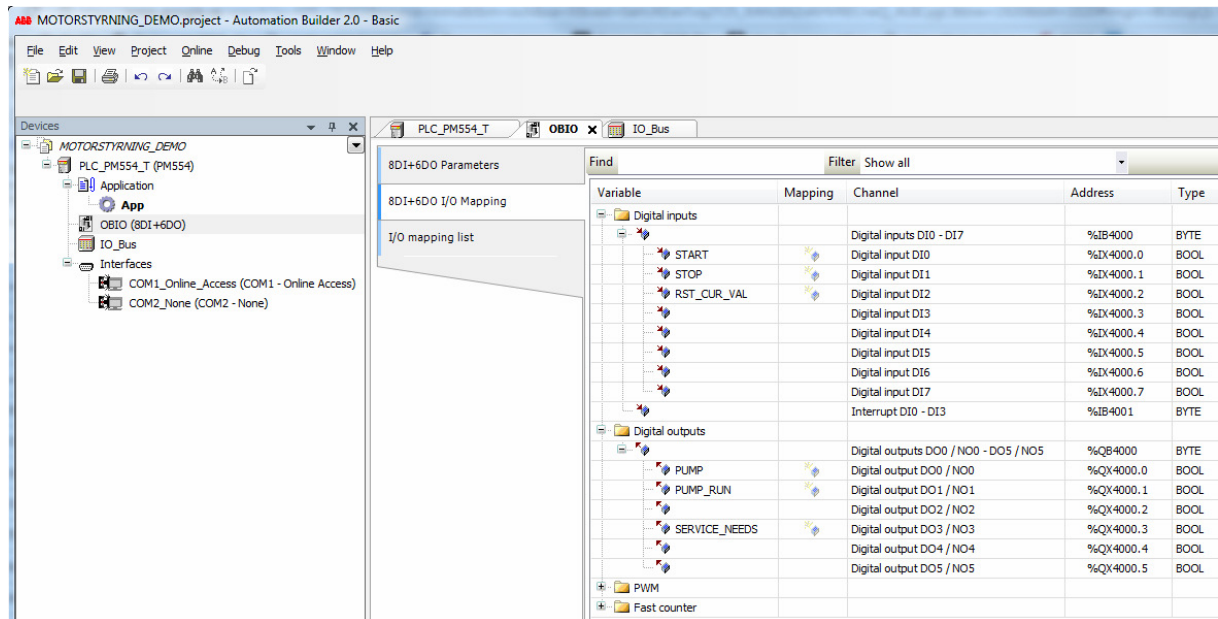
- OBIO

Klicka på

- 8DI+6DO I/O Mapping

Öppna upp Digital input och Digital output genom att klicka på

- +tecknen



Skriv sedan in variabelnamnen enligt:

Digital inputs

START	Digital input DI0	%IX4000.0
STOPP	Digital input DI1	%IX4000.1
RST_CUR_VAL	Digital input DI2	%IX4000.2
	Digital input DI3	%IX4000.3
	Digital input DI4	%IX4000.4
	Digital input DI5	%IX4000.5
	Digital input DI6	%IX4000.6
	Digital input DI7	%IX4000.7

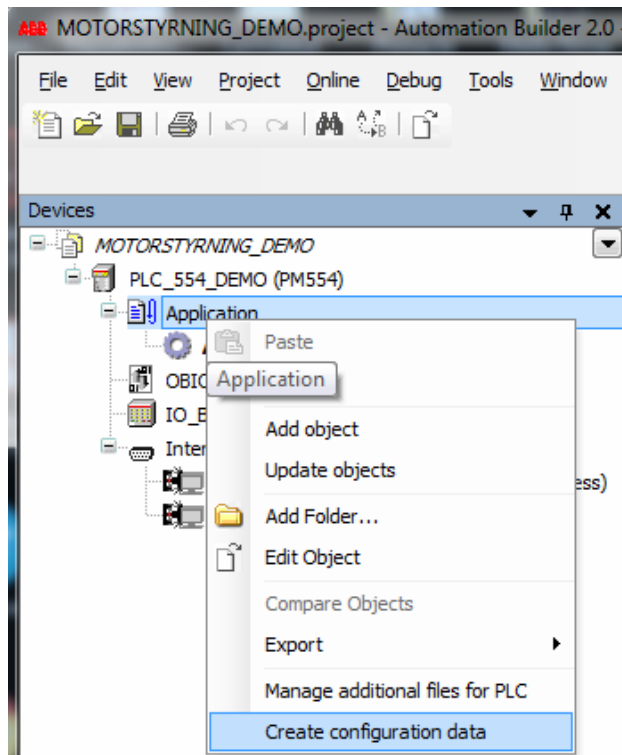
Digital outputs

PUMP	Digital input DO0	%QX4000.0
PUMP_RUN	Digital input DO1	%QX4000.1
	Digital input DO2	%QX4000.2
SERVICE_NEEDS	Digital input DO3	%QX4000.3
	Digital input DO4	%QX4000.4
	Digital input DO5	%QX4000.5

Skapa konfigureringsdata

Högerklicka på

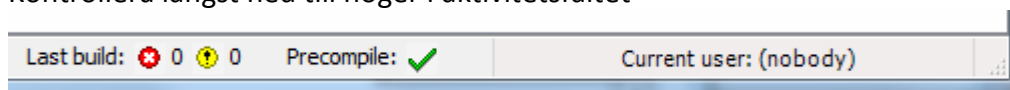
- Application



Klicka på

- Create configuration data

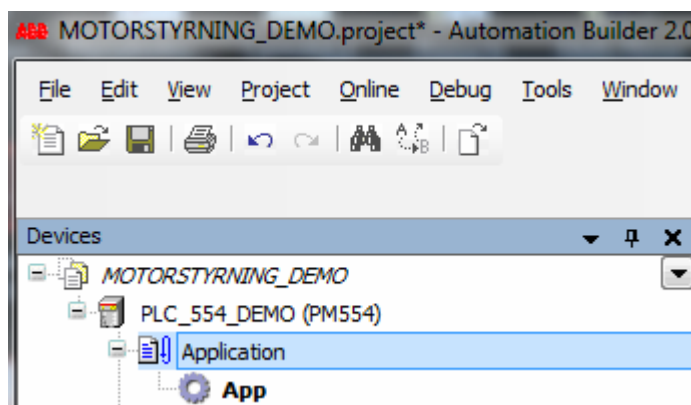
Kontrollera längst ned till höger i aktivitetsfältet



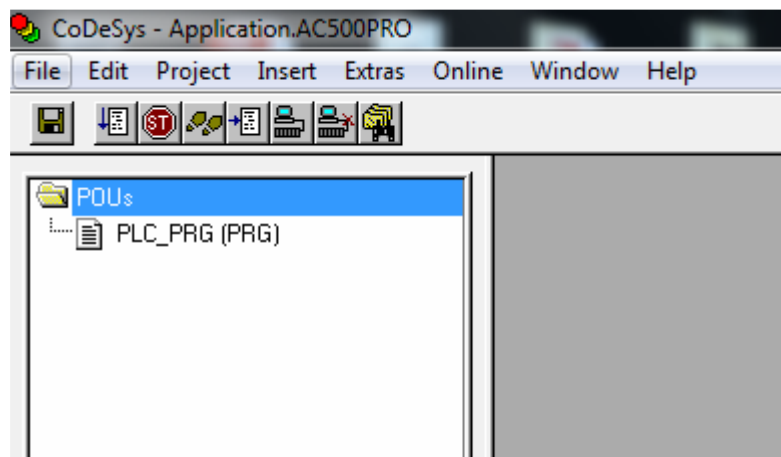
Att Precompiled är i bockad.

Dubbelklicka på

- Application



Nu öppnas CoDeSys



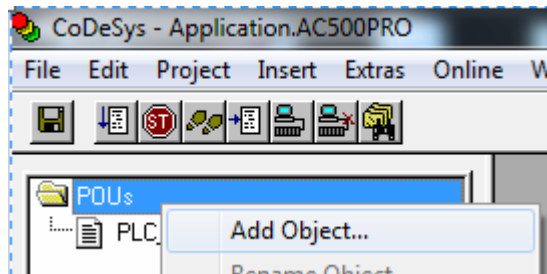
Det är i denna miljö man skapar programmen.

POU - MOTORKRETS_PUMP (LD)

Deklarera en ny POU

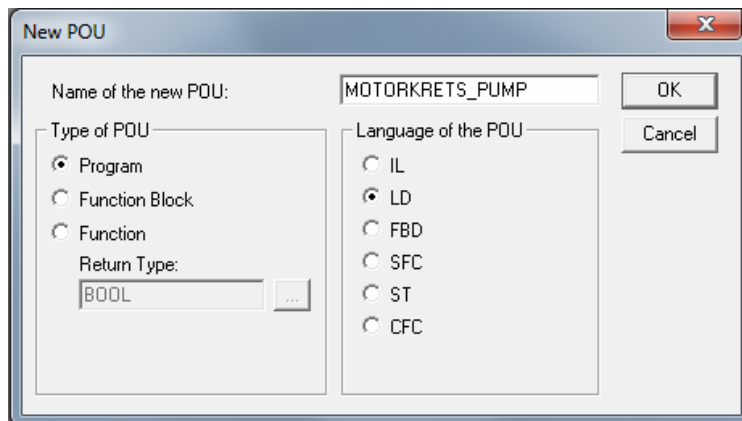
Högerklicka på

- POUs



Och klicka på

- Add Object



Klicka på

- Program
- LD

Och skriv i ett namn MOTORKRETS_PUMP

Klicka på

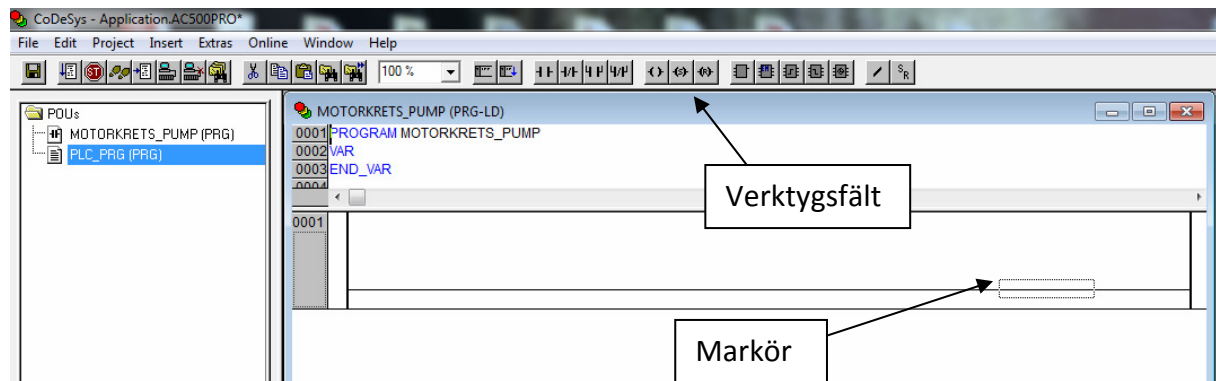
- OK

Tips

Genom att högerklicka i network kommer man åt de flesta kommandon på ett snabbt sätt.

För att följa med i programmeringsexemplet nedan kan det vara bra att titta på de färdiga blocken då och då. De finns i början av detta kompendium.

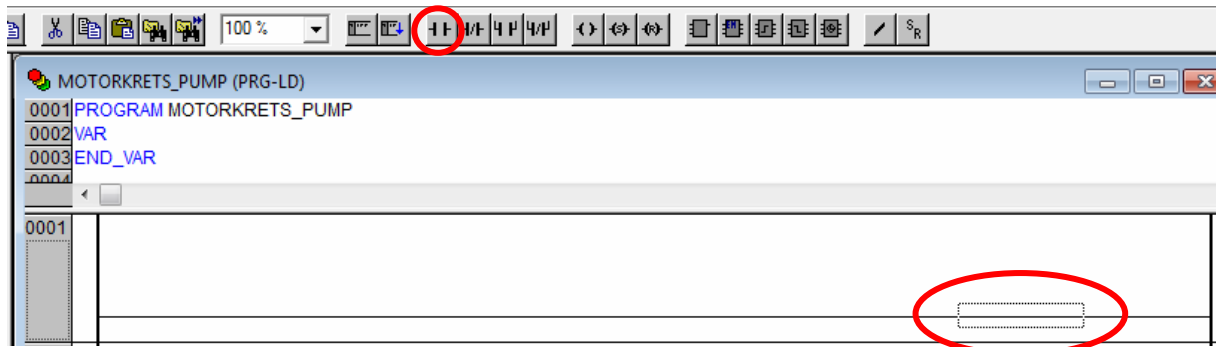
POU programmeringsläget



Markören i network visar var funktionerna infogas när man klicka på verktögsfälten högst upp eller använder högerklick.


Network 1

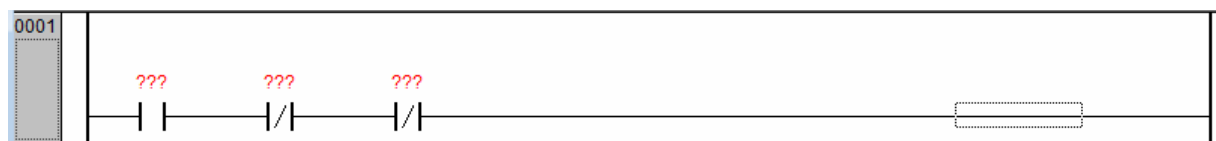
Lägga in första kontakten.




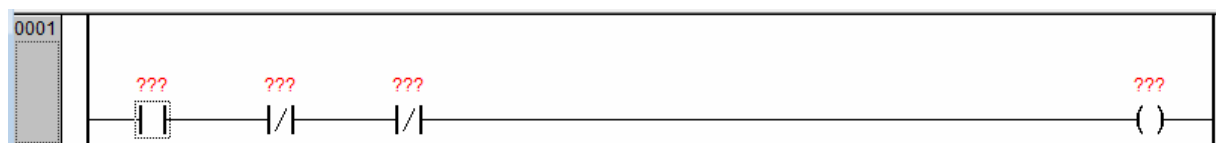
- Klicka på  (Contact) i verktygsfältet




- Klicka på ledningen till höger om kontakten
- Klicka på  (Contact negated) 2 gånger

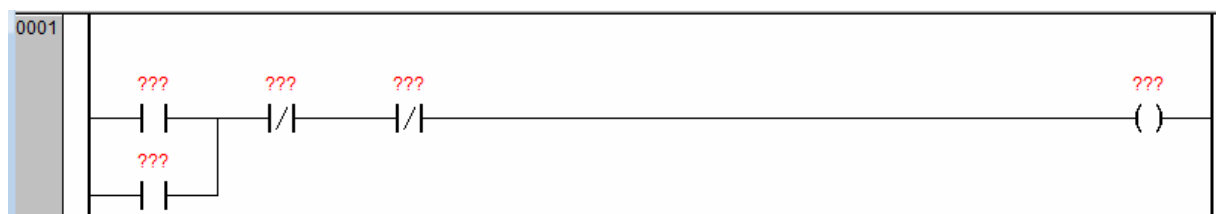


- Klicka på ledningen till höger om den sista kontakten
- Klicka på  (Coil)

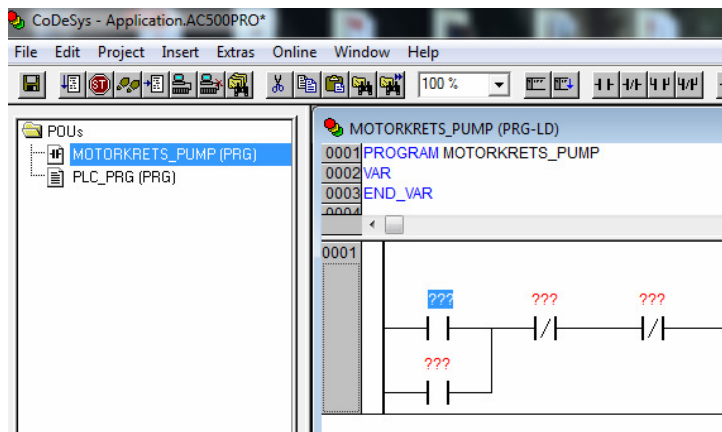


- Klicka på den första kontakten så den markeras
- Klicka på  (Parallel Contact)

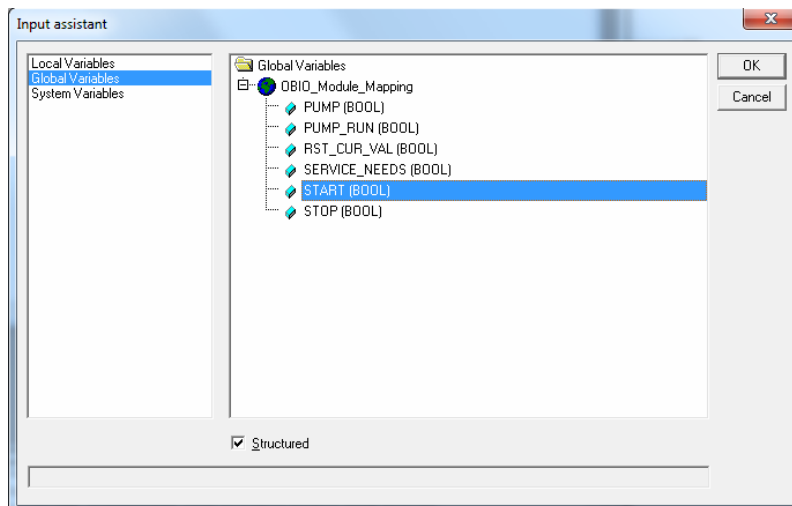
Färdig krets



Lägga in en variabel



- Markera frågetecknet på första variabeln
- Klicka på funktionstangent F2

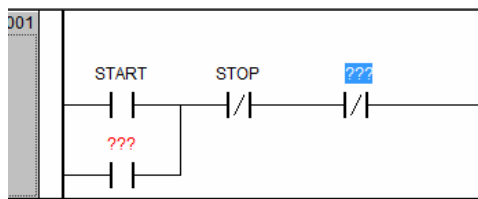


- Klicka på Global Variables till vänster
- Klicka i högerdelen på + tecknet på OBIO_Module_Mapping
- Markera sedan START (BOOL)
- Klicka på OK

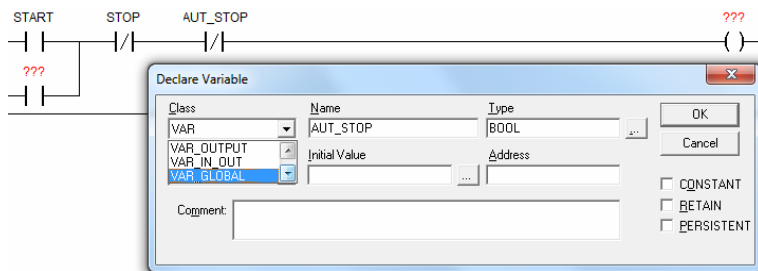
Gör på motsvarande sätt för variabeln STOP

De andra variablerna är interna och måste deklarerars då man skapar de.

Deklarera interna variabler



- Markera frågetecknet på tredje variabeln
- Skriv det önskade variabelnamnet AUT_STOP
- Klicka på Enter




- Välj VAR_GLOBAL i Class

Vill man att variabeln endast ska vara tillgänglig för anrop i denna POU kan man välja att deklarerar den som en VAR

- Kontrollera att rätt Type är vald. Välj BOOL
- Klicka på OK

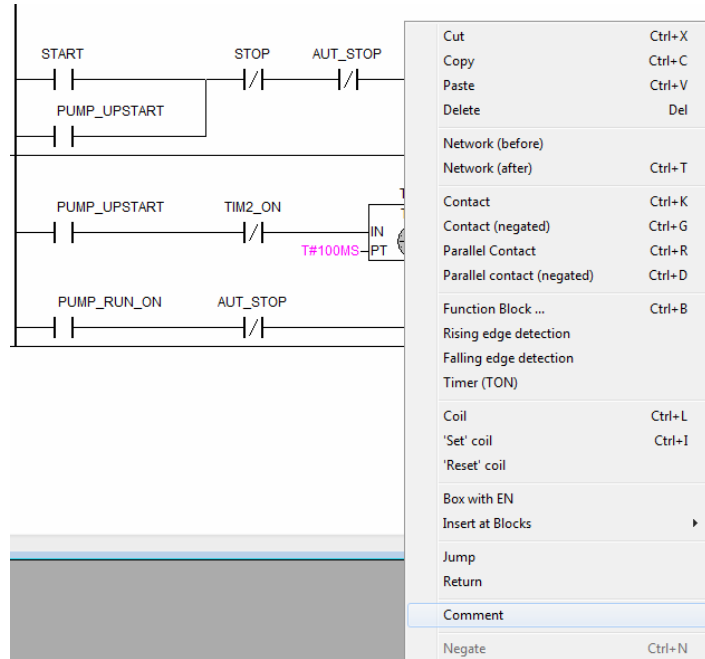
Nu har variabeln deklarerats och finns tillgänglig i projektet. Systemet har själv tilldelat en adress.

Deklarera variabeln PUMP_UPSTART på utgången  på motsvarande sätt.

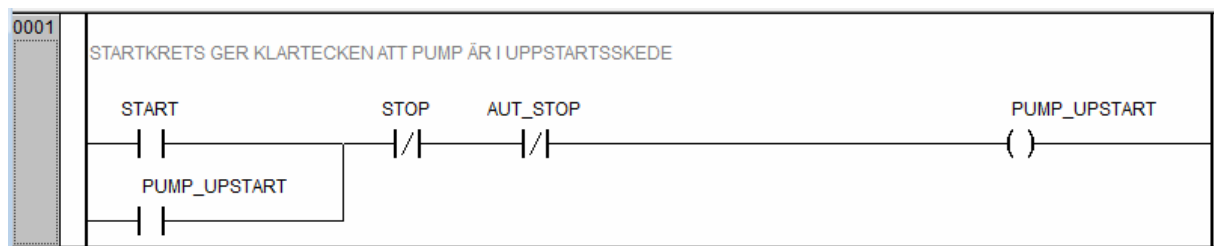
Lägg in variabeln PUMP_UPSTART på parallellkontakten på samma sätt som tidigare.

Infoga kommentar

- Högerklicka i network
- Välj Comment
- Dubbelklicka på texten
Comment och skriv in egen
kommentar
STARTKRETS GER
KLARTECKEN ATT PUMP ÄR
I UPPSTARTSSKEDE



Så här ska det färdiga NETWORK1 se ut.



Tips

Om man vill byta funktionssätt på en kontakt, kan man klicka på NEGATE

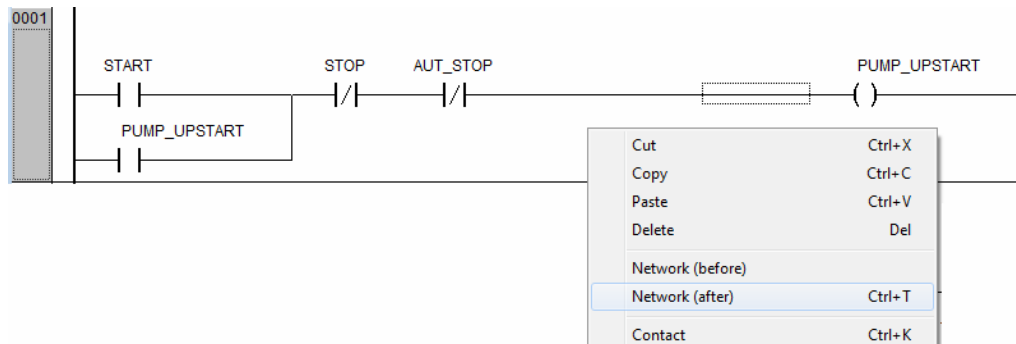


eller högerklicka på kontakten och välja NEGATE

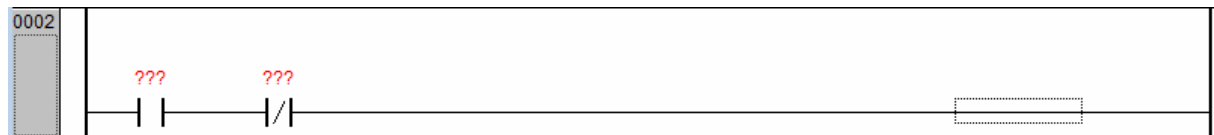
Network 2

Infoga nytt Network

- Högerklicka i NETWORK1
- Klicka på Network (after)

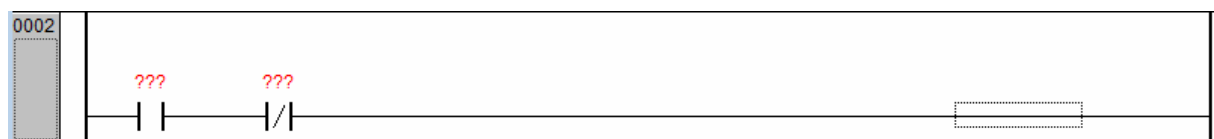
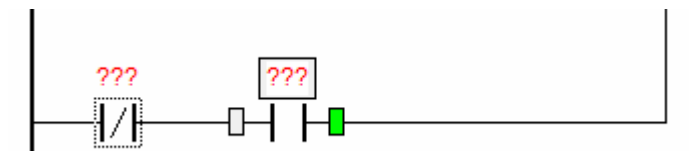


Lägg in 2 kontakter i serie varav den andra ska vara NC (normaly closed).

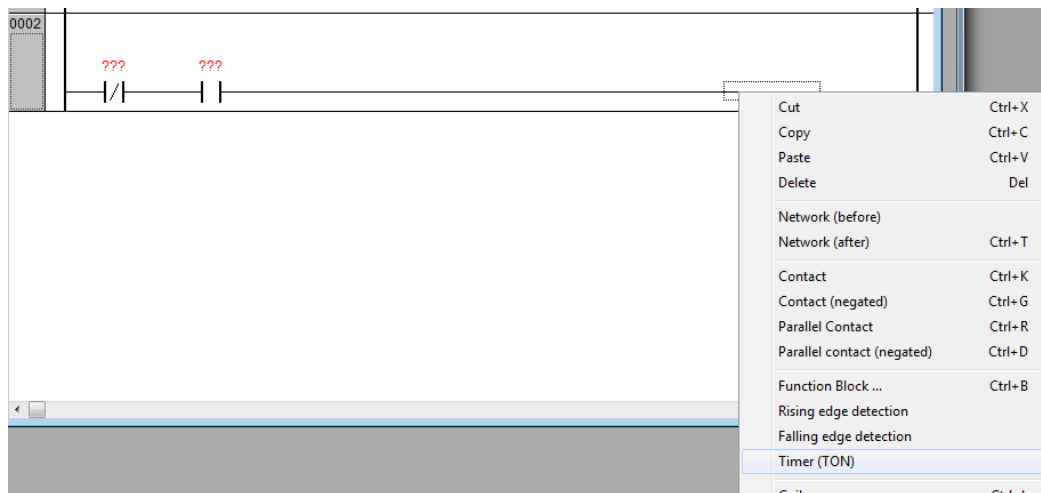


Tips

Om man vill flytta en kontakt klickar man på den och håller ned knappen och drar den till ny plats. När markören förs över ny placering på ny kontakt tänds gröna markeringar. Släpp knappen på den sida du vill placera kontakten.

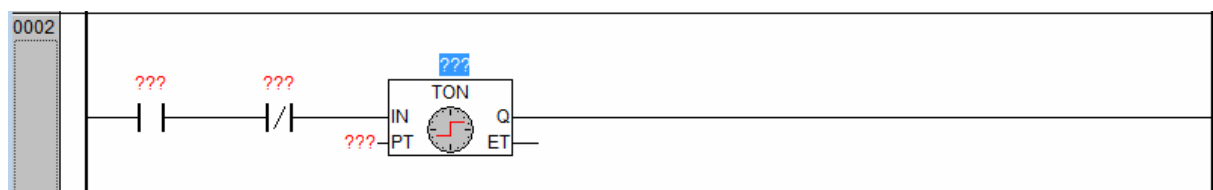
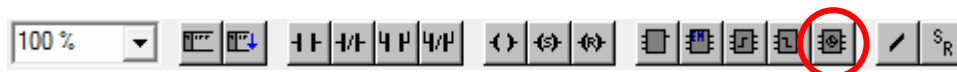


- Klicka på ledningen till höger om den andra kontakten så att markören hamnar där
- Klicka med höger knapp
- Välj Timer (TON)




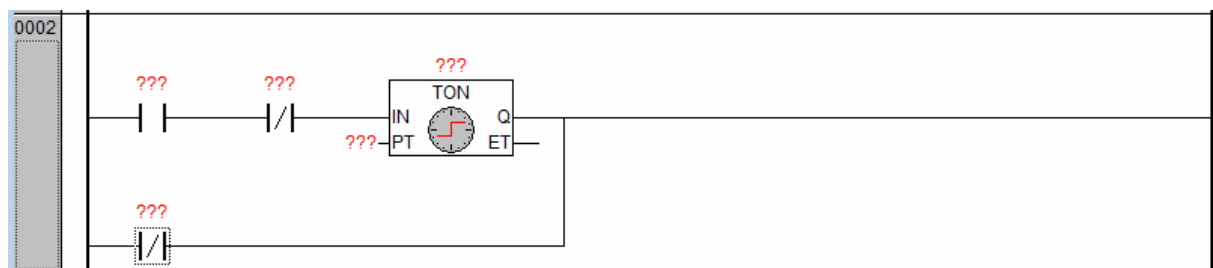
Tips

Man kan även klicka på Timer (TON) i verktygsfältet ovan.




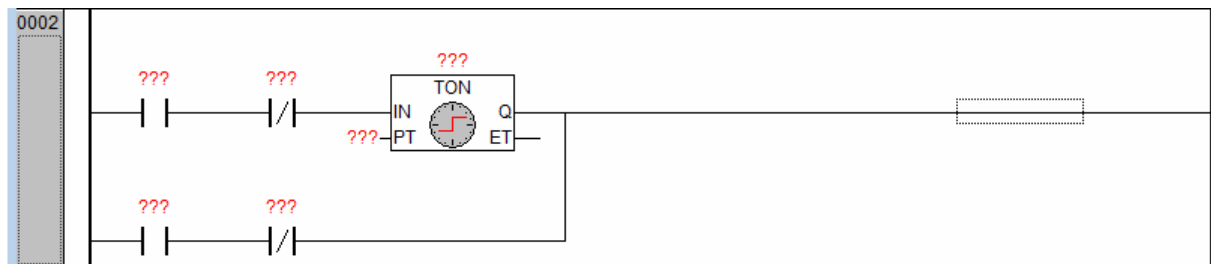
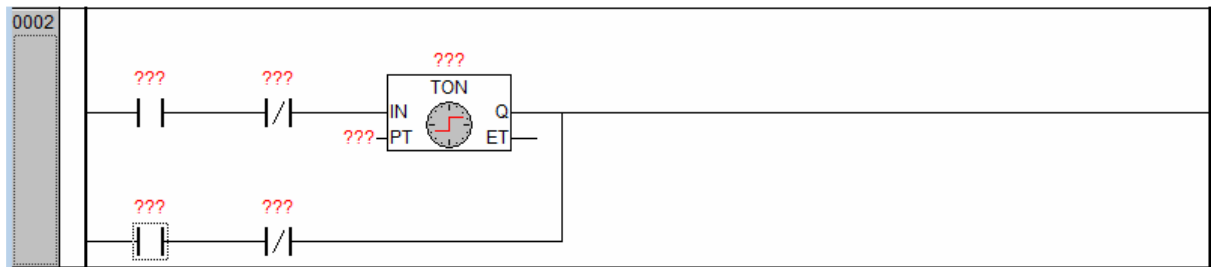
För att rita in den första parallella kontakten.


- Klicka först på den vänstra kontakten
- Håll ned shift-tangenten och klicka på den andra kontakten
- Håll ned shift-tangenten och klicka på timern
- Klicka på  Parallel Kontakt (negated)

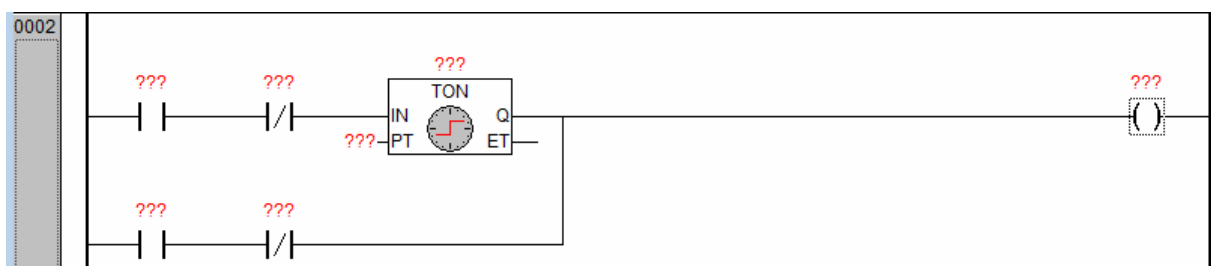



För att rita in den andra parallella kontakten.

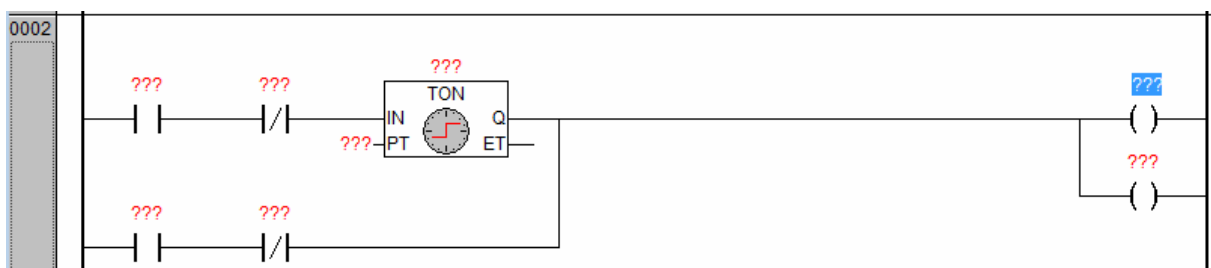
- Klicka sedan på den parallella kontakten
- Klicka sedan på  Contact
Den placeras då före den brytande kontakten



- Klicka på ledningen till höger om timern
- Klicka på  Coil

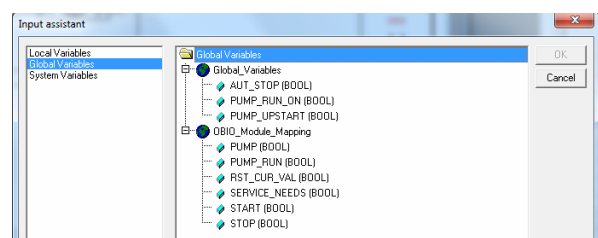


- Klicka på coil så den markeras
- Klicka på coil  igen så placeras den parallellt med den andra



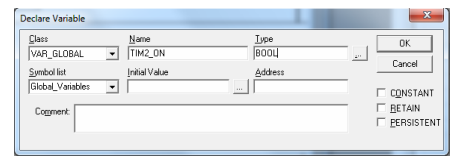
Lägg in variablerna på samma sätt som tidigare med F2 för

- PUMP_UPPSTART
- AUT_STOP
- PUMP



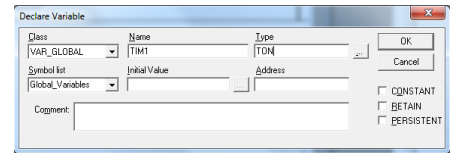
Deklarera övriga variabler som tidigare genom att skriva namnet på frågetecknen och trycka enter för

- TIM2_ON (class=VAR_GLOBAL, type = BOOL)
- PUMP_RUN_ON (class=VAR_GLOBAL, type = BOOL)
- TIM1_ON (class=VAR_GLOBAL, type = BOOL)



Deklarera variablen TIM1 enligt

- TIM_1 (class=VAR_GLOBAL, type = TON)



Skriv på ingångsbenet PT på TIM1

- T#100MS

TON

TON är ett FUNCTION BLOCK och fordrar en minnesarea att spara och data i. Därför deklarerar man denna med ett variabelnamn ovanför själva blocket.

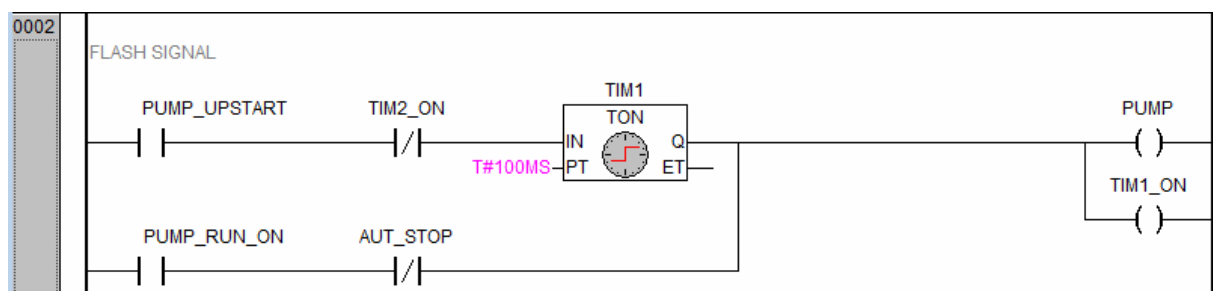
Funktion

T#100ms motsvarar 100msek (0,1 sek). Det innebär att utgången Q blir ett-ställd 0,1 sekunder efter att ingången IN blivit aktiv (1).

Infoga kommentar

- Högerklicka i network
- Välj Comment
- Dubbelklicka på texten Comment och skriv in egen kommentar FLASH SIGNAL

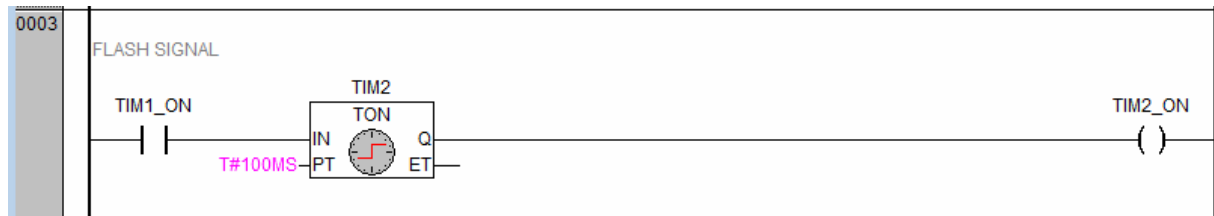
Så här ska det färdiga NETWORK2 se ut.



Network 3

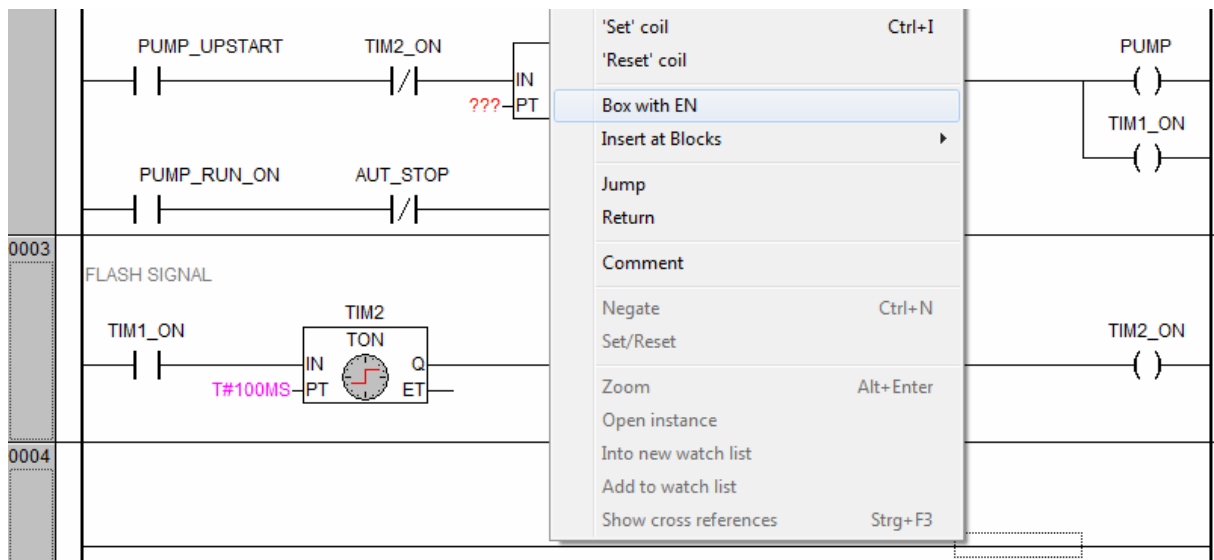
Infoga nytt network (after) och lägg in kontakter och timer och deklarera variabler samt kommentar på motsvarande sätt som tidigare beskrivits.

Så här ska det färdiga network se ut.



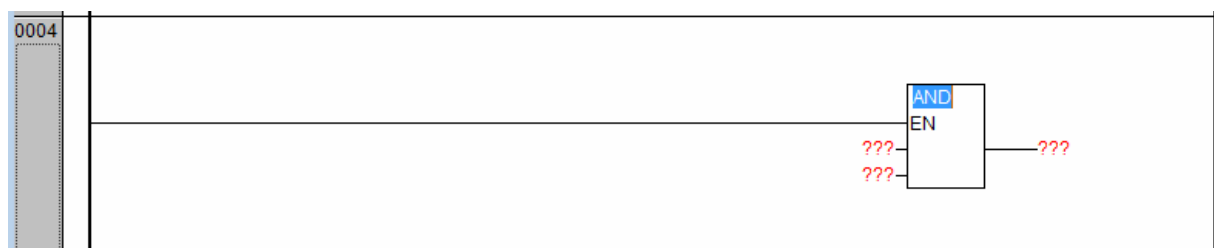
Network 4

Infoga nytt network (after).

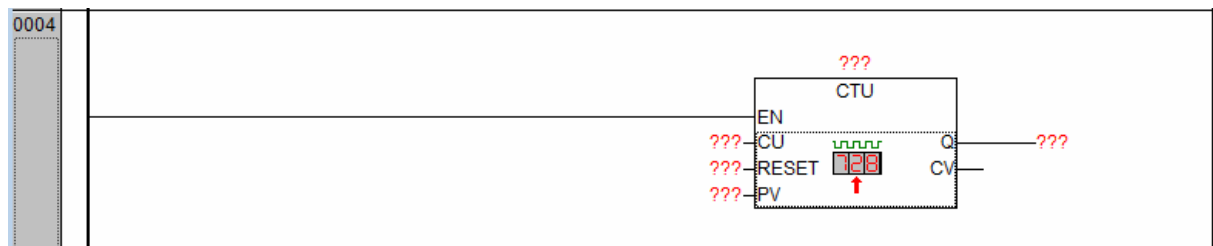


För att rita in räknaren.

- Högerklicka i network
- Välj Box with EN

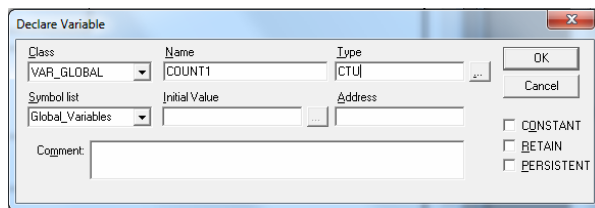


- Klicka på namnet (AND) i boxen
- Skriv CTU (counter upp)



Deklarera räknaren.

- Klicka på frågetecknen på CTU
- Skriv COUNT1
- Tryck enter



- Välj VAR_GLOBAL i Class
- Kontrollera de övriga fälten
- Klicka på OK

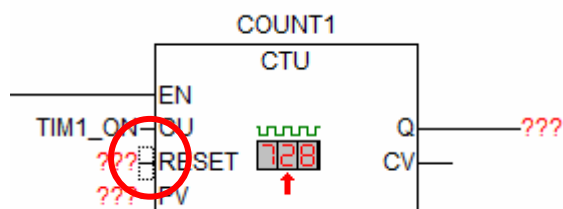
CTU

CTU är en räknare som räknar signaler uppåt d.v.s ökar värdet på varje insignal. Räknarens utgång Q kommer att vara 0 tills det att ingångsbenet CU erhållit det antal signaler som är angivet på ingångsbenet PV. När antalet insignaler överensstämmer med PV-värdet kommer utgången Q att bli 1. För att kunna påbörja en ny räknesekvens måste räknaren först nollställas genom att lägga in en signal (1) på ingångsbenet RESET.

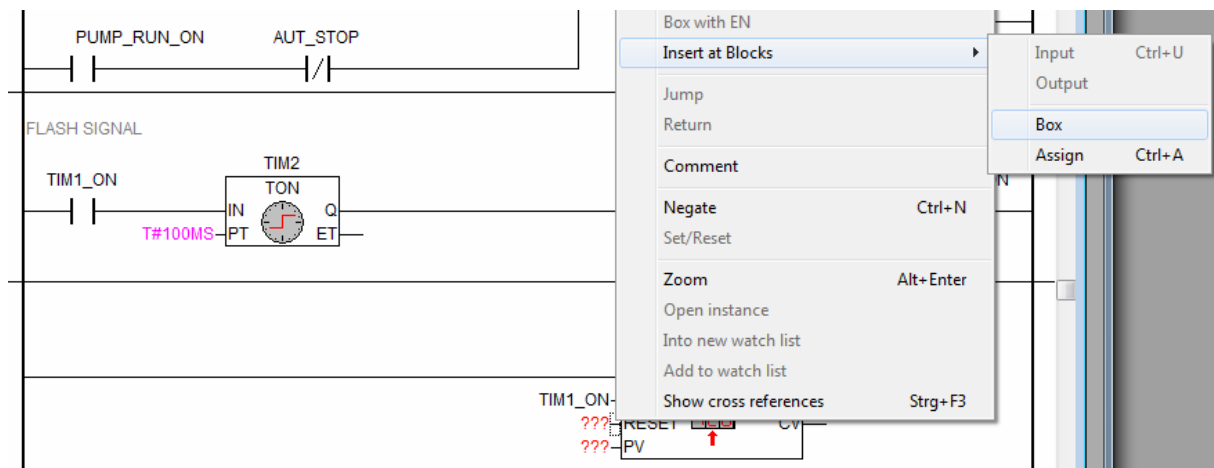
Ingångsbenet CU

- Lägg in TIM1_ON med F2 på ingångsbenet CU

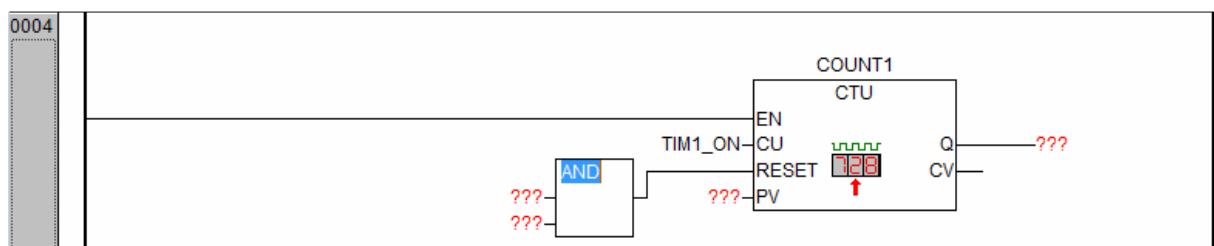
Ingångsbenet RESET



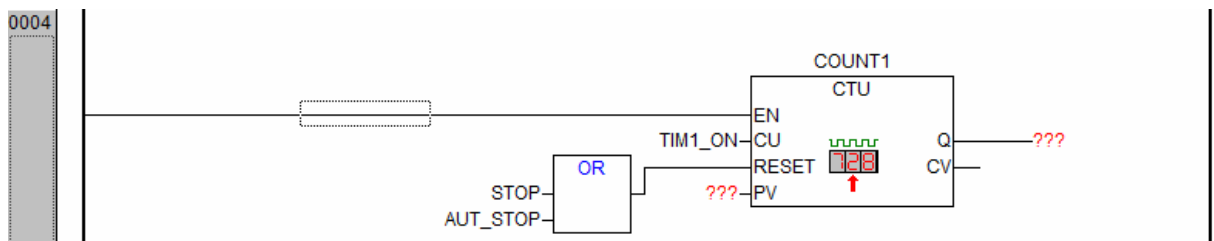
- Klicka på ledningen på ingången RESET
- Högerklicka sedan på den markerade rektangeln



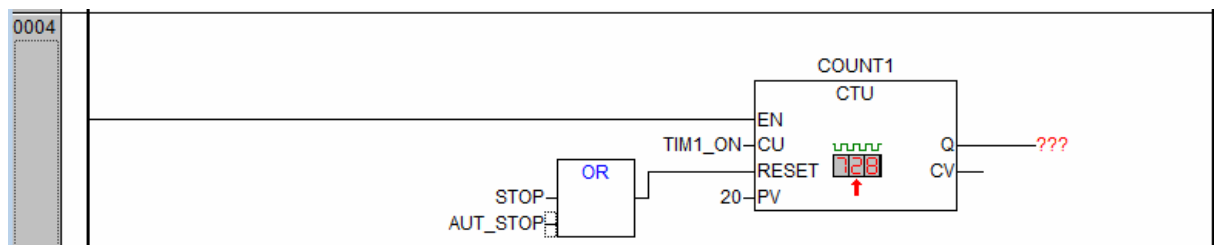
- Välj Insert at Blocks och Box



- Markera namnet i boxen (AND)
- Skriv OR



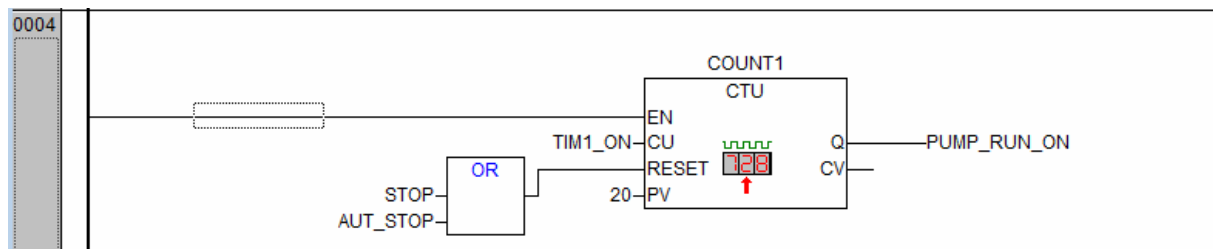
- Lägg in de 2 variablerna STOP och AUT_STOP på OR-grindens ingångar med F2



Ingångsbenet RESET. Anger förvalet för räknaren.

- Skriv 20 på ingångsbenet PV

Utgångsbenet Q.



- Lägg in PUMP_RUN_ON med F2

Utgångsbenet CV.

På utgångsbenet CV kan man lägga en variabel av typen INT. Då kan man läsa av aktuellt värde i räknaren med denna.

Infoga kommentar

- Högerklicka i network
- Välj Comment
- Dubbelklicka på texten Comment och skriv in egen kommentar ANTALET FLASH INNAN PUMP STARTAR

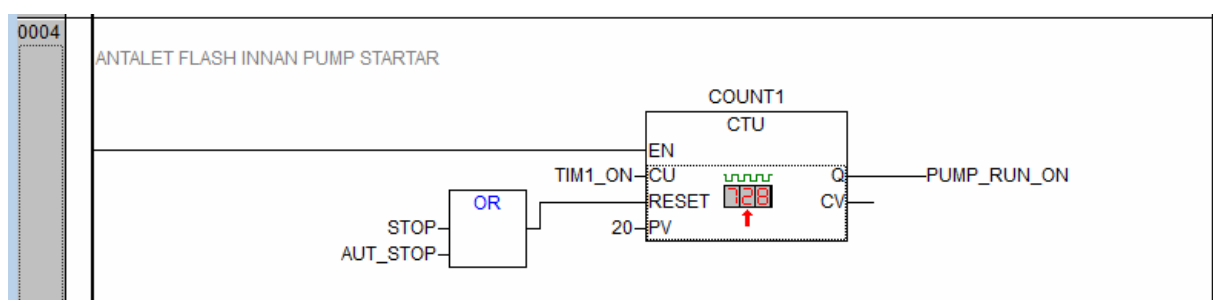
Tips

Function och Function Block

I manualen "DriveSPC_Standard_function_blocks.pdf" finns alla funktioner och funktionsblock beskrivna. Den finns tillgänglig på följande adress

https://library.e.abb.com/public/04663f4640f1aa0ac1257674004539b2/DriveSPC_Standard_function_blocks.pdf

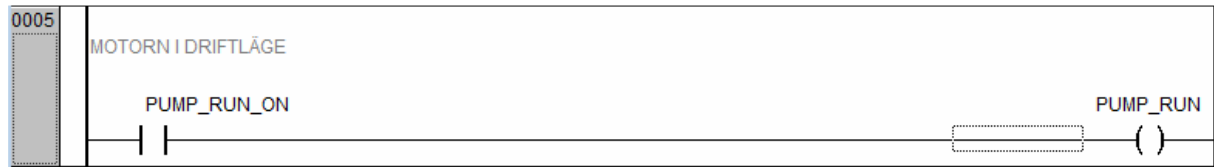
Så här ska det färdiga network se ut.



Network 5

Infoga nytt network (after) och lägg in kontakt och spole. Lägg sedan in variablerna med F2 samt kommentar på motsvarande sätt som tidigare beskrivits.

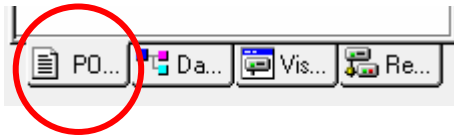
Så här ska det färdiga network se ut.



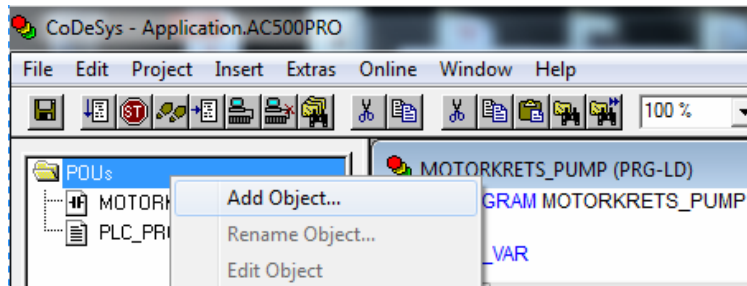
Nu är alla network klara i POU MOTORKRETS_PUMP

POU AUTOSTOP (CFC)

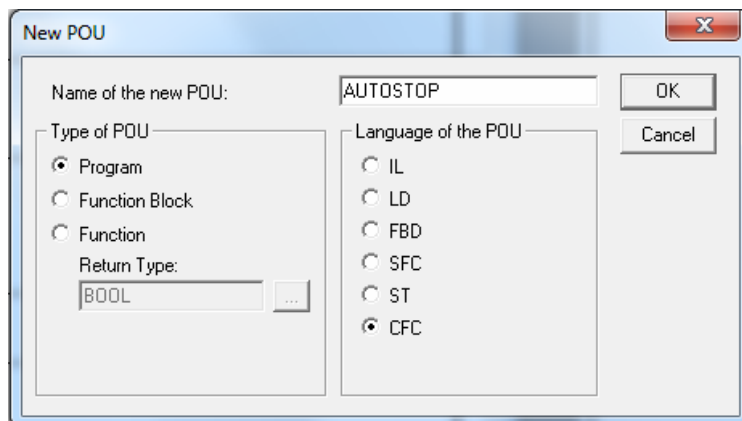
Infoga en ny POU på motsvarande sätt som beskrivits tidigare.
Denna POU ska dock programmeringsspråket CFC tillämpas.



- Klicka på POU's knappen i fönstret längs ned till vänster i CDS



- Högerklicka på den gula mappen POU's
- Välj Add Object...

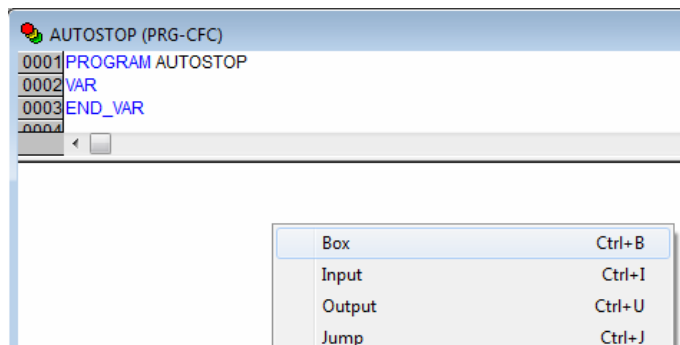


- Markera Program i Type of POU
- Markera CFC i Language of the POU
- Ge namnet AUTOSTOP i fältet för Name of the new POU
- Klicka på OK

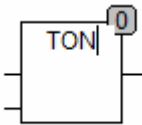
CFC

Continous function chart är en slag variant av function block. Det använder inte network utan alla funktioner och komponenter kan placeras fritt.

Infoga timer



- Högerklicka i network
- Välj Box
- Klicka någonstans i network så den placeras

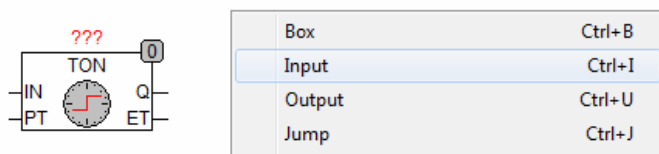


- Markera namnet (AND)
- Skriv TON
- Tryck enter

Tips

I verktygsfältet ovanför network finns alla tillgängliga verktyg. Man kan lägga in objekt antingen genom verktygsfältet eller genom att högerklicka och välja. Det finns också möjlighet att använda kortkommando.

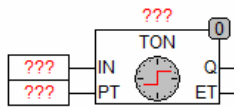
Infoga ingångsvariabler



- Högerklicka i network och välj Input
- Placera ingången på benet IN på timern och klicka där

Gör på motsvarande sätt för ingångsbenet PT.

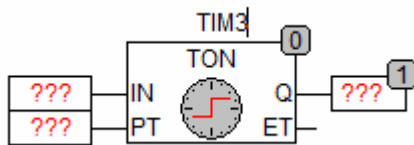
Infoga utgångsvariabel



Box	Ctrl+B
Input	Ctrl+I
Output	Ctrl+U
Jump	Ctrl+J

- Högerklicka i network och välj Output
- Placera utgången på benet Q på timern och klicka där

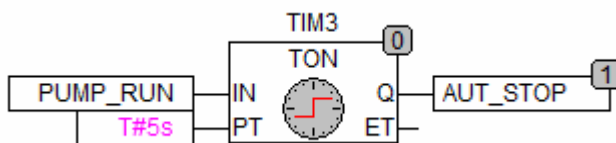
Deklarera TIM3



- Markera frågetecknen överst TON
- Skriv TIM3
- Tryck enter

Kontrollera att uppgifterna stämmer. Class VAR_GLOBAL

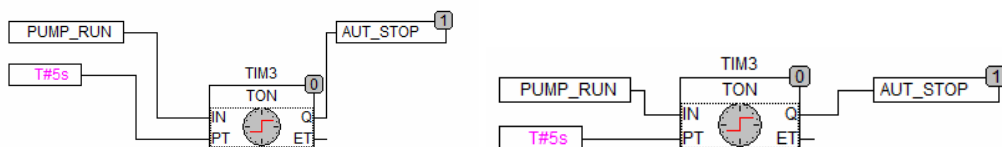
Lägga in variabelnamn



Lägg in variablerna och tidskontanten (5 sek) på motsvarande sätt som tidigare med F2.

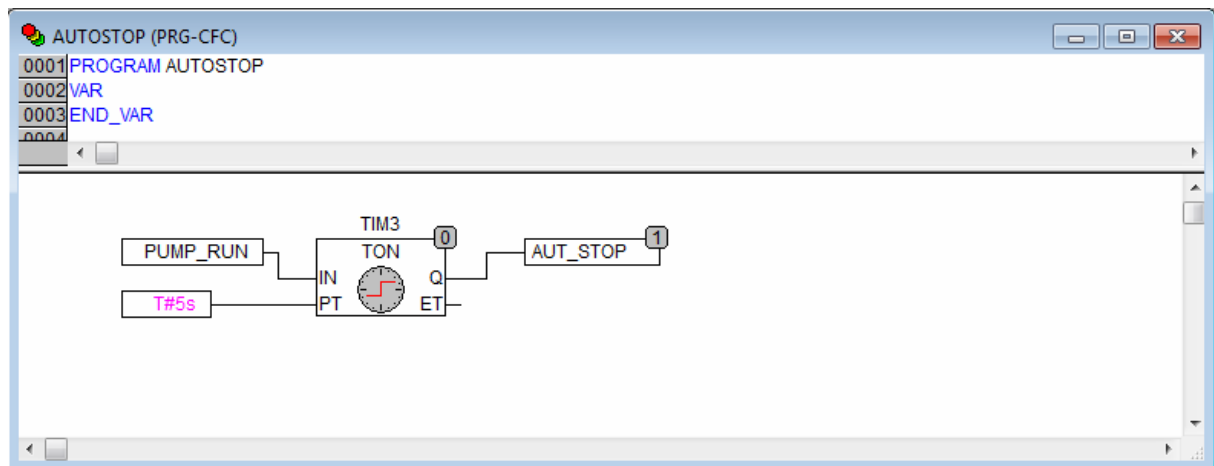
Tips

Om man vill flytta ett objekt markerar med det så att dess rektangel blir markerad.



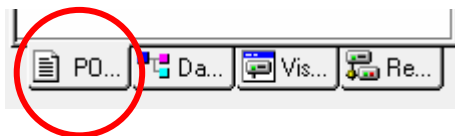
Genom att markera flera objekt kan man flytta flera på samma gång.

Så här ska den färdiga POU se ut.

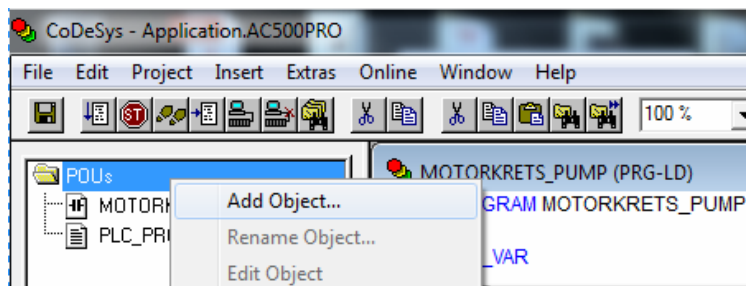


POU DRIFTDATA

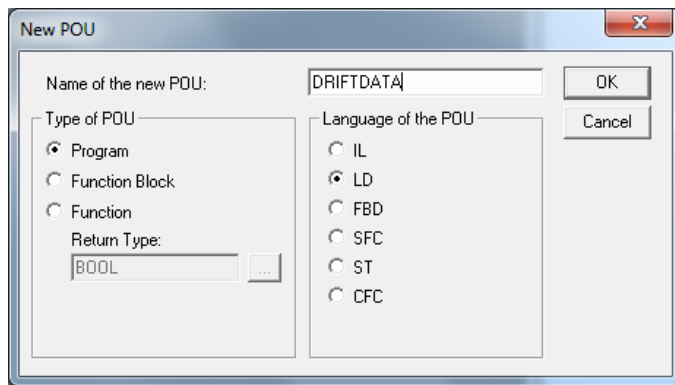
Infoga en ny POU på motsvarande sätt som beskrivits tidigare.
Denna POU ska dock programmeringsspråket LD tillämpas.



- Klicka på POU:s knappen i fönstret längs ned till vänster i CDS

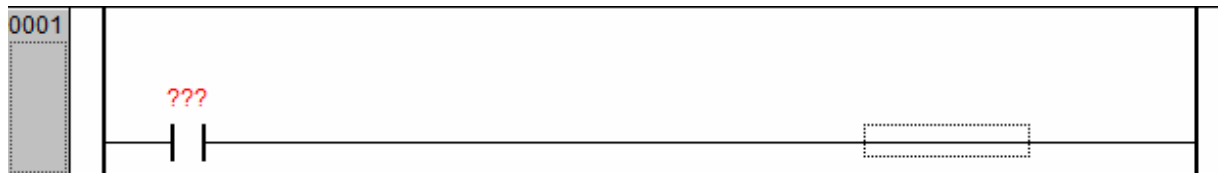


- Högerklicka på den gula mappen POU:s
- Välj Add Object...



- Markera Program i Type of POU
- Markera LD i Language of the POU
- Ge namnet DRIFTDATA i fältet för Name of the new POU
- Klicka på OK

Lägga in kontakt



- Lägg en slutande kontakt
- Klicka på ledningen till höger om kontakten

Lägga in triggpuls. Rising edge detection

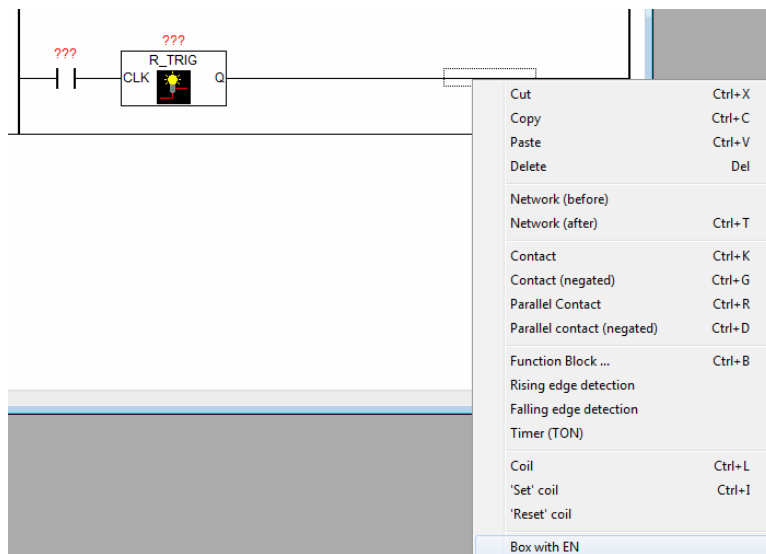


- Klicka på verktyget Rising egde detector i verktygsfältet överst

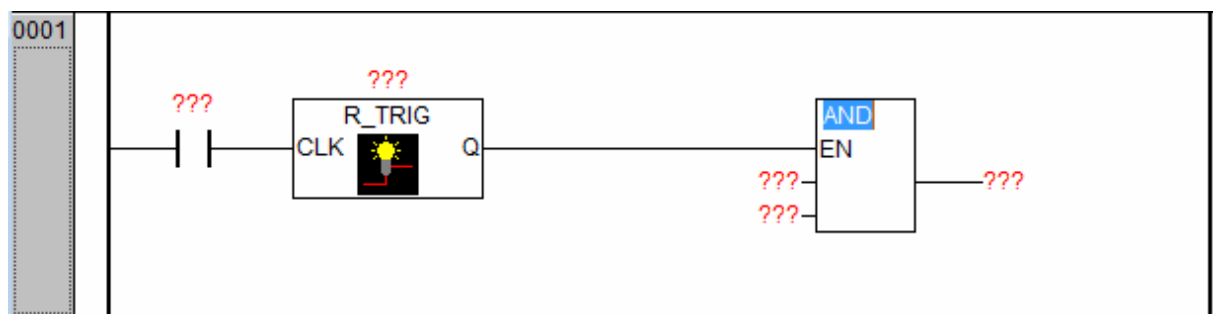
Rising edge detection

Denna funktion ger ut en 1:a på Q under ett programcykelvarv då ingången CLK går hög.

Lägga in box. ADD



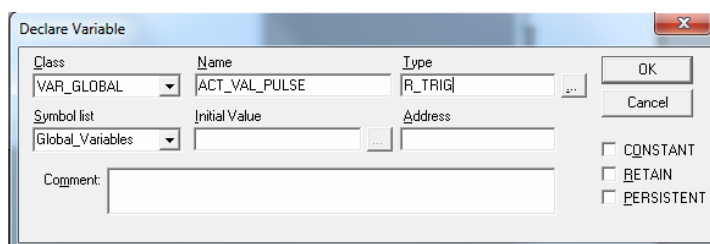
- Högerklicka på ledningen till höger om R_TRIG
- Välj Box with EN



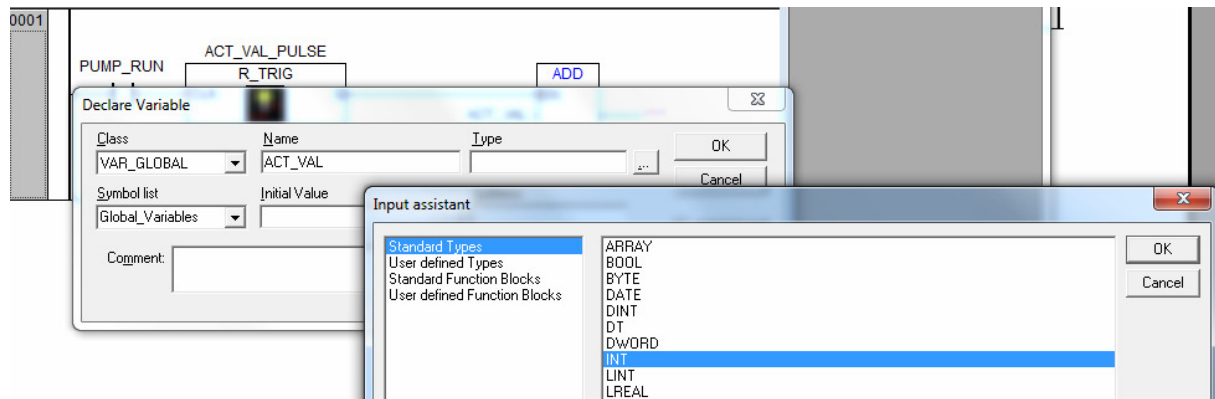
- Markera namnet i boxen (AND)
- Skriv ADD
- Tryck enter


Lägga in och deklarerar variabler

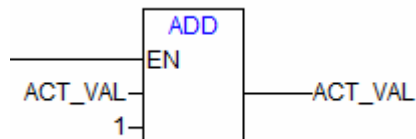
- Lägg in PUMP_RUN på motsvarande sätt som tidigare med F2
- Markera frågetecknen på R_TRIG
- Skriv ACT_VAL_PULSE
- Tryck enter



- Kontrollera uppgifterna i deklarerationsfönstret.
- Klicka på OK
- Markera frågetecknen på översta ingången på ADD
- Skriv ACT_VAL
- Tryck enter

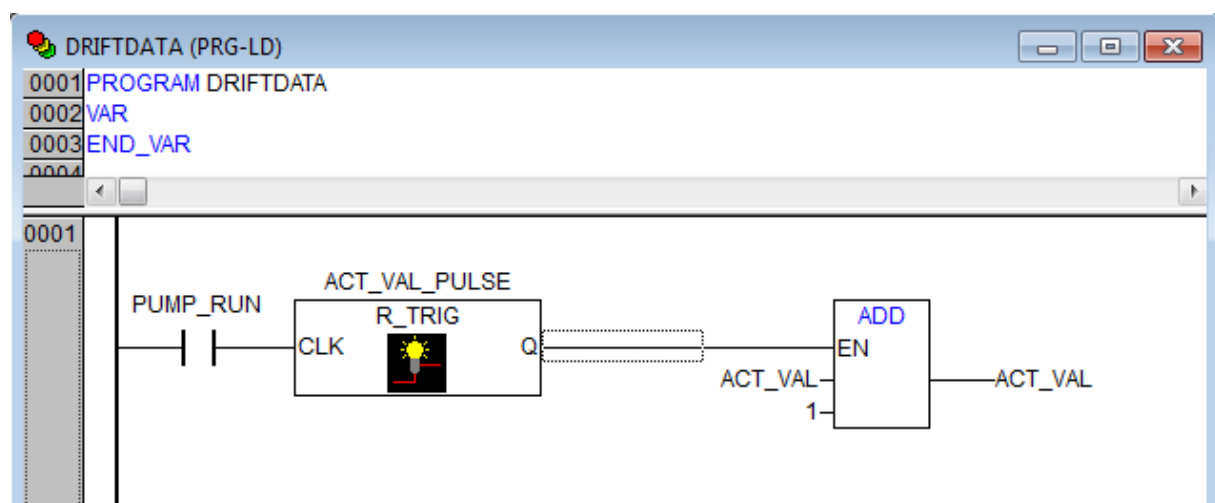


- Välj VAR_GLOBAL i Class
- Klicka på knappen  vid Type
- Välj Standard Type och INT
- Klicka på OK



- Skriv 1 på det nedre ingångsbenet på ADD
- Lägg in ACT_VAL på motsvarande sätt som tidigare med F2

Så här ska den färdiga POU se ut.

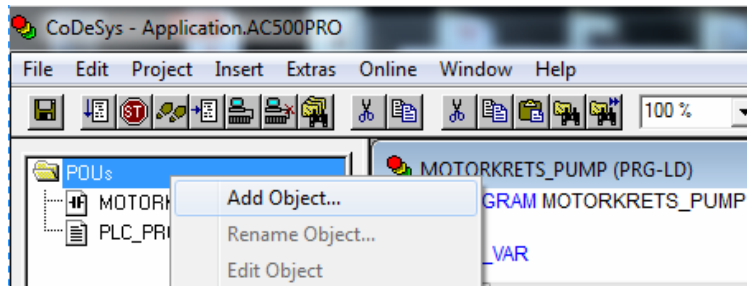


POU RESET_ACT_VALUE (ST)

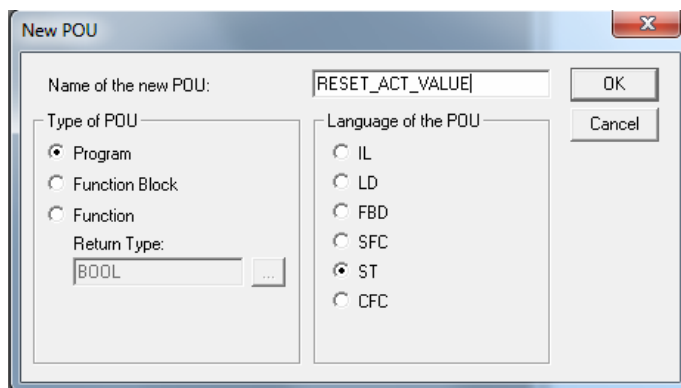
Infoga en ny POU på motsvarande sätt som beskrivits tidigare.
Denna POU ska dock programmeringsspråket ST tillämpas.



- Klicka på POU:s knappen i fönstret längs ned till vänster i CDS



- Högerklicka på den gula mappen POU:s
- Välj Add Object...



- Markera Program i Type of POU
- Markera ST i Language of the POU
- Ge namnet RESET_ACT_VALUE i fältet för Name of the new POU
- Klicka på OK

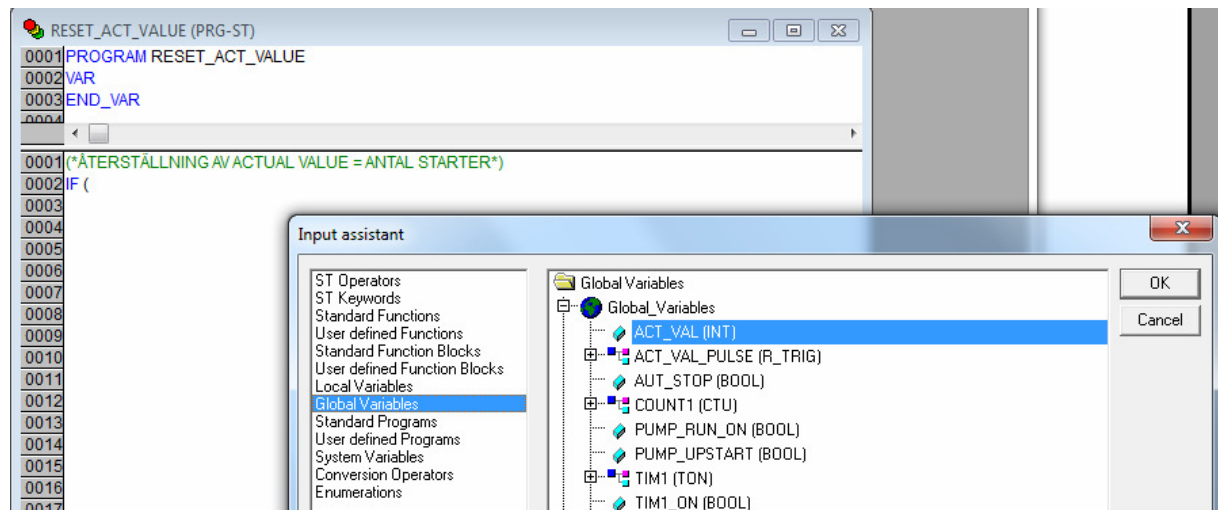
Skriva kommentarer

Text läggs in med (*TEXT*)

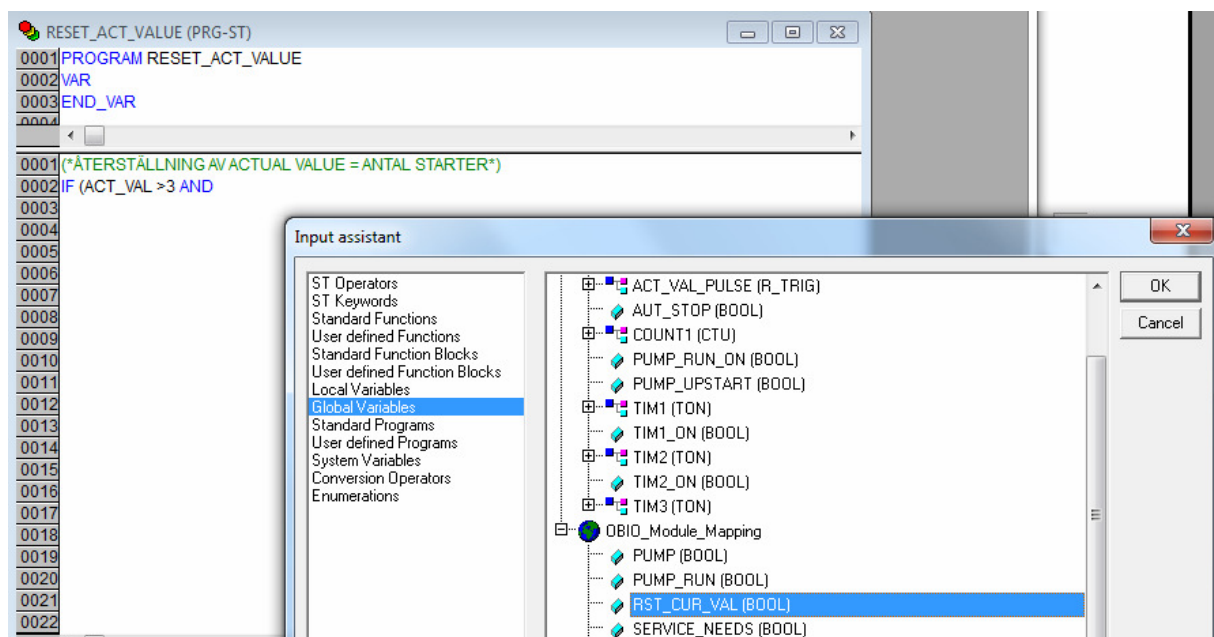
- Skriv (*ÅTERSTÄLLNING AV ACTUAL VALUE = ANTAL STARTER*)

Lägga in en IF-funktion

- Skriv IF (
- Klicka på F2
- Välj variabel ACT_VAL i listan



- Tillse att öppna upp mappen GlobalVariables till vänster
Det går även att skriva variabelnamnet direkt men då måste det avstavas exakt och med rätt germen/versal
- Klicka på OK
- Fortsätt skriva >3 AND
- Klicka på F2



- Välj variabel RST_CUR_VAL i listan
- Tillse att öppna upp mappen GlobalVariables till vänster
- Klicka på OK
- Fortsätt skriva =1) THEN
- Klicka på enter
- Klicka på F2
- Välj in variabel ACT_VAL på motsvarande sätt som tidigare
- Skriv :=0;
- Tryck enter

- Skriv ELSE
- Tryck enter
- Klicka på F2
- Välj in variabel ACT_VAL på motsvarande sätt som tidigare
- Skriv :=
- Klicka på F2
- Välj in variabel ACT_VAL på motsvarande sätt som tidigare
- Skriv ;
- Tryck enter
- Skriv END_IF;

Så här ska den färdiga POU se ut.

```

0001 PROGRAM RESET_ACT_VALUE
0002 VAR
0003 END_VAR
0004
0005 (*ÅTERSTÄLLNING AV ACTUAL VALUE = ANTAL STARTER*)
0006 IF (ACT_VAL >3 AND RST_CUR_VAL=1) THEN
0007 ACT_VAL:=0;
0008 ELSE
0009 ACT_VAL:=ACT_VAL;
0010 END_IF;
0011

```