

Tutorial

Siemens PLC

Simatic S7-300
STEP7
WinCC Flexible

DE HAAGSE
HOGESCHOOL

SIEMENS

De Haagse Hogeschool
Academie voor TIS / Delft
9 mei 2013

J.E.J. op den Brouw

Inhoudsopgave

1.	INLEIDING	4
2.	DE PC	6
3.	DE PLC	8
3.1	BESCHRIJVING PLC	8
3.1.1	<i>Processormodule CPU315F-2 PN/DP</i>	9
3.1.2	<i>Voedingsmodule PS307</i>	9
3.1.3	<i>Digitale I/O module SM323</i>	10
3.1.4	<i>Analoge I/O module SM334</i>	10
3.1.5	<i>Simulatie I/O module SM 374</i>	11
3.2	PROGRAMMEERTALEN	12
3.3	GEBRUIK PLC	13
4.	TUTORIAL STEP7 EN LAD.....	14
4.1	PROJECT AANMAKEN	14
4.2	NETWERK CONFIGUREREN	16
4.3	PLC CONFIGUREREN.....	17
4.4	CONFIGURATIE DOWNLOADEN.....	22
4.5	SYMBOLEN DEFINIËREN.....	25
4.6	PROGRAMMA INVOEREN.....	26
4.7	LADEN PLC PROGRAMMA	31
4.8	MONITOREN VAN HET PROGRAMMA	32
5.	TUTORIAL S7-GRAPH.....	34
5.1	PROJECT AANMAKEN	34
5.2	CONFIGURATIE EN BLOCKS UIT PLC HALEN	35
5.3	SYMBOLENLIJST OPSTELLEN	37
5.4	EEN BLOK VOORAF	37
5.5	BLOKKEN, STAPPEN EN OVERGANGEN	38
5.6	AANMAKEN FUNCTION BLOCK.....	39
5.7	STARTEN S7-GRAPH EDITOR	40
5.8	INVOEREN EERSTE STAP	41
5.9	INVOEREN VOLGENDE STAPPEN	42
5.10	INVOEREN OB1	44
5.11	LADEN PLC MET PROGRAMMA.....	45
5.12	VARIABELENTABEL	45
5.13	MONITOREN VAN HET PROGRAMMA EN VARIABELEN	47
5.14	EXTRA OPDRACHT	47
6.	TUTORIAL SCL.....	49
6.1	ANALOGE INGANG EN TEMPERATUUR METEN	49
6.2	PROJECT AANMAKEN	50
6.3	CONFIGURATIE PLC	50
6.4	SYMBOLENLIJST	53
6.5	SCL BRONBESTAND AANMAKEN	53

6.6	INVOEREN SCL-CODE.....	54
6.7	COMPILEREN SCL-CODE	56
6.8	TWEEDE SCL-BRONBESTAND.....	57
6.9	OB1 PROGRAMMA.....	58
6.10	LADEN PLC PROGRAMMA	59
6.11	MONITOREN TEMPERATUUR	59
7.	TUTORIAL WINCC FLEXIBLE	61
7.1	AANMAKEN NIEUW PROJECT	61
7.2	CONFIGURATIE UPLOADEN	62
7.3	HMI STATION PLAATSEN	64
7.4	MPI-NETWERK	65
7.5	HMI-STATION CONFIGUREREN	66
7.6	SYMBOLENLIJST	70
7.7	STARTEN WINCC FLEXIBLE.....	71
7.8	DEVICE SETTINGS	72
7.9	CYCLUSTIJD TOEVOEGEN.....	73
7.10	VERBINDING MET PLC MAKEN	74
7.11	TAGS TOEVOEGEN	74
7.12	SCHERM OPBOUWEN.....	76
7.13	STARTEN RUNTIME.....	81
7.14	AFSLUITEN RUNTIME.....	82
8.	TUTORIAL DATA BLOCK.....	83
8.1	AANMAKEN DATA BLOCK.....	83
8.2	EEN STUK JE SCL-CODE	85
8.3	SYMBOLEN, OB1 EN DOWNLOADEN	86
8.4	MONITORING	87
8.5	INSTANCE DATA BLOCK	88
9.	TIPS, TRICKS & TROUBLESHOOT	89
9.1	PROJECTEN ARCHIVEREN.....	89
9.2	PROJECTEN INLEZEN	91
9.3	PG/PC INTERFACE.....	93
9.3.1	<i>PC/PG interface via de USB adapter.....</i>	<i>93</i>
9.3.2	<i>PC/PG interface omzetten.....</i>	<i>95</i>
9.4	PLC DIAGNOSTIEK.....	96
9.5	ONLINE BLOKKEN IN DE PLC VERWIJDEREN	97
9.6	CLOCK MEMORY BYTE.....	98
10.	BIJLAGEN	99
	BIJLAGE 1 – LAD-PROGRAMMA VERKEERSLICHTSYSTEEM.....	99
	BIJLAGE 2 – GRAPH PROGRAMMA	102
	BIJLAGE 3 – PLC BLOCKS, INPUT- EN OUTPUTADRESSEN, MEMORY	104
	BIJLAGE 4 – ENIGE FORMULES VOOR TEMPERATUURMETING	105
	BIJLAGE 5 – SCL-CODE TEMPERATUURCONVERSIE EN MIN-MAX TESTEN	106
	BIJLAGE 6 – REFERENTIES	110

Voor suggesties en/of opmerkingen over deze handleiding kan je je wenden tot J. op den Brouw, kamer 411, of je kunt E-mail versturen naar: J.E.J.opdenBrouw@hhs.nl.

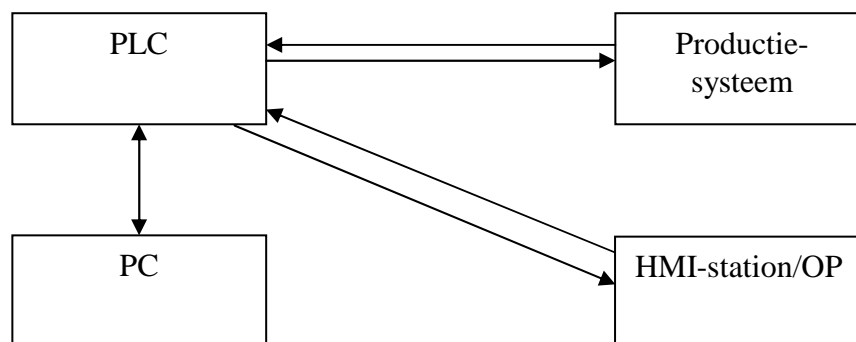
1. Inleiding

Het zal de meeste mensen niet opvallen, maar veel van onze producten worden gemaakt in fabrieken. We denken hierbij aan bijvoorbeeld mobiele telefoons, microprocessors maar ook colaflessen (en het vullen ervan).

Het vervaardigen van deze producten wordt, omdat er massaproductie nodig is, gedaan door middel van een productielijn, in de volksmond lopende band genoemd. Vroeger waren hier grote groepen arbeiders voor nodig, tegenwoordig worden veel van deze klussen gedaan door machines.

Deze machines moeten bestuurd worden. Het besturen wordt gedaan door Programmable Logic Controllers, afgekort tot PLC¹. Een PLC is te beschouwen als een computer met gespecialiseerde I/O. De computer, bestaande uit een microprocessor en geheugen, is te programmeren. De I/O wordt gebruikt om de machines aan te sturen en informatie betreffende de voortgang van het proces te vergaren.

De Academie voor Technology, Innovation and Society / Delft beschikt over een schaalmodel van een productiesysteem van Festo en PLC's van Siemens. Deze beide componenten worden verderop toegelicht. Zie figuur 1-1.



Figuur 1-1

Noot: de PC wordt voor meerdere doeleinden gebruikt.

¹ In de Verenigde Staten wordt veelal de term Programmable Controller gebruikt, afgekort tot PC. Dit geeft echter verwarring omdat een ander veel voorkomend apparaat, de Personal Computer, óók wordt afgekort tot PC.

Een korte introductie van de blokken:

- PLC (Programmable Logic Controller), de aanstuurder van (een deel van) het productiesysteem.,
- Productiesysteem, de te besturen productieomgeving, bv. van de firma Festo,
- PC, de computer waarop de software voor de PLC wordt ontwikkeld, tevens monitoring systeem.
- HMI-station/Operator Panel, een terminal waarop de engineer/operator het productieproces kan volgen.

Tussen de diverse blokken is communicatie mogelijk:

- PLC↔Productiesysteem, meetsignalen en stuursignalen (gezien vanuit de PLC).
- PLC↔PC, programmeerkabel, communicatie d.m.v. MPI of Ethernet. Deze kabel kan voor meer doeleinden gebruikt worden, maar in eerste instantie voor programmeren.
- PLC↔HMI-station/OP, communicatie d.m.v. MPI, Profibus en Profinet.

Deze handleiding begeleidt de cursist aan de hand van diverse tutorials door de veelzijdigheid van de STEP7-programmeeromgeving. De eerste tutorial laat een ladderprogramma zien voor een eenvoudig verkeerslichtsysteem. In de tweede tutorial wordt het programma opnieuw ingevoerd, maar nu in S7-Graph, tevens is hier een uitbreiding wat betreft functionaliteit van het programma. De derde tutorial betreft iets geheel anders; met behulp van een NTC-weerstand en een weerstand wordt een thermometer ontwikkeld. Het bijbehorende programma is in SCL geschreven. De vierde tutorial gaat over WinCC Flexible, het SCADA-pakket. De thermometer uit de vierde tutorial dient als basis. De laatste tutorial gaat over het gebruik van Data Blocks. Als laatste volgen enkele Tips, Tricks & Troubleshoot.

Hoe moet dit document gelezen worden:

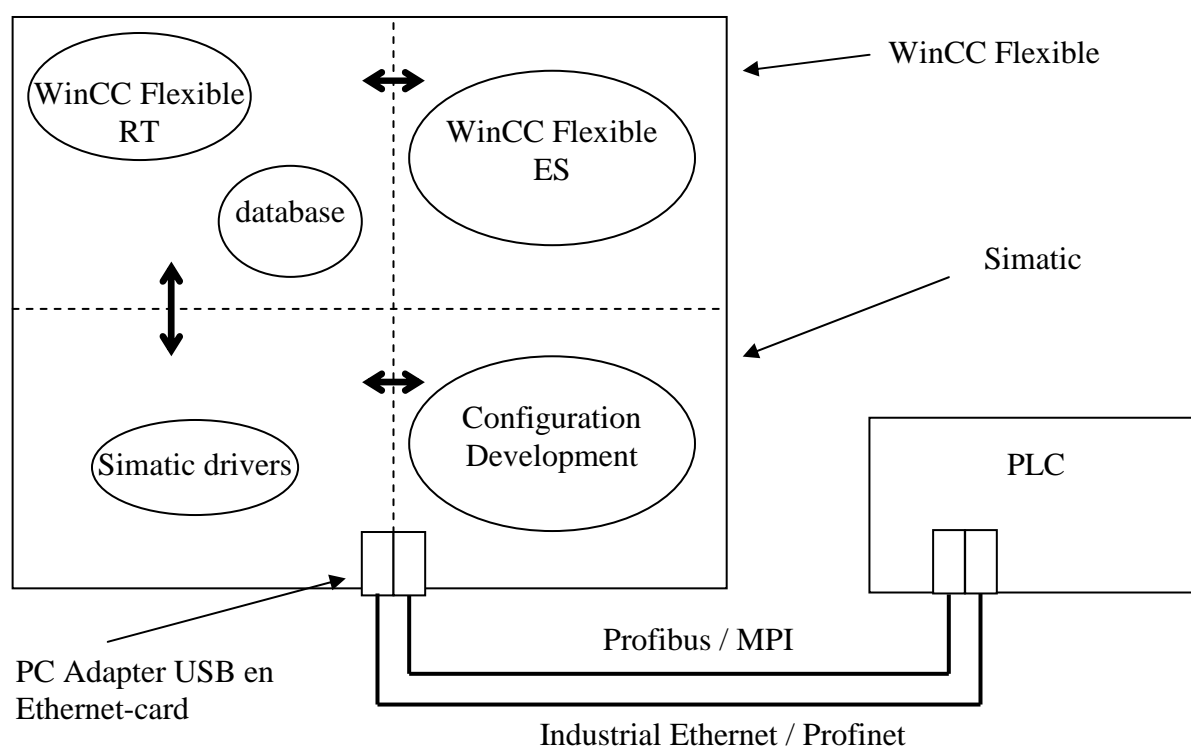
In de eerste twee tutorials wordt de lezer “aan de hand meegenomen”. Van bijna alles wat de lezer moet invullen of aanklikken is een *screenshot* weergegeven. Vanwege de layout zijn sommige screenshots na de daarwerkelijke tekst geplaatst, bijvoorbeeld op de volgende bladzijde. De lezer kan het best een alinea per keer lezen om een beeld te krijgen van de te verrichten handelingen.

Noot betreffende de plaats waar de projectbestanden moeten worden opgeslagen:

Maak geen gebruik van een USB-stick tijdens het werken met de Simatic software. Er worden zeer veel bestanden aangemaakt en er is veel verkeer tussen harddisk en de programmatuur. Het gebruik van de H:-schijf kan wel (is niet echt onderzocht). Gebruik de D:-schijf en archiveer naar de H:-schijf of USB-stick. Archiveren wordt besproken in hoofdstuk 9.

2. De PC

De PC is een gewone IBM compatible computer en heeft in dit project meerdere functies: het is een ontwikkelstation om de PLC te configureren en programmeren, het is een ontwikkelstation voor de SCADA software en het is een monitoring station voor SCADA. Dit is in figuur 2-1 schematisch weergegeven.



Figuur 2-1

Aan de PC is een PC Adapter USB van Siemens gekoppeld. Hiervoor zijn drivers geïnstalleerd. De PC is met de PLC verbonden via MPI, een Siemens-eigen bus en protocol. Deze interface kan omgeschakeld worden naar Profibus, een gestandaardiseerd bussysteem dat gebruikt wordt in de automatiseringstechniek. Beide maken gebruik van het RS-485 protocol. De PC is ook uitgerust met een Ethernetkaart. Communicatie tussen de PLC en PC, bv als HMI-station, kan ook via de Ethernetkaart. Op de PC is de volgende software geïnstalleerd.

Windows XP SP2

Eén van de Operating Systems van Microsoft, aangevuld met de laatste Service Packs.

Simatic STEP 7 5.4 Professional 2006

Dit is de software van Siemens waarmee het PLC-programma ontwikkeld wordt en waarmee dat programma in de PLC geladen wordt. Dit laden gebeurt via de MPI-bus. De professional versie heeft naast de vertrouwde programmeertalen LAD, FDB en STL ook Graph en SCL.

WinCC Flexible 2005 Engineering Station

Deze software van Siemens wordt gebruikt voor procesvisualisatie (HMI = Human Machine Interface) en toezichhouderscontrole (SCADA = Supervisory Control And Data Acquisition). WinCC Flexible ES is het deel van het totale pakket waarmee een *runtime* ontworpen wordt. Hier kan je dus knoppen, I/O-velden, etc. aanmaken en acties koppelen aan die knoppen en I/O velden. Is een runtime ontworpen, dan wordt het gecompileerd en kan het draaien in de WinCC Flexible Runtime (RT).

WinCC Flexible 2005 Runtime

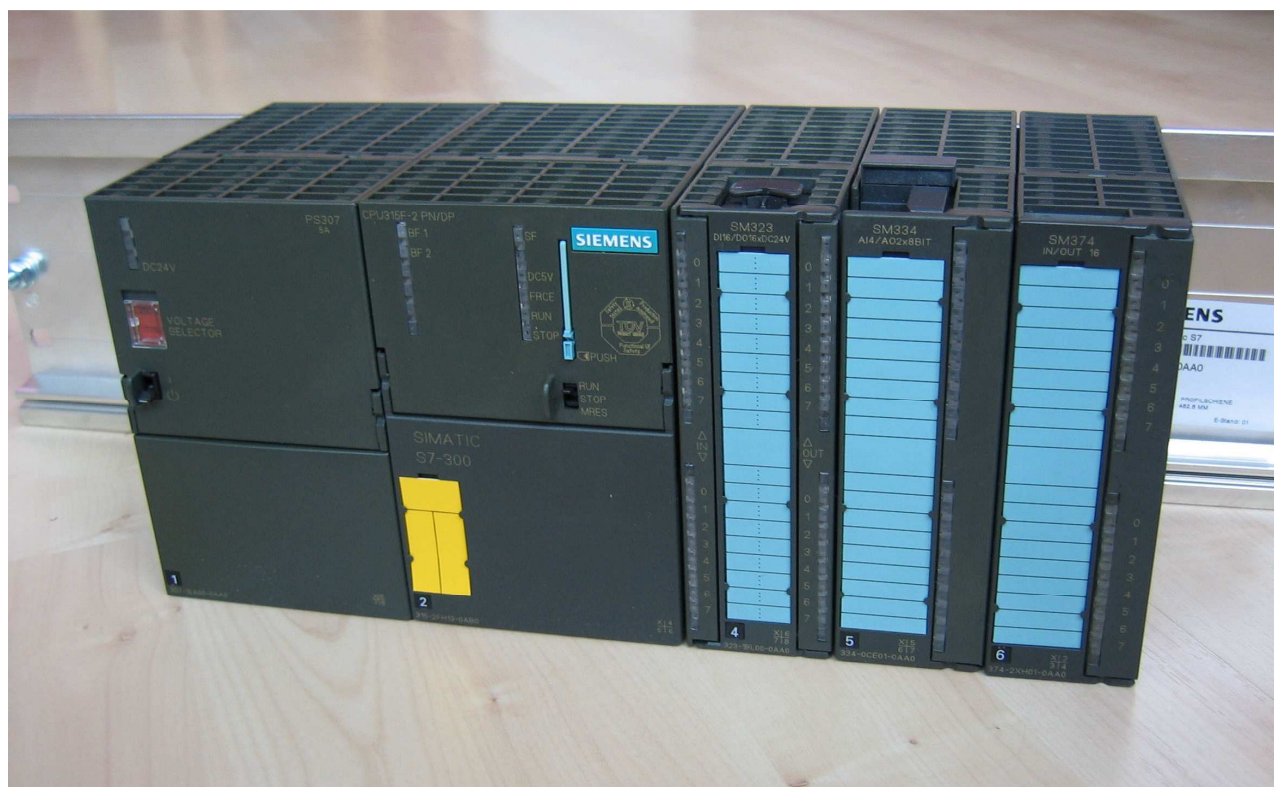
WinCC Flexible Runtime is de component die de gecompileerde code uitvoert en daadwerkelijk interactie met de PLC vertoont. Het zal dus gegevens uit de PLC opvragen en bijvoorbeeld in een trenddatabase loggen. Zo kan de gebruiker informatie over langere tijd te vergaren en opslaan om bijvoorbeeld historische trends te analyseren, zodat het productieproces verbeterd kan worden. De PC is te gebruiken als HMI-station.

3. De PLC

Het aansturen van fabrieksmachines wordt gedaan met een Programmable Logic Controller, afgekort tot PLC. We maken gebruik van één uit de familie van de S7-300 serie van Siemens.

3.1 Beschrijving PLC

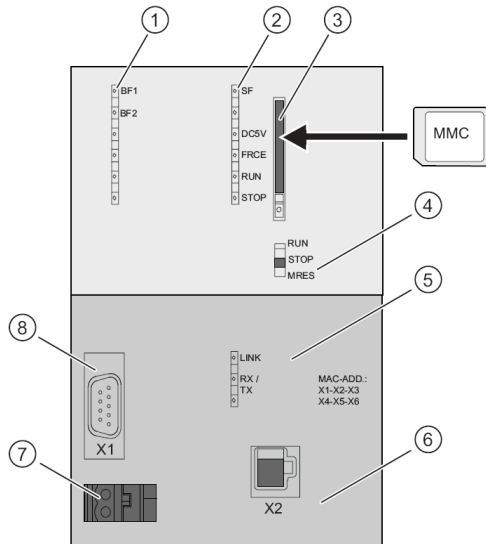
Het exemplaar dat tijdens dit blok wordt gebruikt is de CPU315F-2 PN/DP, gecombineerd met voeding, digitale en analoge I/O-module en een simulatiemodule. Een foto van de opstelling is te zien in figuur 3-1.



Figuur 3-1

3.1.1 Processormodule CPU315F-2 PN/DP

Dit is het hart van het systeem. Op de foto is deze module in het middel gesitueerd. Het bevat een CPU-eenheid en een FLASH-opslagkaart. De CPU kan worden geprogrammeerd door middel van STEP7 software. De module heeft naast een MPI/Profibus-aansluiting ook een Ethernet aansluiting voor Industrial Ethernet of Profinet.



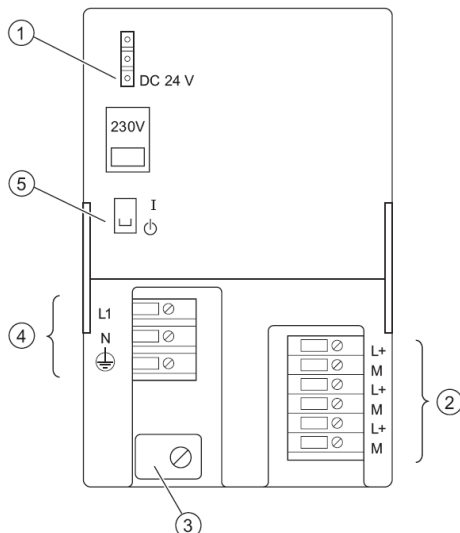
Figuur 3-2

De volgende onderdelen zijn (figuur 3-2):

1. Bus Fail; statusleds betreffende de bussen.
2. Diverse statusleds, oa. System Fail, Run en Stop mode.
3. Het MMC-slot. Hierin wordt de MMC-kaart geplaatst. Zonder deze kaart werkt de PLC niet.
4. Operation Switch. Zet de PLC in RUN of STOP.
5. Leds van de Ethernet Link.
6. De Ethernet/Profinet-aansluiting.
7. Voedingsspanning aansluiting (+24V)
8. De MPI/Profibus-aansluiting

3.1.2 Voedingsmodule PS307

Het geheel wordt gevoed met een aparte voeding, links op de foto. De voeding kan 5A leveren en is krachtig genoeg om de CPU en de I/O-modules aan te sturen.



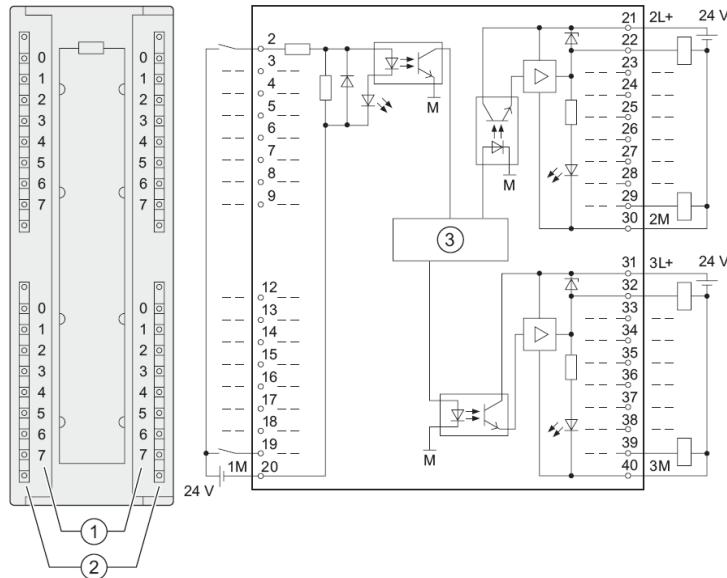
Figuur 3-3

De volgende onderdelen zijn (figuur 3-3):

1. “24 Volt aanwezig”-lamp.
2. Aansluitingen voor 24V DC.
3. Trekontlasting.
4. Spanningaansluiting 230V AC en aarde.
5. 24 V DC aan-uit schakelaar.

3.1.3 Digitale I/O module SM323

Deze module, rechts naast de CPU-module, heeft 16 digitale ingangen en uitgangen. De module werkt op 24V gelijkspanning. Voor zowel de ingangen als de uitgangen geldt: een logische '0' komt overeen met 0 V, een logische '1' komt overeen met 24 V. De uitgangen kunnen maximaal 0,5 A leveren.



De onderdelen zijn (figuur 3-4):

1. Kanaalnummer.
2. Statusled (groen indien logische '1').
3. Backplane

Figuur 3-4

Links op figuur 3-4 is de voorkant van de module afgebeeld. Links zijn de ingangen, rechts de uitgangen gesitueerd.

3.1.4 Analoge I/O module SM334

Deze module staat rechts naast de digitale I/O-module op de foto. Het heeft 4 analoge ingangen en 2 analoge uitgangen. De ingangen kunnen d.m.v spanning of stroom worden aangestuurd. De uitgangen kunnen een spanning of een stroom leveren. Het spanningsbereik ligt tussen 0 en 10 V, het stroombereik tussen 0 en 20 mA. Aangezien de PLC verder een digitaal systeem is, moeten de analoge waarden worden geconverteerd. Er wordt gebruik gemaakt van 8 bits resolutie voor zowel ingangen als uitgangen.

De digitale waarden bestaan allemaal uit 16 bit, ook al worden maar 8 bits effectief gebruikt. De analoge spanningen zijn allemaal positief, het tekenbit is dan ook 0. Daarna volgen de 8 bits. De overige bits zijn allemaal 0. Zie ook tabel 3-1.

Bitnummer	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Waarde	0	v	v	v	v	v	v	v	v	0	0	0	0	0	0	0

Tabel 3-1

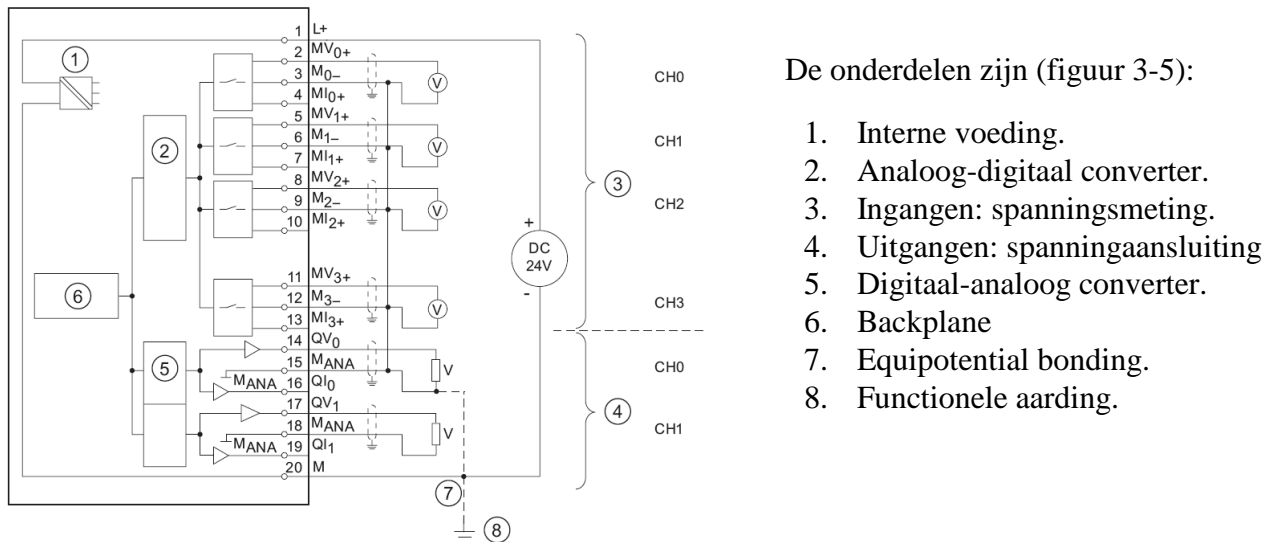
Voor zowel spanning als stroom geldt een "rated range", het bereik waarin normaal gewerkt wordt. Mocht een spanning of stroom daarboven komen, dan wordt over de "overshoot range" gesproken. Bij een nog hogere spanning of stroom wordt gesproken over "overflow". Zie ook tabel 3-2.

Decimaal	Hexadecimaal	Spanning	Stroom	Opmerking
32767	7FFF	11,852 V*	23,70 mA*	Overflow*
32512	7F00			
32511	7EFF	11,759 V	23,52 mA	Overshoot
27649	6C01			
27648	6C00	10,000 V	20,00 mA	Rated range
20736	5100	7,500 V	15,00 mA	
128	80	46,296 mV	92,59 uA	
0	0	0V	0 mA	

* zie tekst

Tabel 3-2

Indien een uitgang in de overflow range wordt gestuurd, levert de uitgang een spanning van 0V of een stroom van 0 mA. Indien een ingang in de overflow range wordt gestuurd levert dit een waarde van 7FFF op.



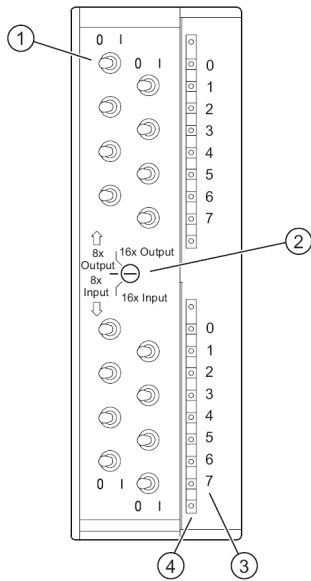
Figuur 3-5

3.1.5 Simulatie I/O module SM 374

Geheel rechts op de foto is een I/O-module te zien, waarmee digitale ingangen en uitgangen gesimuleerd kunnen worden. Nu is simuleren niet helemaal het juiste woord, het zijn wel gewoon in- en uitgangen. Alleen zijn de ingangen te bedienen door een gebruiker die hiermee de werking van een machine kan simuleren. De module is te gebruiken als:

- 16x ingang via schakelaars,
- 16x uitgang via leds,
- 8x ingang en 8x uitgang gecombineerd.

In deze laatste modus wordt de module gebruikt tijdens de tutorials.



Figuur 3-6

De onderdelen in figuur 3-6 zijn:

1. Schakelaars.
2. Modus-selector.
3. Kanaalnummer.
4. Leds

Tijdens de tutorials wordt de 8x input / 8x output modus gebruikt. De selector moet daarvoor in het midden staan. Het bovenste gedeelte werkt als 8x output; de schakelaars hebben geen functie. Het onderste gedeelte werkt als 8x input; de leds geven de stand van de schakelaars weer.

3.2 Programmeertalen

De PLC is ontstaan in de energie- en automatiseringstechniek en niet bij de “echte” computerhoek. De programmeurs hadden geen opleiding tot computerprogrammeurs gevolgd. Hierdoor ontbreekt een echte programmeertaal. In de loop der jaren zijn diverse methoden van programmeren ontworpen. Noot: hieronder volgt slechts een korte beschrijving. Voor meer informatie, zie het boek en sheets.

De internationale norm IEC 61131-3² beschrijft een aantal programmeertalen. Hieronder een overzicht. In de opsomming wordt eerste de IEC naam gegeven en tussen de haakjes de IEC afkorting en de afkorting die Siemens gebruikt.

Ladder Diagram (LD, LAD)

Deze taal bestaat uit een set van symbolische instructies die op een grafische wijze worden gepresenteerd. Er zijn vijf categorieën van instructies: relais³, timer/counter, rekenkundig, data manipulatie en programmabesturing. Deze symbolen kunnen zo worden opgesteld dat de gewenste werking in het geheugen wordt ingevoerd. Ladderdiagrammen worden gebruikt voor het besturen van (kontakt-)uitgangen gebaseerd op ingangscondities.

Function Block Diagram (FBD, FBD)

Dit is een grafische taal waarbij de logische functies als blokken worden weergegeven. Naast de bekende AND en OR zijn er ook blokken voor teller, timers enz. Het is een alternatief voor ladderdiagrammen.

Instruction List (IL, STL)

Dit is het best te omschrijven als een assemblertaal voor PLC's. Diverse instructies zoals L (load) en T (output) zijn voor handen. Hierin is het mogelijk om zeer nauwkeurig de werking van een

² Zie <http://www.plcopen.org>

³ Vroeger waren de uitgangen van relais voorzien. Tegenwoordig worden de uitgangen ook met transistoren of thyristoren uitgevoerd. Men heeft echter de oude terminologie aangehouden.

programma te beschrijven omdat de instructies één op één worden vertaald naar machinetaal. STL staat voor Statement List. STL is niet conform de IEC-norm.

Structured Text (ST, SCL)

Deze taal lijkt erg op Pascal. Er kan geprogrammeerd worden als een echte programmeertaal, compleet met IF's, functieaanroepen, verschillende datatypes etc. SCL staat voor Structured Control Language. SCL is conform de IEC norm, en heeft enkele uitbreidingen.

Sequential Function Chart (SFC, Graph)

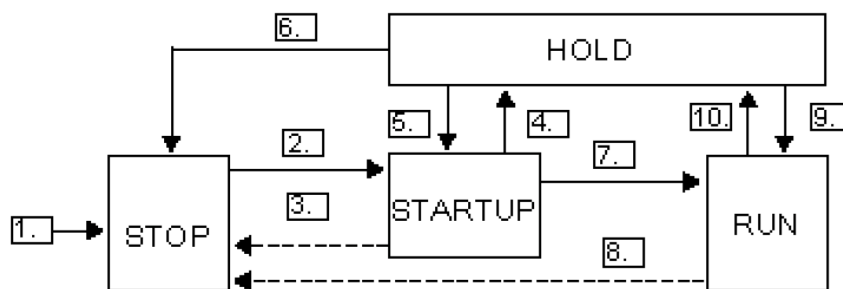
Ook dit is weer een grafische taal. Het is te vergelijken met het beschrijven van een toestandsmachine uit de digitale techniek; de condities van de overgangen worden beschreven d.m.v. LD's of FDB's. Aftakkingen (Branch) en parallelisme (Simultaneous Branch) zijn mogelijk. Siemens heeft een variant, S7 Graph genaamd, die voldoet aan de IEC 61131-3 norm. Daarvoor moet wel een optie worden geactiveerd.

Siemens levert een softwareapplicatie, genaamd STEP7 Professional, waarin al deze programmeertalen zijn opgenomen.

3.3 Gebruik PLC

De PLC is een robuust apparaat. Dat neemt niet weg dat het gebruik zorgvuldig dient te geschieden. De PLC kan tijdens de tutorials en werkzaamheden gewoon onder spanning staan. Wel is het raadzaam om de PLC in STOP te zetten als er wijzigingen in het geheugen worden gedaan. Hierbij valt te denken aan configuratie-aanpassing, nieuwe blokken verzenden.

Hieronder staan de operating modes van de PLC afgebeeld. Zie figuur 3-7.



Figuur 3-7

1.	Na aanzetten komt de CPU in STOP.	2.	De CPU gaat naar STARTUP als de schakelaar in RUN wordt gezet.
3.	De CPU komt in STOP als de schakelaar in STOP wordt gezet of de PLC een fout detecteert.	4.	Hold mode (wordt niet besproken).
5.	Hold mode (wordt niet besproken).	6.	Hold mode (wordt niet besproken).
7.	De CPU komt in RUN als alles goed is gegaan.	8.	De CPU komt in STOP als de schakelaar in STOP wordt gezet of de PLC een fout detecteert.
9.	Hold mode (wordt niet besproken).	10.	Hold mode (wordt niet besproken).

4. Tutorial STEP7 en LAD

In deze tutorial wordt uitgelegd hoe een project moet worden aangemaakt. Daarin wordt een eenvoudig verkeerslichtsysteem ingevoerd. Eerst wordt een nieuw project aangemaakt, dan worden de netwerk- en hardwareconfiguratie van de PLC opgesteld. Daarna wordt de symbolenlijst aangemaakt zodat het programmeren wat eenvoudiger wordt. Vervolgens wordt het programma ingevoerd en als laatste wordt het programma online gevolgd (*monitoren*).

STEP7 is een groot programma met zeer veel mogelijkheden. In deze tutorial wordt slechts een deel van de mogelijkheden getoond.

4.1 Project aanmaken

Klik op het pictogram Simatic Manager. Zie figuur 4-1. De manager wordt gestart.



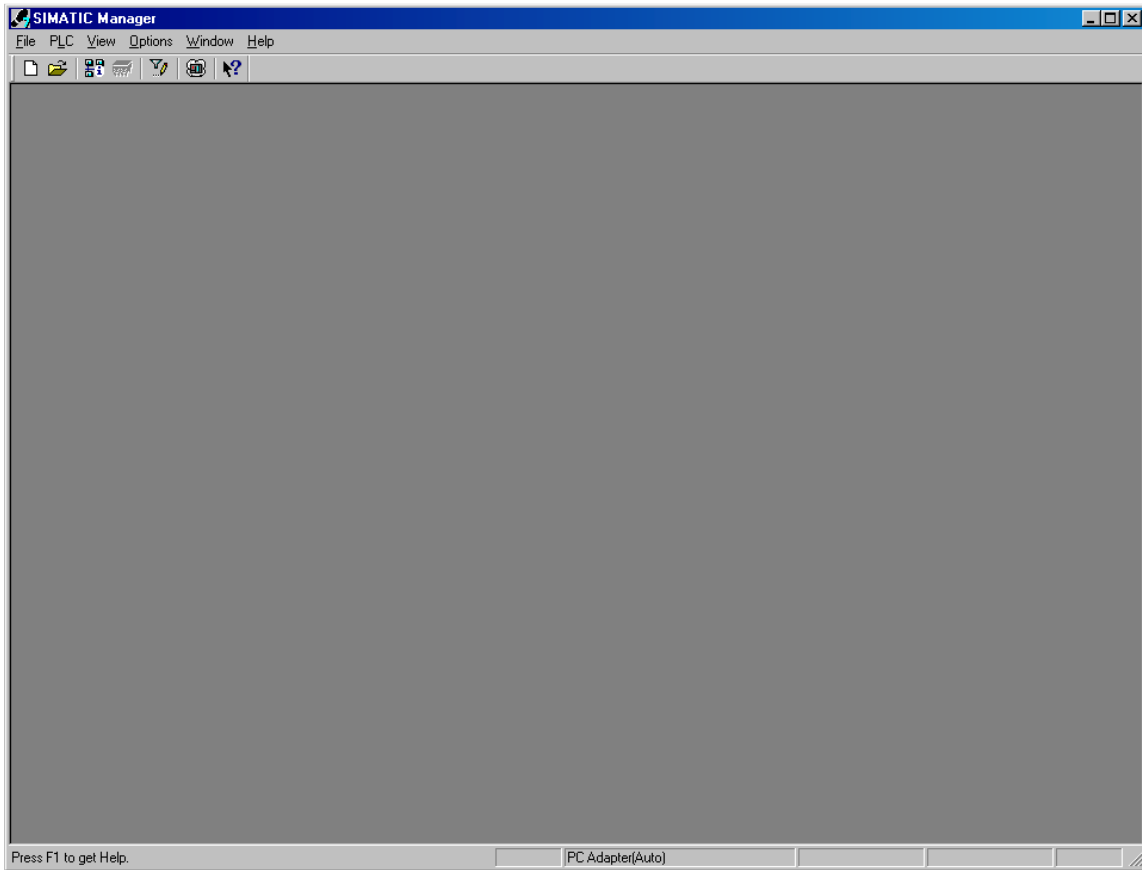
Figuur 4-1

Direct hierna kan de Project Wizard verschijnen, maar dat hoeft niet. Deze wordt niet gebruikt, dus klik op **Cancel**. Zie figuur 4-2.



Figuur 4-2

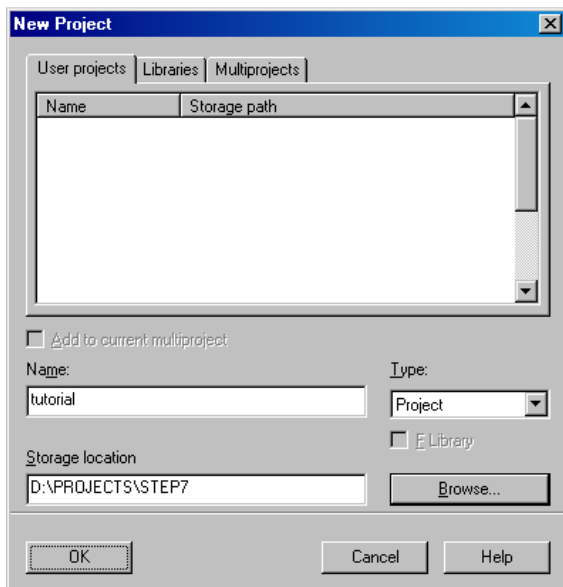
Nu verschijnt de Simatic Manager die verder leeg is. Zie figuur 4-3.



Figuur 4-3

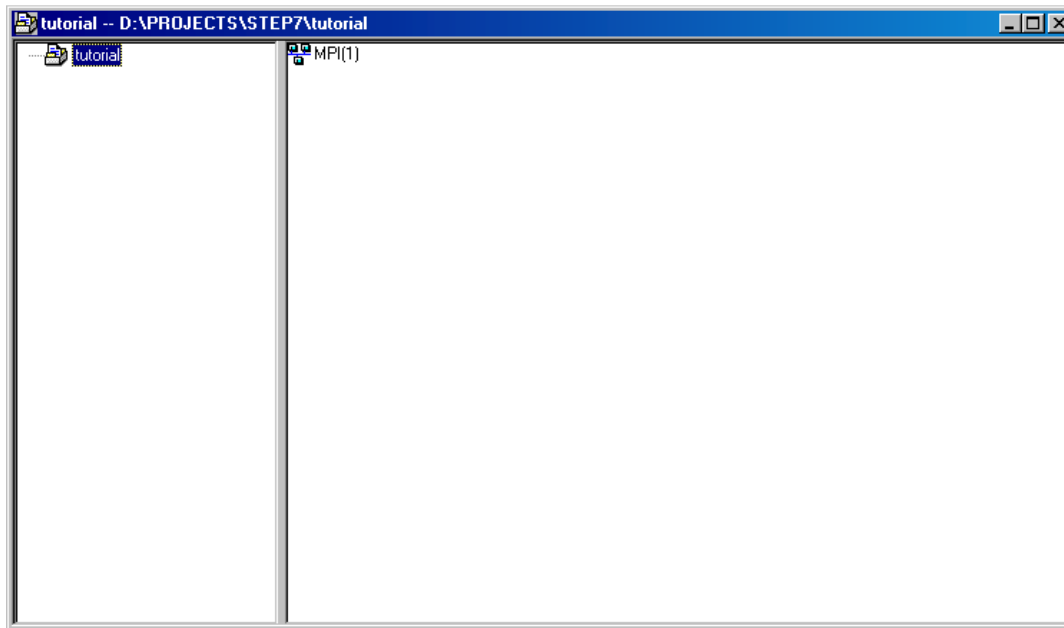
Klik nu op **File**→**New** om een nieuw project te starten. Er wordt een venster geopend zoals in figuur 4-4. Vul de projectnaam **tutorial** in, vul eventueel de map in, en zorg dat de rest overeenkomt met de figuur. Klik op **OK**.

Noot: gebruik geen USB-stick tijdens deze tutorial, later kan het natuurlijk wel dienen als opslag.



Figuur 4-4

Er verschijnt een vrijwel leeg projectscherm (figuur 4-5).



Figuur 4-5

4.2 Netwerk configureren ⁴

Eerst moet nu het netwerk waarin de PLC zich bevindt worden ingevoerd. Dubbelklik op de naam MPI (1), of klik op het pictogram **Configure Network** (figuur 4-6).



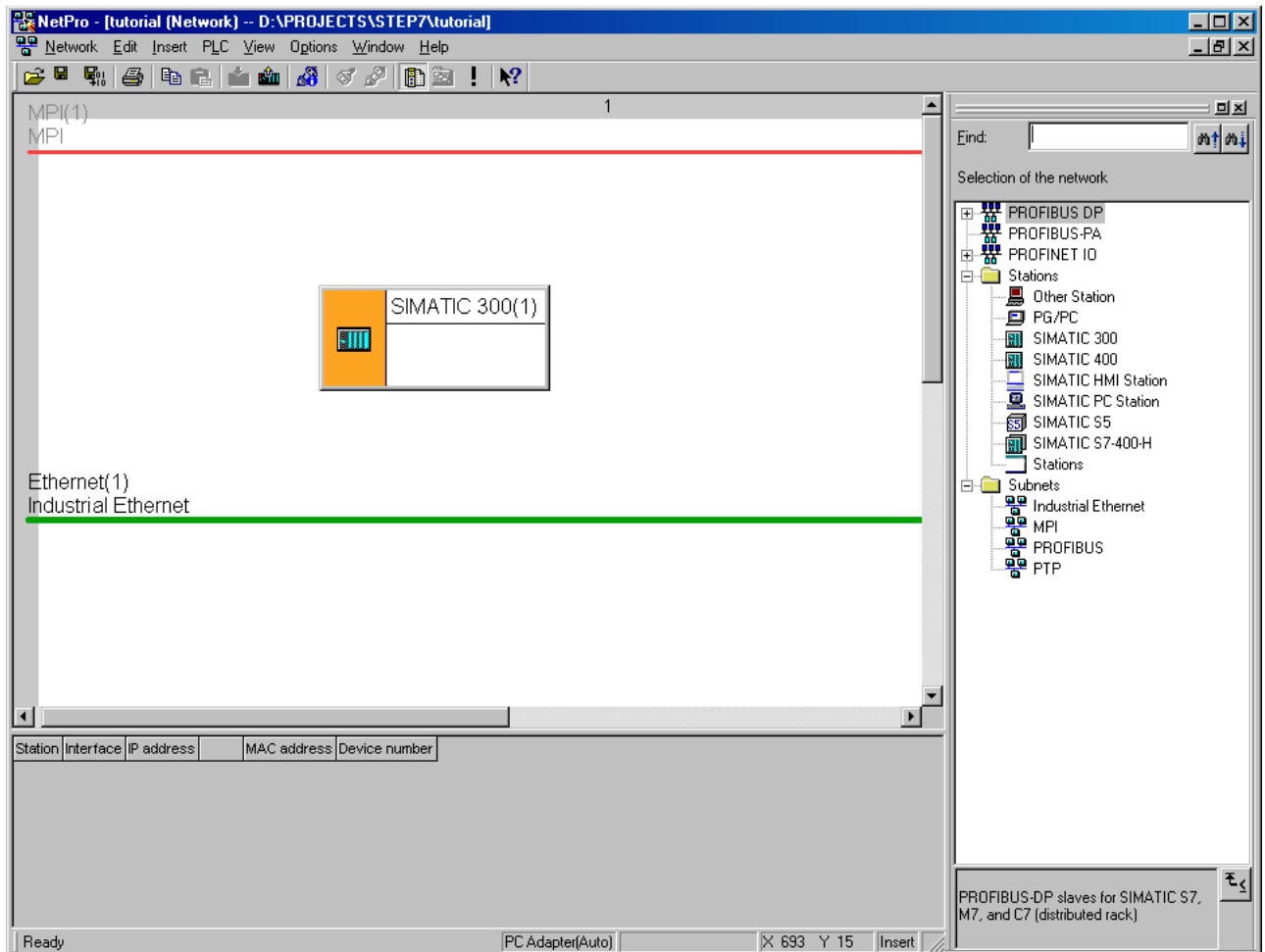
Figuur 4-6

Het programma `Netpro` wordt gestart met daarin een oranje-rode lijn die de MPI-bus voorstelt. Aan de rechterkant van `Netpro` bevinden zich de onderdelen die aan het netwerk kunnen worden toegevoegd. Aangezien de PLC ook een Ethernet-aansluiting heeft, moet er een Industrial Ethernet worden toegevoegd. Dit is de groene lijn in figuur 4-7. Klik rechts op `Subnets`→`Industrial Ethernet` en sleep dit onder de MPI-bus.

Daarnaast moet ook een S7-300 PLC worden toegevoegd. Voor `Netpro` is alleen van belang dat het type wordt opgegeven; de exacte configuratie van de PLC wordt m.b.v een ander programma ingevoerd. Voeg een `SIMATIC 300` toe zoals is weergegeven in figuur 4-7.

Noot: het programma plaatst de onderdelen links op het scherm, deze kunnen niet zichtbaar zijn. Met de scroll-bars kan dan worden gezocht naar de geplaatste onderdelen.

⁴ Hoewel dit een eenvoudige setup is, waarin deze stap kan worden overgeslagen, wordt toch het netwerk geconfigureerd. Bij grote omgevingen is deze stap wel nodig.

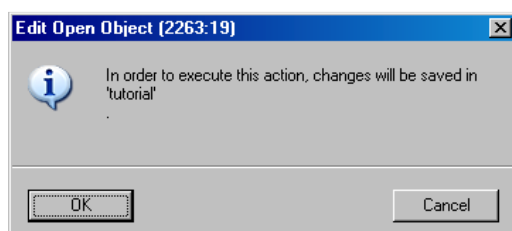


Figuur 4-7

4.3 PLC configureren

De PLC moet nu geconfigureerd worden. Hierbij wordt opgegeven waar de PLC uit bestaat en hoe de adressering van de ingangen en uitgangen in elkaar zit.

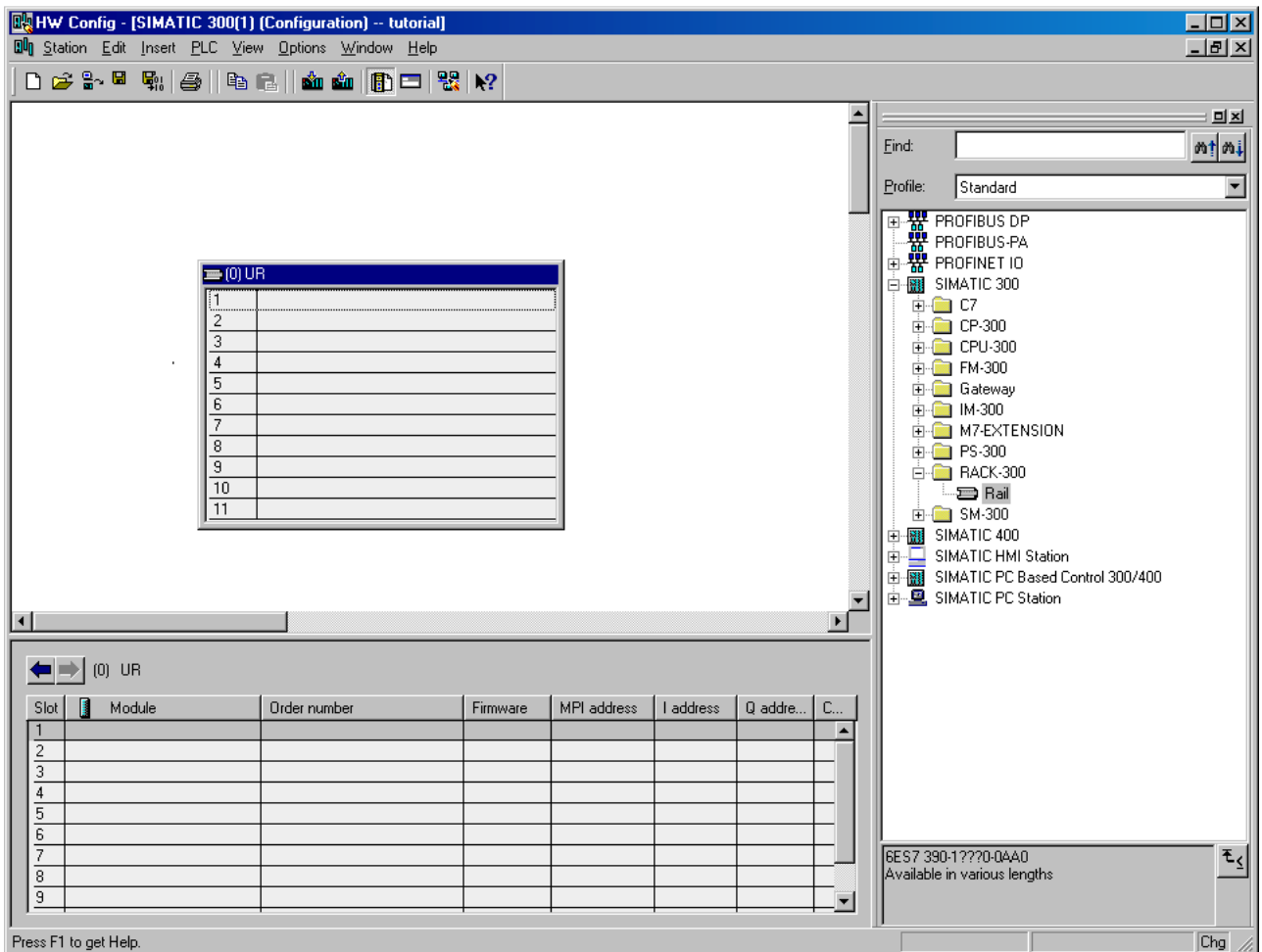
Dubbelklik op het onderdeel SIMATIC 300(1) uit figuur 4-7⁵. Er volgt nu een dialoog waarin wordt aangegeven dat het netwerk wordt opgeslagen. Klik op **OK**. Zie ook figuur 4-8.



Figuur 4-8

⁵ Dit kan ook door eerst Netpro af te sluiten en via de Simatic Manager de hardware van de PLC te configureren.

Het hardware-configuratieprogramma HWConfig wordt gestart. Als eerste moet een rack worden toegevoegd. Dit is te kiezen in het rechterscherm via SIMATIC 300→RACK-300→Rail. Het resultaat is te zien in figuur 4-9.



Figuur 4-9

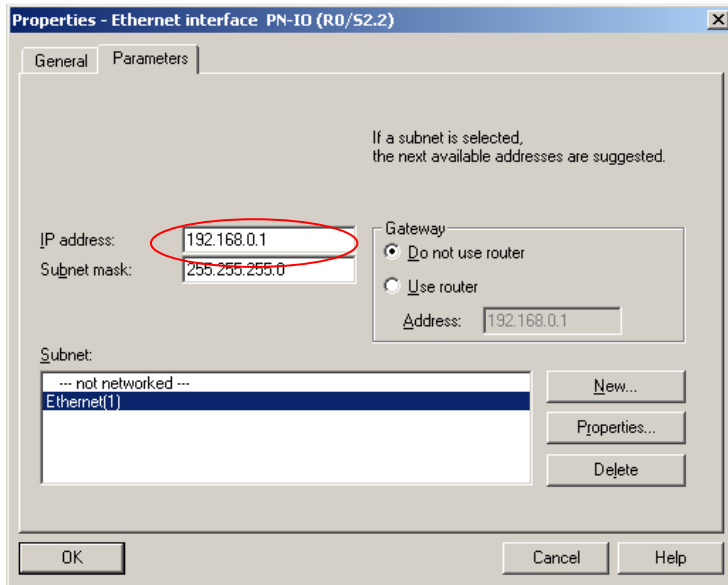
Nu moeten de voeding, de CPU-module, de digitale I/O, analoge I/O en de simulatiemodule worden toegevoegd.

Selecteer via SIMATIC 300→PS-300→PS 307 5A de voeding PS 307 5A en plaats deze in regel 1 (slot 1). Selecteer daarna de CPU 315F-2 PN/DP eenheid en sleep die in slot 2. Alleen in deze slot mag de CPU voorkomen. Het ordernummer is 6ES7 315-2FH13-0AB0, het versienummer is 2.5.

Let op het juiste Siemens ordernummer en versienummer!

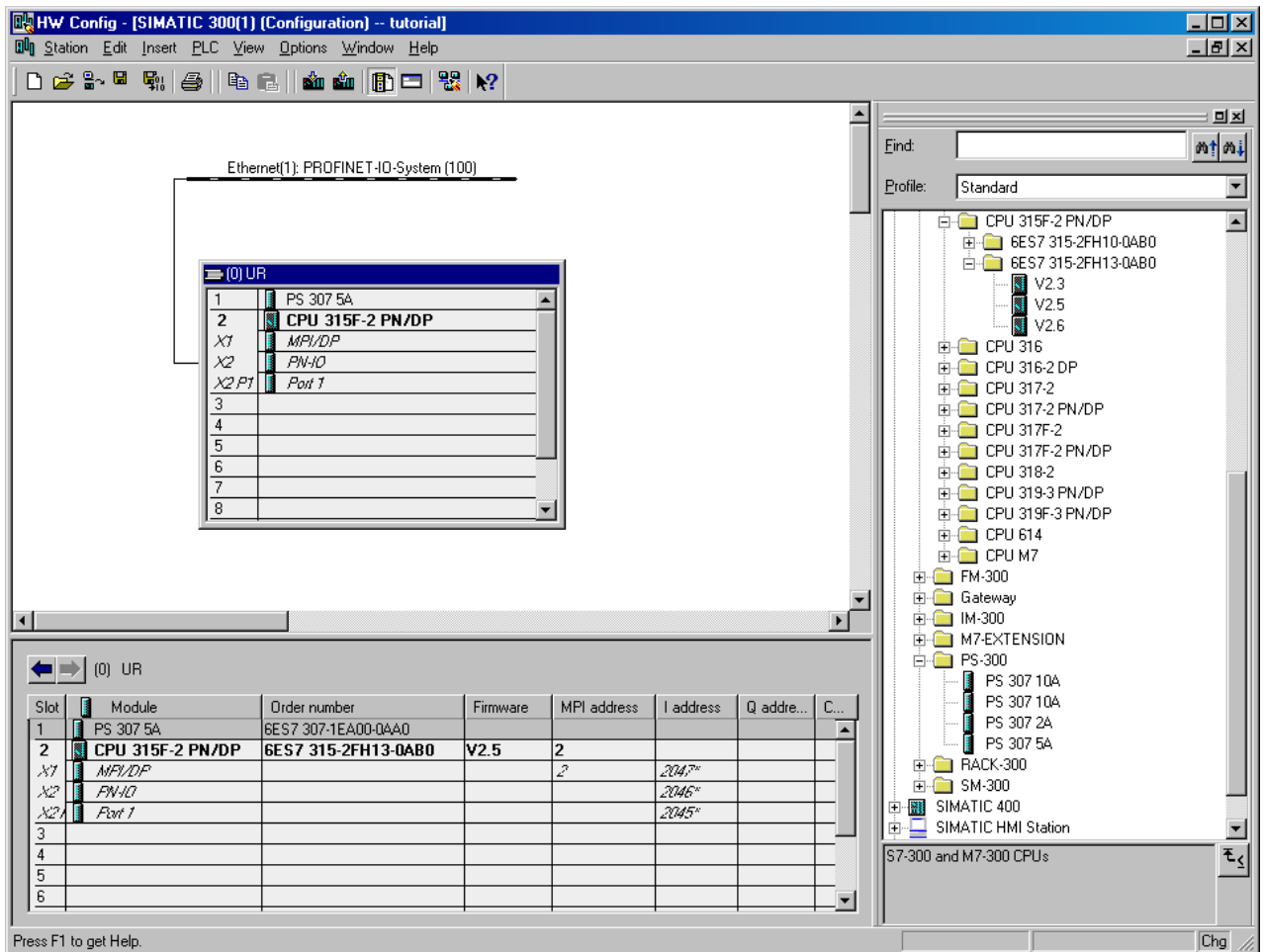
Voordat de CPU in slot 2 zichtbaar is, moeten eerst enige gegevens worden ingevuld voor het IP-adres en het MPI-adres. Hiervoor worden nieuwe invoerdialogen gestart. Zie figuur 4-10.

Vul het IP-adres in zoals is aangegeven in figuur 4-10. Selecteer ook Ethernet (1) zodat de CPU aan dit netwerk verbonden wordt. Klik daarna op OK.



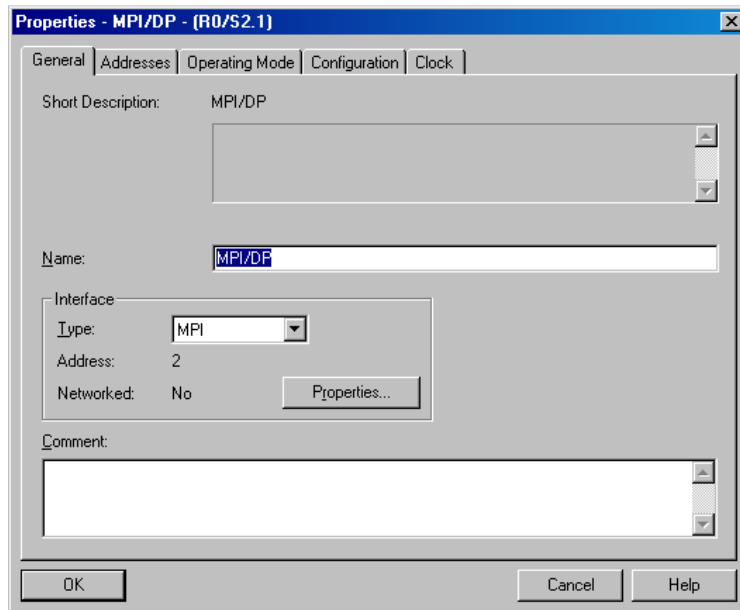
Figuur 4-10

In figuur 4-11 is te zien dat de CPU-module geplaatst is, evenals de Profinet-verbinding.



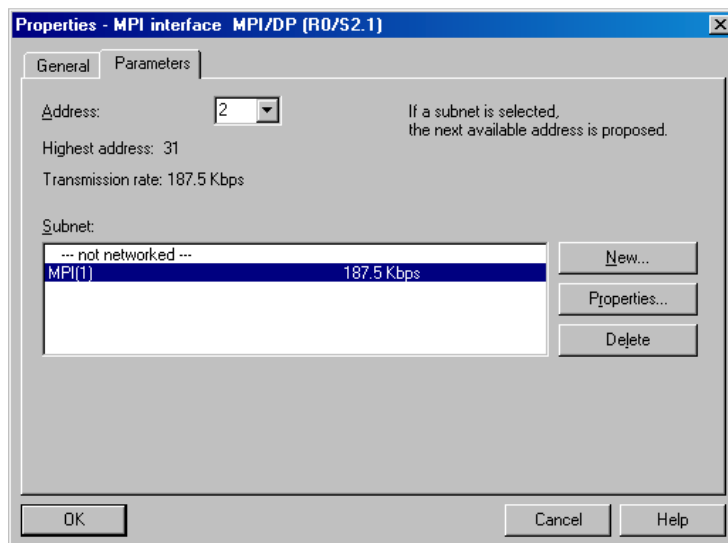
Figuur 4-11

Nu moet ook nog de MPI-interface gekoppeld worden aan de MPI-bus. Dubbelklik in slot 2 onder onderdeel X1 op MPI /DP (zie figuur 4-10). Nu volgt een dialoogvenster als figuur 4-12. Zorg ervoor dat de type op MPI staat. Klik daarna op **Properties**.



Figuur 4-12

Er volgt een nieuw venster zoals figuur 4-13. Kies 2 als adres en selecteer MPI (1) als netwerk.



Figuur 4-13

Klik daarna op **OK**. Nu wordt het dialoog uit figuur 4-12 weer zichtbaar. Klik hier ook weer **OK**.

Als laatste moeten de drie modules worden toegevoegd. Selecteer uit het rechterpaneel via SIMATIC 300→SM-300 deze modules.

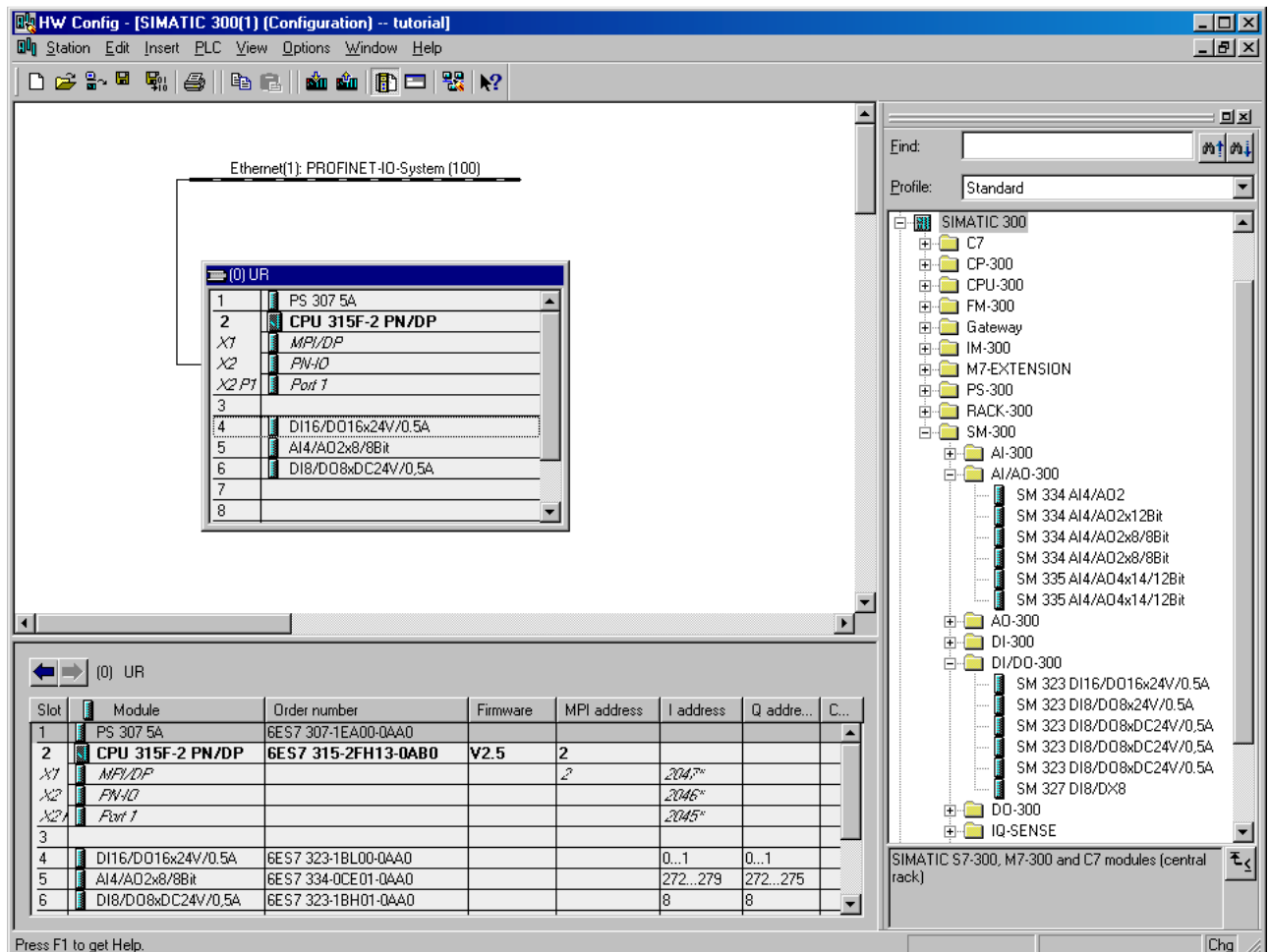
Selecteer de juiste modules! Let goed op de simulatiemodule!

Het ordernummer voor de digitale I/O-module is 6ES7 323-1BL00-0AA0 (slot 4).

Het ordernummer voor de analoge I/O-module is 6ES7 334-0CE01-0AA0 (slot 5).

Het ordernummer voor de simulatiemodule is 6ES7 323-1BH01-0AA0 (slot 6).

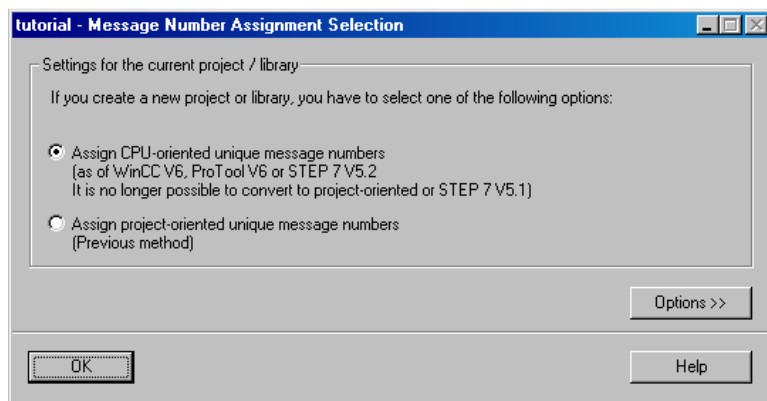
Zie het één en ander in figuur 4-14.



Figuur 4-14

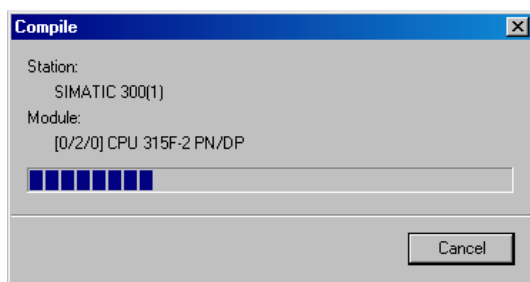
De configuratie is nu klaar en kan in de PLC worden geladen. Kies nu via het menu **Station**→**Save and Compile**.

Eerst wordt nog gevraagd hoe de CPU-berichten moeten worden opgesteld. Daar wordt geen gebruik van gemaakt. Kies de eerste mogelijkheid en klik dan op **OK**. Dit dialoog verschijnt éénmalig. Zie figuur 4-15.



Figuur 4-15

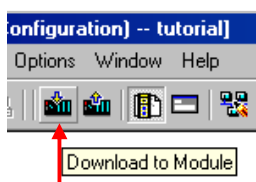
Er volgt een venster waarin de voortgang wordt weergegeven. Zie figuur 4-16.



Figuur 4-16

4.4 Configuratie downloaden

Zet de PLC nu eerst in STOP door de operating mode schakelaar van RUN naar STOP te zetten. Nu kan de configuratie naar de PLC worden gestuurd via het menu **PLC**→**Download** of via het pictogram in figuur 4-17.



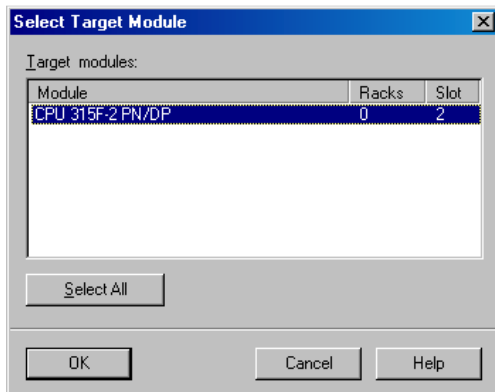
Figuur 4-17

Selecteer in het volgende venster welk *target module* moet worden geladen. Er is er maar één, dus dat is eenvoudig. Klik op **OK**. Zie figuur 4-18.

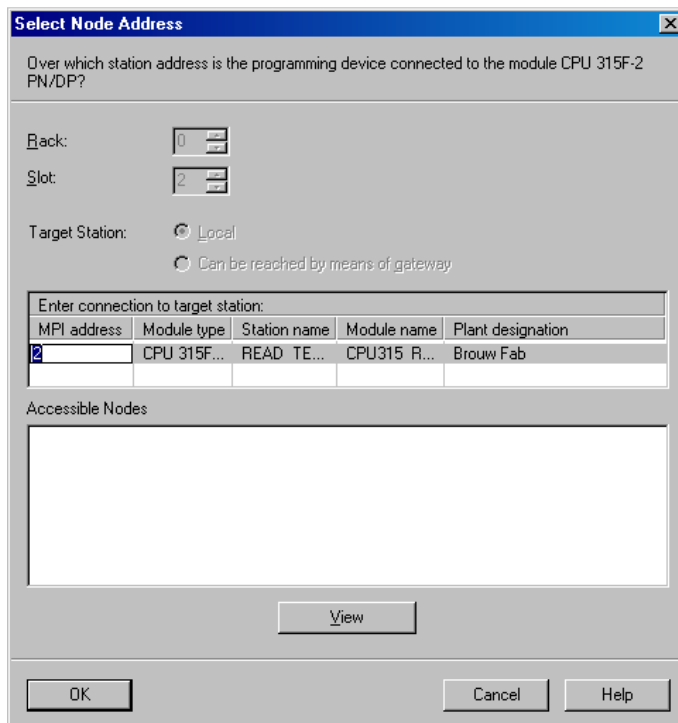
Daarna volgt een dialoog waarin het MPI-adres van de target module kan worden geselecteerd. Adres 2 is al geselecteerd. Klik op **OK** (figuur 4-19).

Figuur 4-20 laat zien wat er gebeurt als de CPU-module die geconfigureerd is, anders is dan die geprogrammeerd is. De *online* en *offline*-versie verschillen. In deze figuur zijn onder andere de station- en modulenaam anders. Zorg er in ieder geval voor dat het ordernummer en

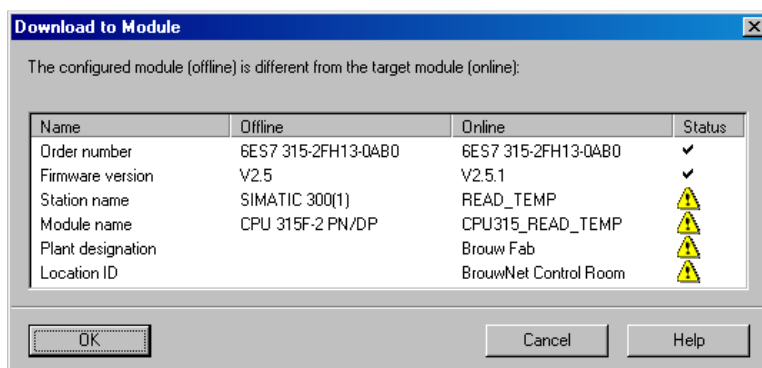
versienummer van de CPU correct is. Anders kan de CPU niet functioneren. Deze figuur hoeft niet te verschijnen.



Figuur 4-18

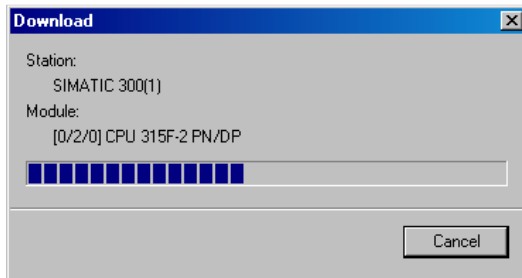


Figuur 4-19



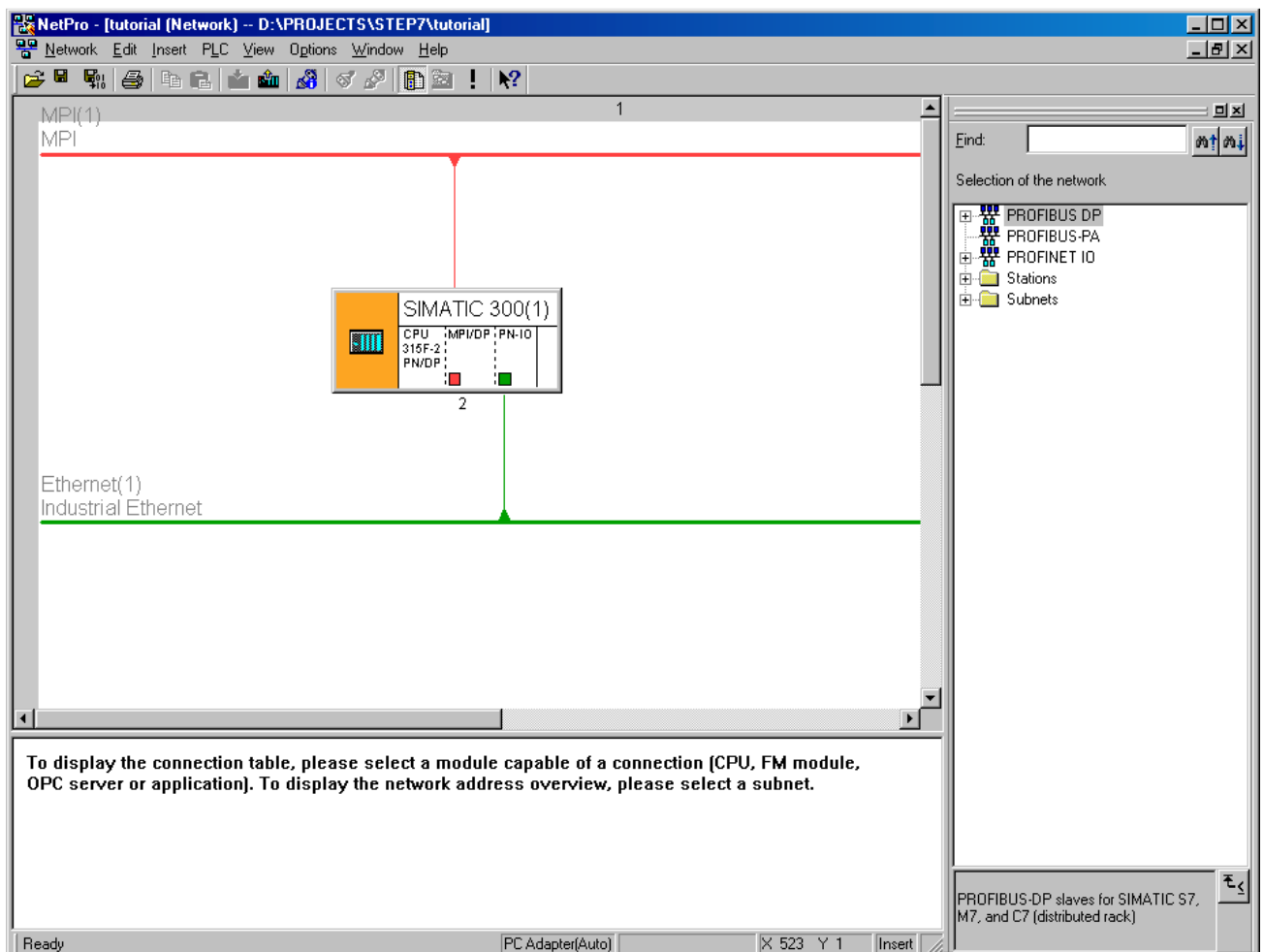
Figuur 4-20

Het laden wordt nu gestart. Dit duurt enkele seconden. Zie figuur 4-21. Als het laden niet lukt, kan het zijn dat de *PG/PC Interface* verkeerd is ingesteld. Raadpleeg dan de docent.



Figuur 4-21

Sluit de hardwareconfiguratie af via menu **Station**→**Exit**. Er wordt weer terug gekeerd naar het netwerkprogramma **Netpro**. Hierin is duidelijk te zien dat de PLC verbonden is aan het MPI- en Ethernet-netwerk. Zie figuur 4-22.

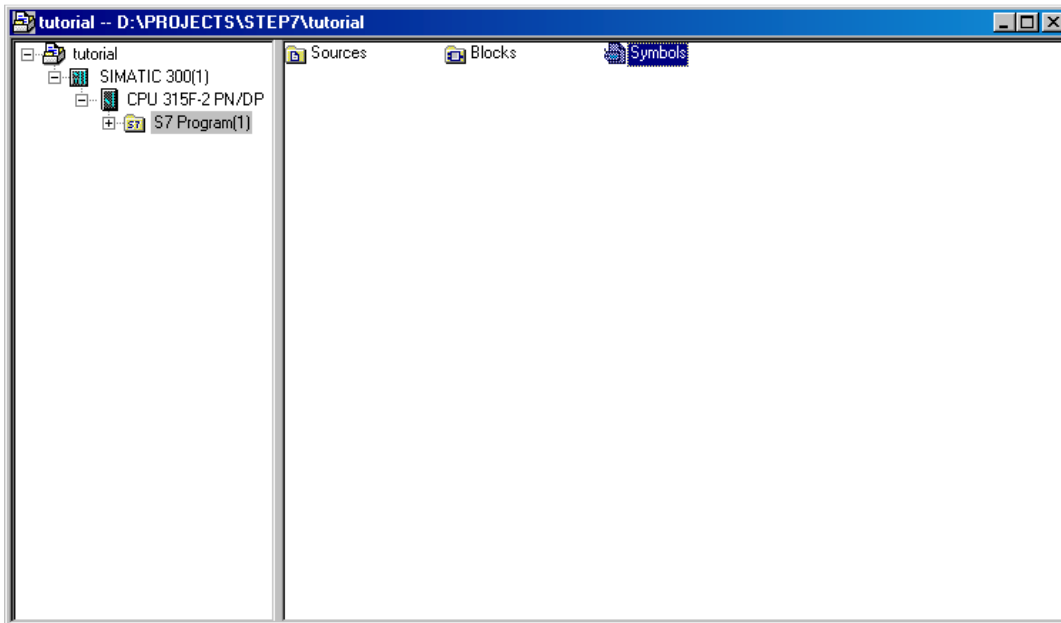


Figuur 4-22

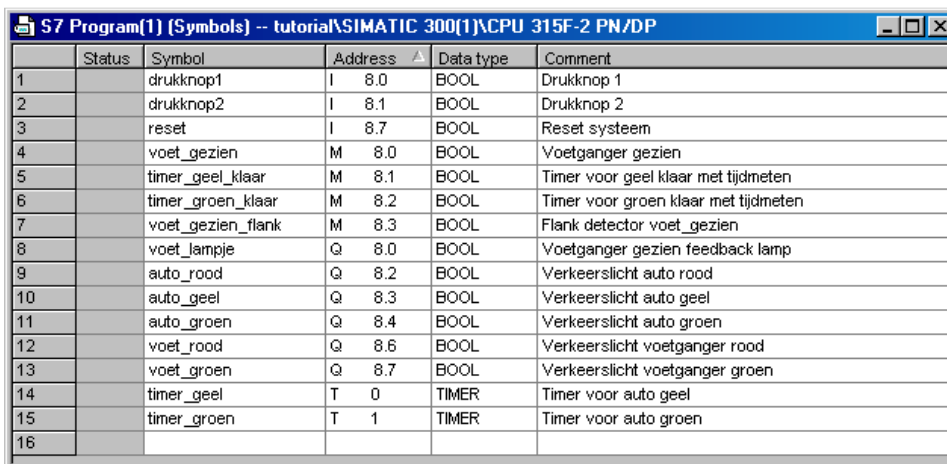
De hardwareconfiguratie is nu klaar. Netpro kan worden afgesloten via het menu **Network**→**Exit**.

4.5 Symbolen definiëren

Het onthouden van de adressen als ingangen, uitgangen en timers, is een lastige en vermoeiende bezigheid. Gelukkig kan een symbolenlijst worden opgesteld. Er kan dan gerefereerd worden met de symbolische naam in plaats van het eigenlijke adres. Dubbelklik in de projectscherm op Symbols (figuur 4-23). De symbolen-editor wordt dan gestart. Voer de symbolen in zoals is weergegeven in figuur 4-24 (het veld Data Type wordt door de editor zelf ingevuld). Sluit de editor af, de definities worden automatisch opgeslagen. Soms verschijnt een dialoog zoals in figuur 4-25. Kies dan **Yes**.

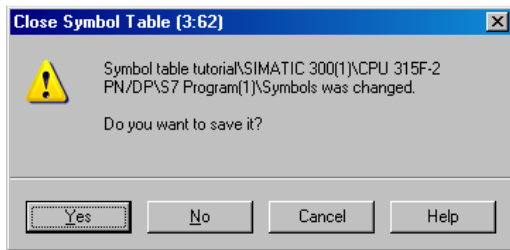


Figuur 4-23

The screenshot shows the Symbols editor window titled 'S7 Program(1) (Symbols) -- tutorial\SIMATIC 300(1)\CPU 315F-2 PN/DP'. It contains a table with 16 rows of symbol definitions. The columns are Status, Symbol, Address, Data type, and Comment.

	Status	Symbol	Address	Data type	Comment
1		drukknop1	I 8.0	BOOL	Drukknop 1
2		drukknop2	I 8.1	BOOL	Drukknop 2
3		reset	I 8.7	BOOL	Reset systeem
4		voet_gezien	M 8.0	BOOL	Voetganger gezien
5		timer_geel_klaar	M 8.1	BOOL	Timer voor geel klaar met tijdmeter
6		timer_groen_klaar	M 8.2	BOOL	Timer voor groen klaar met tijdmeter
7		voet_gezien_flank	M 8.3	BOOL	Flank detector voet_gezien
8		voet_lampje	Q 8.0	BOOL	Voetganger gezien feedback lamp
9		auto_rood	Q 8.2	BOOL	Verkeerslicht auto rood
10		auto_geel	Q 8.3	BOOL	Verkeerslicht auto geel
11		auto_groen	Q 8.4	BOOL	Verkeerslicht auto groen
12		voet_rood	Q 8.6	BOOL	Verkeerslicht voetganger rood
13		voet_groen	Q 8.7	BOOL	Verkeerslicht voetganger groen
14		timer_geel	T 0	TIMER	Timer voor auto geel
15		timer_groen	T 1	TIMER	Timer voor auto groen
16					

Figuur 4-24



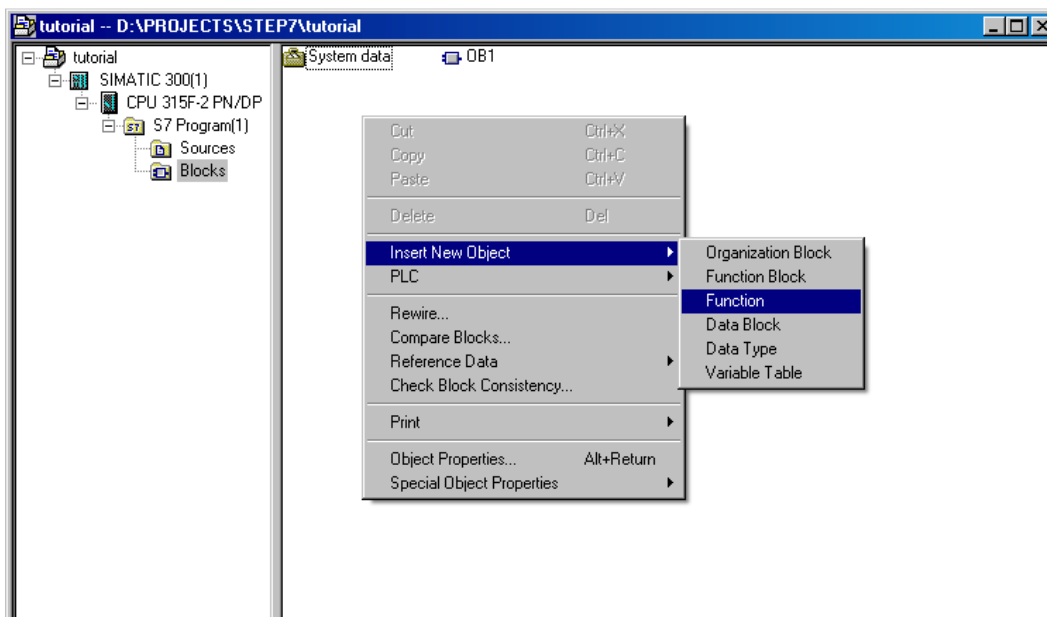
Figuur 4-25

4.6 Programma invoeren

Zoals bekend moet een PLC-programma altijd Organization Block 1 (OB1) bevatten, ook al roept deze niets anders aan dan Functions (FC) en Function Blocks (FB).

Eerst wordt FC1 geprogrammeerd met het eigenlijke verkeerslichtprogramma, vervolgens wordt OB1 geprogrammeerd met daarin een aanroep van FC1. Deze volgorde is van belang.

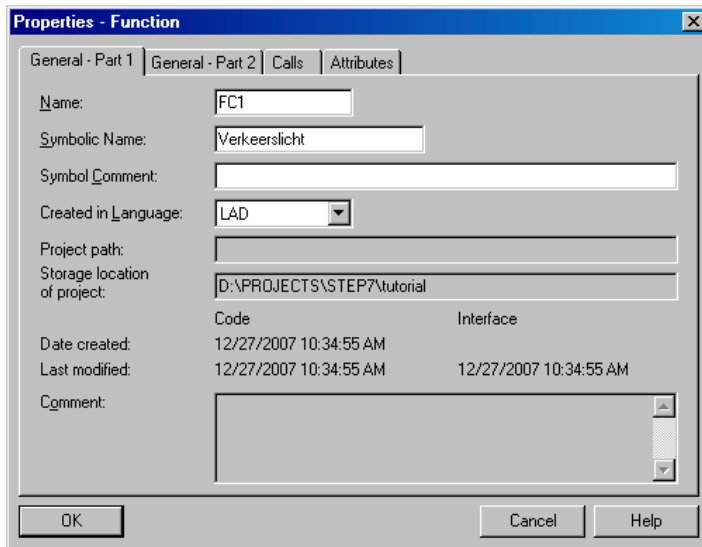
Vanuit de Simatic Manager moet FC1 aangemaakt worden. Selecteer in het linkerscherm **Blocks**, en klik in het rechterscherf (onder **System Data** en **OB1**) op de **rechter** muisknop. Er wordt een contextgevoelig menu geopend en selecteer daarmee **Insert New Object** → **Function**. Zie figuur 4-26.



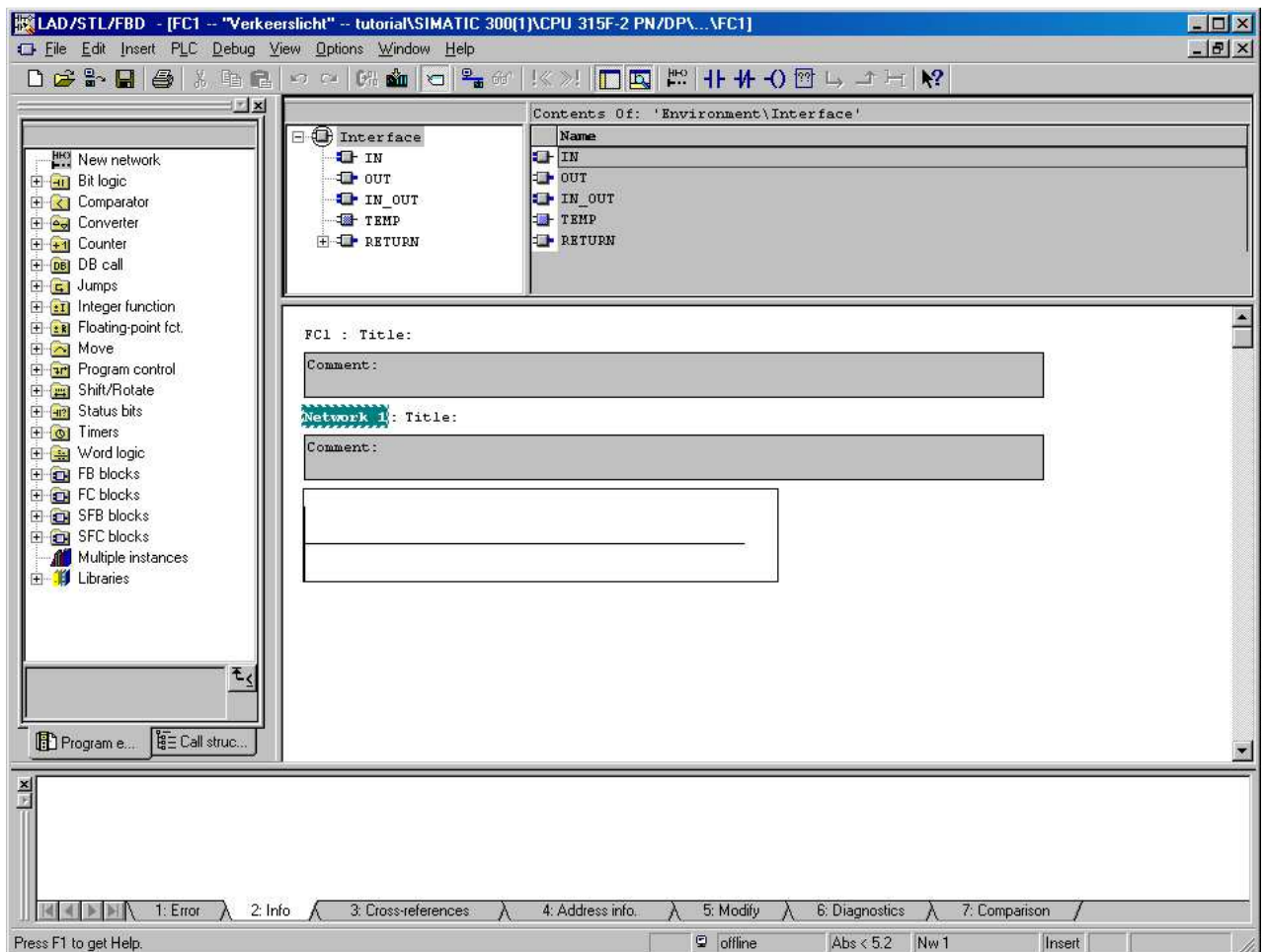
Figuur 4-26

Nu kunnen allerlei gegevens (eigenschappen) worden ingevoerd. Vul dit in zoals is weergegeven in figuur 4-27. Let erop dat als programmeertaal LAD wordt geselecteerd. Klik op **OK**.

In de projectscherf staat nu FC1. Dubbelklik hier op en de LAD-editor wordt geopend. Zie figuur 4-28.



Figuur 4-27



Figuur 4-28

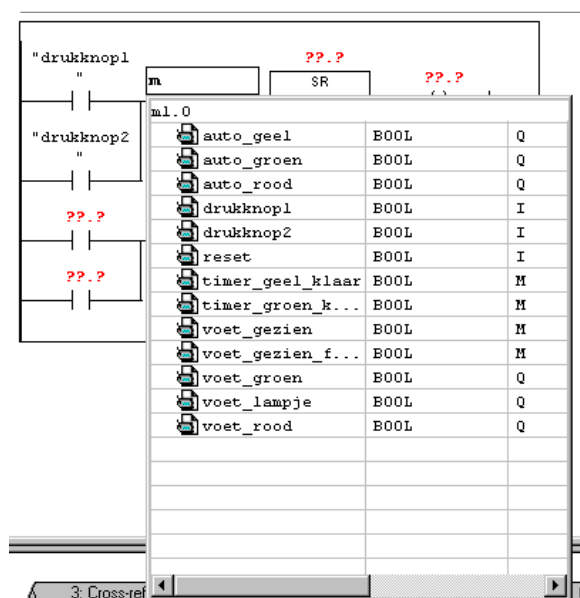


Figuur 4-29

Een ladderdiagram bestaat uit één of meerdere netwerken. Per netwerk kan één logische schakeling (*rung*) geprogrammeerd worden met één of meerder uitgangen. Een netwerk kan worden toegevoegd door middel van het eerste, meest linkse pictogram uit figuur 4-29. Als ingangen kunnen PLC-ingangen en merkers dienen. Dit wordt aangegeven met contacten. De tweede en derde pictogram zijn resp. *normally open* en *normally closed contacts*. Als uitgangen (*coils*) kunnen PLC-uitgangen of merkers gebruikt worden⁶. Deze worden toegevoegd met het vierde pictogram. Een logische OR wordt gemaakt met behulp van de laatste twee pictogrammen.

Het programmeren vergt enige routine. Om ingangen in te voegen moet b.v. eerst een rung van een netwerk worden geselecteerd worden zodat deze groen wordt. Dan worden de pictogrammen uit figuur 4-29 blauw en kunnen ze worden ingevoerd. Niet alle mogelijkheden kunnen via de pictogrammen worden ingevoerd. Aan de linkerkant staan alle mogelijkheden, zoals SR-elementen, timers, counters en rekenkundige operaties.

Tijdens het invoeren van adressen zoals M0.0 verschijnt de symbolenlijst. Dan kan eenvoudig het juiste symbool worden gekozen. Zie figuur 4-30.



Figuur 4-30

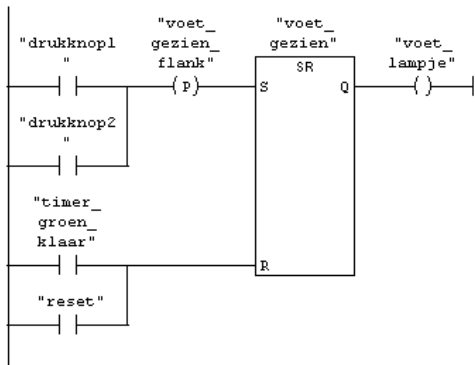
Indien iets niet ingevoerd kan worden, zoals een BYTE-adres bij een contact, wordt dit rood afgebeeld en kan de laddereditor niet worden afgesloten. Ook als adressen niet zijn ingevuld, zoals in figuur 4-30 zichtbaar, kan de editor niet worden afgesloten.



Als voorbeeld is het eerste netwerk in figuur 4-31 afgebeeld. Het complete programma is opgenomen in bijlage 1. Voor correcte werking moet het hele programma ingevoerd worden.

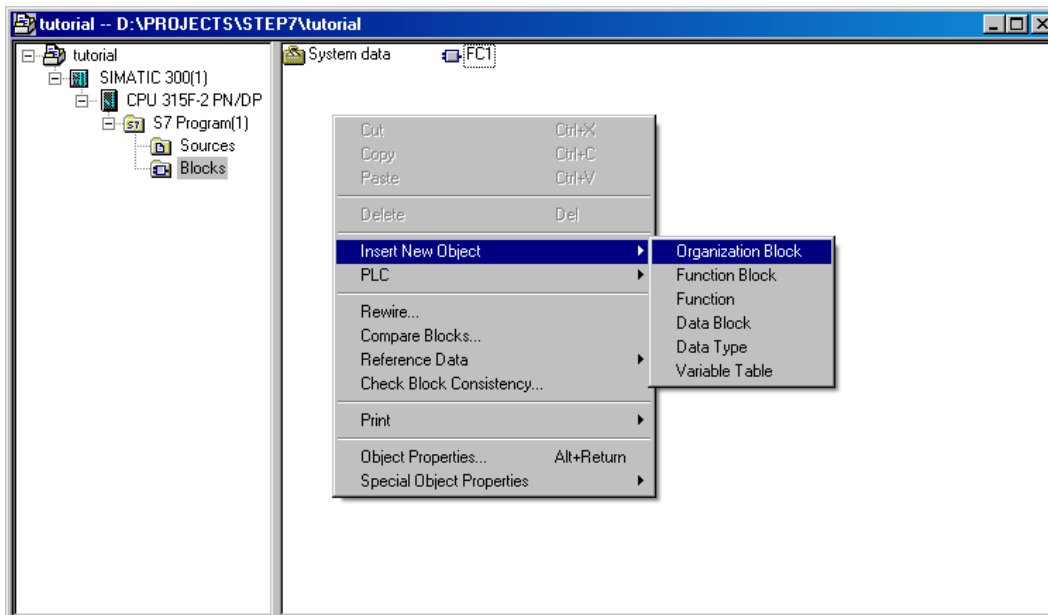
Sluit de editor af via **File**→**Exit** als het hele ladderdiagram is ingevoerd.

⁶ Ook Q, DB en L kunnen als ingangen en uitgangen dienen.



Figuur 4-31

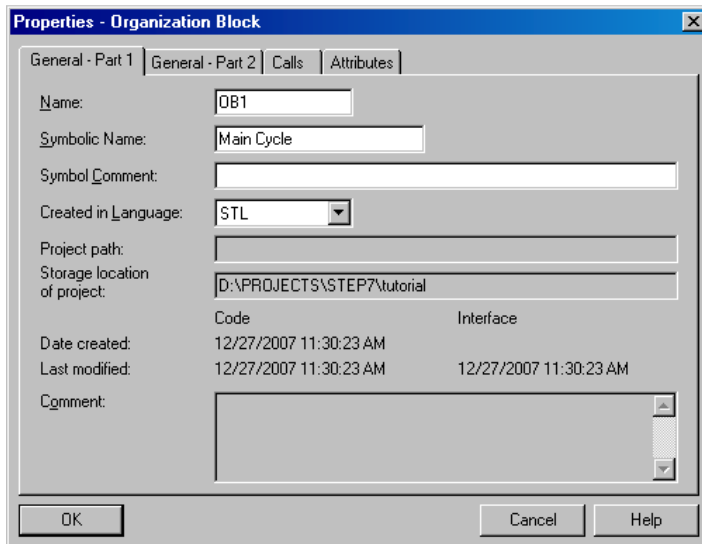
Nu FC1 ingevoerd is, moet OB1 nog worden geprogrammeerd. OB1 moet alleen FC1 aanroepen. Dit wordt gedaan in STL. OB1 is al aangemaakt; dat gebeurt bij het aanmaken van een nieuw project. Als OB1 niet bestaat, voeg het dan toe via **Insert New Object**→**Organization Block**. OB1 verschijnt nu in het overzicht in projectscherm. Zie figuur 4-32.



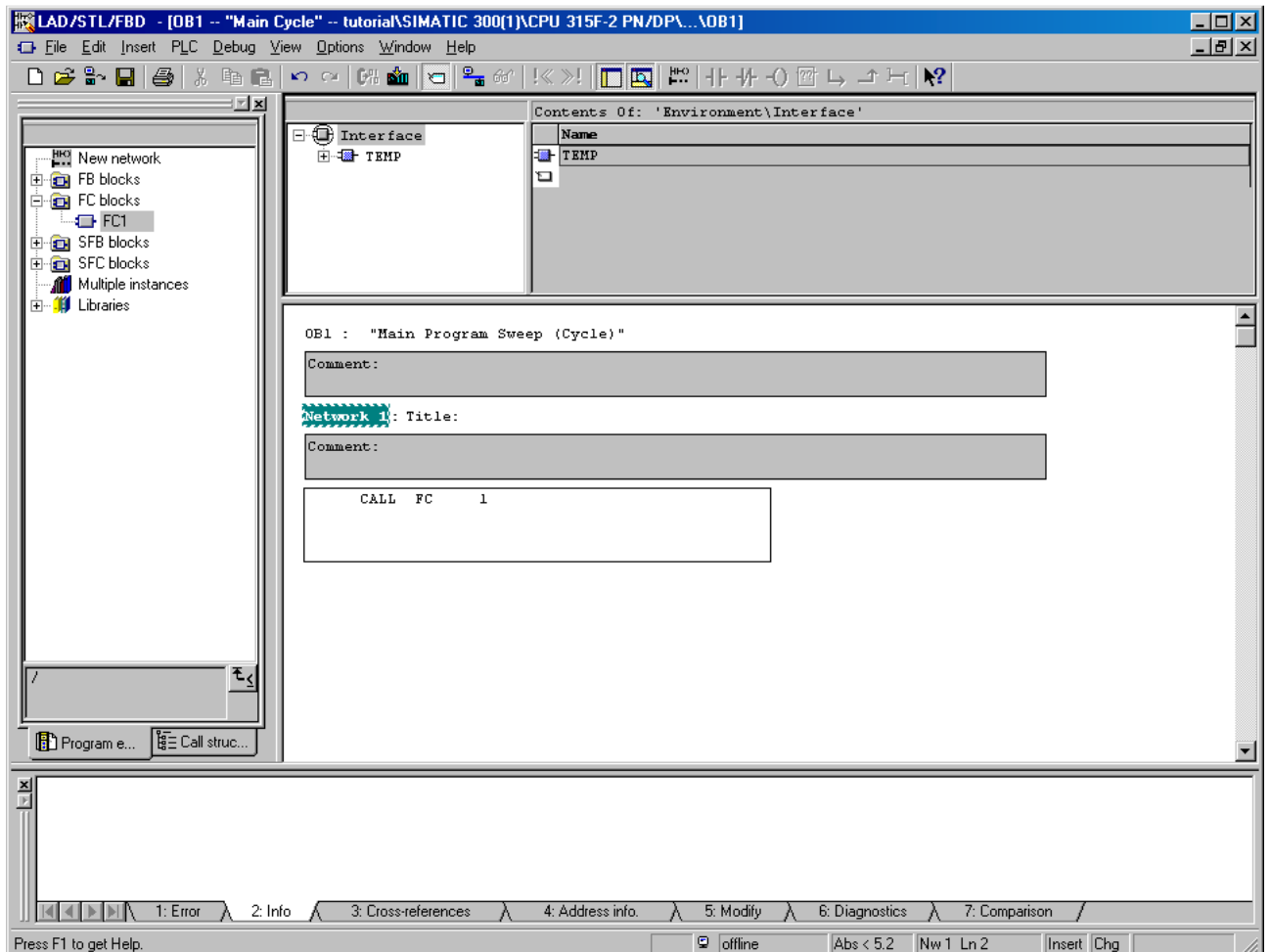
Figuur 4-32

Als OB1 al bestaat, wijzig dan de eigenschappen door eerst OB1 te selecteren. Klik nu op de rechtermuisknop en selecteer **Object Properties**. Vul de eigenschappen in voor OB1 zoals is aangegeven in figuur 4-33 en klik op OK. Dubbelklik op OB1 om de STL-editor te openen⁷. Vul nu als enige regel de aanroep naar FC1 in. Zie figuur 4-34. Sla het ingevoerde programma op via **File**→**Save** en sluit de editor af via het menu **File**→**Exit**.

⁷ De oplettende lezer ziet dat de STL-editor en de ladder-editor dezelfde zijn. Dit is correct. Via het menu **View** kan worden geschakeld tussen STL, LAD en FBD.



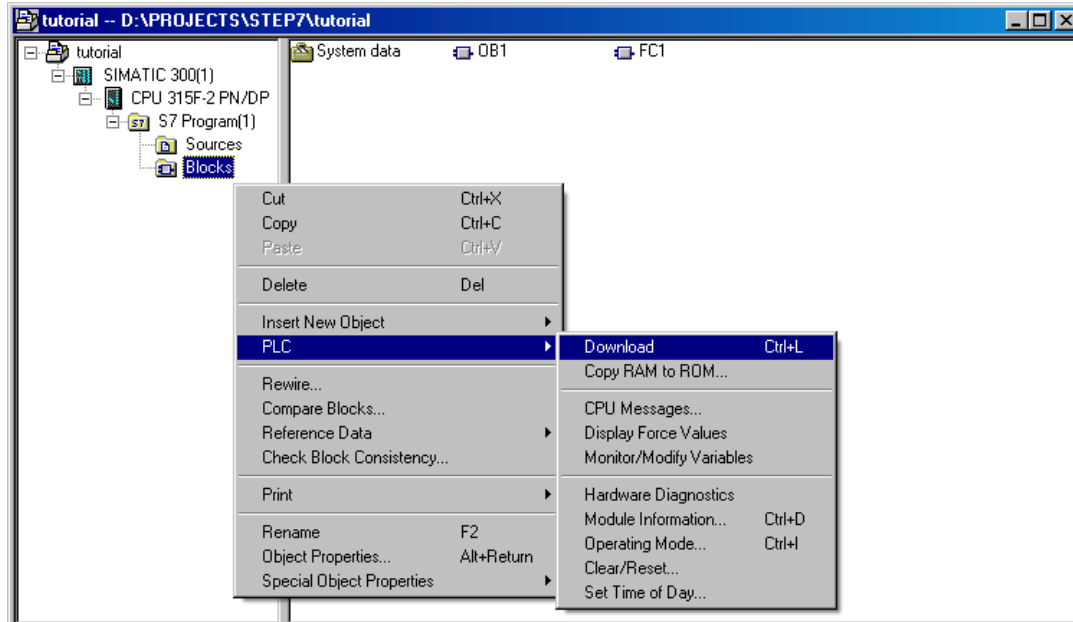
Figuur 4-33



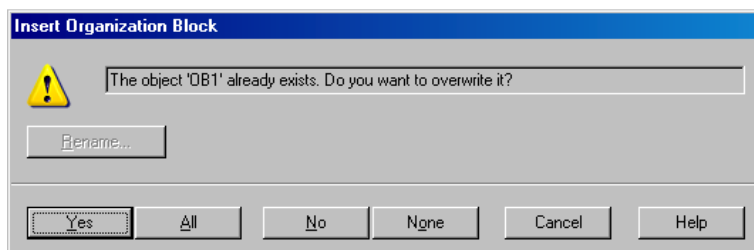
Figuur 4-34

4.7 Laden PLC programma

Nu moeten de diverse onderdelen in de PLC geladen worden. Zet de PLC eerst in STOP. Klik met de rechtermuisknop op **Blocks** in de Simatic Manager en selecteer **PLC**→**download** (figuur 4-35). Het laden wordt nu begonnen. Als blokken, die geladen moeten worden, al in de PLC aanwezig zijn, zal een bevestiging worden gevraagd (figuur 4-36). Klik op de knop **All**, zodat niet voor elk blok een aparte vraag wordt gesteld.

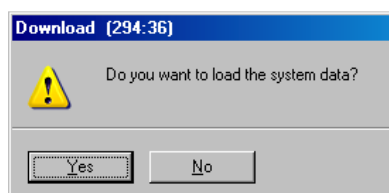


Figuur 4-35

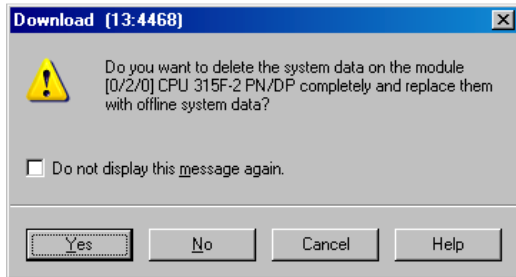


Figuur 4-36

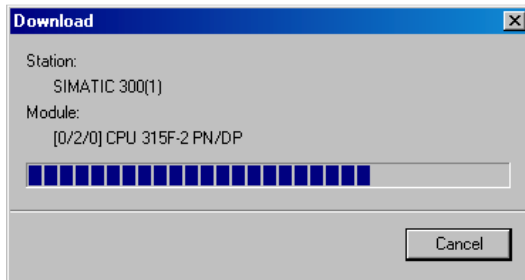
Omdat alle blokken geselecteerd zijn zal ook de System Data worden geladen. Dit mag overgeslagen worden door op **No** te klikken. Als op **Yes** wordt geklikt, volgt er nog een aantal dialogen. Zie de figuren 4-37 tot en met 4-39. De figuren 4-40 en 4-41 verschijnen alleen als de PLC in RUN staat.



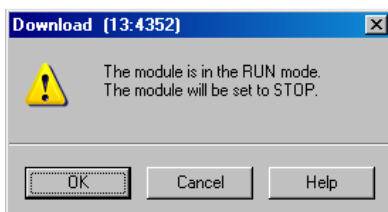
Figuur 4-37



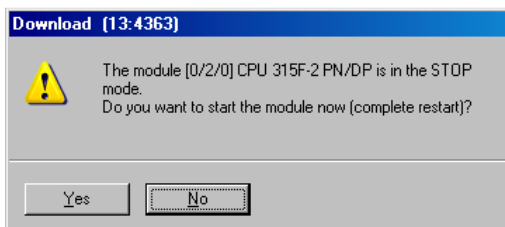
Figuur 4-38



Figuur 4-39



Figuur 4-40



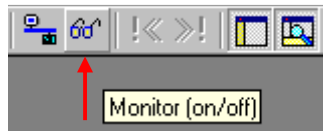
Figuur 4-41

4.8 Monitoren van het programma

Het programma is nu geladen; de werking kan worden getest. Start de PLC door operating mode schakelaar van STOP naar RUN te schakelen. De PLC gaat nu starten. Daarna kunnen de schakelaars op de simulatiemodule gebruikt worden om het programma te testen. Raadpleeg de docent.

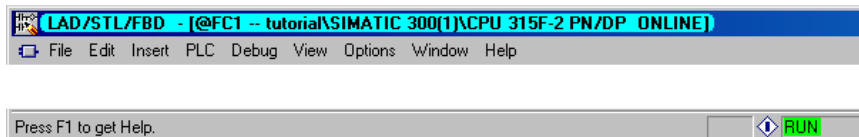
Het programma kan ook gevolgd (*monitor*) worden via de ladder- of STL-editor. Dubbelklik hiervoor op FC1 om deze bouwsteen te openen in de editor.

Start het monitoren via het menu **Debug** → **Monitor** of klik op het brilletje zoals afgebeeld in figuur 4-42.



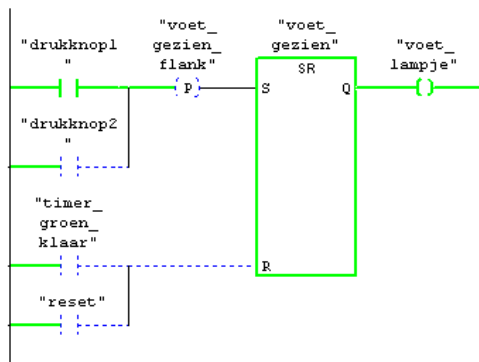
Figuur 4-42

De kleur van de titelbalk van de editor wordt nu lichtblauw ten teken dat het hier om een online verbinding gaat. Onderaan de editor is een groene, lopende balk zichtbaar. Zie figuur 4-43.

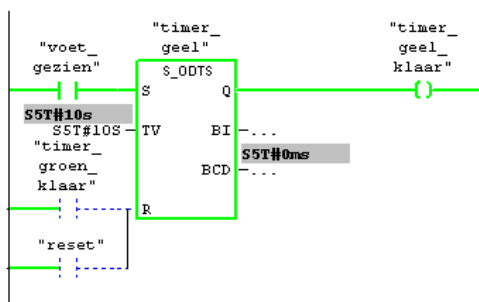


Figuur 4-43

Hieronder zijn twee afbeeldingen zichtbaar van netwerk 1 en netwerk 2 tijdens het draaien van het programma (figuren 4-44 en 4-45).



Figuur 4-44



Figuur 4-45

De tutorial is hiermee ten einde. De cursist wordt uitgedaagd het eenvoudige verkeerslichtsysteem uit te breiden met b.v. een mogelijkheid tot het laten knipperen van het gele licht indien het systeem niet operationeel is. Hiervoor is een extra schakelaar nodig.

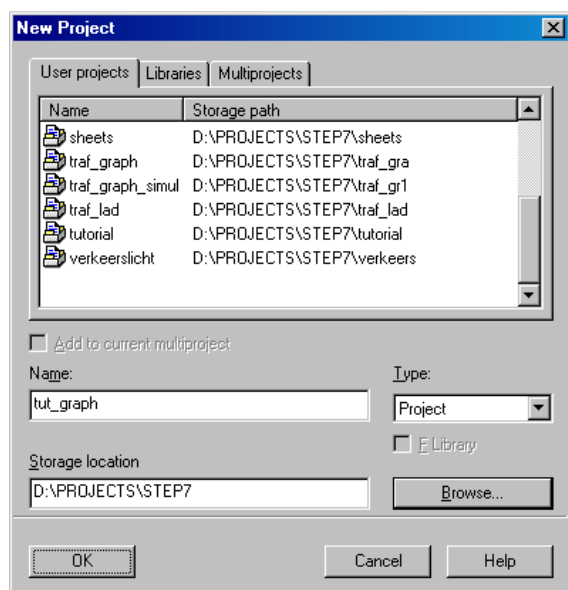
5. Tutorial S7-Graph

Veel productieprocessen kunnen worden opgelost met een afloop. Een (deel-)product volgt een vast aantal stappen waarin iets gebeurt. Deze stappen zijn goed van elkaar te onderscheiden. Zo'n programma kan goed worden geprogrammeerd als een toestandsmachine. S7-Graph is een taal waarin zo'n toestandsmachine kan worden geprogrammeerd.

In deze tutorial wordt uitgelegd hoe een S7-Graph programma ingevoerd moet worden. Eerst wordt de configuratie van de PLC ingevoerd, vervolgens een klein ladder-programma voor conditionering van de ingangen, dan het S7-Graph programma. Als laatste wordt dit programma en enkele variabelen gevolgd met de monitor.

5.1 Project aanmaken

Begin een nieuw project zoals is aangegeven in het vorige hoofdstuk en noem het project tut_graph. Zie figuur 5-1.



Figuur 5-1

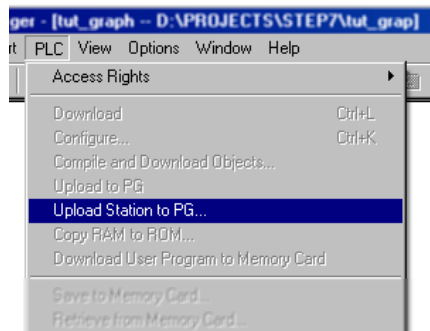
Vul de bekende gegevens in en klik op **OK**. Het nieuwe project wordt aangemaakt.

Eerst moet de MPI-bus $MPI(1)$ verwijderd worden, deze wordt straks opgehaald uit de configuratie. Indien deze stap wordt overgeslagen, zijn straks twee MPI-bussen zichtbaar.

5.2 Configuratie en Blocks uit PLC halen

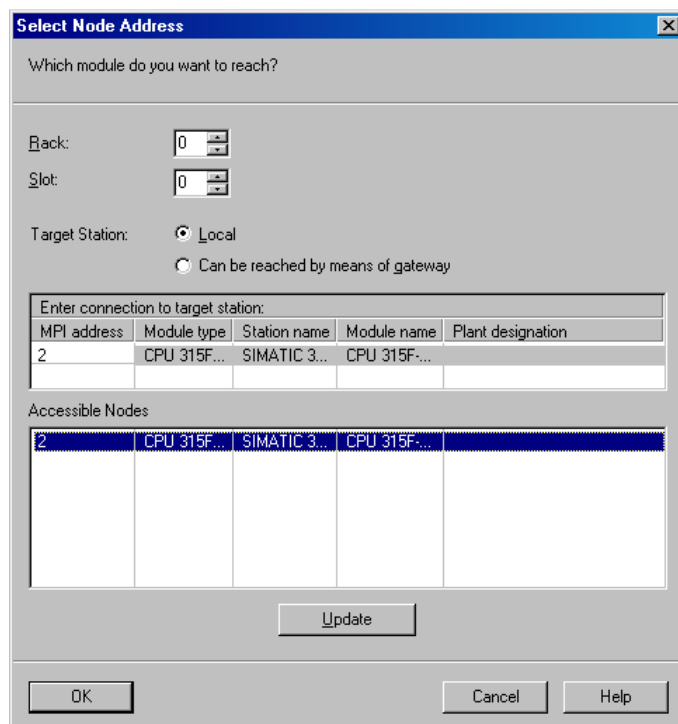
In het vorige hoofdstuk is een nieuwe configuratie van de PLC opgezet en geladen in de PLC. Deze configuratie kan teruggehaald worden uit de PLC en zo gebruikt worden in het nieuwe project⁸. Hiervoor moeten de configuratiegegevens *ge-upload* worden vanuit de PLC naar het project. Hierbij worden ook alle programma's mee gekopieerd (OB, FC etc). De symbolenlijst is niet in de PLC opgeslagen, die moet opnieuw worden toegevoegd.

Doe een upload via het menu **PLC**→**Upload Station to PG**. Zie figuur 5-2.



Figuur 5-2

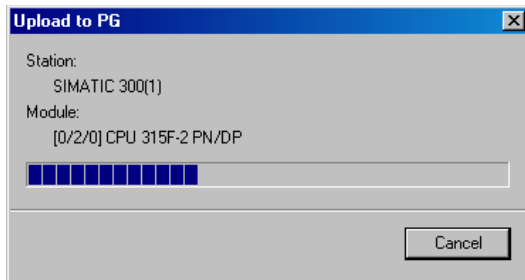
Er verschijnt nu een scherm met daarin alle aangesloten PLC's. Midden onderaan het scherm verschijnt een knop **View**. Als er geen PLC's zichtbaar zijn, klik hier dan op. Na een tijdje verschijnt onder **Accessible Nodes** de PLC die op het practicum is aangesloten. Selecteer de gegeven PLC en klik op **OK**. Zie figuur 5-3.



Figuur 5-3

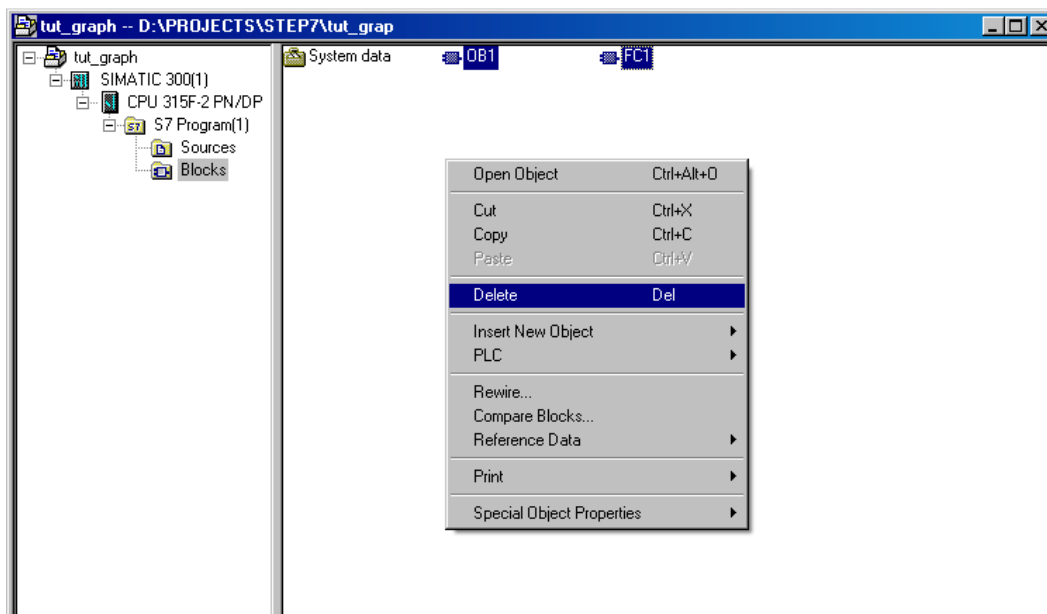
⁸ Natuurlijk wordt de laatst geladen configuratie opgehaald.

De configuratie wordt nu uit de PLC geladen en in het project opgenomen. Zie figuur 5-4.



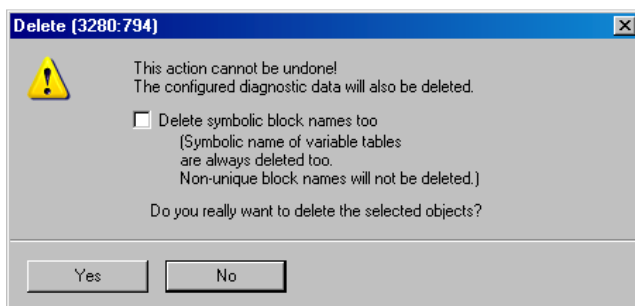
Figuur 5-4

In het projectscherm is nu de PLC, de MPI-bus én de Ethernetbus zichtbaar. Als er twee MPI-bussen zichtbaar zijn, moet er één verwijderd worden. Dit kan worden gedaan via het netwerkconfiguratie-programma Netpro. De programma- en datablokken zijn ook uit de PLC gehaald. Deze moeten verwijderd worden. Zie figuur 5-5 als voorbeeld.



Figuur 5-5

Nu wordt gevraagd of de blokken verwijderd moeten worden. Klik op **Yes**. Zie figuur 5-6.



Figuur 5-6

De configuratie is nu afgerond. Via de programma's Netpro en HWConfig kunnen eventuele wijzigingen aangebracht worden.

5.3 Symbolenlijst opstellen

Natuurlijk moet ook voor deze tutorial een symbolenlijst opgesteld worden. In figuur 5-7 zijn de symbolen vermeld. Let vooral op aantal_gedrukt, dit is een merkerwoord van het type INT.

	Status	Symbol	Address	Data type	Comment
1		drukknop1	I 8.0	BOOL	Drukknop 1
2		drukknop2	I 8.1	BOOL	Drukknop 2
3		knipknop	I 8.2	BOOL	Drukknop voor knippen geel licht
4		reset	I 8.7	BOOL	Reset systeem
5		voet_gezien	M 8.0	BOOL	Voetganger gezien
6		knip_gezien	M 8.1	BOOL	Knipperschakelaar gezien
7		voet_gezien_flank	M 8.2	BOOL	Flankdetector voet_gezien
8		knip_gezien_flank	M 8.3	BOOL	Flankdetector knip_gezien
9		aantal_gedrukt	MW 20	INT	Aantal keer dat er op een drukknoop is gedrukt
10		voet_lampje	Q 8.0	BOOL	Voetganger gezien feedback lamp
11		auto_rood	Q 8.2	BOOL	Verkeerslicht auto rood
12		auto_geel	Q 8.3	BOOL	Verkeerslicht auto geel
13		auto_groen	Q 8.4	BOOL	Verkeerslicht auto groen
14		voet_rood	Q 8.6	BOOL	Verkeerslicht voetganger rood
15		voet_groen	Q 8.7	BOOL	Verkeerslicht voetganger groen
16					

Figuur 5-7

5.4 Een blok vooraf

Eerst wordt een ladderprogramma ingevoerd wat samenwerkt met het S7-Graph programma. Het conditioneert de ingangen, zodat de invoer via de drukknoppen makkelijk kunnen worden verwerkt. Maak hiervoor FC2 aan, vul de gegevens in en voer het ladderprogramma in en klik daarna op **OK**. Zie de figuren 5-8 en 5-9.

Properties - Function

General - Part 1 | General - Part 2 | Calls | Attributes

Name: FC2

Symbolic Name: DrukknopConditioner

Symbol Comment:

Created in Language: LAD

Project path:

Storage location of project: D:\PROJECTS\STEP7\tut_grap

	Code	Interface
Date created:	12/30/2007 09:06:35 AM	
Last modified:	12/30/2007 09:06:35 AM	12/30/2007 09:06:35 AM

Comment:

OK Cancel Help

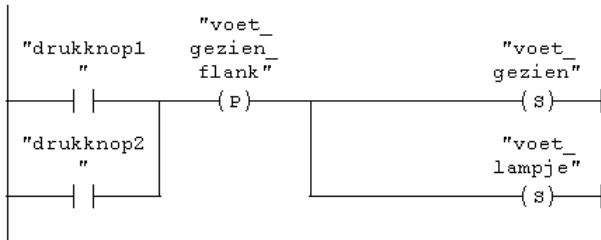
Figuur 5-8

FC2 : Drukknopconditioner

Deze FC bestaat uit twee rungs, een voor de voetgangersdrukknoppen en een voor de knipperdrukknop. Zie aldaar.

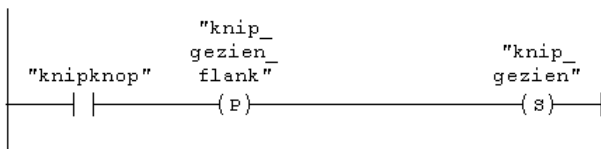
Network 1: Voetganger gezien

Dit netwerk zorgt voor conditionering van het gebruik van de drukknooppotten door voetgangers. De flank is nodig omdat als een drukknop continue wordt ingedrukt, de verkeerslichtsysteem "rondjes" blijft draaien. De merker "voet_gezien" wordt gereset in het verkeerslichtsysteem-programma.



Network 2: Knipperschakelaar gezien

Deze rung zorgt voor conditionering van de knipperdrukknop. Als hier op wordt gedrukt zal het systeem na afloop van een cyclus naar de knipperstand gaan. Om terug te keren moet nog eens op de knipperknop worden gedrukt.



Figuur 5-9

5.5 Blokken, stappen en overgangen

Naar de gebruiker toe laat de CPU en bijbehorend geheugen zich afbeelden als programmeerbare blokken. Er zijn verschillende typen, zie hiervoor bijlage 3. Voorlopig is genoeg te weten dat elk gebruikersprogramma moet starten in Organization Block 1 (OB1) en dat S7-Graph programma's worden opgeslagen in een Function Block (FB) met bijbehorend Data Block (DB).

Een S7-Graph programma bestaat uit een aantal stappen (*steps*) en overgangen (*transitions*) tussen de stappen. Dit geheel wordt in S7-Graph een sequencer genoemd. In een stap kunnen diverse operaties worden uitgevoerd, zoals het aanzetten van een uitgang of het starten van een teller. Aan de overgangen kunnen voorwaarden worden verbonden zoals "ga verder als ingang 1 actief is".

De oplettende lezer ziet in dit alles een overeenkomst met toestandmachines uit de digitale techniek.

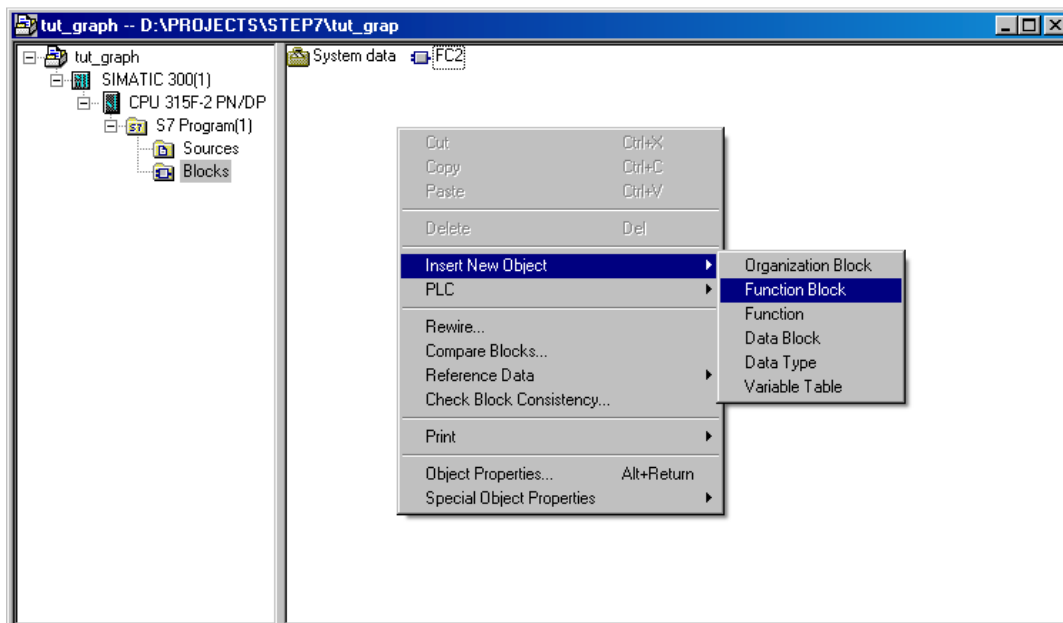
Eigenlijk is S7-Graph niets anders dan een frontend die de ingevoerde sequencer omzet naar STL. Het is dus eigenlijk een compiler.

5.6 Aanmaken Function Block

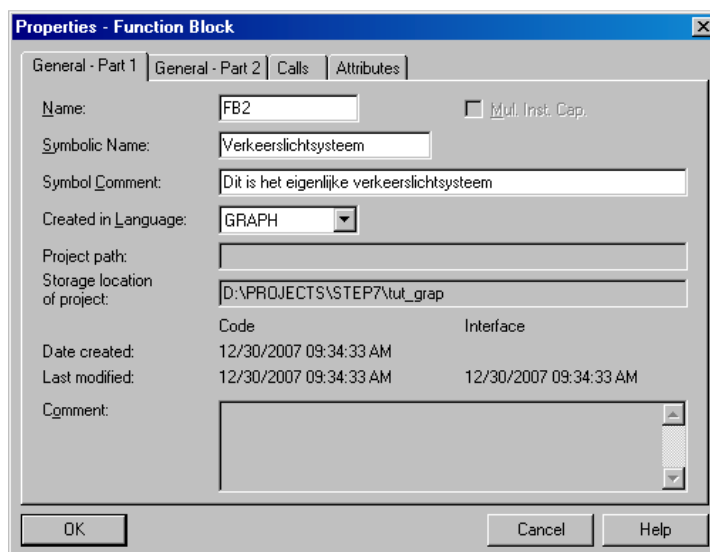
Een S7-Graph programma kan alleen in een FB (Function Block) ingevoerd worden. Aan een Function Block wordt een Data Block gekoppeld.

Eerst moet een FB worden aangemaakt waarin het S7-Graph programma kan “leven”. Klik in de Simatic Manager links **Blocks** open. Klik nu met de rechtermuisknop om een **Function Block** in te voegen. Zie figuur 5-10. Er wordt een dialoogvenster geopend waarin de eigenschappen van de FB kunnen worden vastgelegd (figuur 5-11).

Vul bij **Name** FB2 in. In **Symbolic Name** kan een naam worden ingevoerd. Deze naam komt dan terug in de symbolentabel. **Commentaar** gaat in **Symbol Comment**. Kies verder als taal **Graph** (**Created in Language: GRAPH**). Klik op **OK** om de FB aan te maken.



Figuur 5-10

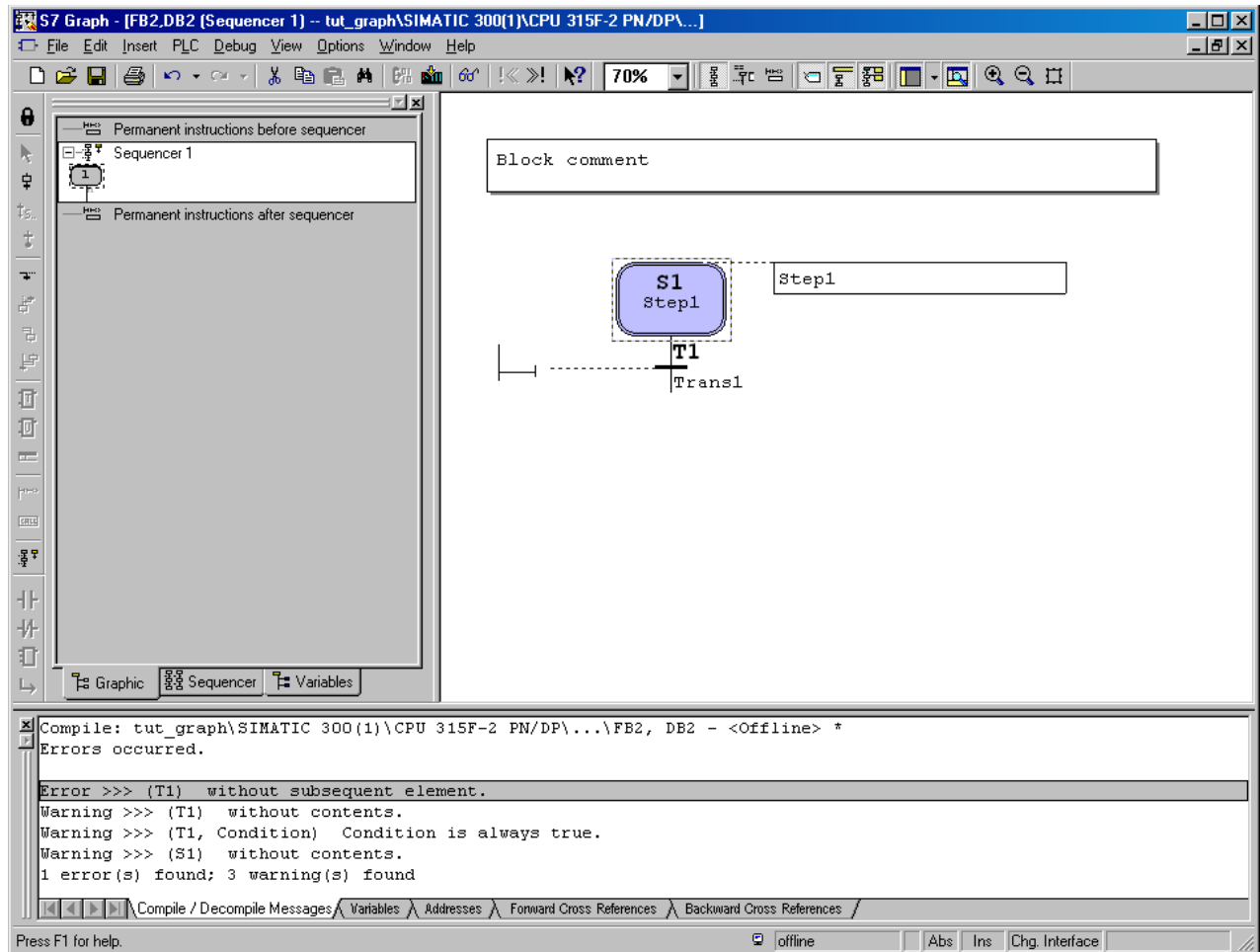


Figuur 5-11

5.7 Starten S7-Graph editor

Nu is in de Simatic Manager FB2 te vinden. Dubbelklik hierop om de S7-Graph editor met FB2 te openen. Het openings scherm is afgebeeld in figuur 5-12.

Voor meer informatie over het editen, zie het document “S7-Graph - Programming Sequential Control Systems”, hoofdstuk 6.

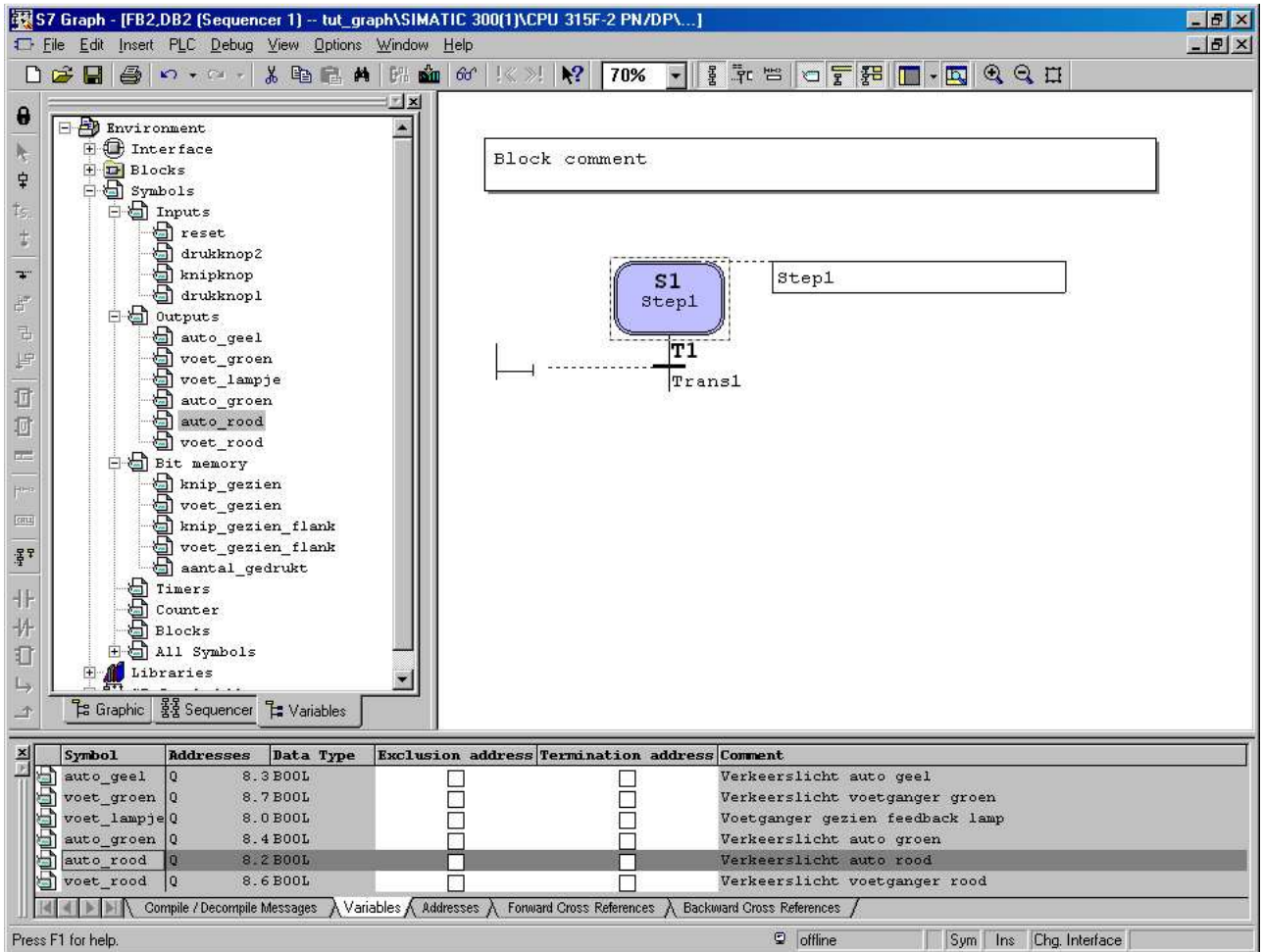


Figuur 5-12

Het scherm is in drie panelen verdeeld. Linksboven is een overzicht van de sequencer inclusief de permanente instructies te zien (tabblad *Graphic*). Er kan echter ook voor het tabblad *Variables* gekozen worden (figuur 5-13). Rechtsboven is het edit-paneel. Hierin worden de aanpassingen verricht. Onder is console-paneel. Hier zijn onder andere systeemberichten te vinden zoals compiler-uitvoer, gegevens van variabelen en gebruikte adressen in een stap.

In het edit-paneel wordt gelijk de eerste stap geplaatst. De dubbele ring geeft aan dat het hier om een beginstap (*initial step*) gaat. Als de sequencer wordt herstart (zie paragraaf 5-13), is dit de eerste stap.

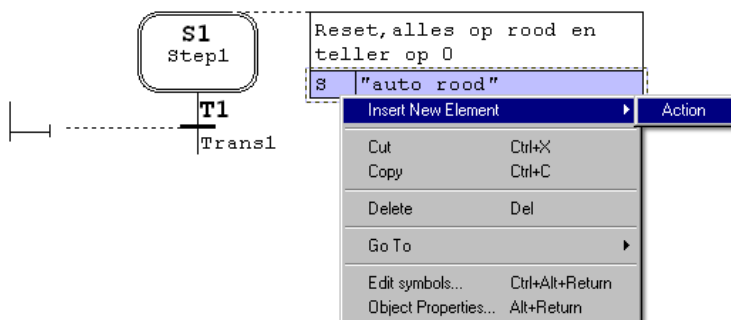
Let op: er moet precies één initial step in het programma aanwezig zijn. Een veel voorkomend probleem is dat tijdens het programmeren de initial step wordt verwijderd, en er geen nieuwe wordt aangemerkt. De sequencer start dan niet als de PLC het programma begint te verwerken.



Figuur 5-13

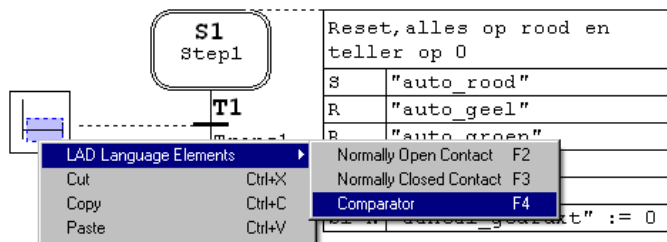
5.8 Invoeren eerste stap

Het gebruik van de editor vergt, net als bij de laddereditor, enige routine. Als vuistregel geldt: als iets veranderd, ingevoegd of verwijderd moet worden, selecteer dan eerst het onderdeel en voer dan de operatie uit. Namen van variabelen, zoals `auto_rood`, kunnen zonder aanhalingstekens worden ingevoerd, de editor plaatst ze zelf. Hiervoor moet wel de optie **View→Display With→Symbol List** aanstaan. In figuur 5-14 is te zien hoe een actie wordt ingevoerd. Selecteer de actiekop, klik op de rechter muisknop en selecteer **Insert New Element→Action**.



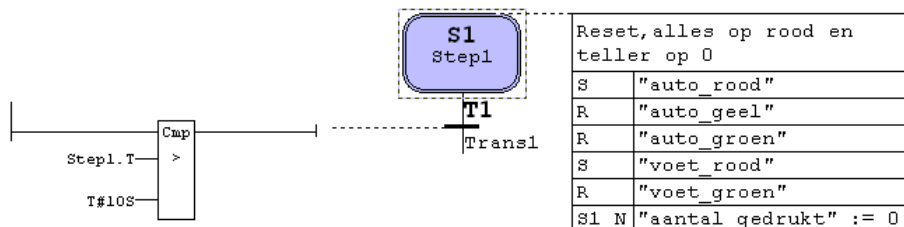
Figuur 5-14

In figuur 5-15 is te zien hoe een overgangsconditie, in dit geval een vergelijkfunctie, wordt ingevoerd. Ook dit vergt enige routine.



Figuur 5-15

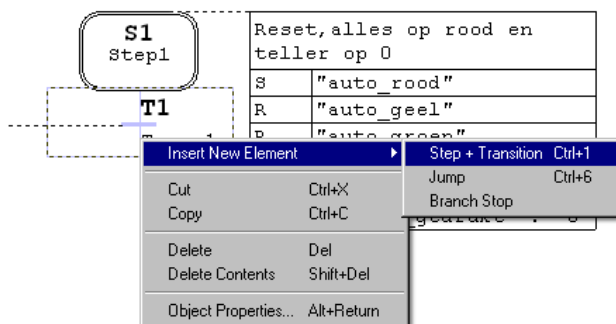
In figuur 5-16 is de complete stap met acties en overgangsconditie te zien. Programmeer de eerste stap precies als in deze figuur.



Figuur 5-16

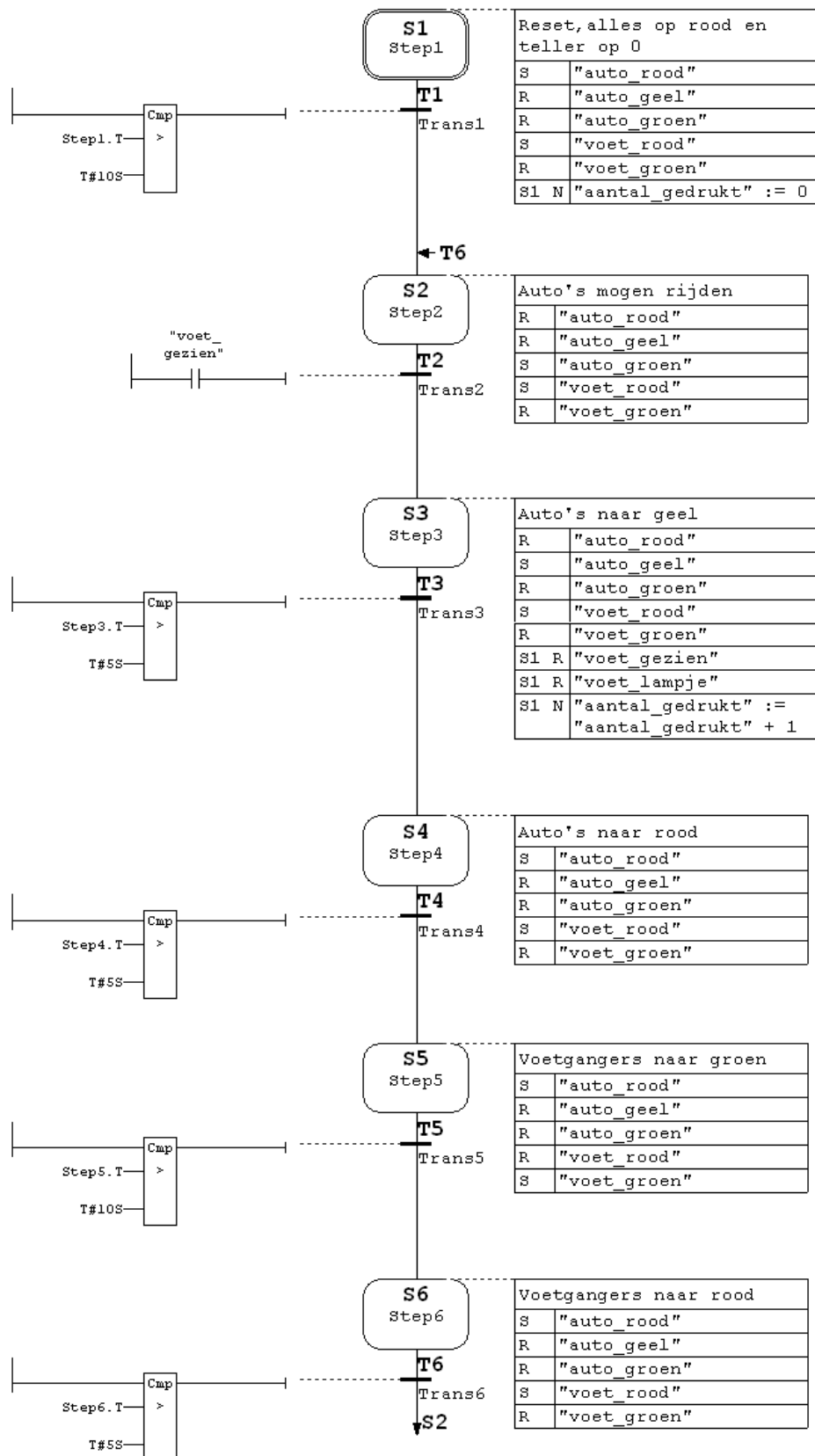
5.9 Invoeren volgende stappen

Aan de eerste stap moet nu de vervolgstap gekoppeld worden. Selecteer hiervoor de overgang (niet de conditie), klik op de rechter muisknop en selecteer **Insert New Element** → **Step + Transition**. Een nieuwe stap wordt nu toegevoegd. Zie figuur 5-17.



Figuur 5-17

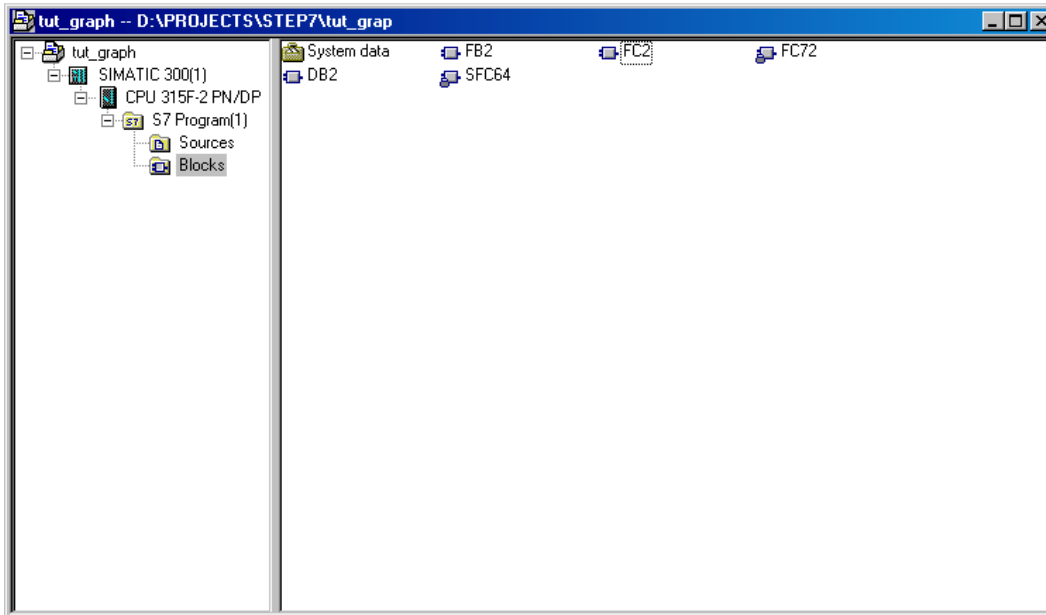
In figuur 5-18 zijn alle stappen, acties en overgangscondities afgebeeld. Voer deze zo in. Let op de laatste stap (stap 6). Daar wordt een *jump* gemaakt naar stap 2. Boven stap 2 is dan een horizontale pijl te zien. Deze is pas te zien na de plaatsing van de jump.



Figuur 5-18

Als alles ingevoerd is, moet programma nog opgeslagen en gecompileerd worden. Start deze actie via het menu **File**→**Save**. Hierna kan de editor worden afgesloten.

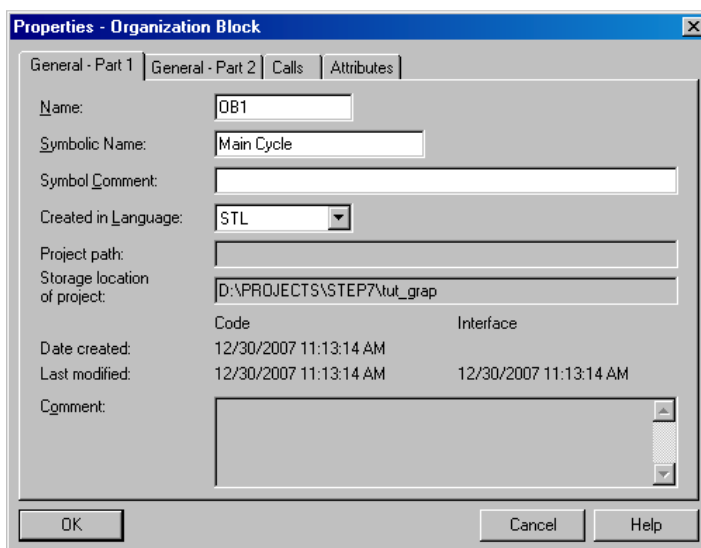
In het projectscherm zijn nu FB2 en andere blokken te zien (figuur 5-19). DB2 is de Data Block dat bij FB2 hoort. Dat wordt door de S7-Graph-compiler geregeld. De code in FB2 gebruikt dus DB2 als geheugen. FC72 (Function) is een blok waarin generieke (voor ieder hetzelfde) code wordt geplaatst voor S7-Graph ingevoerde blokken. SFC64 (System Function) wordt door FB2 gebruikt om de verlopen tijd bij te houden.



Figuur 5-19

5.10 Invoeren OB1

Nu moet nog OB1 ingevoerd worden, die FC2 en FB2 moet aanroepen. Zonder OB1 kan de PLC geen programma draaien. OB1 wordt geprogrammeerd in STL. Zie figuur 5-20.



Figuur 5-20

Het programma van OB1 bevat twee aanroepen (*call*), één naar FC2 en één naar FB2, die DB2 gebruikt. Let heel goed op de aanroep van FB2. Zodra dit wordt ingevoerd, zal de STL-editor gegevens toevoegen. Dit zijn de formele parameters van FB2. Alleen INIT_SQ moet gekoppeld worden aan symbool `reset`, de rest wordt niet gebruikt. De reset kan dan gebruikt worden om de sequencer te herstarten. Zie figuur 5-21 voor de inhoud van OB1. Sla de gegevens op en sluit de editor.

```

OB1 : "Main Program Sweep (Cycle)"
Comment:

Network 1 : Title:
Comment:

CALL FC 2

Network 2 : Title:
Comment:

CALL FB 2 , DB2
OFF_SQ :=
INIT_SQ := "reset"
ACK_EF :=
S_PREV :=
S_NEXT :=
SW_AUTO :=
SW_TAP :=
SW_MAN :=
S_SEL :=
S_ON :=
S_OFF :=
T_PUSH :=
S_NO :=
S_MORE :=
S_ACTIVE:=
ERR_FLT :=
AUTO_ON :=
TAP_ON :=
MAN_ON :=

```

Figuur 5-21

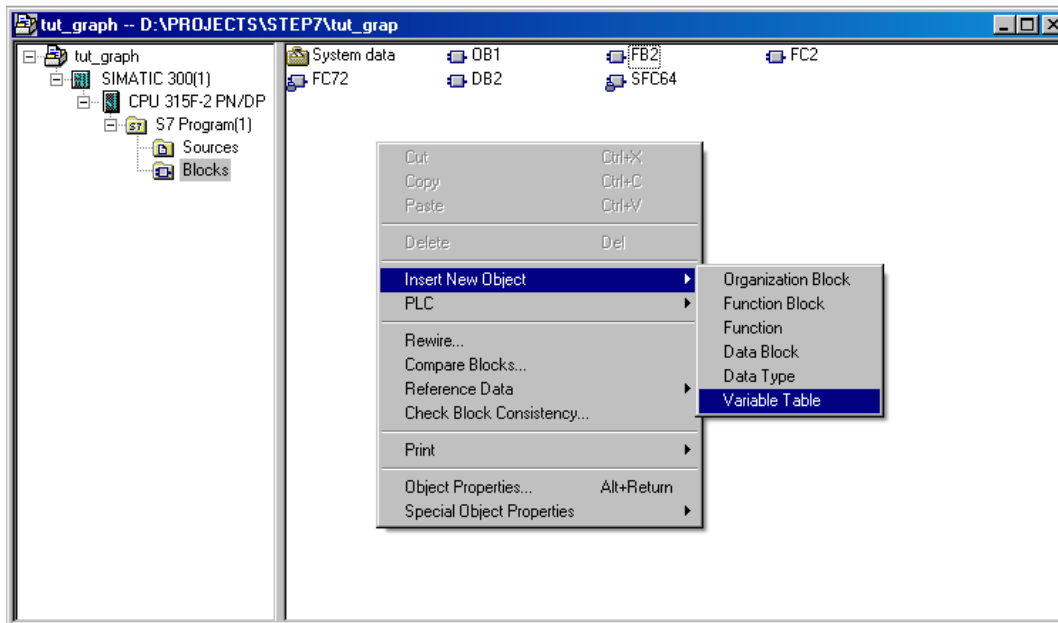
5.11 Laden PLC met programma

Het laden van het programma gaat zoals is beschreven in hoofdstuk 4. Zorg ervoor dat de PLC in STOP staat. Selecteer de blokken OB1, FC2, FB2, DB2 en FC72 en laadt ze in de PLC. Blok SFC64 is reeds in de PLC aanwezig.

5.12 Variabelentabel

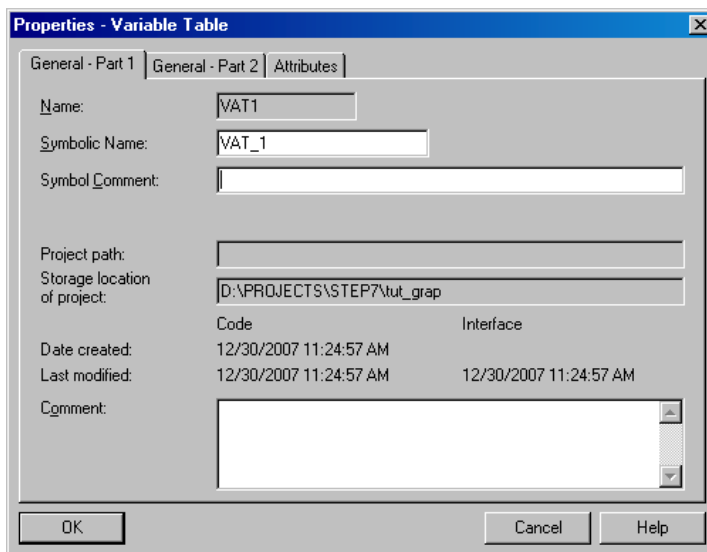
Met een variabelentabel is het makkelijk om variabelen (adressen) in de PLC te monitoren. Ze kunnen dan ook gewijzigd worden, zelfs ingangen (denk hierbij aan de *Process Image Table*). Maak in het projectscherm een tabel aan met **Insert New Object→Variable Table**. Zie figuur 5-22.

Noot: de variabelentabel kan niet in de PLC geladen worden.



Figuur 5-22

In figuur 5-23 zijn de eigenschappen weergegeven. Vul in naar eigen inzicht en klik op **OK**.



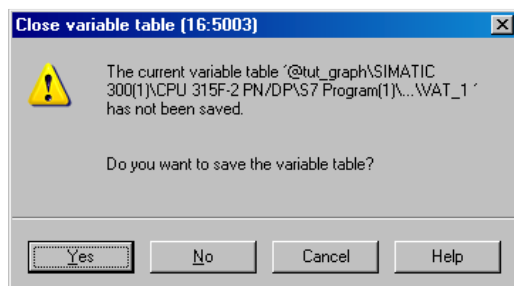
Figuur 5-23

Nadat de eigenschappen zijn ingevuld, moeten de variabelen worden ingevoerd. Dit kan door symbolen in te vullen, of direct adressen. Daarna kan voor variabele aantal_gedrukt het afbeeldformaat op DEC gezet worden. Zie figuur 5-24

Sla de tabel op via het menu **Table**→**Exit**. Er verschijnt een dialoog als in figuur 5-25.

	Address	Symbol	Display format	Status value	Modify value
1	MW 20	"aantal_gedrukt"	DEC		
2	Q 8.2	"auto_rood"	BOOL		
3	Q 8.3	"auto_geel"	BOOL		
4	Q 8.4	"auto_groen"	BOOL		
5	Q 8.6	"voet_rood"	BOOL		
6	Q 8.7	"voet_groen"	BOOL		
7	I 8.7	"reset"	BOOL		true
8					

Figuur 5-24



Figuur 5-25

5.13 Monitoren van het programma en variabelen

Het S7-Graph-programma kan gemonitord worden, d.w.z. gevolgd worden. De S7-Graph-editor bevat hiervoor een speciale modus. Open de editor door op FB2 te dubbelklikken. Aan de bovenrand is een pictogram met een brilletje te vinden. Klik hier één keer op. De editor zal nu communiceren met de PLC en weergeven in welke stap de sequencer zich bevindt (groen gekleurde stap). Ook de conditie van de transitie wordt weergegeven. Zie figuur 5-26.

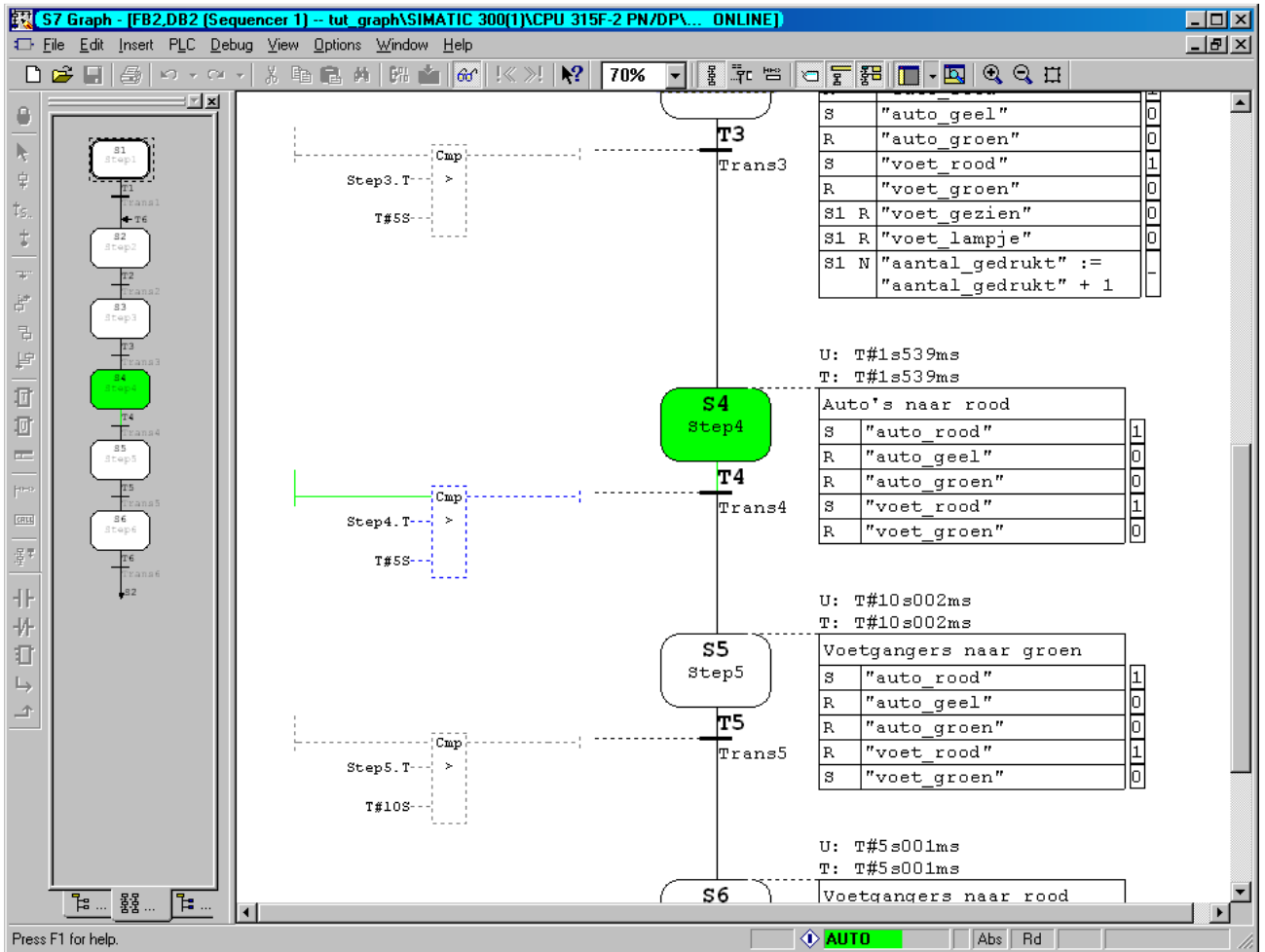
De variabelen kunnen op eenzelfde wijze gemonitord worden. Open de variabelentabel VAT1 en klik op het brilletje. De variabelen worden dan continue bijgewerkt met de meest actuele waarden.

Het is ook mogelijk om waarden *in* een variabele te plaatsen, bijvoorbeeld een merker of een uitgang. Ingangen kunnen ook worden aangepast, want de aanpassing wordt gedaan in de *process image table* en natuurlijk niet op de ingangen zelf. Zo kan bijvoorbeeld een reset van de sequencer worden uitgevoerd vanaf de PC als er geen fysieke schakelaar op de ingang is aangesloten. Vul in de kolom *Modify Value* het getal 1 of het woord `true` in en klik op de pictogram met de bliksemschicht (naast het brilletje).

Zie voor een overzicht figuur 5-27.

5.14 Extra opdracht

In bijlage 2 is het programma uitgebreid met een knippermogelijkheid. Voer dit gedeelte nog in.



Figuur 5-26

The screenshot shows the Variable Manager window for the program. It displays a table of variables with their addresses, symbols, display formats, and current status values.

Address	Symbol	Display format	Status value	Modify value
M/W 20	"aantal_gedrukt"	DEC	6	
Q 8.2	"auto_rood"	BOOL	true	
Q 8.3	"auto_geel"	BOOL	false	
Q 8.4	"auto_groen"	BOOL	false	
Q 8.6	"voet_rood"	BOOL	true	
Q 8.7	"voet_groen"	BOOL	false	
I 8.7	"reset"	BOOL	false	true

Figuur 5-27

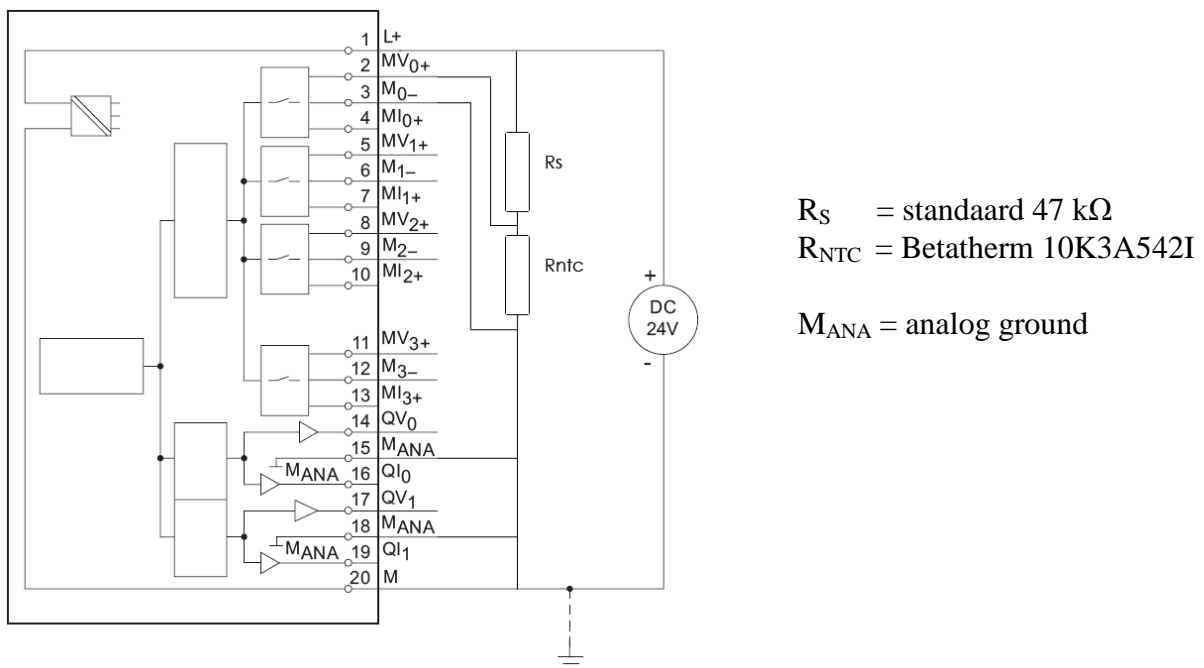
6. Tutorial SCL

In deze tutorial wordt de programmeertaal SCL (Structured Command Language)⁹ gebruikt. SCL is een variant op de IEC-taal ST¹⁰. De taal lijkt erg veel op Pascal. Het is een gestructureerde taal waarin mogelijkheden als lussen en beslissingen eenvoudig kunnen worden ingevoerd. Het werken met complexe datastructuren als arrays¹¹ en structures¹² is veel eenvoudiger dan in STL of ladder.

Voor programmeurs van andere talen zoals C zal het werken met SCL eenvoudig uitpakken. Deze tutorial laat slechts een klein deel van SCL zien. Er wordt niet ingegaan op de syntax en mogelijkheden. Zie hiervoor de SCL-handleiding.

6.1 Analoge ingang en temperatuur meten

In deze tutorial wordt gebruik gemaakt van een analoge ingang. Er wordt een thermometer gebouwd met behulp van een NTC-weerstand. De module is al aangesloten op een 24 V voeding. Tevens wordt er een alarmering ingebouwd, die in hoofdstuk 7 weer wordt gebruikt.



Figuur 6-1

⁹ http://www.automation.siemens.com/simatic/industriesoftware/html_76/produkte/software-s7-scl.htm

¹⁰ http://en.wikipedia.org/wiki/Structured_text

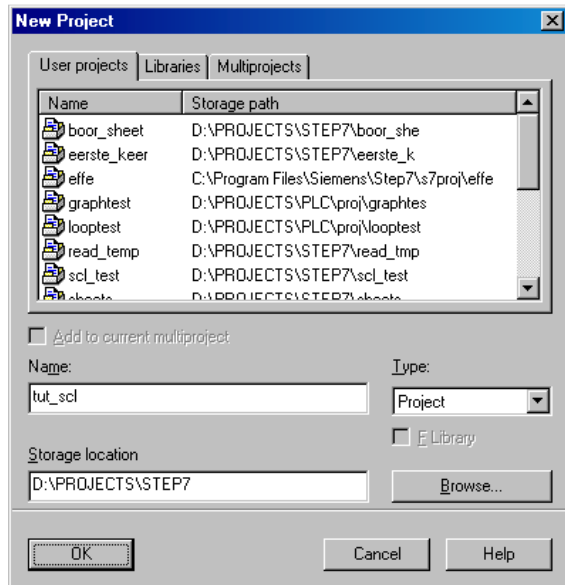
¹¹ <http://en.wikipedia.org/wiki/Array>

¹² <http://en.wikipedia.org/wiki/Struct>

Let op: op de analoge ingang mag maximaal 10 V aangeboden worden!
Zie bijlage 4 voor de afleiding van de formule voor de temperatuur.
Als alternatief voor de NTC-weerstand kan een potmeter gebruikt worden.

6.2 Project aanmaken

Maak een project aan met de naam `tut_scl`. Zie figuur 6-2.



Figuur 6-2

6.3 Configuratie PLC

De configuratie van de PLC kan opnieuw worden ingevoerd of, net zoals in de vorige tutorial, worden opgehaald uit de PLC. Dit keer wordt wel de naam van de PLC veranderd in `SIMATIC_300_THR`. Later wordt dit project uitgebreid met een HMI-station (zie hoofdstuk 7).

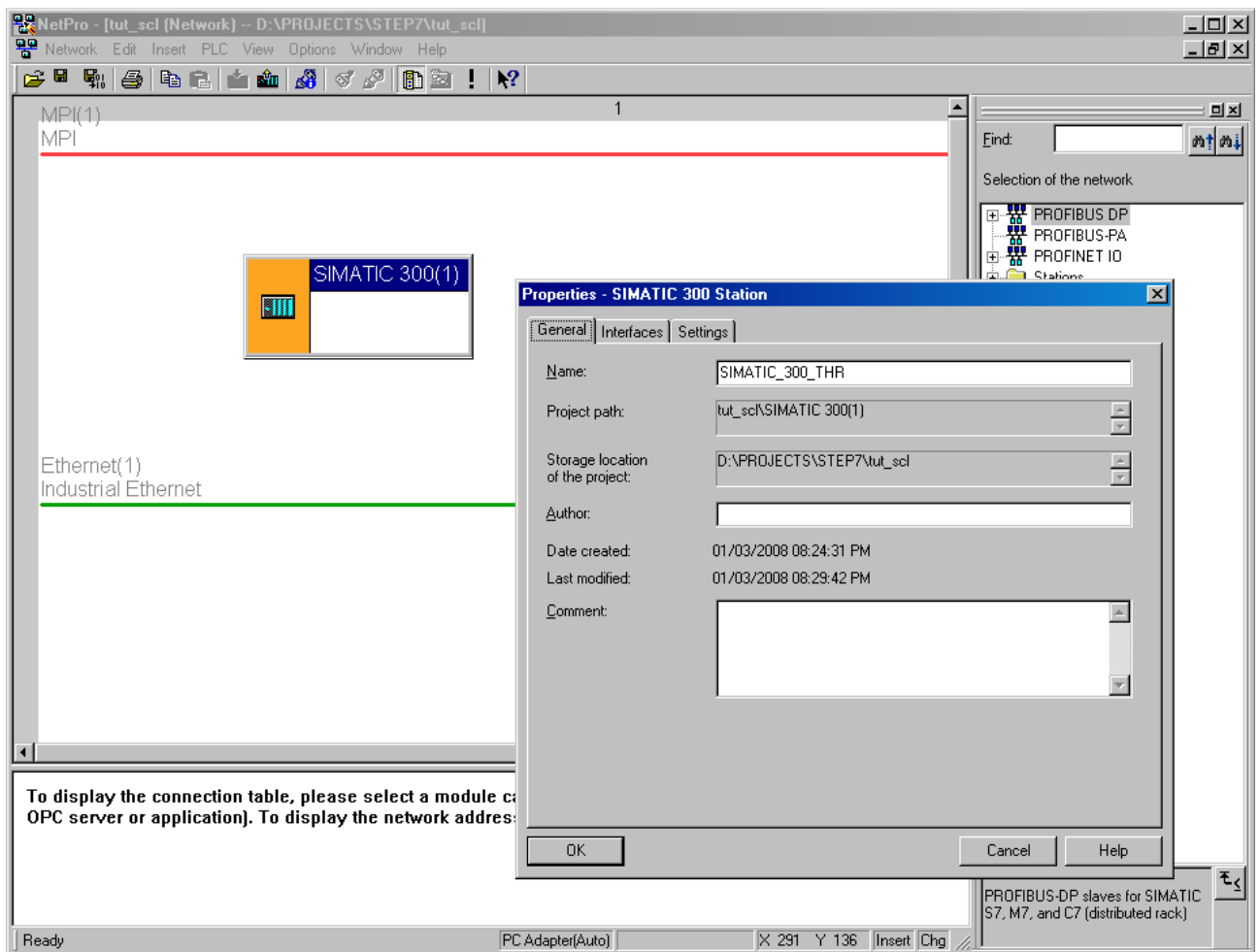
Voer nu eerst de configuratie in of haal deze op. Start daarna het netwerkconfiguratieprogramma `Netpro` (dubbelklik op de MPI-bus). Selecteer de PLC-naam en klik op de rechter muisknop. Kies dan **Object Properties**. Vul dan de naam in. Zie figuur 6-3.

Soms verschijnt er een dialoog zoals figuur 6-4. Klik dan op **OK**.

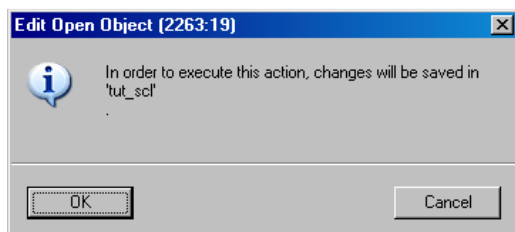
Daarna moet de verandering opgeslagen worden. Doe dit via **Network** → **Save and Compile**. Er verschijnt een dialoog (figuur 6-5). Selecteer **Save and check everything**.

Figuur 6-6 geeft een beeld van hoe het netwerk er na de naamsverandering uitziet.

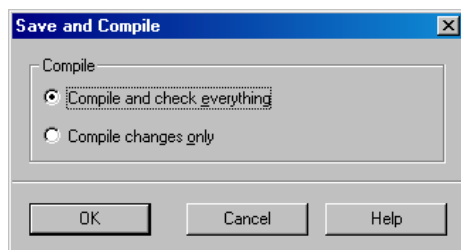
Download de configuratie naar de PLC met **PLC** → **Download to Current Project** → **Selected Stations**. Aangezien de PLC-naam anders is dan die in de PLC is opgeslagen, zal figuur 6-7 verschijnen. Hierin is het verschil te zien. Klik op **OK**.



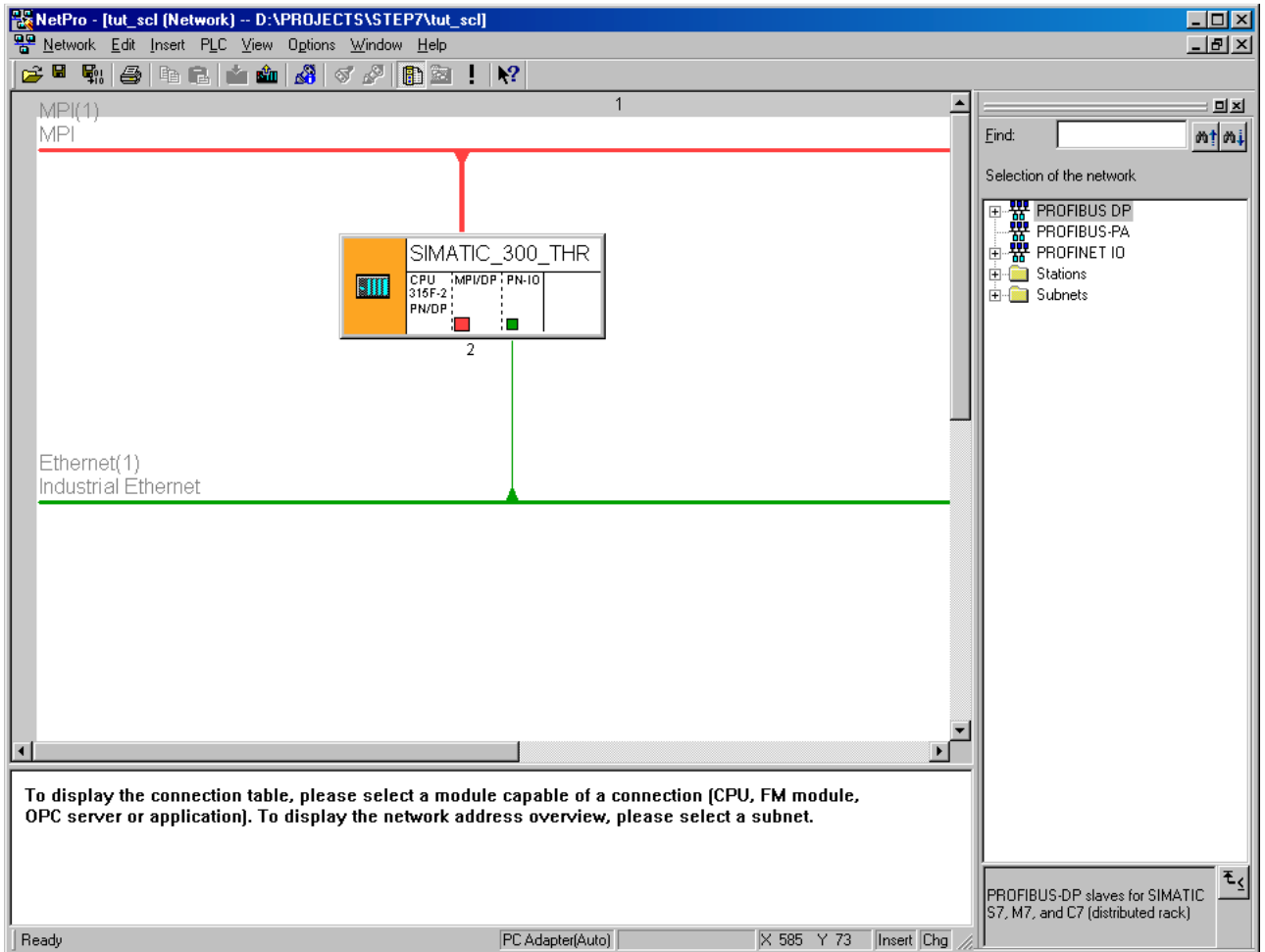
Figuur 6-3



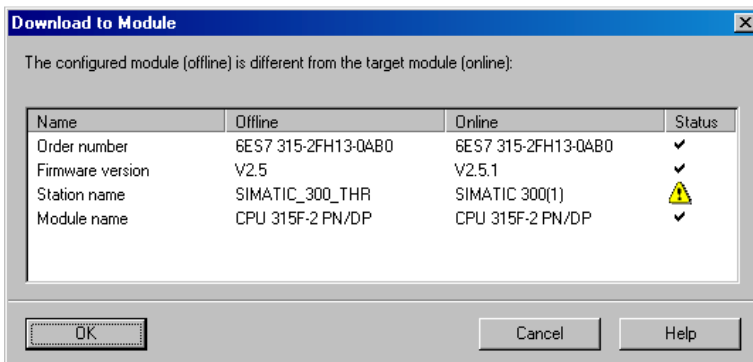
Figuur 6-4



Figuur 6-5



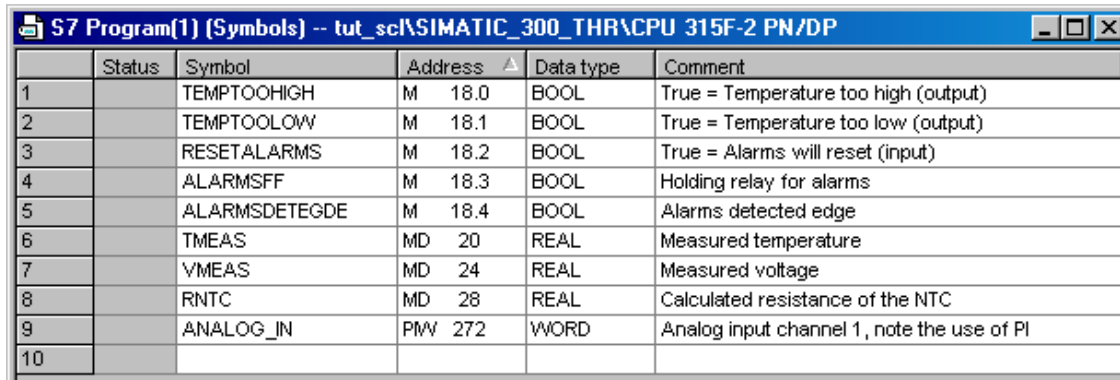
Figuur 6-6



Figuur 6-7

6.4 Symbolenlijst

De symbolenlijst is erg kort. In figuur 6-8 is de invulling te zien. Let op het adres van symbool ANALOG_IN. Dit is PIW272 en niet IW272. Maak de symbolenlijst aan op de gebruikelijke manier. Verder is er een aantal merkerbits gedefinieerd. Die worden gebruikt voor de alarmering. Merk op dat de symboolnamen volledig uit hoofdletters bestaan. Dit is niet verplicht.

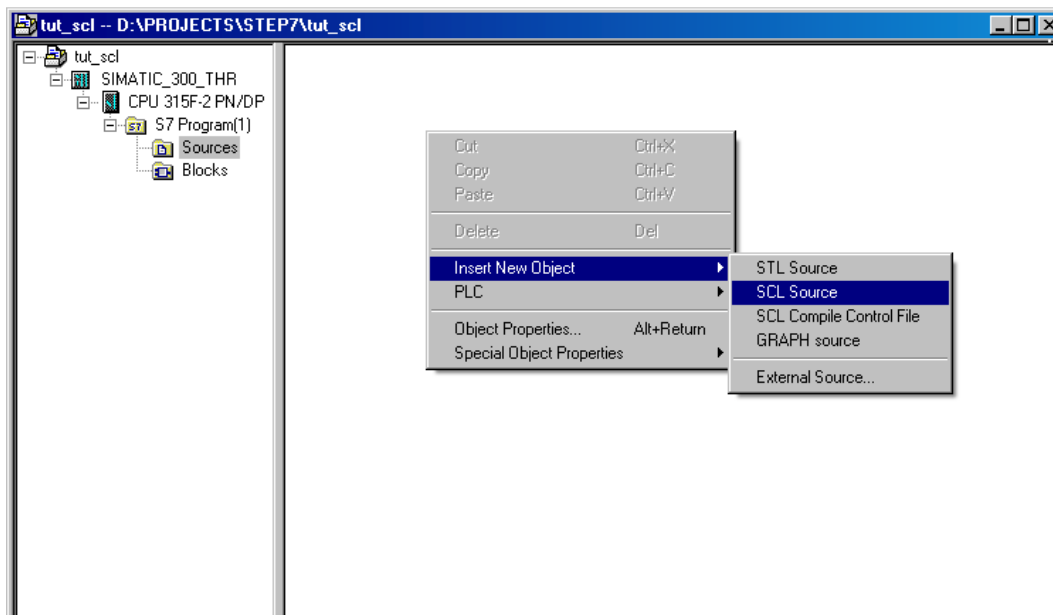


	Status	Symbol	Address	Data type	Comment
1		TEMPTOOHIGH	M 18.0	BOOL	True = Temperature too high (output)
2		TEMPTOLOW	M 18.1	BOOL	True = Temperature too low (output)
3		RESETALARMS	M 18.2	BOOL	True = Alarms will reset (input)
4		ALARMSFF	M 18.3	BOOL	Holding relay for alarms
5		ALARMSDETEGDE	M 18.4	BOOL	Alarms detected edge
6		TMEAS	MD 20	REAL	Measured temperature
7		VMEAS	MD 24	REAL	Measured voltage
8		RNTC	MD 28	REAL	Calculated resistance of the NTC
9		ANALOG_IN	PIW 272	WORD	Analog input channel 1, note the use of PI
10					

Figuur 6-8

6.5 SCL bronbestand aanmaken

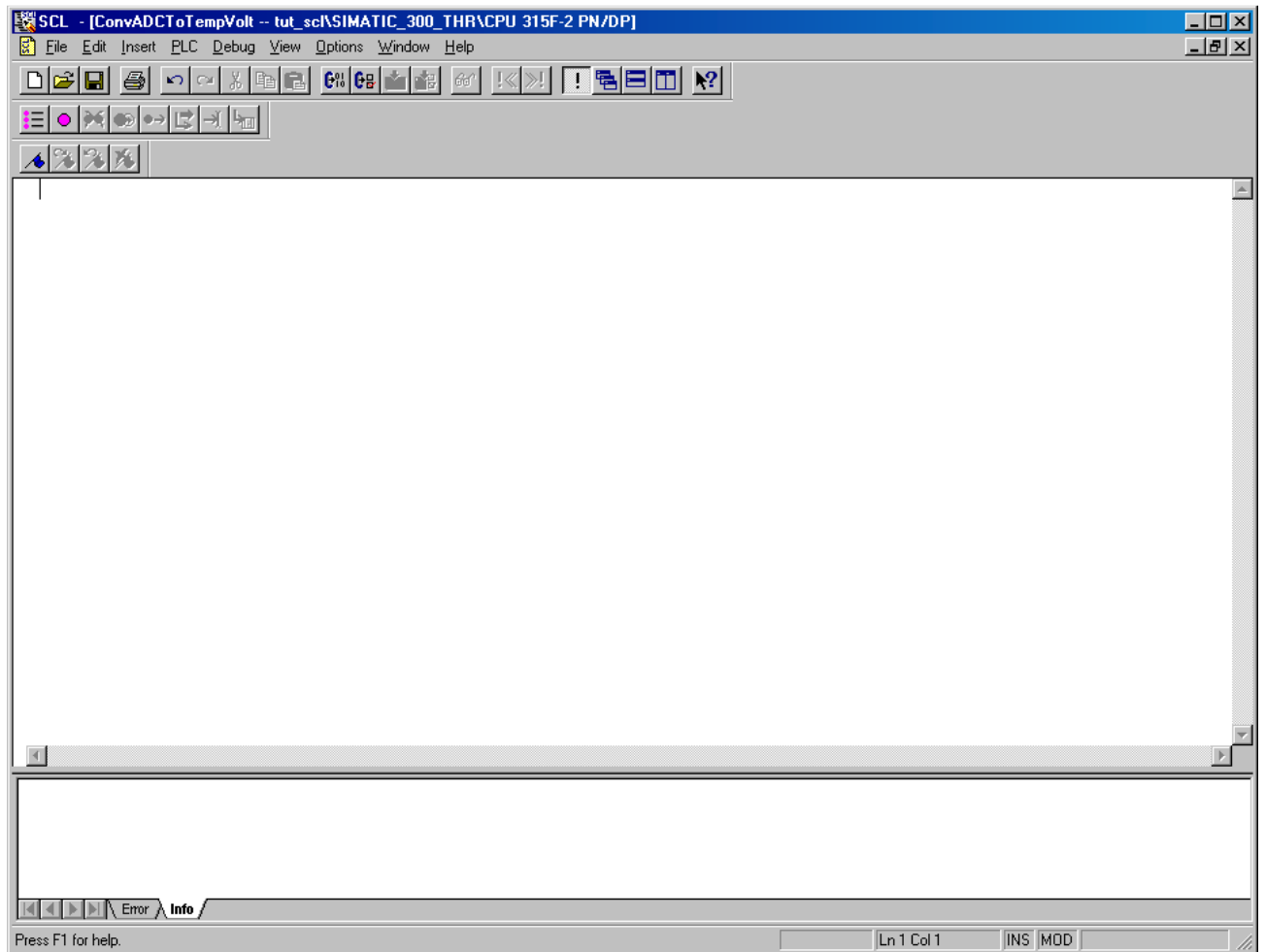
Het programmeren van SCL-code kan niet direct via de bekende weg. Bij het aanmaken van een OB, FB of FC is de taaloptie SCL niet beschikbaar. Het aanmaken van een SCL-bronbestand gaat via Sources. In het rechter paneel van het projectscherm moet op de rechter muisknop worden geklikt en selecteer dan **Insert New Object** → **SCL Source** (zie figuur 6-9). Er verschijnt een object met de naam SCL Source (1). Verander deze naam in ConvADCTempVolt.



Figuur 6-9

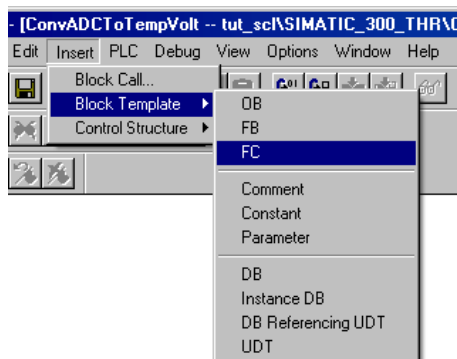
6.6 Invoeren SCL-code

Dubbelklik op de naam om de SCL-editor¹³ te starten. Er volgt een leeg scherm. Zie figuur 6-10.



Figuur 6-10

Voeg nu via het menu **Insert**→**Block Template**→**FC** de structuurcode voor een FC (figuur 6-11). Deze structuur is te zien in figuur 6-12.



Figuur 6-11

¹³ Met deze editor kunnen ook STL- en Graph-bestanden worden gemaakt, dit wordt niet verder uitgewerkt.

```

FUNCTION FCxxx : INT

VAR_TEMP
  // Temporary Variables

END_VAR

  // Statement Section
  ;
  FCxxx := 100;
END_FUNCTION

```

Figuur 6-12

Het programma komt in FC3 en er komt een REAL terug, dus moet dat worden aangepast. In figuur 6-13 is de complete code. In bijlage 5 is de code nog eens terug te vinden, maar nu inclusief commentaar.

```

FUNCTION FC3 : REAL                                // Function 3

CONST
  VPOWER := 24.0;                                  // Power Supply
  VANALOG_MAX := 10.0;                             // Max analog voltage
  ADCVALUE_AT_VANALOG_MAX := 27648;              // ADC value code @ ...
  A := 0.0235664;                                  // A parameter in Ohm
  B := 3863.65;                                    // B parameter in K
  RS := 47000.0;                                   // Series resistor of 47 KOhm
  RIN := 100000.0;                                 // Input resistance of analog input
  K_TO_C := 273.15;                                // Convert to Celsius
END_CONST

VAR_INPUT
  ADC_VALUE : WORD;                                // The input from the analog channel
END_VAR

VAR_OUTPUT
  VMEAS : REAL;                                   // Voltage measured
  RNTC : REAL;                                    // Calculated resistance of NTC
END_VAR

VAR_TEMP
  Z : REAL;                                       // Conversions
  ZI : INT;
END_VAR

BEGIN
  // ADC value to integer
  ZI := WORD_TO_INT(ADC_VALUE);
  // Voltage measured
  VMEAS := (1.0*ZI)/ADCVALUE_AT_VANALOG_MAX * VANALOG_MAX;
  // Calculate Z as function of ...
  Z := VMEAS/VPOWER;

  // Rntc in Ohms. Function found with the aid of Maple 10.
  RNTC := -(VMEAS*RS*RIN / (-VPOWER*RIN+VMEAS*RIN+VMEAS*RS));

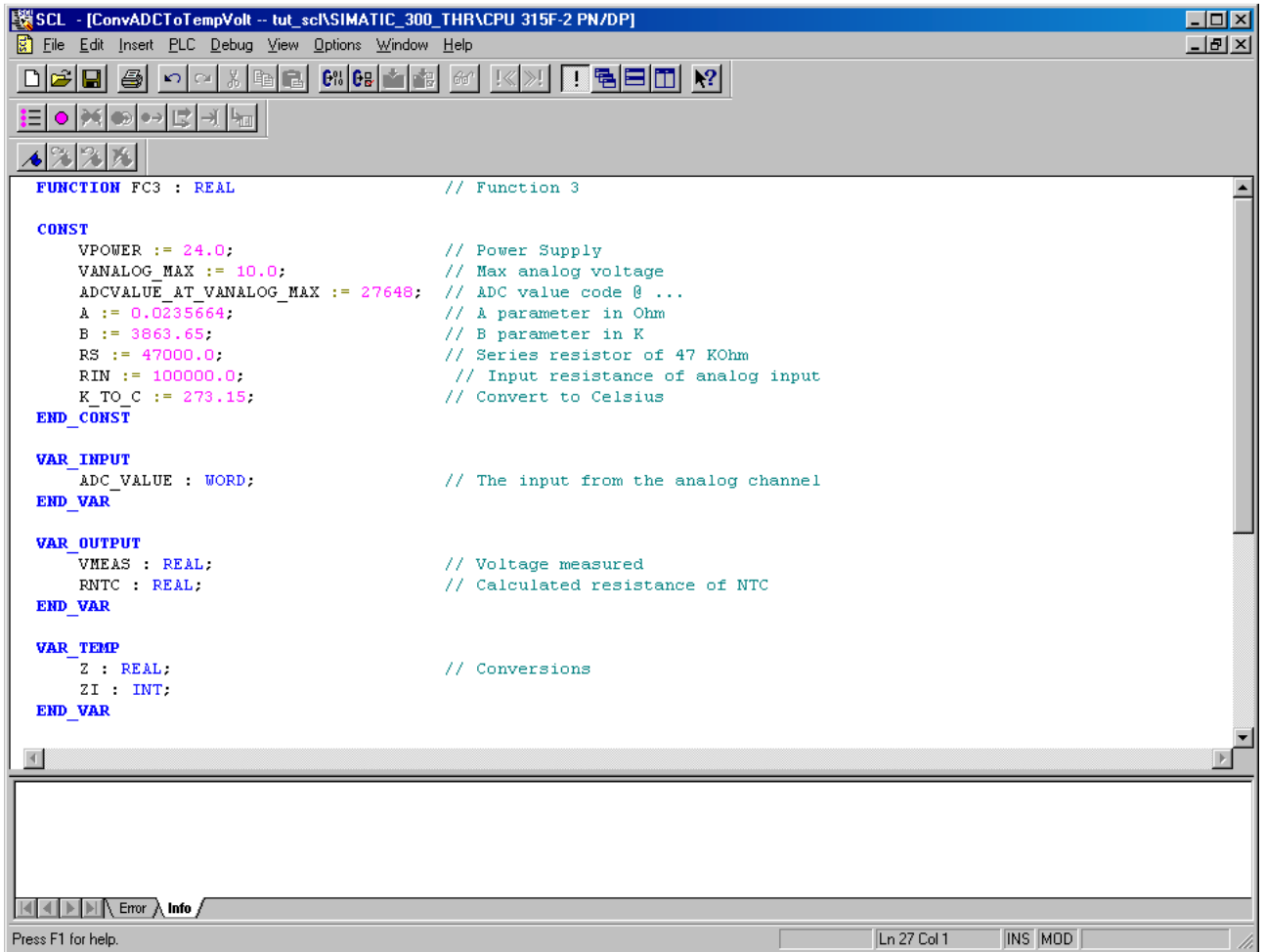
  // Temperature in degree C. Function found with the aid of Maple 10.
  // This is the return value of the function.
  FC3 := B / LN( -1.0*Z*RS*RIN / (A*(-RIN+Z*RS+Z*RIN)) ) - K_TO_C;

END_FUNCTION

```

Figuur 6-13

In figuur 6-14 is een deel van het programma terug te vinden in de SCL-editor.

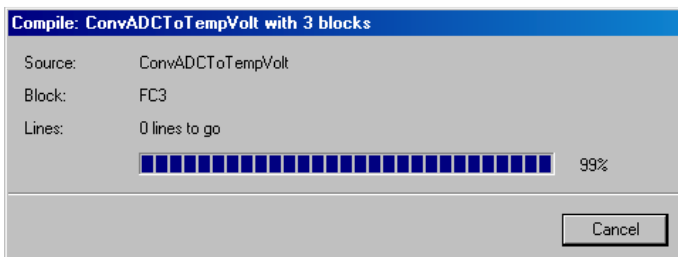


Figuur 6-14

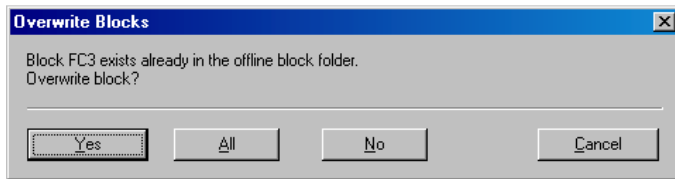
6.7 Compileren SCL-code

Nu het programma ingevoerd en compleet is, moet het gecompileerd worden. De *compiler* vertaalt de SCL-code naar STL, de assemblertaal voor de PLC.

Selecteer het menu **File**→**Compile** (sneller is via **Ctrl+B**). De compiler gaat nu aan het werk. De compileervoortgang wordt weergegeven, zie figuur 6-15. Als blok FC3 al bestaat (en dat komt natuurlijk best vaak voor), zal gevraagd worden of dit blok moet worden overschreven. Klik dan op **Yes** of **All**. Zie figuur 6-16.

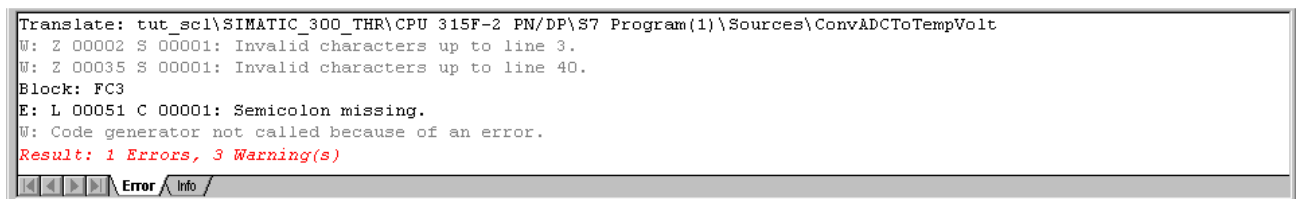


Figuur 6-15



Figuur 6-16

Eventuele fouten worden gerapporteerd en moeten verbeterd worden. Zie voor een voorbeeld van een mislukte compilatie figuur 6-17.

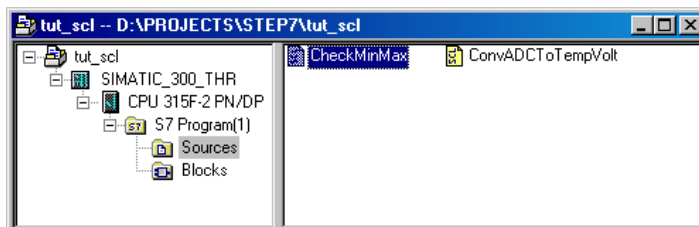


Figuur 6-17

6.8 Tweede SCL-bronbestand

Er wordt nu een tweede SCL-bestand aangemaakt. De code wordt in FC4 geplaatst en zet twee merkerbits indien de temperatuur te hoog of te laag is.

Maak eerst een nieuw SCL-bestand aan en noem dit CheckMinMax. Zie figuur 6-18.



Figuur 6-18

De code is weergegeven in figuur 6-19. Let vooral op de teruggaveparameter van FC4, deze is van het type VOID, oftewel er is geen teruggaveparameter. De teruggave wordt geregeld via twee VAR_OUTPUT-parameters.

De code van CheckMinMax is zeer eenvoudig. Het zijn slechts twee simpele IF-statements. Hiermee is ook goed te zien dat SCL gebruikt kan worden voor simpele beslissingen. Natuurlijk had de code veel eenvoudiger gekund en doorgewinterde PLC-programmeurs hadden dit in LAD opgelost.

In bijlage 5 is de code nog eens afgebeeld.

```

FUNCTION FC4 : VOID

CONST
  C_TEMPTOOHIGH := 30.0;           // Higher bound temperature
  C_TEMPTOOLOW  := 15.0;           // Lower bound temperature
END_CONST

VAR_INPUT
  TEMPERATURE : REAL;             // The current temperature
END_VAR

VAR_OUTPUT
  TEMPTOOHIGH : BOOL;             // Signals temperature too high
  TEMPTOOLOW  : BOOL;             // Signals temperature too high
END_VAR

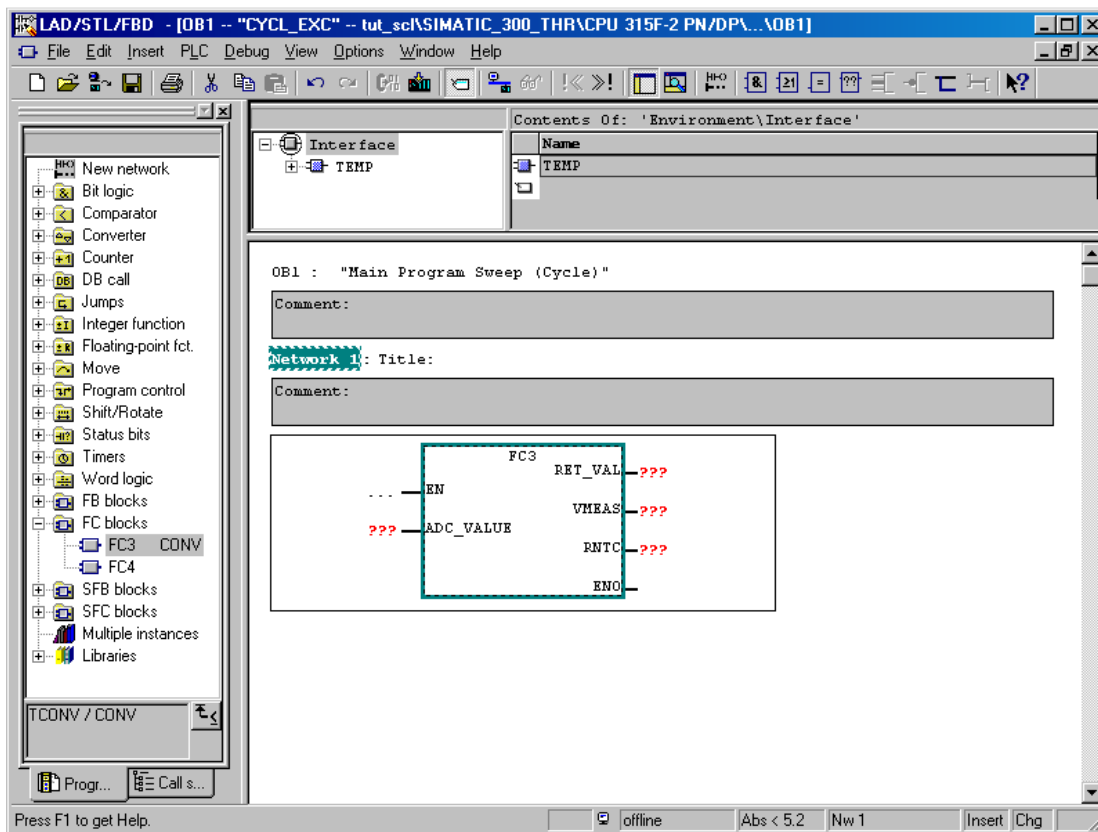
BEGIN
  IF TEMPERATURE >= C_TEMPTOOHIGH THEN // If current temperature too high...
    TEMPTOOHIGH := true;           // signal it.
  ELSE
    TEMPTOOHIGH := false;
  END_IF;
  IF TEMPERATURE <= C_TEMPTOOLOW THEN // If current temperature too low...
    TEMPTOOLOW := true;           // signal it.
  ELSE
    TEMPTOOLOW := false;
  END_IF;
END_FUNCTION

```

Figuur 6-19

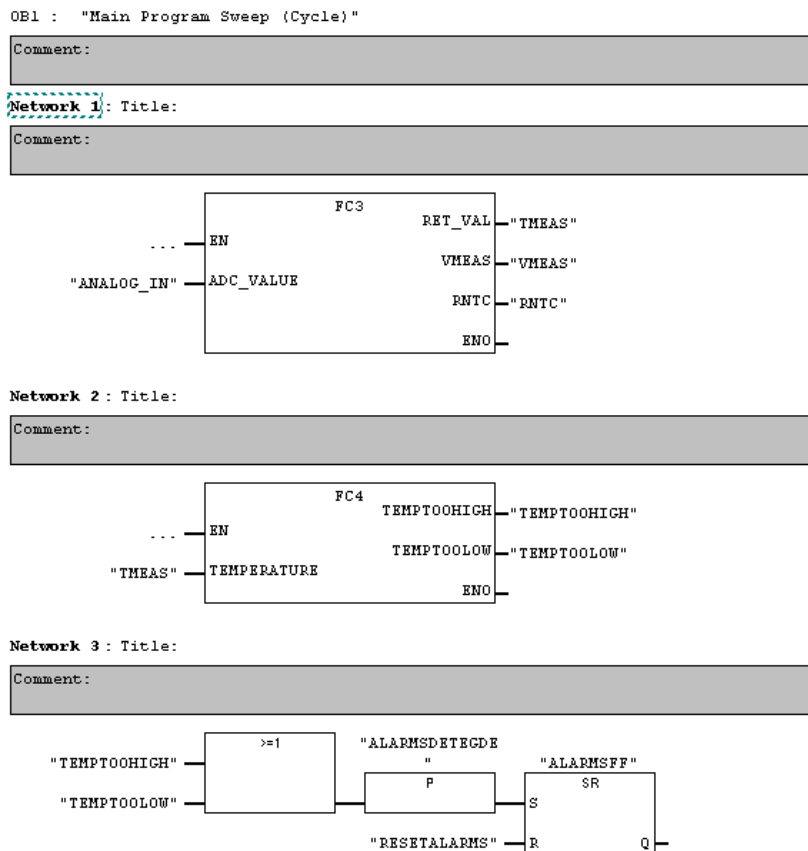
6.9 OB1 programma

Nu de SCL-routine klaar zijn moet ze via OB1 worden aangeroepen met de juiste parameters. Maak OB1 op de bekende wijze aan (via Blocks) en start de LAD-editor. Sleep vanuit linker kader onder FC Blocks FC3 naar edit-kader. Zie figuur 6-20.



Figuur 6-20

Na het plaatsen van het FC-blok geeft de editor automatisch de formele parameters weer met rode vraagtekens op de plaatsen waar de actuele parameters moeten worden ingevoerd. In figuur 6-21 zijn deze ingevuld. Tevens wordt FC4 ook aangeroepen en is er een stukje FBD gemaakt voor de alarmering¹⁴.



Figuur 6-21

Sla OB1 op de bekende manier op.

6.10 Laden PLC programma

Dit gaat weer op de bekende wijze. Zorg ervoor dat de PLC in STOP staat en laadt OB1, FC3 en FC4.

6.11 Monitoren temperatuur

Het monitoren kan eenvoudig met een variabelentabel. Maak de tabel aan op de bekende wijze en ga naar de monitor-modus (klik op het brilletje). Let op dat een aantal gegevens in floating point genoteerd zijn. Zie figuur 6-22.

¹⁴ De lezer wordt uitgedaagd om netwerk 3 te bestuderen.

	Address	Symbol	Display format	Status value	Modify value
1	MD 20	"TMEAS"	FLOATING_POINT	31.64032	
2	MD 24	"VMEAS"	FLOATING_POINT	3.101852	
3	MD 28	"RNTC"	FLOATING_POINT	7543.841	
4	PIW 272	"ANALOG_IN"	HEX	VW#16#2180	
5	M 18.0	"TEMPTOOHIGH"	BOOL	true	
6	M 18.1	"TEMPTOLOW"	BOOL	false	
7	M 18.2	"RESETALARMS"	BOOL	false	true
8	M 18.3	"ALARMSFF"	BOOL	true	
9					

Figuur 6-22

In figuur 6-22 is te zien dat de temperatuur boven de 30 °C is gestegen. De merkers TEMPTOOHIGH en ALARMSFF zijn beide hoog (`true`). Er is nu een alarm getriggerd. Als de temperatuur nu onder de 30 °C zakt, zal TEMPTOOHIGH weer laag (`false`) worden, maar ALARMSFF blijft hoog. Om deze laag te krijgen moet RESETALARMS op even hoog en daarna weer laag worden. Dit kan makkelijk via de VAT. Merk op dat de alarmering via merkers gaat en niet via uitgangen. In de tutorial van WinCC Flexible wordt nog terug gekomen op de alarmen.

7. Tutorial WinCC flexible

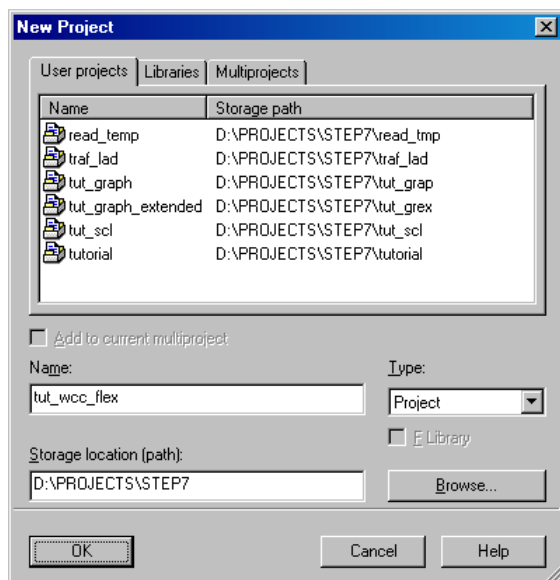
Met behulp van WinCC flexible kan informatie uit de PLC op een zogenaamde Operator Panel worden afgebeeld en het is mogelijk om het panel informatie in de PLC te brengen. Op deze manier kan een fabriek schematisch worden afgebeeld. Dit visualiseren, controleren en besturen wordt ook wel SCADA¹⁵ genoemd, *Supervisory Control And Data Acquisition*.

In het PLC-lab zijn er echter geen operator panels, hiervoor wordt de PC gebruikt. Deze tutorial borduurt voort op de SCL-tutorial.

In deze tutorial worden slechts een paar mogelijkheden van WinCC Flexible gedemonstreerd. Als eerste wordt in het STEP7-project een HMI-station toegevoegd. Daarna wordt het geconfigureerd, zoals een PLC. Vervolgens worden een paar uitvoervelden en een knop aangemaakt in de WinCC Flexible editor en als laatste wordt de Runtime gestart.

7.1 Aanmaken nieuw project¹⁶

Begin een nieuw project zoals is aangegeven in het vorige hoofdstuk en noem het project `tut_wincc_flex`. Zie figuur 7-1. Vul de bekende gegevens in.



Figuur 7-1

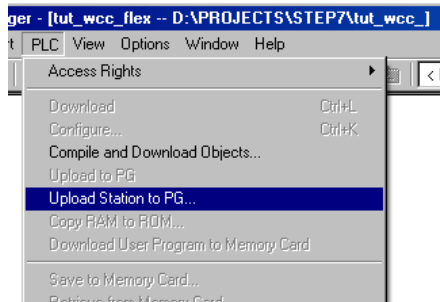
¹⁵ <http://nl.wikipedia.org/wiki/SCADA>

¹⁶ Deze stap kan worden overgeslagen als deze tutorial direct volgt op de SCL-tutorial. De projectnaam blijft dan `tut_scl`. Ga dan verder met paragraaf 7.3.

7.2 Configuratie uploaden

In het vorige hoofdstuk is een nieuwe configuratie van de PLC opgezet en geladen in de PLC. Deze configuratie kan teruggehaald worden uit de PLC en zo gebruikt worden in het nieuwe project. Hiervoor moeten de configuratiegegevens *ge-upload* worden vanuit de PLC naar het project. Hierbij worden ook alle programma's mee gekopieerd (OB, FC etc).

Doe een upload via het menu **PLC**→**Upload Station to PG**. Zie figuur 7-2.

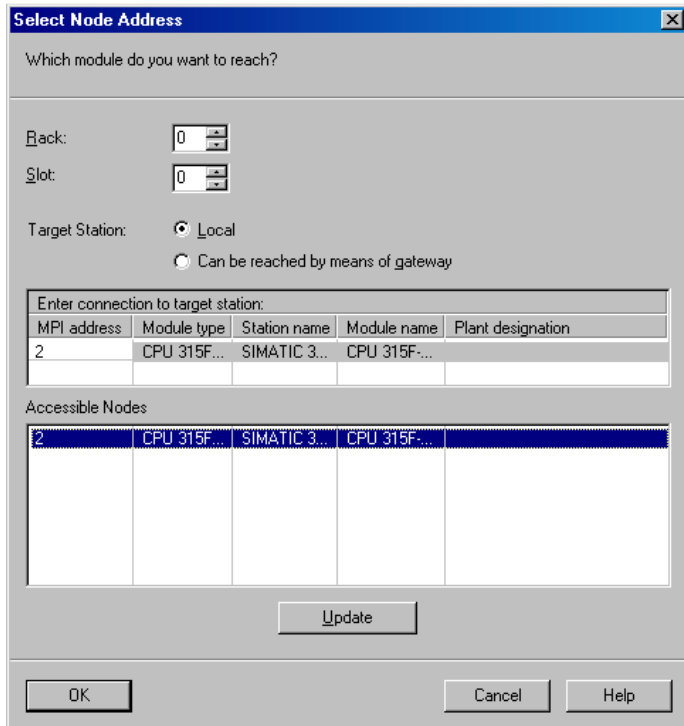


Figuur 7-2

Er verschijnt nu een scherm met daarin alle aangesloten PLC's. Midden onderaan het scherm verschijnt een knop **View**. Als er geen PLC's zichtbaar zijn, klik hier dan op. Na een tijdje verschijnt onder *Accessible Nodes* de PLC die op het practicum is aangesloten. Selecteer de gegeven PLC en klik op **OK**. Zie figuur 7-3.

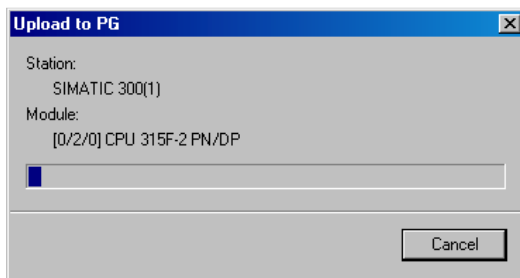
Let er wel op dat alleen de blokken (*Blocks*) worden gekopieerd. De symbolentabel (*Symbols*) en de bronbestanden (*Sources*) worden niet gekopieerd. Dit komt omdat in de PLC alleen gecompileerde code is opgeslagen. Als er toch veranderingen moeten worden gemaakt aan de programmatuur levert dat problemen op.

Het is mogelijk om de blokken, bronbestanden en symbolentabel te kopiëren uit een ander project. Open dit andere project, selecteer *Blocks*, *Sources* en *Symbols* en sleep ze op de juiste plaats in het huidige project.



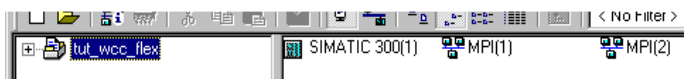
Figuur 7-3

De configuratie wordt nu uit de PLC geladen en in het project opgenomen. Zie figuur 7-4.



Figuur 7-4

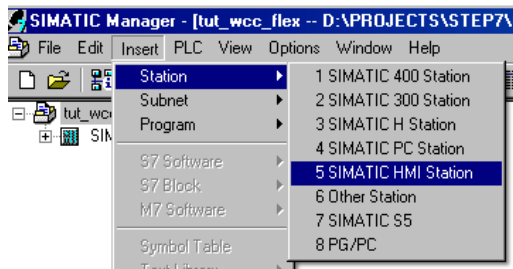
Het projectoverzicht is nu de PLC én de MPI-bus zichtbaar. Als er twee MPI-bussen zichtbaar zijn, moet er één verwijderd worden. Zie figuur 7-5 als voorbeeld.



Figuur 7-5

7.3 HMI station plaatsen

Voeg nu via het menu **Insert**→**Station**→**SIMATIC HMI Station** een HMI-station toe aan het project. Zie figuur 7-6.

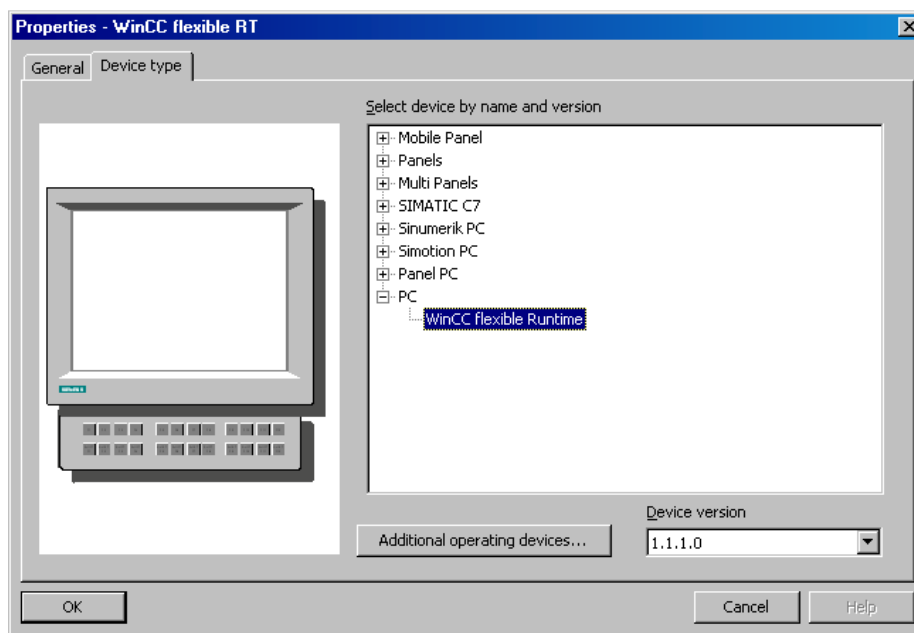


Figuur 7-6

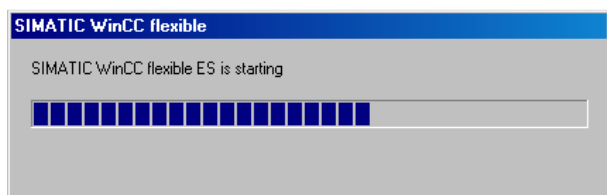
Er wordt nu een venster geopend waarin het type HMI-station kan worden gekozen. Kies hier voor **PC**→**WinCC flexible Runtime**. Zie figuur 7-7.

Nu wordt de WinCC flexible Engineering Station software gestart, zie figuur 7-8.

Let op: dit kost enige tijd. Heb geduld! Ga desnoods even koffie drinken!



Figuur 7-7



Figuur 7-8

Hierna is het HMI station toegevoegd aan het project (zie figuur 7-9).



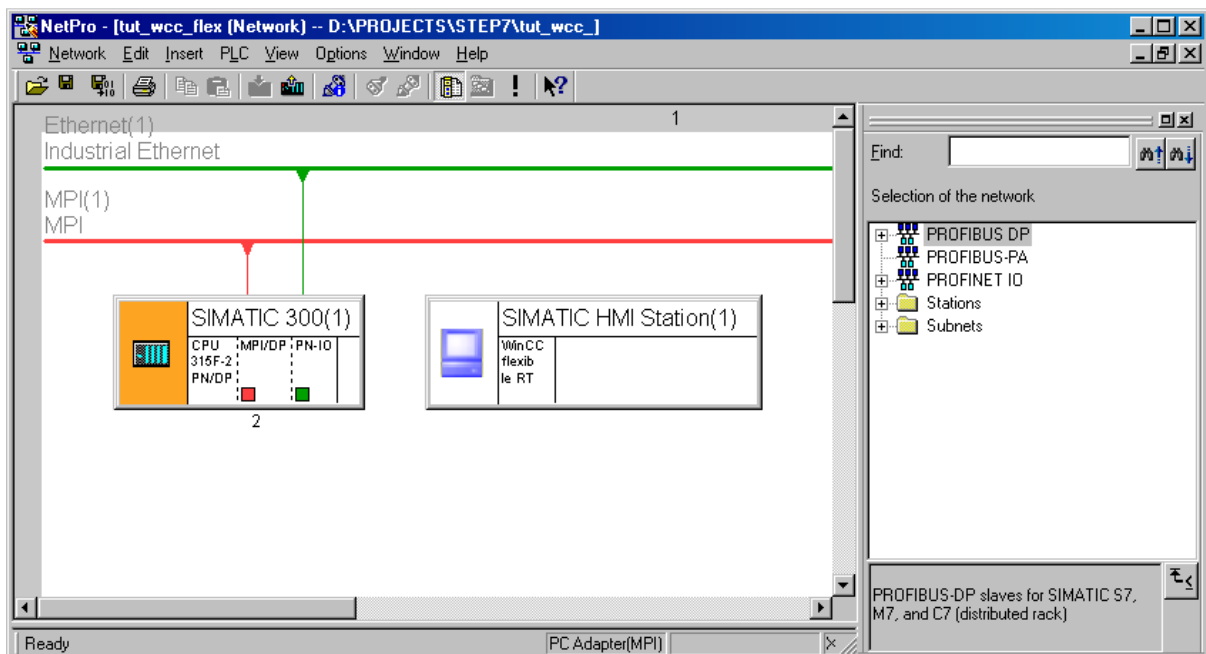
Figuur 7-9

7.4 MPI-netwerk

Nu moeten de PLC en het HMI station verbonden worden via het MPI-netwerk. Dit wordt gedaan via het programma NetPro, wat onderdeel is van STEP7. Dubbelklik op MPI (1) in het projectvenster (figuur 7-9).

Nu wordt NetPro geopend met hierin de MPI-bus, de Ethernet-bus, de PLC en het HMI-station. Deze laatste twee zijn nog niet aangesloten op het netwerk (als dat wel zo is, kan deze stap worden overgeslagen).

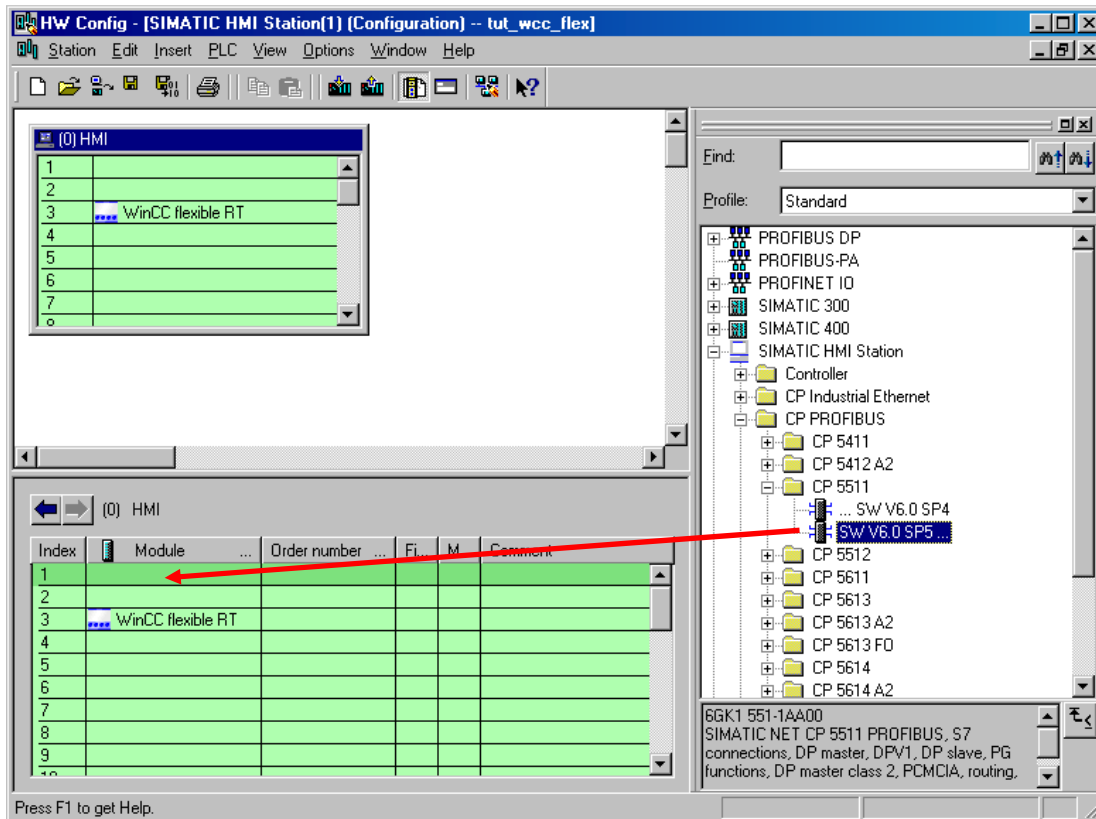
Om nu de PLC te koppelen aan de MPI-bus, moet het volgende gedaan worden: klik met de muis op de rode vierkantje en houd de muisknop ingedrukt. Sleep nu naar de MPI-bus en laat de muisknop vervolgens los. De PLC is dan verbonden met het netwerk. Zie figuur 7-10.



Figuur 7-10

7.5 HMI-station configureren

Het HMI-station moet, net als de PLC, geconfigureerd worden. Dubbelklik nu op het HMI station. De hardwareconfiguratie kan nu worden ingevoerd. Zie figuur 7-11.



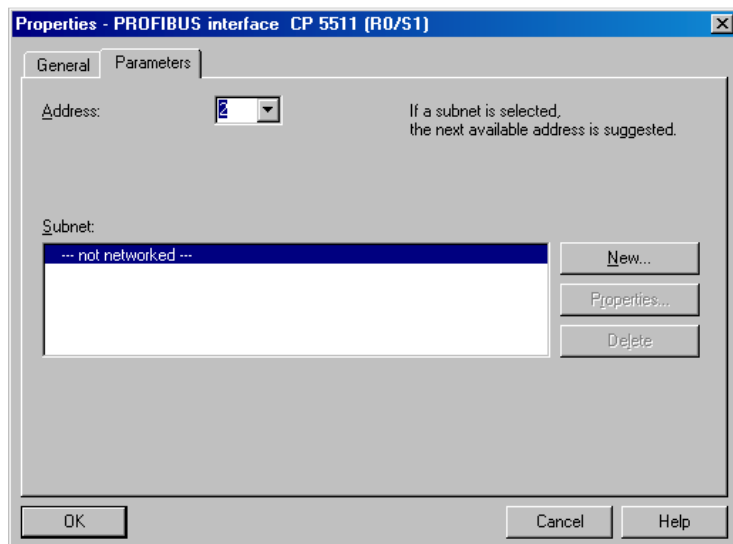
Figuur 7-11

Noot: Het idee is dat een SIMATIC HMI-station net als een S7-300 een rack en slots heeft. Deze worden Virtual Rack en Virtual Slots genoemd. Hierdoor blijven een HMI station en een S7-300 op dezelfde manier te configureren.

Zoek in het venster aan de rechterkant naar de CP5511-netwerkkkaart via SIMATIC HMI Station→CP Profibus→CP5511 en kies de tweede regel. Sleep de geselecteerde regel naar Index 1. Zie de rode pijl.

Noot: Merk op dat in de PC, want daar draait straks de WinCC Flexible Runtime op, helemaal geen CP5511-kaart zit! De PC is met de PLC verbonden via een USB-adapter. De PC zorgt er automatisch voor dat de communicatie op juiste wijze in gang wordt gezet.

Zodra dat gebeurd is, volgt een eigenschappen scherm voor Profibus-instellingen van de CP5511-netwerkkkaart. Klik nu op **OK** zonder iets te veranderen. Zie figuur 7-12.



Figuur 7-12

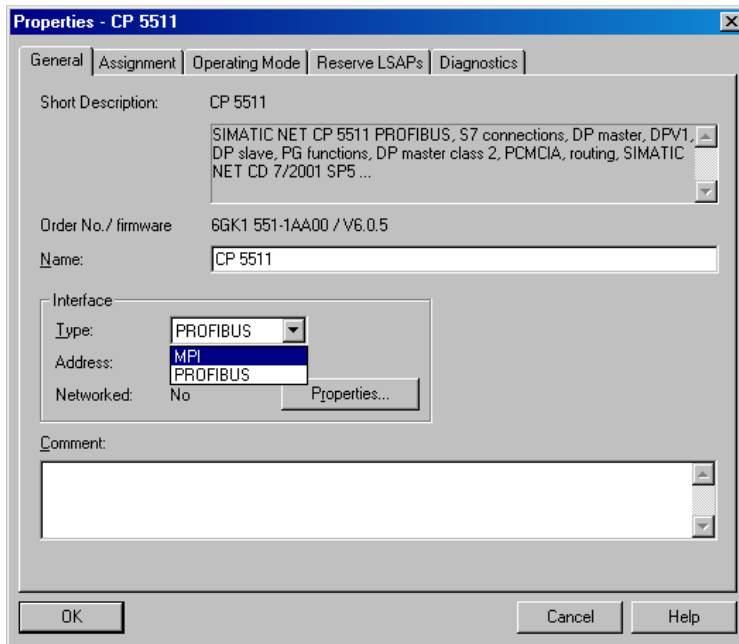
De CP5511 is nu zichtbaar op Index 1 in het configuratiescherm. Dubbelklik hier nu op om weer de eigenschappen te openen, maar kies nu bij *Interface* voor de MPI-bus. Zie figuur 7-13.

Er wordt dan een bevestigingsvenster geopend met de melding dat de Profibus-instellingen verloren zullen gaan en dat de netwerkkaart aan de MPI-bus wordt gekoppeld. Klik op *Yes*. Zie figuur 7-14.

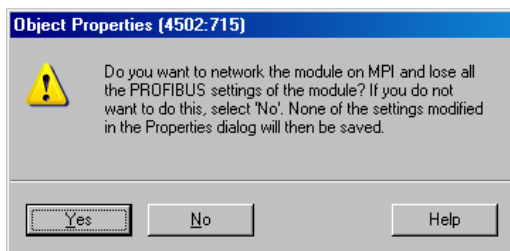
Klik vervolgens in het *Interface* kader op **Properties**. Nu worden de eigenschappen van de MPI-interface getoond. Als subnet moet nu MPI (1) gekozen worden. In het voorbeeld wordt verder netwerkadres 3 gekozen, maar dit mag ook een ander zijn met uitzondering van 0 en 2. Zie figuur 7-15.

Klik nu een aantal keer op **OK** en sla de nieuwe configuratie op met **Station→Save and Compile**. Sluit de hardwareconfiguratie af via menu **Station→Exit** of **Alt+F4**.

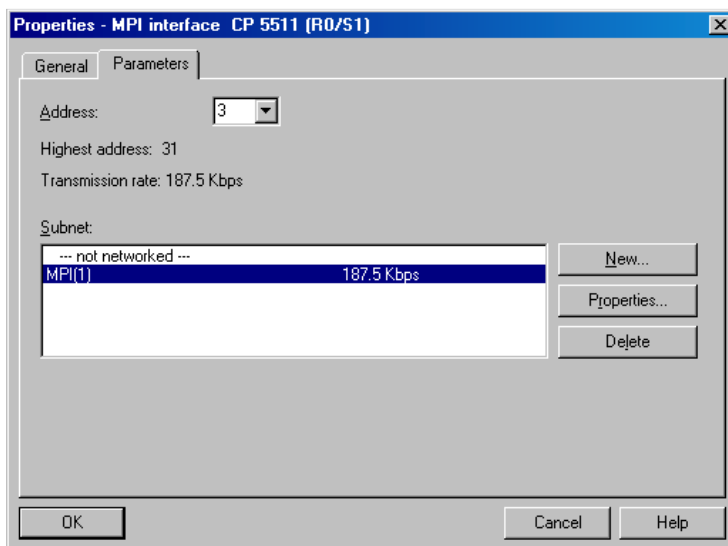
Er komt nog een melding of de wijzigingen moeten worden opgeslagen en daarna wordt een compileerslag gedaan.



Figuur 7-13

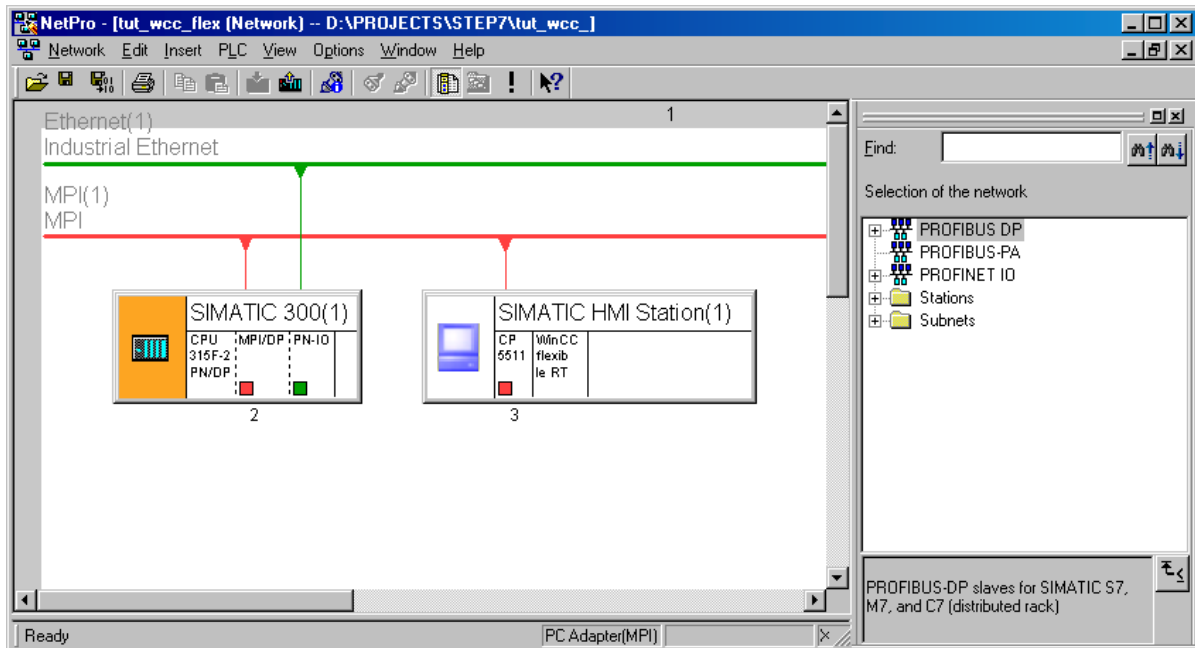


Figuur 7-14



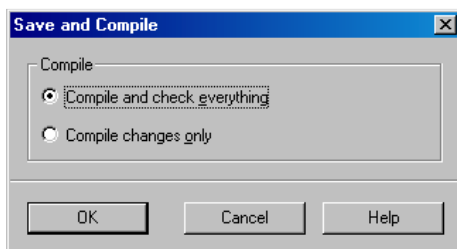
Figuur 7-15

Nu is de koppeling van het HMI-station en de MPI-bus klaar. In de volgende figuur is dat goed te zien (figuur 7-16).



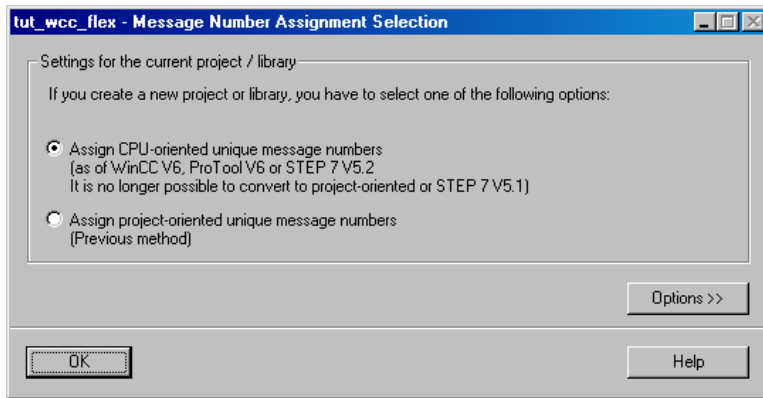
Figuur 7-16

Selecteer via het menu **Network**→**Save and Compile**. Er volgt een dialoog waarin gekozen moet worden voor **Compile and check everything**. Zie figuur 7-17.



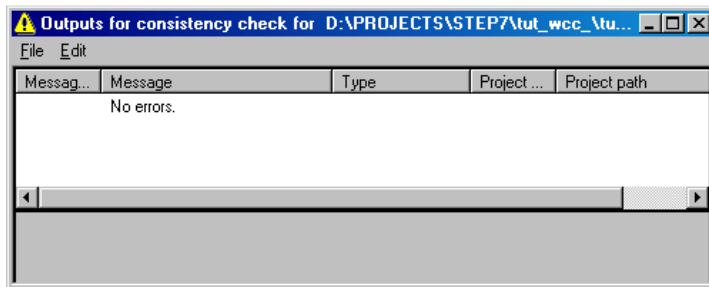
Figuur 7-17

Indien het om een nieuw opgezet project gaat, wordt nog even gevraagd hoe de berichten in de CPU gecodeerd moeten worden (figuur 7-18). Selecteer hier de eerste optie. Deze stap wordt overgeslagen als het al een keer eerder gevraagd is, bijvoorbeeld bij een oud project.



Figuur 7-18

Als alles goed gaat volgt figuur 7-19.



Figuur 7-19

7.6 Symbolenlijst

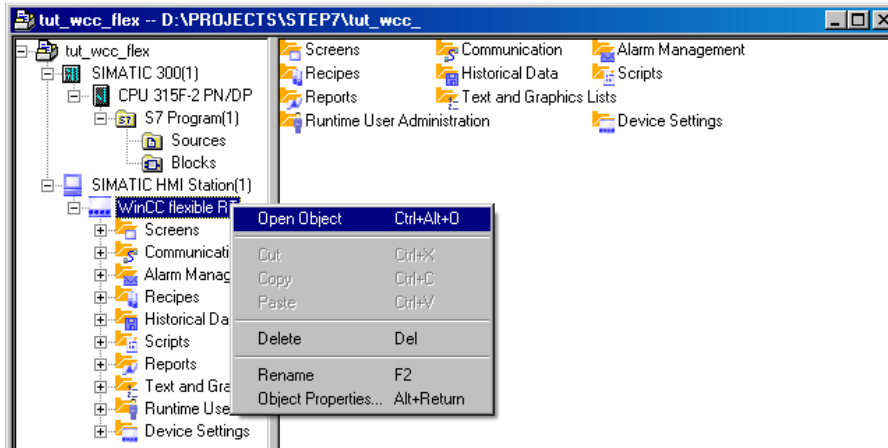
Net als in de vorige tutorial moet een symbolenlijst gemaakt worden. Zie ook paragraaf 6.4. In figuur 7-20 worden de symbolen weergegeven.

	Status	Symbol	Address	Data type	Comment
1		TEMPTOOHIGH	M 18.0	BOOL	True = Temperature too high (output)
2		TEMPTOOLOW	M 18.1	BOOL	True = Temperature too low (output)
3		RESETALARMS	M 18.2	BOOL	True = Alarms will reset (input)
4		ALARMSFF	M 18.3	BOOL	Holding relay for alarms
5		ALARMSDETEGDE	M 18.4	BOOL	Alarms detected edge
6		TMEAS	MD 20	REAL	Measured temperature
7		VMEAS	MD 24	REAL	Measured voltage
8		RNTC	MD 28	REAL	Calculated resistance of the NTC
9		ANALOG_IN	PMW 272	WORD	Analog input channel 1, note the use of PI
10					

Figuur 7-20

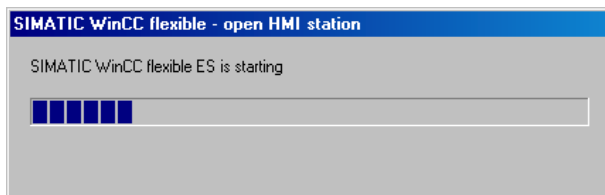
7.7 Starten WinCC flexible

Zoek in de Simatic Manager naar WinCC flexible RT. Dit is te vinden in het rechterscherm via **tut_wcc_flex**→**SIMATIC HMI Station(1)**→**WinCC flexible RT**. Selecteer deze regel en klik op de rechtermuisknop (contextmenu). Selecteer dan **Open Object**. Zie ook figuur 7-21.



Figuur 7-21

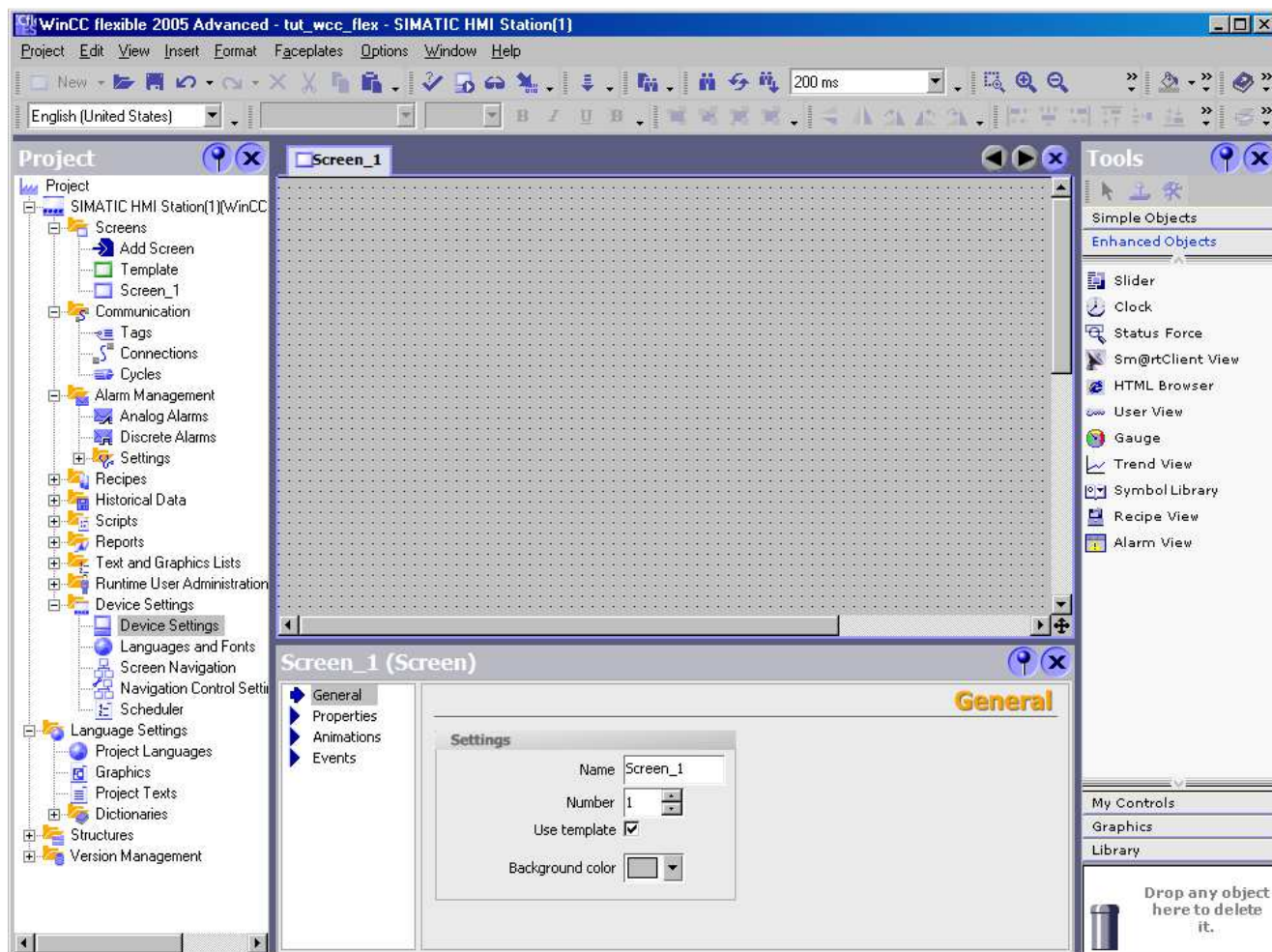
WinCC flexible wordt nu gestart. Dit is te zien aan figuur 7-22.



Figuur 7-22

Let op: heb geduld, dit kan enkele minuten duren! Tijd voor een kopje koffie!

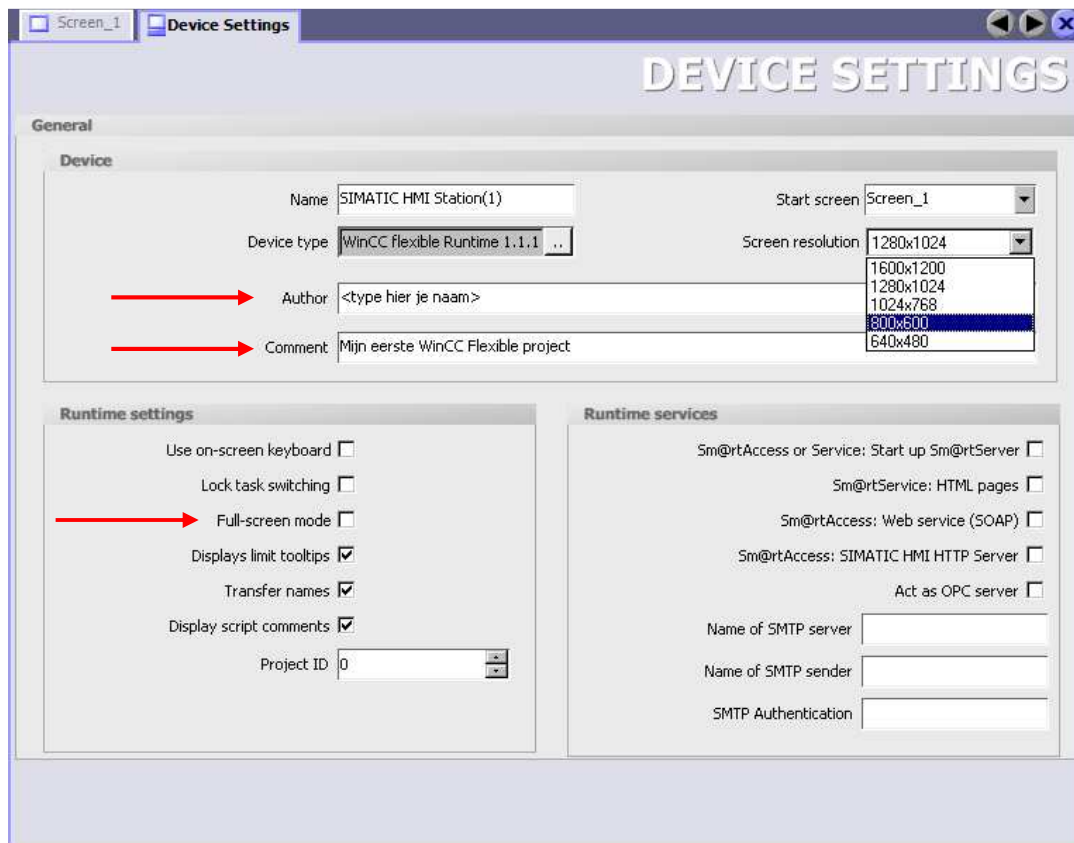
In figuur 7-23 wordt een overzicht gegeven van het WinCC Flexible scherm. Aan de linkerkant staan de onderdelen die in het project kunnen voorkomen (niet alles hoeft te worden gebruikt). Rechts staan de objecten die in het project kunnen worden gebruikt. Middenboven is het scherm wat geopend is afgebeeld (dat gekozen zijn aan de linkerkant), en onder een contextgevoelig scherm dat informatie geeft over (objecten op) het scherm erboven.



Figuur 7-23

7.8 Device settings

Klik in het linkerscherm op `Device Settings`. In het scherm verschijnt nu een lijstje met opties. Eén daarvan is weer `Device Settings`. Dubbelklik hier op en een nieuw scherm wordt geopend, waarin diverse gegevens kunnen worden ingevuld. Zo dient de schermgrootte te worden aangepast naar 800 bij 600 en kan de naam van de programmeur worden ingevuld. Zorg dat het vinkje bij `Full-screen Mode` weg is. De naam van het startscherm kan worden geselecteerd. Deze tutorial gebruikt echter maar één scherm. Zie figuur 7-24.



Figuur 7-24

7.9 Cyclustijd toevoegen

Het opvragen van de gegevens uit de PLC door het HMI-station wordt cyclisch uitgevoerd. De cyclustijd hiervoor is instelbaar. Een seconde is echter te langzaam en 100 ms is weer zo snel dat het teveel belasting voor de PLC kost. Er wordt een eigen cyclustijd aangemaakt van 200 ms.

Dubbelklik in het rechterscherf op `Communications` zodat een lijst wordt geopend. Dubbelklik op `Cycles` om het invoeren mogelijk te maken. Voeg nu een nieuwe cycle toe zoals in figuur 7-25 is opgegeven. Deze cyclustijd zal later worden gebruikt.

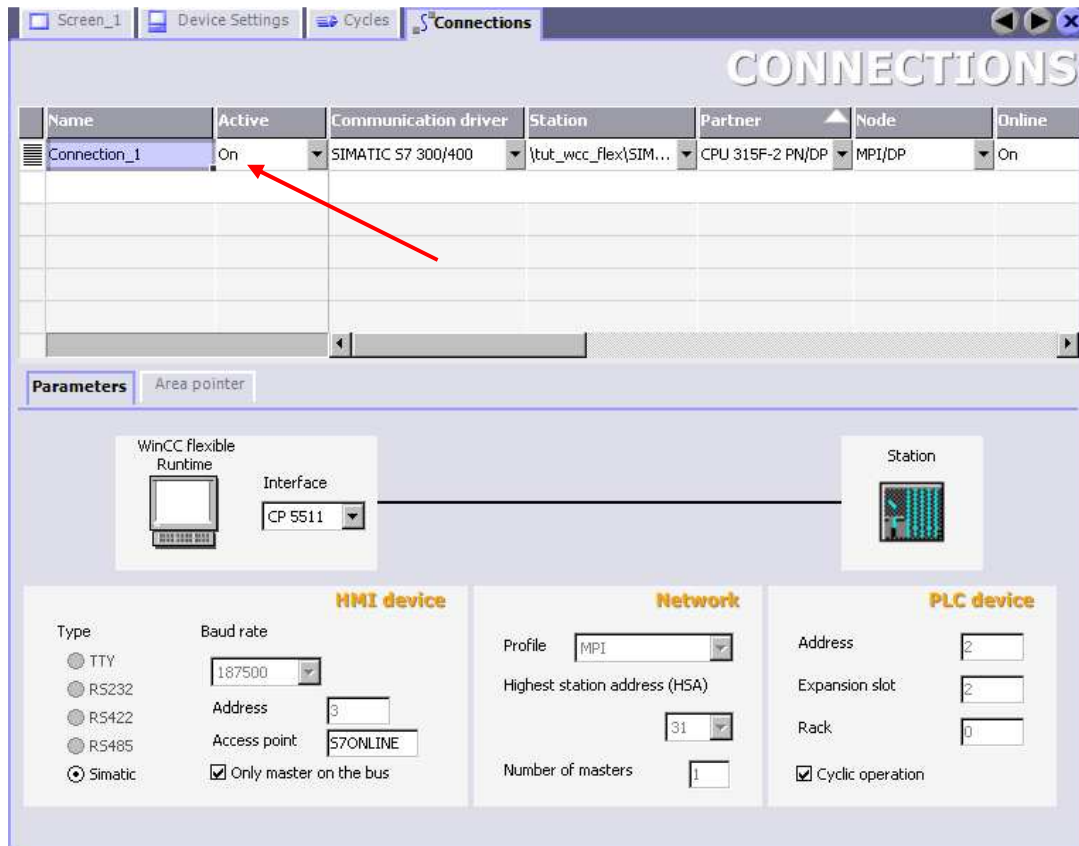
Cycle time	Cycle unit	Name	Last changed	Comment
1	Hour	1 h	10/31/2008 8:5...	
1	Minute	1 min	10/31/2008 8:5...	
1	Second	1 s	10/31/2008 8:5...	
10	Second	10 s	10/31/2008 8:5...	
100	Millisecond	100 ms	10/31/2008 8:5...	
2	Second	2 s	10/31/2008 8:5...	
5	Second	5 s	10/31/2008 8:5...	
500	Millisecond	500 ms	10/31/2008 8:5...	
200	Millisecond	200 ms	10/31/2008 8:0... Self defined 200 ms cycle	

Figuur 7-25

7.10 Verbinding met PLC maken

In hetzelfde lijstje staat ook `Connections`. Hier kunnen verbindingen met PLC's worden opgenomen, maximaal acht. In het lab is er maar één gekoppeld aan de MPI-bus.

Dubbelklik op `Connections` om het scherm te openen. Automatisch wordt de verbinding in de eerste regel gepresenteerd onder de naam `Connection_1`. Let er op dat de verbinding actief moet zijn, anders werkt het geheel niet. Zie rode pijl in figuur 7-26.



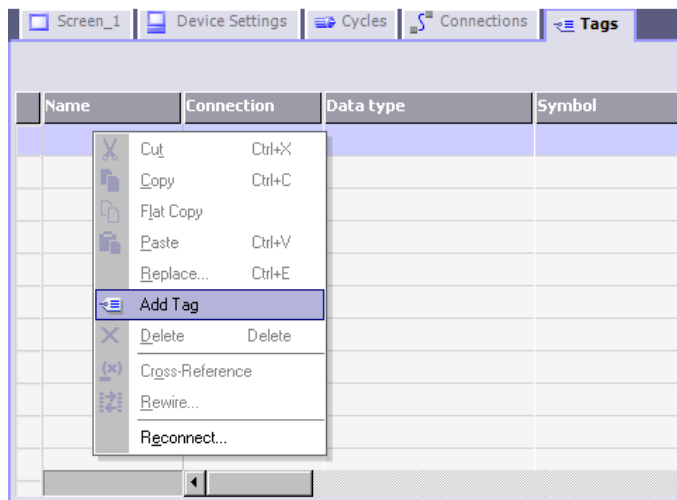
Figuur 7-26

Onder in de figuur is diverse informatie beschikbaar zoals de interface en adressen.

7.11 Tags toevoegen

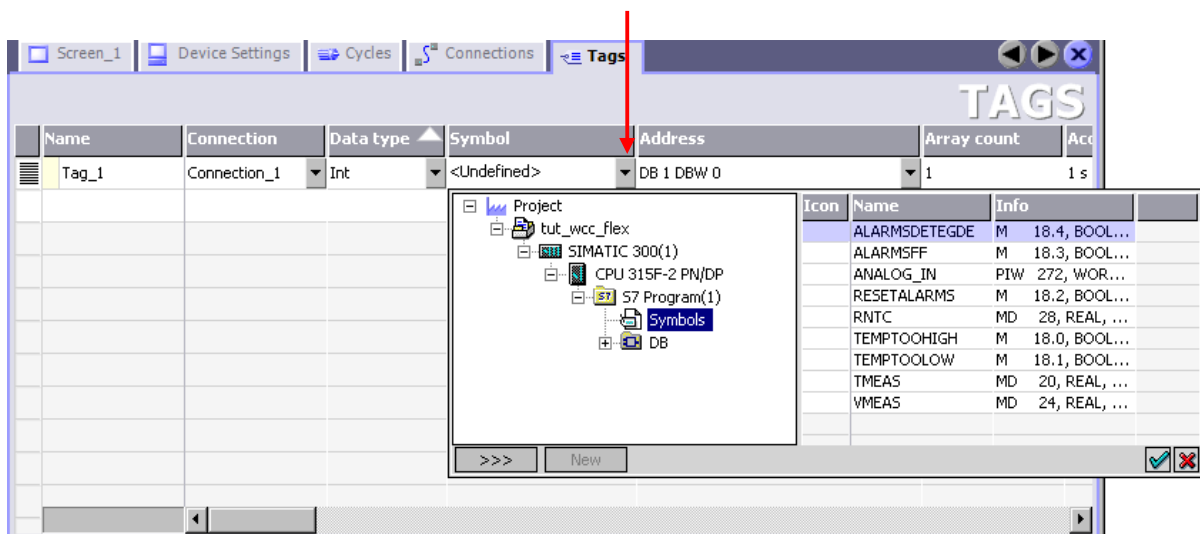
Via tags kan informatie uit de PLC gehaald worden en in het WinCC flexible systeem verder worden verwerkt. Tags kunnen direct zijn zoals `M0.0` of `I125.2`, maar er kan ook gebruik gemaakt worden van de symbolentabel.

Dubbelklik in het rechterscherm op `Tags`. Het `Tags` scherm wordt geopend. Open nu het contextmenu (rechtermuisknop) en kies `Add Tag`. Zie figuur 7-27.



Figuur 7-27

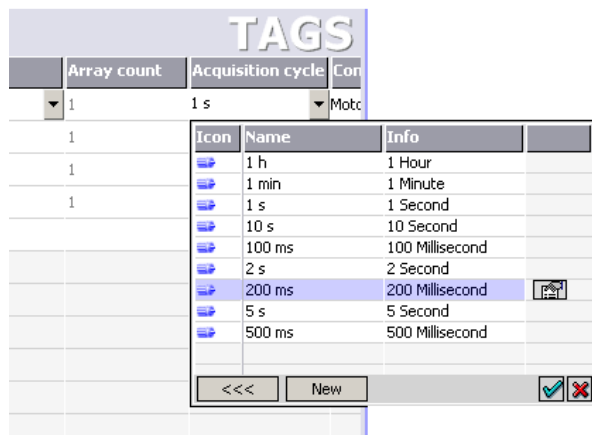
Nu wordt automatisch een standaard invulling aan de tag gegeven, die veranderd moet worden. Klik aan de rechterkant van het vakje onder de kolom Symbol op het pijltje voor een *dropdown*-menu (zie de rode pijl). Er wordt nu een klein scherm getoond met daarin het STEP7-project. Vouw dit helemaal uit totdat het onderdeel Symbols te zien is. Aan de rechterkant kan nu uit de symbolen gekozen worden. Zie figuur 7-28.



Figuur 7-28

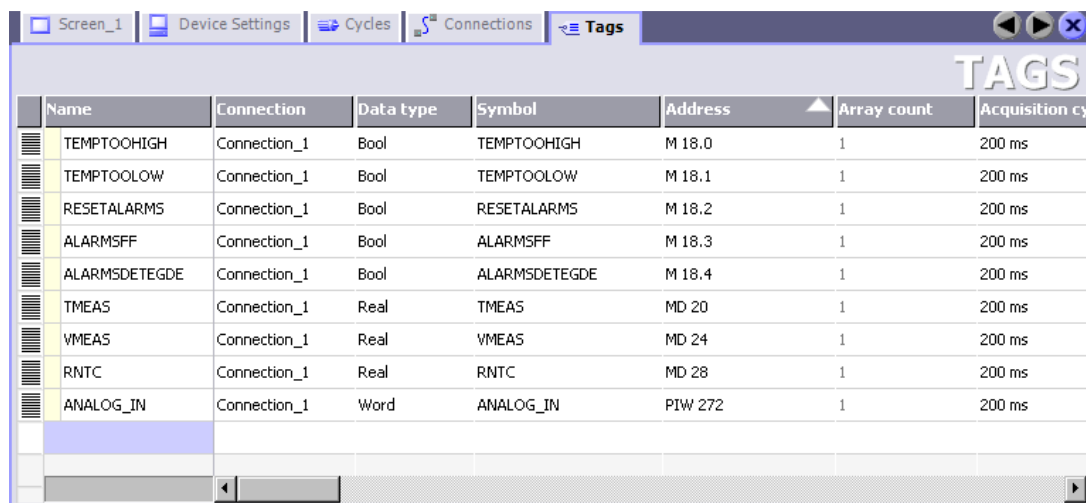
De tag-naam en het datatype worden automatisch overgenomen uit de symbolentabel. Op deze manier kunnen ook variabelen uit Data Blocks worden gekoppeld aan een tag.

Als laatste moeten en cyclustijden worden aangepast; die staan standaard op 1 seconde. Onder de kolom Acquisition cycle kunnen deze worden ingevuld. Kies voor de 200ms cyclustijd. Zie figuur 7-29.



Figuur 7-29

Voeg nu alle tags toe. Zie figuur 7-30.



Figuur 7-30

7.12 Scherm opbouwen

Het laatste onderdeel is het scherm. Hierop zijn buttons, I/O fields, sliders, switches, bars etc. te plaatsen.

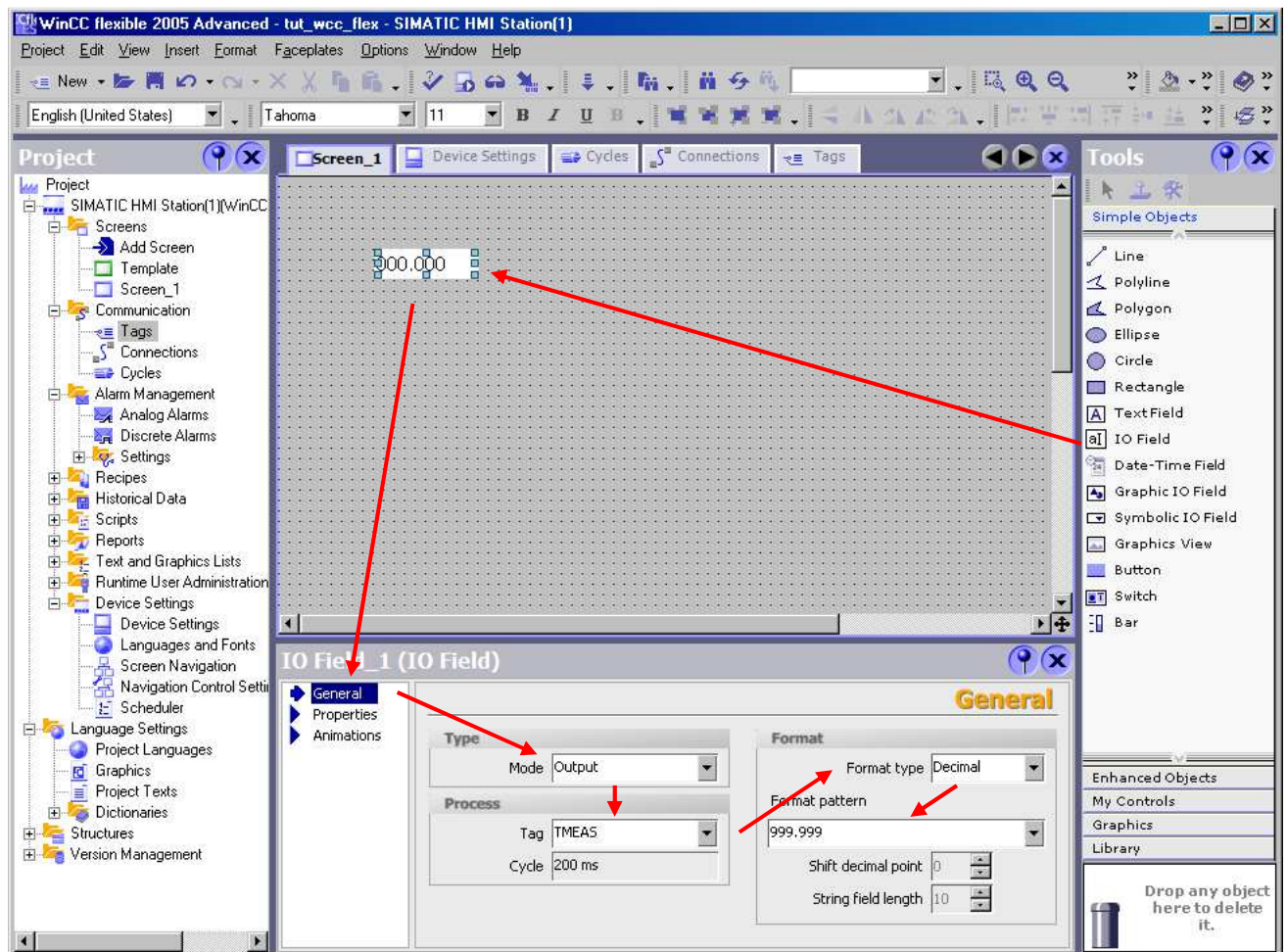
Op het scherm worden drie I/O fields geplaatst. Daarop worden de temperatuur (TMEAS), de spanning over de NTC (VMEAS) en de weerstandswaarde van de NTC (RNTC) afgebeeld.

Daarna worden er die cirkels geplaatst die gebruikt gaan worden als signaleringslampen (één van de leuke trucs die je met objecten kan doen).

Vervolgens wordt er een knop geplaatst waarmee het alarmsignaal kan worden gehonoreerd (*acknowledge*, in jargon ook kwiteren genoemd).

Als laatste wordt er een “ouderwede” meter geplaatst waarop de temperatuur te zien is.

Plaats nu als eerste een I/O Field op het scherm. Selecteer in het eigenschappenscherf onderaan General. Selecteer als Mode nu Output, en bij Tag de tag TMEAS. Het formaat is Decimal en het patroon is 999.999. Zie de rode pijlen in figuur 7-31.



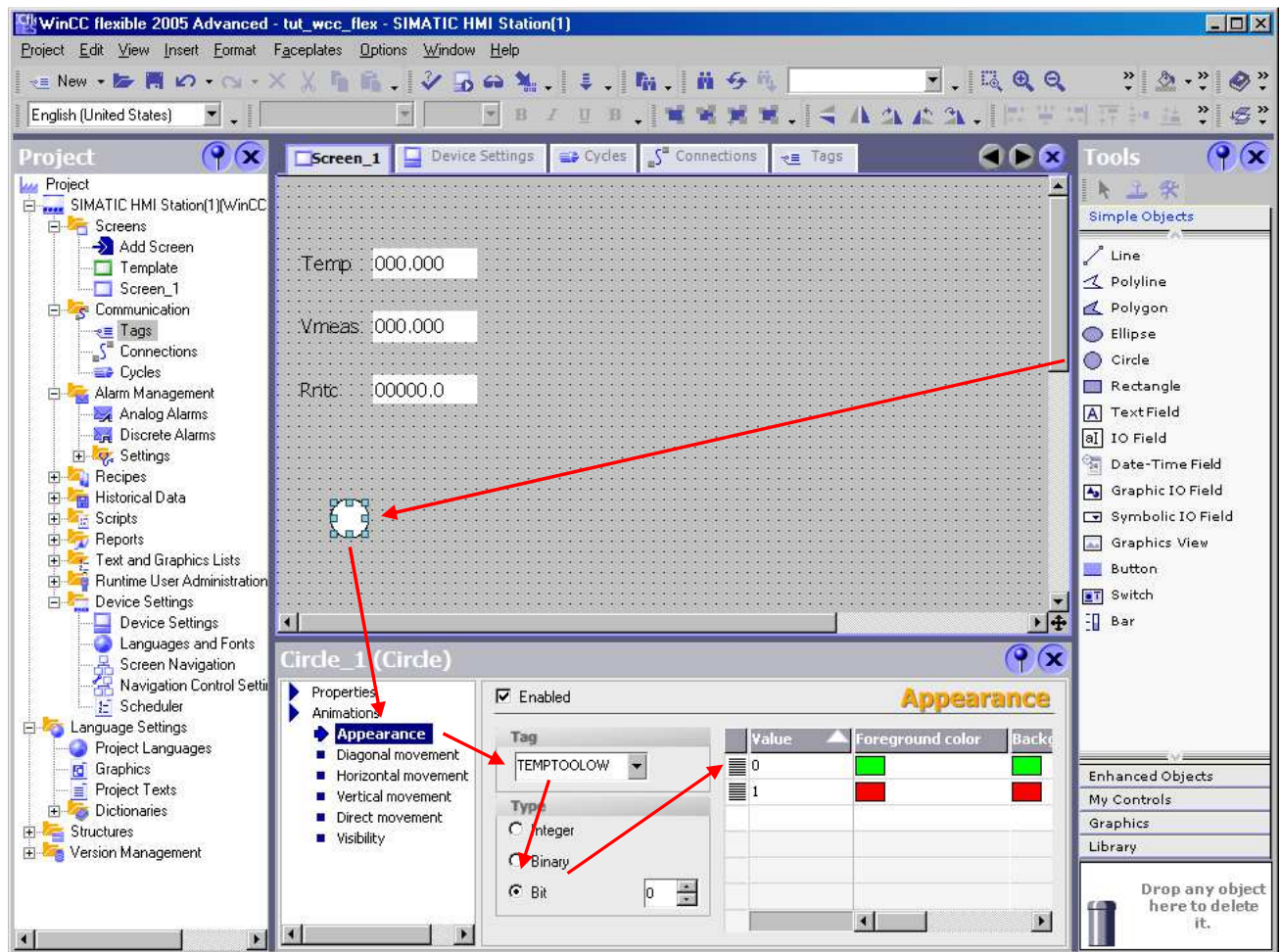
Figuur 7-31

Dit wordt ook gedaan voor de tags VMEAS en RNTC. Bij RNTC moet het patroon echter op 99999.9 staan, anders kan straks de waarde niet worden gerepresenteerd. Plaats ook tekst (Text field) bij de I/O Fields. Zie alvast figuur 7-32.

Om de alarmen weer te geven worden cirkels geplaatst die dienst doet als signaleringlampen. De kleur van de cirkel is aan te passen via de waarde van een tag. Zo kan de cirkel groen worden als een tag de (binaire) waarde 0 heeft en rood voor de waarde 1. Dit wordt gedaan via een *animation*.

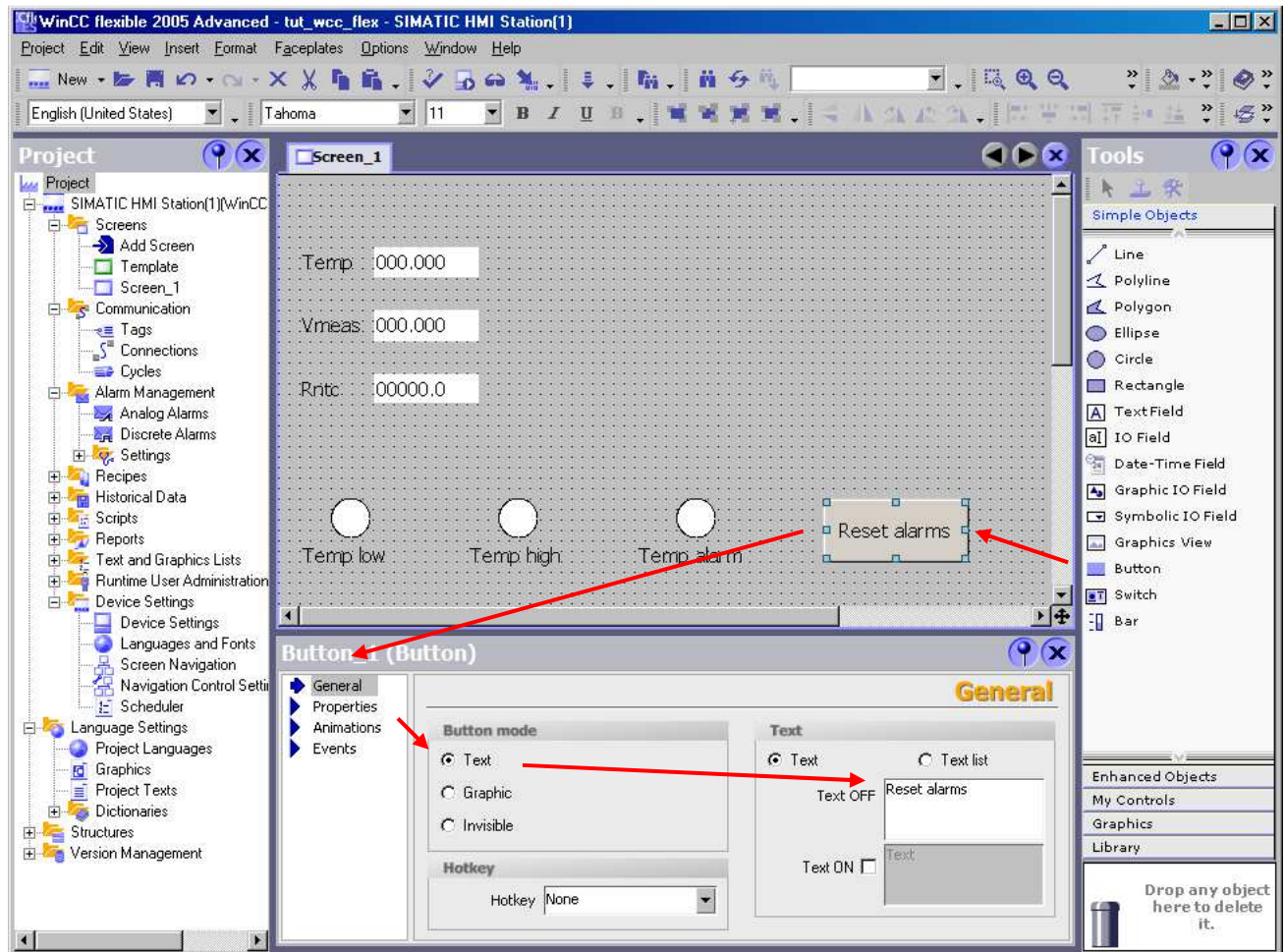
Sleep vanuit het rechterscherf een cirkel naar het grijze vlak. Onderaan zie je alle eigenschappen van de cirkel. Klik op Animations en daarna op Appearance. Vul onder Tag de naam van TEMPTOOLOW in, hier wordt weer een scherm geopend zoals in figuur 7-28. Selecteer daarna het type Bit (STEP7 noemt het een Bool).

Onder het kopje Value moeten nu twee regels worden aangemaakt. Dit kan het beste worden gedaan door per regel even te dubbelklikken. Er verschijnen nu twee regels met 0 en 1 onder Value, maar nog niet de goede kleuren. Door te dubbelklikken op een kleurvlakje verschijnt een selectieschermpje. Kies dan voor rood en groen. Zie figuur 7-32.



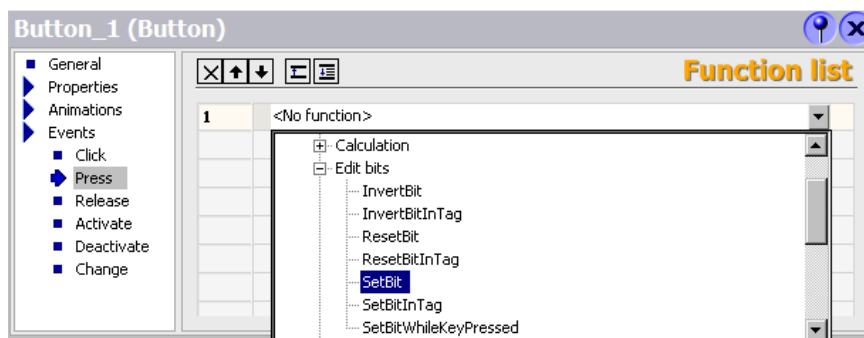
Figuur 7-32

Plaats nu een knop (Button). Onderaan worden de eigenschappen van de knop weergegeven. Vul deze in zoals is weergegeven in figuur 7-33.



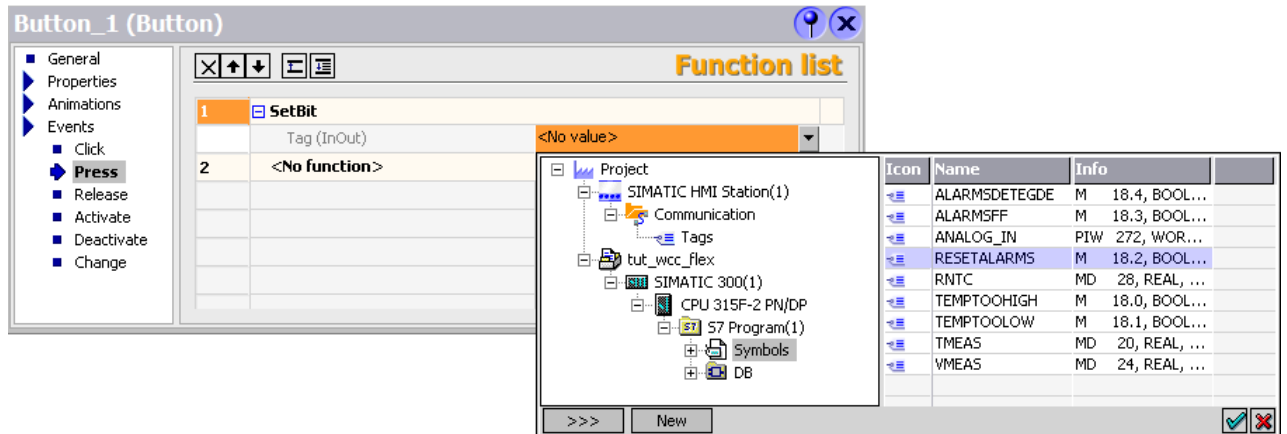
Figuur 7-33

Klik nu op Events. Kies nu onderdeel Press. Onder de lijst Function List kan nu voor een voorgeprogrammeerde functie worden gekozen. Kies uit de lijst nu de functie SetBit. Zie figuur 7-34.



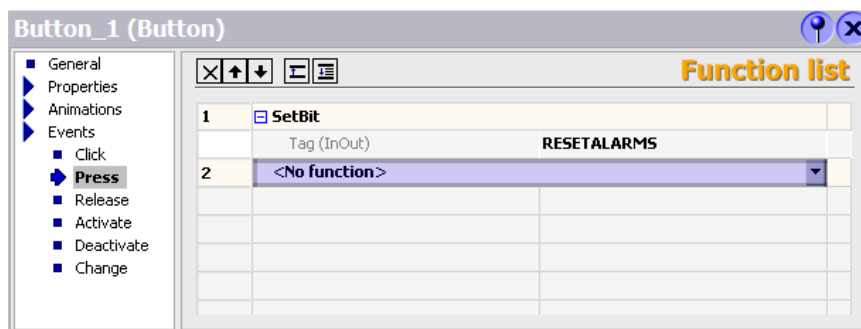
Figuur 7-34

Er wordt nu een regel aangemaakt met de functie SetBit. Deze moet op een tag werken, maar die is nog niet ingevuld. Op de tweede regel staat achter Tag (InOut) nu <No value>. Klik hier nu op en vul de tagnaam RESETALARMS in. Er komt een selectiescherm zoals in figuur 7-28. Zie figuur 7-35.



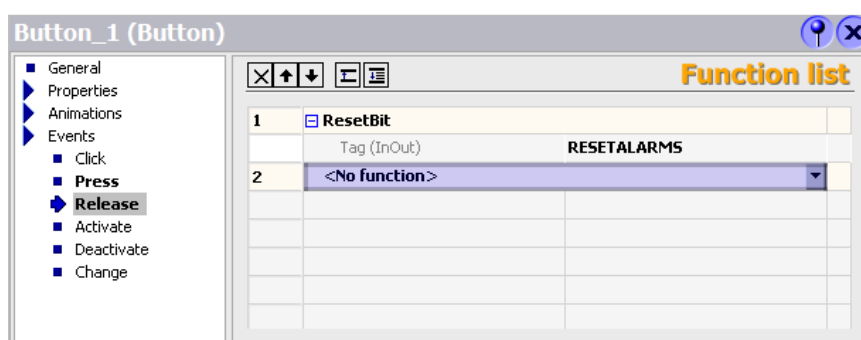
Figuur 7-35

Figuur 7-36 toont het resultaat.



Figuur 7-36

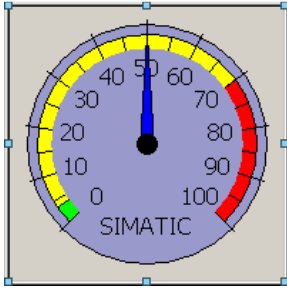
Hetzelfde moet gedaan worden maar nu voor Events→Release en de functie ResetBit. Zie figuur 7-37.



Figuur 7-37

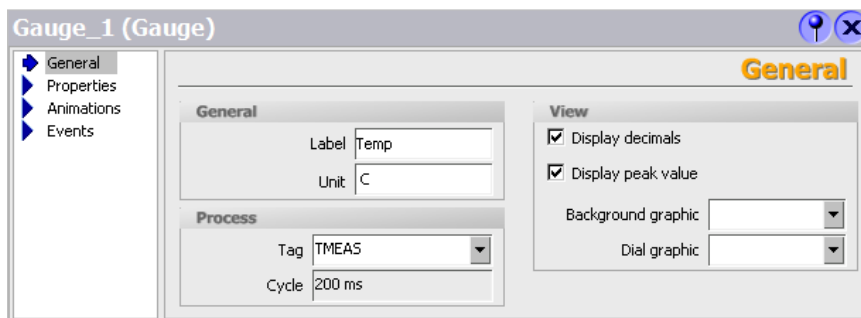
Als laatste wordt een *gauge* geplaatst, die er uit ziet als een ouderwetse manometer. De meter heeft een bereik van 10 °C tot 40 °C, en drie kleurenbanden voor het aangeven van de status.

De gauge is in het linkerscherm te vinden onder Enhanced Objects. Plaats de gauge boven de knop. Na plaatsing ziet de gauge er uit als in figuur 7-38.

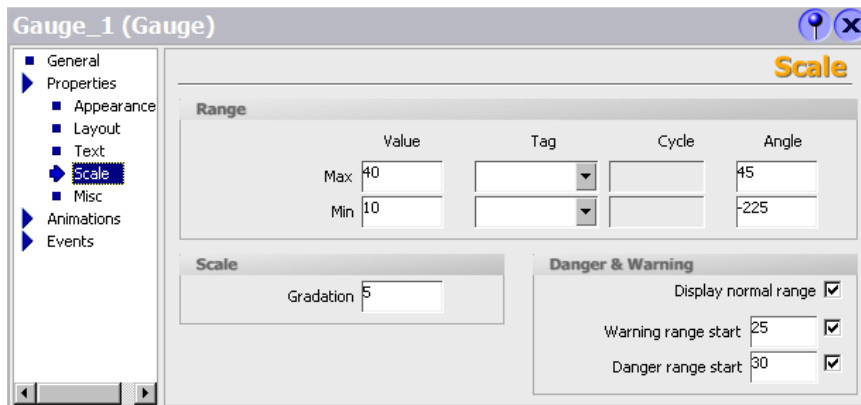


Figuur 7-38

Vul onder General van het eigenschappenscherf van de gauge in zoals is opgegeven in figuur 7-39. Selecteer daarna Properties en vul deze in zoals is aangegeven in figuur 7-40. Tevens is te zien hoe de gauge er uitziet.



Figuur 7-39



Figuur 7-40

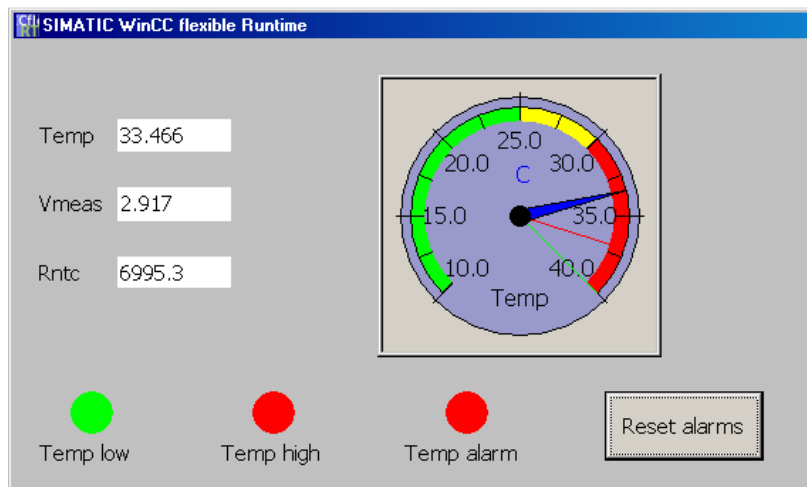
7.13 Starten Runtime

Nu alles ingevuld is, kan een *consistency check* gedaan worden. Hiermee wordt gekeken of het project consistent is. Start de check via Project→Compiler→Check Consistency. Indien het project in orde is, wordt niets gemeld. Zie figuur 7-41.

Time	Category	Description
23:34:31.42	Compiler	Compiling started ...
23:34:31.42	Compiler	Linking target 'SIMATIC HMI Station(1)' ...
23:34:31.42	Compiler	Number of PowerTags used: 7
23:34:31.42	Compiler	Succeeded with 0 error(s), 0 warning(s).
23:34:31.42	Compiler	Time stamp: 1-11-2008 23:34 - used 66112 bytes
23:34:31.85	Compiler	Compiling finished!

Figuur 7-41

Start nu de runtime via **Project**→**Compiler**→**Start Runtime**. Het “programma” wordt nu gecompileerd en de runtime wordt gestart. Dit kan lang duren. In figuur 7-42 is de Runtime te zien. Probeer nu de temperatuur boven de 30 °C te krijgen. Probeer daarna het alarm te kwiteren.



Figuur 7-42

7.14 Afsluiten Runtime

De Runtime kan worden afgesloten door de toetscombinatie **Alt+F4** in te drukken.

8. Tutorial Data Block

In deze tutorial wordt even kort gedemonstreerd hoe een data block wordt opgezet en gebruikt. Als eerste wordt een data block aangemaakt binnen de Simatic Manager, daarna wordt het ingevuld met een array. Er wordt een kort stukje SCL-code gegeven waarin het data block wordt gebruikt.

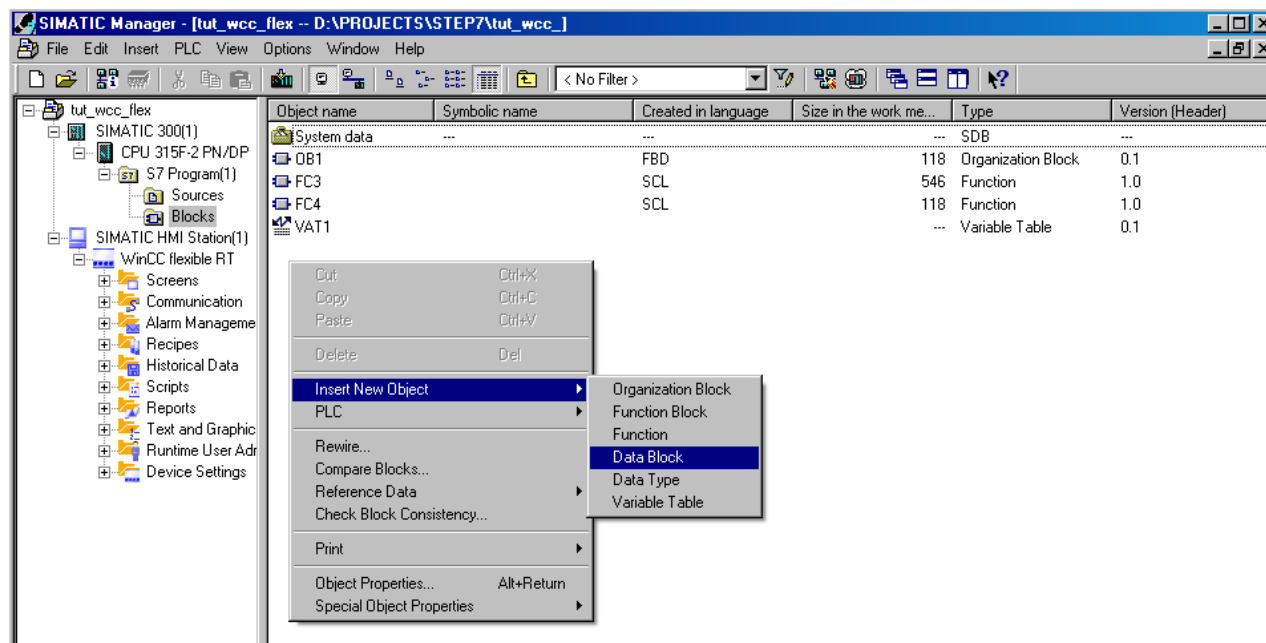
Als project wordt het SCL-project of het WinCC Flexible project gekozen.

Noot: bij deze tutorial wordt verwacht dat de cursist al enige kennis heeft van STEP7 en de vorige tutorials. Een aantal handelingen wordt niet meer expliciet uitgelegd. Houdt daar rekening mee.

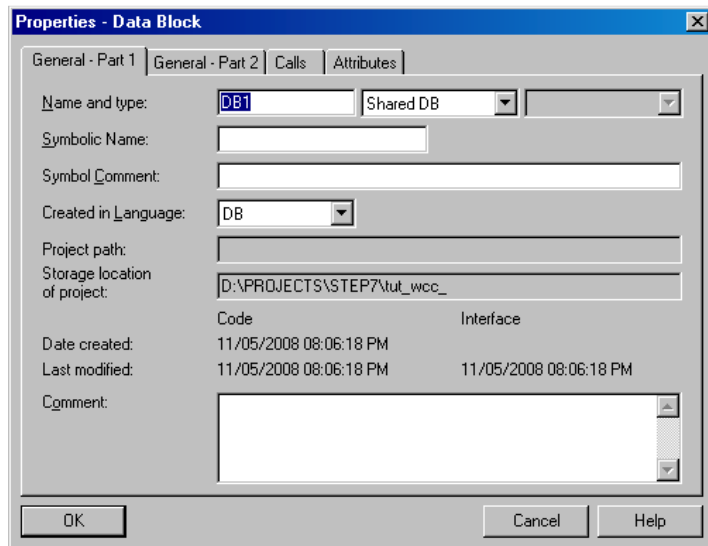
8.1 Aanmaken Data Block

Open één van de genoemde projecten en maak een nieuw data block aan in de Simatic Manager. Zie figuur 8-1.

Kies als naam DB1. Eventueel kan een symbolische naam gekozen worden. Zie figuur 8-2.

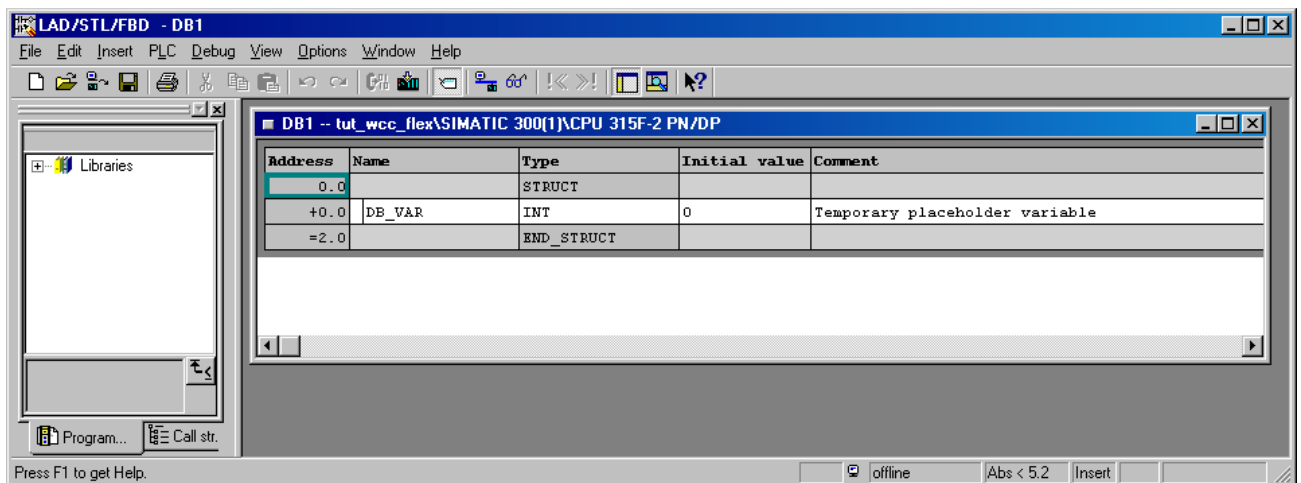


Figuur 8-1



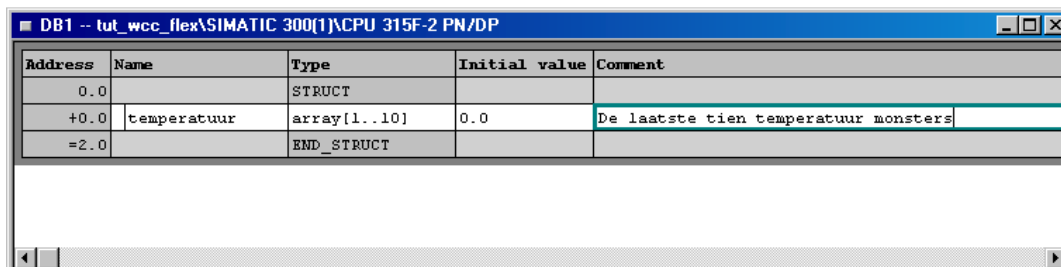
Figuur 8-2

Bevestig met **OK** het aanmaken van het data block. Er wordt nu een venster geopend met daarin één declaratie (dit is de *initial view* van het data block). Zie figuur 8-3.



Figuur 8-3

Vul de eerste regel over de initial view heen. Zie figuur 8-4.



Figuur 8-4

Na een druk op de return-toets verschijnt onder `array[1..10]` een leeg wit vakje. Vul daar het woord `REAL` in en sluit af met een druk op de return-toets. Het geheel wordt nu door de editor aangepast in de juiste vorm: de keywords worden met hoofdletters geschreven en de initiële waarde van reals wordt in wetenschappelijke notatie omgezet. Zie figuur 8-5.

Address	Name	Type	Initial value	Comment
*0.0		STRUCT		
+0.0	temperatuur	ARRAY[1..10]	0.000000e+000	De laatste tien temperatuur monsters
*4.0		REAL		
=40.0		END_STRUCT		

Figuur 8-5

Voeg er nu nog een integer bij met de naam `index`. Let op dat deze als initiële waarde 1 moet hebben. Zie figuur 8-6.

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	temperatuur	ARRAY[1..10]	0.000000e+000	De laatste tien temperatuur monsters
*4.0		REAL		
+40.0	index	INT	1	Waar komt een nieuwe meetwaarde?
=42.0		END_STRUCT		

Figuur 8-6

Sluit de editor af door middel van **File**→**Save**.

8.2 Een stuk je SCL-code

Nu moet er natuurlijk iets met dat data block gedaan worden. Er volgt nu een stuk je SCL-code dat twee onderdelen bevat. Het eerste deel zorgt ervoor dat een gemeten temperatuurwaarde wordt opgeslagen in de array, het tweede onderdeel sommeert de tien opgeslagen waarden uit de array en deelt het door tien. Zo wordt het rekenkundig gemiddelde berekend.

Maak een SCL-broncodebestand aan met de naam `RecordAndAvgTemp`. Vul de SCL-code in zoals in opgegeven in figuur 8-7. Compileer de code en kijk onder `Blocks` of `FC8` is aangemaakt. Zie bijlage 5 voor de volledige code.

```

FUNCTION FC8 : REAL

VAR_INPUT
    record_temp : BOOL;
    current_temp : REAL;
END_VAR

VAR_TEMP
    loop_index : INT;
    cumu_temp : REAL;
END_VAR

BEGIN
    IF record_temp THEN
        DB1.temperatuur[DB1.index] := current_temp;
        DB1.index := DB1.index + 1;
        IF DB1.index > 10 THEN
            DB1.index := 1;
        END_IF;
    END_IF;

    cumu_temp := 0.0;

    FOR loop_index:= 1 TO 10 BY 1 DO
        cumu_temp := cumu_temp + DB1.temperatuur[loop_index];
    END_FOR;

    FC8 := cumu_temp / 10.0;
END_FUNCTION

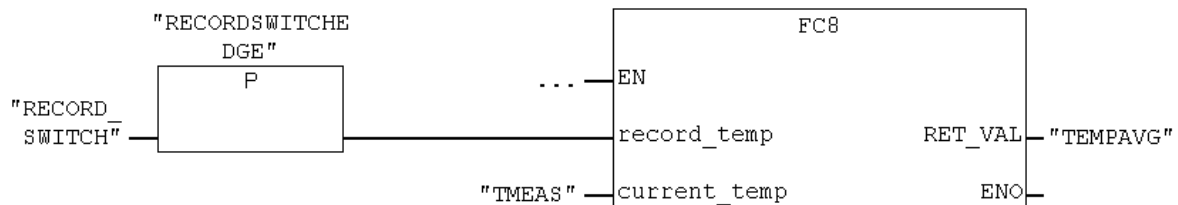
```

Figuur 8-7

8.3 Symbolen, OB1 en downloaden

Het functieblok moet gestart worden vanuit OB1. Dit wordt gedaan door middel van een schakelaar op de simulatiemodule. De functie geeft een real-waarde terug. Die moet ergens worden opgeslagen. Vul de symbolentabel aan met nieuwe variabelen: twee booleans genaamd RECORDSWITCH en RECORDSWITCHEDEGE en real genaamd TEMPAVG. Let er op dat de geheugenplaatsen niet overlappen met andere variabelen!

Vervolgens moet in OB1 nog de functie-aanroep geprogrammeerd worden. Pas OB1 aan zodat de functie wordt aangeroepen (waarom is de flankdetector nodig?). Zie figuur 8-8.

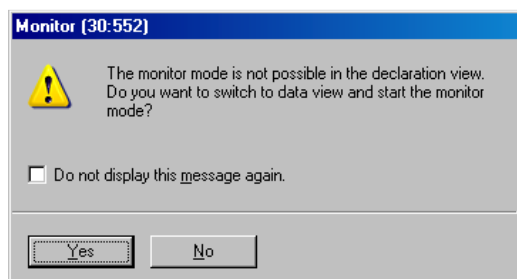


Figuur 8-8

Download nu de nieuwe en aangepaste blokken OB1, DB1 en FC8. Omdat de volgorde van downloaden niet vast staat, kan het zijn dat FC8 vóór DB1 geladen worden. Als nu de PLC in RUN staat én FC8 wordt direct na laden uitgevoerd (OB1 is dus net geladen), dan gaat de PLC in STOP omdat DB1 nog niet geladen is en FC8 een aanroep doet naar DB1. Zet daarom de PLC eerst in STOP. Na het downloaden natuurlijk de PLC weer in RUN zetten.

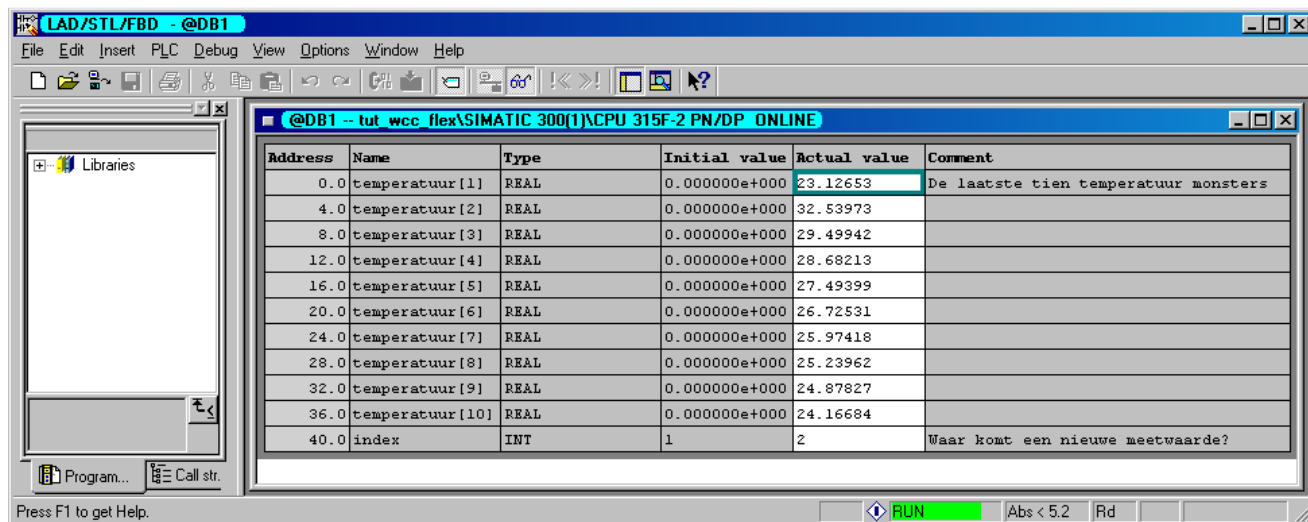
8.4 Monitoring

Nu kan de SCL-code getest worden. Hiervoor moet het data block geopend worden. Klik na openen op het bekende brilletje. Er volgt een waarschuwing. Klik op **Yes**. Zie figuur 8-9.



Figuur 8-9

Nu volgt de data view in *online mode* van DB1. Hierin zijn de actuele waarden van de diverse variabelen te zien. Merk op dat index op 2 staat en alle array-onderdelen een waarde (anders dan de initiële waarde 0.0) hebben. Zo te zien is er al meer dan tien keer een waarde opgeslagen. Hier wordt immers gebruik gemaakt van een circulaire buffer¹⁷. Zie figuur 8-10.



Figuur 8-10

Natuurlijk moet ook nog de gemiddelde temperatuur gemonitord worden. Dit kan met via een VAT. Zie figuur 8-11.

¹⁷ http://en.wikipedia.org/wiki/Circular_buffer

Address	Symbol	Display format	Status value	Modify value
I 8.0	"RECORD_SWITCH"	BOOL	true	
M 18.0	"TEMPTOOHIGH"	BOOL	false	
M 18.1	"TEMPTOOLOW"	BOOL	false	
M 18.2	"RESETALARMS"	BOOL	false	
M 18.3	"ALARMSFF"	BOOL	true	
M 18.4	"ALARMSDETEGDE"	BOOL	false	
MD 20	"TMEAS"	FLOATING_POINT	21.45813	
MD 24	"VMEAS"	FLOATING_POINT	4.351852	
MD 28	"RNTC"	FLOATING_POINT	11691.38	
MD 32	"TEMPAVG"	FLOATING_POINT	26.8326	
PMW 272	"ANALOG_IN"	HEX	VW#16#2F00	
12				

Figuur 8-11

8.5 Instance Data Block

In deze tutorial is gebruikt gemaakt van een *shared data block*. Het data block wordt echter alleen door FC8 gebruikt; geen ander programmablok werkt met de data uit DB1. Dan is het makkelijker om een Function Block (FB) te gebruiken met daaraan gekoppeld een *instance data block*. Zo kan de declaratie van de array en index gewoon binnen de FB gedaan worden; hiervoor moeten STAT-variabelen worden aangemaakt. Deze STAT-variabelen komen dan in de Instance DB terecht. Bij referentie naar de STAT-variabelen hoeft de DB-qualifier niet meer te worden opgegeven.

In bijlage 5 is de code van FB8 en DB8 te vinden. FB8 lijkt erg veel op FC8 uit figuur 8-7. FB's hebben echter geen teruggaveparameter maar moeten dat met een OUTPUT-parameter doen. De declaratie van de array en index wordt nu gedaan binnen de beschrijving van de FB. Het data block DB8 is leeg. Dat is ook de bedoeling, want de parameters van FB8 worden automatisch in DB8 geplaatst. Wel moet worden opgegeven dan DB8 bij FB8 hoort. Merk op dat de in- en out-parameters ook in het data block aanwezig zijn. Zie figuur 8-12.

Address	Declaration	Name	Type	Initial value	Actual value	Comment
0.0	in	record_temp	BOOL	FALSE	FALSE	Should we record?
2.0	in	current_temp	REAL	0.000000e+000	0.000000e+000	The current temperature
6.0	out	avg_temp	REAL	0.000000e+000	0.000000e+000	The average temp returns
10.0	stat	temperature[1]	REAL	0.000000e+000	0.000000e+000	Special treat: 10 items of temperature initialized to 0.0
14.0	stat	temperature[2]	REAL	0.000000e+000	0.000000e+000	
18.0	stat	temperature[3]	REAL	0.000000e+000	0.000000e+000	
22.0	stat	temperature[4]	REAL	0.000000e+000	0.000000e+000	
26.0	stat	temperature[5]	REAL	0.000000e+000	0.000000e+000	
30.0	stat	temperature[6]	REAL	0.000000e+000	0.000000e+000	
34.0	stat	temperature[7]	REAL	0.000000e+000	0.000000e+000	
38.0	stat	temperature[8]	REAL	0.000000e+000	0.000000e+000	
42.0	stat	temperature[9]	REAL	0.000000e+000	0.000000e+000	
46.0	stat	temperature[10]	REAL	0.000000e+000	0.000000e+000	
50.0	stat	index	INT	1	1	Index into the array, initialized to 1

Figuur 8-12

9. Tips, tricks & troubleshoot

In dit hoofdstuk worden wat zaken behandeld die niet direct in een tutorial thuishoren maar wel handig zijn om te weten. Zo wordt uitgelegd hoe je projecten kan archiveren en gearchiveerde projecten kan inlezen. Het veranderen van PC/PG interface is ook nog wel eens nodig als bijvoorbeeld met de Ethernet-verbinding moet worden gewerkt. De PLC kan in STOP gaan na een fout, maar wat is er nu gebeurd? Dat is te zien via diagnostiek.

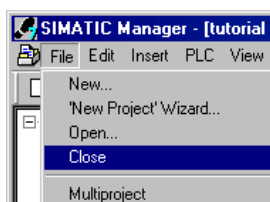
9.1 Projecten archiveren

Via de Simatic Manager is het mogelijk om projecten te archiveren. Alle bestanden van het project worden dan door middel van compressie in één bestand geplaatst. Dit archief, wat in feite een gewoon ZIP-bestand is, kan dan worden opgeslagen op bijvoorbeeld een USB-stick.

Noot: maak nooit zelf een archief aan, altijd via de Simatic Manager.

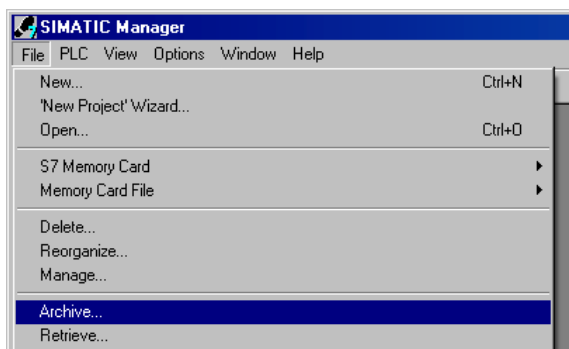
Noot: er mogen géén spaties in map- en bestandsnaam voorkomen.

Sluit eerst het project af door middel van **File**→**Close**. Zie figuur 8-1.



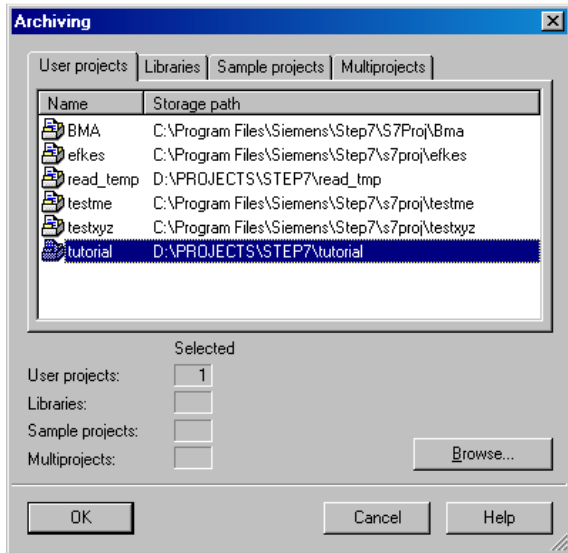
Figuur 9-1

Kies daarna **File**→**Archive**. Zie figuur 9-2.



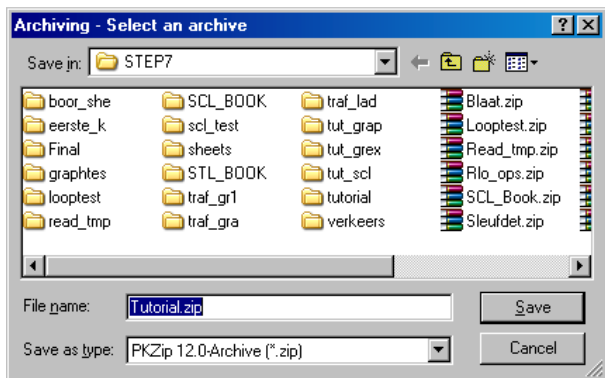
Figuur 9-2

Vervolgens wordt gevraagd om een project te selecteren. Selecteer het juiste project en klik op **OK**. Noot: het project moet afgesloten zijn! Zie figuur 9-3.



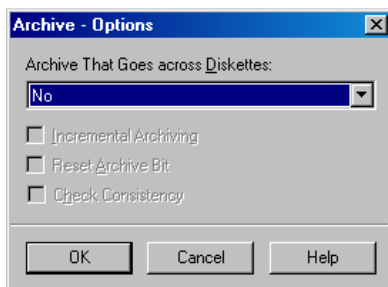
Figuur 9-3

Nu wordt er gevraagd wat de naam is van het nieuw te maken archief. Kies een naam die nog niet bestaat, anders wordt het oude archief zonder pardon overschreven. Zie figuur 9-4.



Figuur 9-4

Dan wordt er gevraagd of er gearchiveerd wordt op diskettes (jawel!). Kies **No**. Zie figuur 9-4.



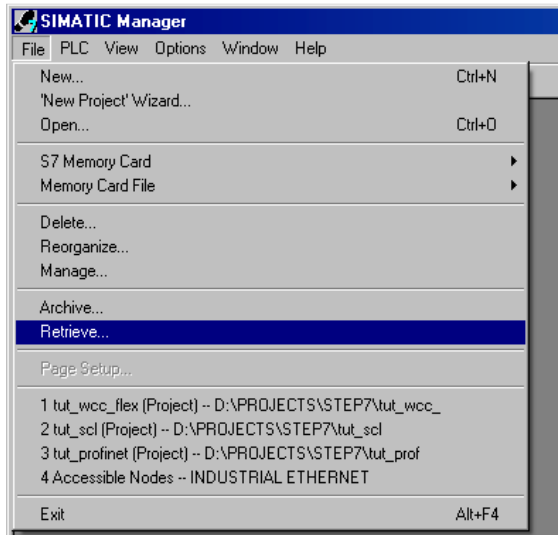
Figuur 9-5

Daarna wordt het project gearchiveerd in een nieuw geopende DOS-box. Hier is geen figuur van.

9.2 Projecten inlezen

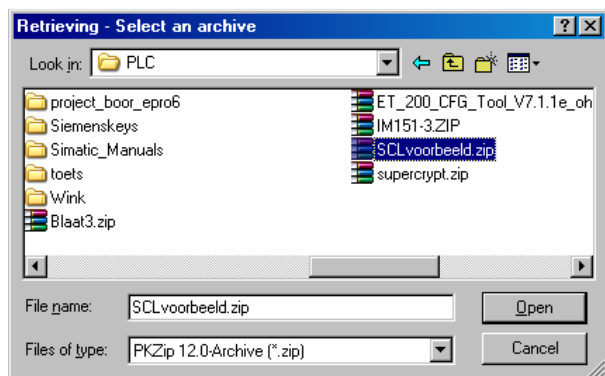
Gearchiveerde projecten kunnen weer worden ingelezen. Daarna zijn ze te openen via de Simatic Manager.

Klik in de Simatic Manager op **File**→**Retrieve**. Zie figuur 9-6.



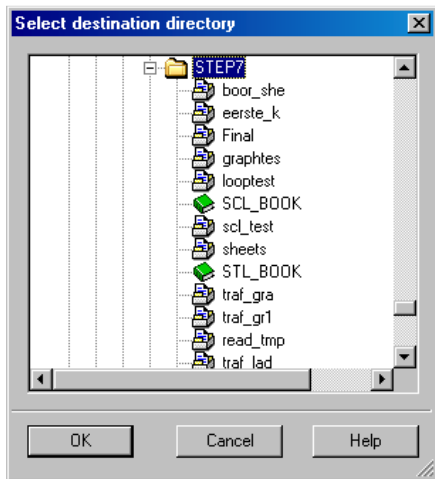
Figuur 9-6

Selecteer nu het juiste archief en klik op **Open**. Zie figuur 9-7.



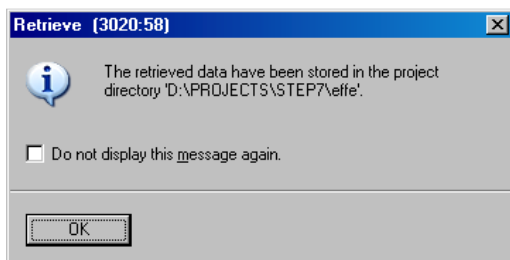
Figuur 9-7

Nu moet worden opgegeven waar de bestanden van het archief worden geplaatst. Zoek een geschikte directory en klik op **OK**. Zie figuur 9-8.



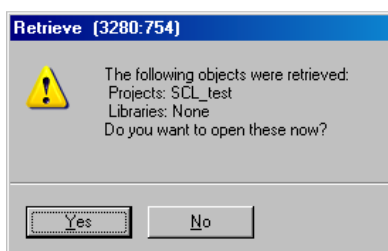
Figuur 9-8

Het archief wordt geopend en de bestanden worden ingelezen. Dit is te zien in een aparte DOS-box die gestart wordt (geen figuur). Nadat het archief helemaal geplaatst is volgt een melding. Klik op **OK**. Zie figuur 9-9.



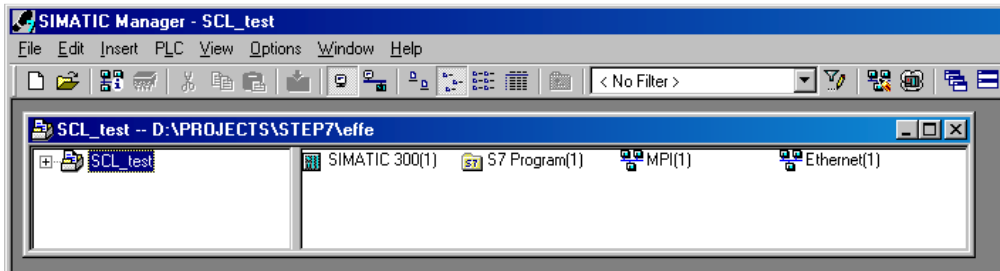
figuur 9-9

Dan wordt gevraagd of het project gelijk geopend moet worden. Klik op **Yes** om te openen. Zie figuur 9-10.



Figuur 9-10

Het project wordt geopend. Zie als voorbeeld figuur 9-11.



Figuur 9-11

9.3 PG/PC interface

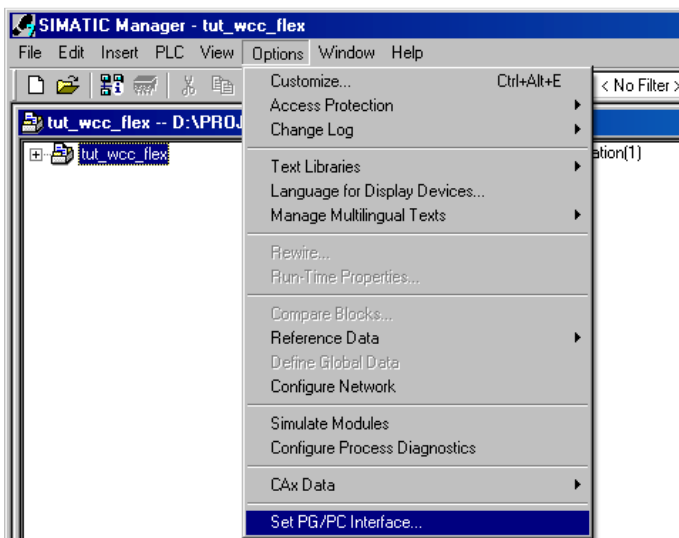
De PG/PC¹⁸ interface wordt voor diverse doeleinden gebruikt. Het is het logische koppelstuk tussen de PG/PC en de PLC- en HMI-hardware. De PLC kan er mee worden geconfigureerd en geprogrammeerd, en kan er mee worden gevolgd door middel van een *online* verbinding. HMI-stations kunnen er mee worden geprogrammeerd.

Hardwarematig zijn er aan de PG/PC-kant verschillende mogelijkheden. Er kan gekozen worden voor zogenaamde CP-kaarten, communicatiekaarten van Siemens. Denk hierbij aan de CP511 en CP5611. Het kan een seriële interface zijn, of een Ethernet-aansluiting. In het lab zijn ook USB-adapters beschikbaar.

Afhankelijk van de PLC en communicatiemogelijkheden moet de juiste interface worden ingesteld. Hieronder volgt een korte tutorial over het selecteren van een nieuwe interface.

9.3.1 PC/PG interface via de USB adapter

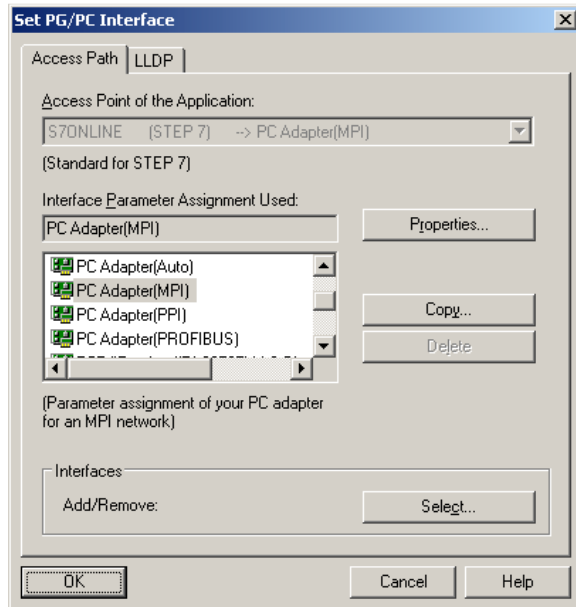
Kies in de Simatic Manager **Options**→**Set PG/PC Interface**. Zie figuur 9-12.



Figuur 9-12

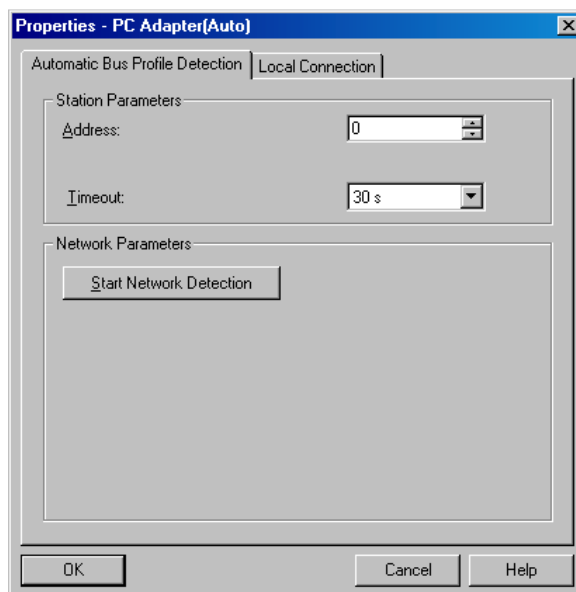
¹⁸ De term PG staat voor Programmier Gerät. De term PG wordt eigenlijk gebruikt om de PC/laptop/computer aan te geven waarmee de PLC geconfigureerd en geprogrammeerd wordt.

Er wordt een venster geopend waarin de huidige geselecteerde interface te zien is. In dit geval is het de PC Adapter (USB) in Auto modus (figuur 9-13). De adapter zoekt dan zelf uit of er een verbinding kan worden met de PLC kan worden gelegd via MPI, Profibus of PtP. Gebruik deze modus liever niet. Het kan problemen opleveren met het online volgen van variabelen en programmacode.



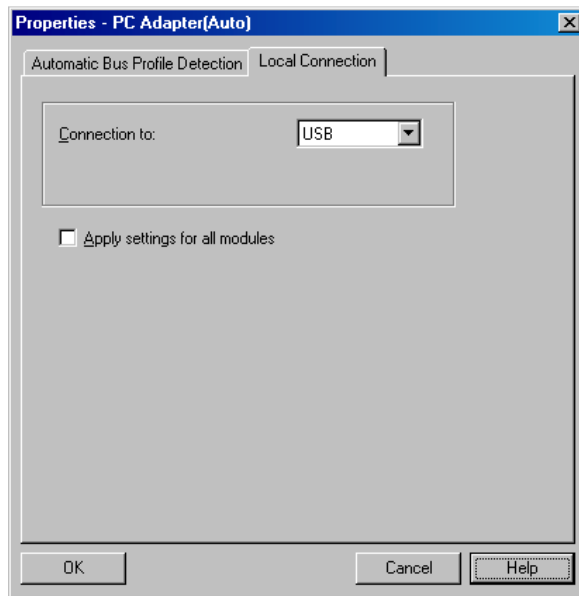
Figuur 9-13

Als de adapter niet werkt, kan het zijn dat een verkeerde poort is ingesteld. Klik dan op **Properties**. Er opent nu een nieuw venster (figuur 9-14). De parameters moeten precies als in de figuur ingesteld zijn.



Figuur 9-14

Onder het tabblad `Local Connection` kan de huidige poort gewijzigd worden. Deze moet als USB ingesteld staan. Zie figuur 9-15.

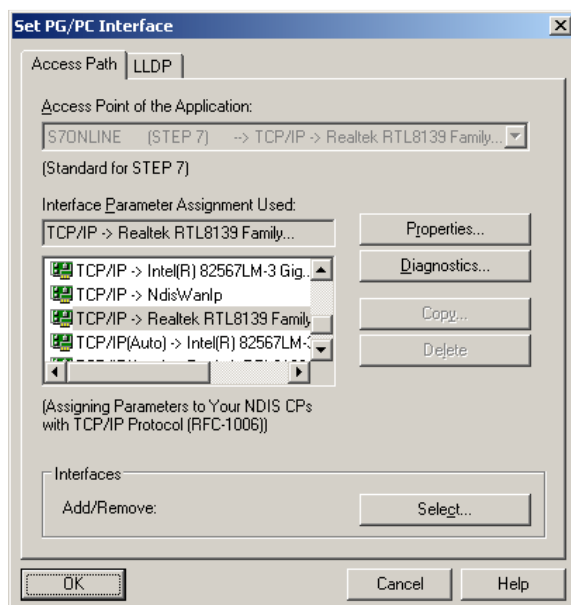


Figuur 9-15

9.3.2 PC/PG interface omzetten

De PLC in het lab zijn geschikt om via Ethernet te communiceren. Dat heeft als voordeel dat de netwerkcomponenten die al beschikbaar zijn zoals routers en switches, gebruikt kunnen worden. De PLC is configureren, programmeren en monitoren via Ethernet.

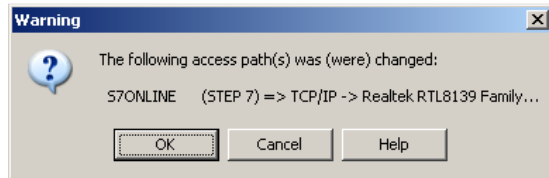
Kies in de Simatic Manager **Options**→**Set PG/PC Interface**. Zie figuur 9-12. Selecteer nu de TCP/IP interface van de Ethernetkaart waarmee de PLC bereikt kan worden (figuur 9-16).



Figuur 9-16

Noot: omdat er veel verschillende Ethernetkaarten bestaan, zal figuur 9-16 enigzins afwijken van de werkelijkheid.

Klik nu op **OK**. Er volgt een dialoog, zie figuur 9-17. Klik op **OK** om de Ethernet-interface te gebruiken.

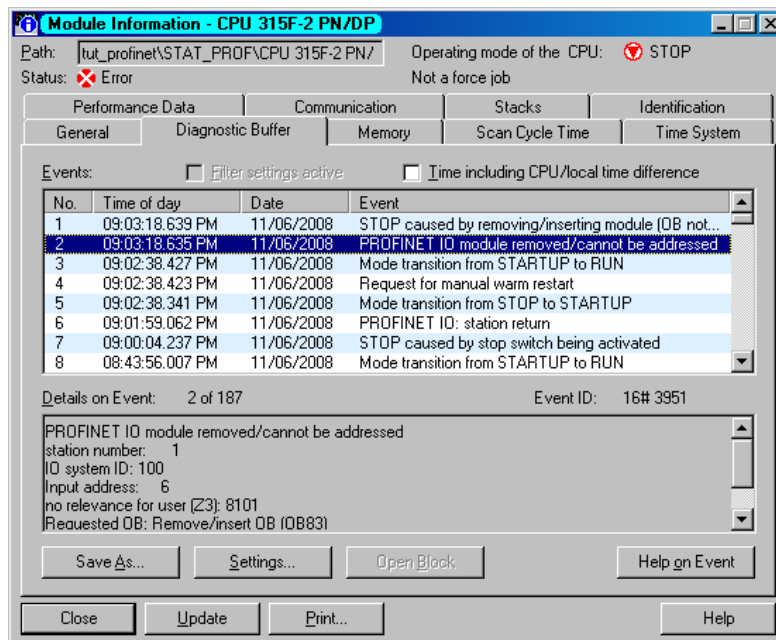


Figuur 9-17

9.4 PLC diagnostiek

Als de PLC onverwacht in STOP gaat, is het meestal goed fout. Maar waarom is de PLC in STOP gegaan? Er zijn veel verschillende problemen mogelijk. Een BCD-conversie kan zijn mislukt, een *I/O-slave* kan offline gegaan zijn, er kan een programmeerfout zijn gemaakt.

De PLC registreert al deze gebeurtenissen. Ze kunnen worden opgevraagd via het diagnostische venster. Selecteer in de Simatic Manager eerst de PLC. Klik nu op **PLC→Diagnostic/Setting→Module Information**. De diagnostics venster wordt geopend op tabblad General. Selecteer tabblad Diagnostic Buffer. In figuur 9-18 is te zien dat de PLC in STOP ging omdat een Profinet-module verwijderd werd.

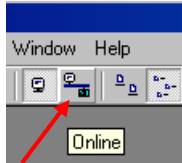


Figuur 9-18

9.5 Online blokken in de PLC verwijderen

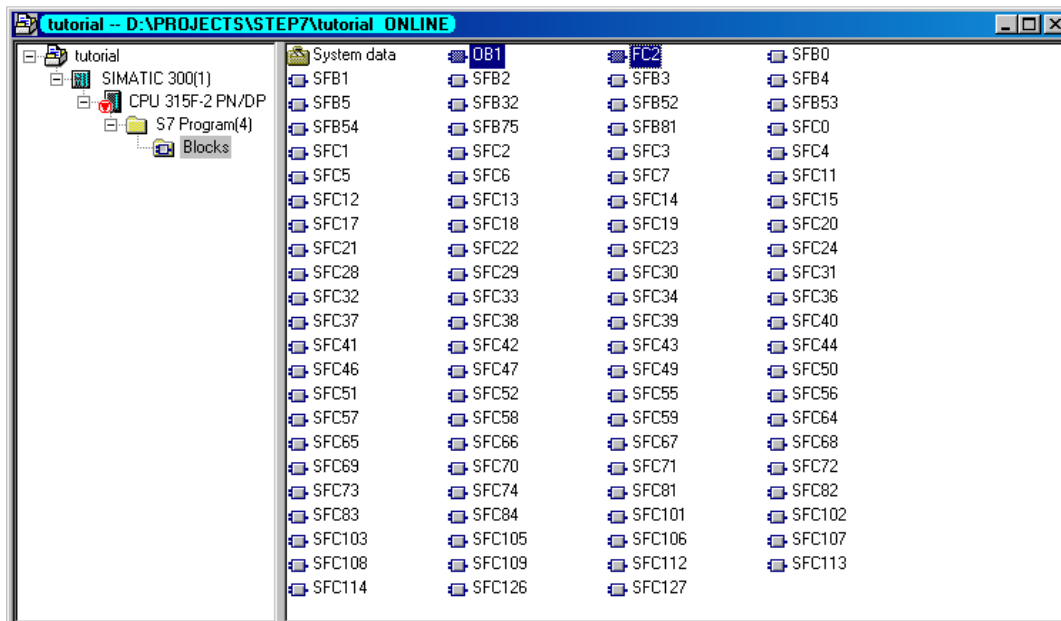
Het is mogelijk om de blokken die ooit naar de PLC zijn verstuurd (download) te verwijderen. Deze blokken staan namelijk niet echt in de PLC maar op de MMC-kaart. Als de PLC in RUN gaat, worden de benodigde blokken in het werkgeheugen van de PLC geladen. Nu kan het zijn dat er zoveel blokken op de MMC-kaart staan, dat nieuwe blokken niet verstuurd kunnen worden. Via de *online block view* kunnen (niet benodigde) blokken verwijderd worden.

Selecteer in de Simatic Manager de map Blocks. Klik dan op **View→Online** of klik op het Online-pictogram (figuur 9-19).

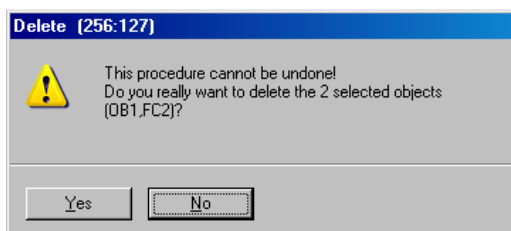


Figuur 9-19

Er opent zich een venster waarin de online blokken te zien zijn. Selecteer alle blokken, behalve de System Data, SFC's en SDB's en verwijder ze. Zie figuur 9-20 en 9-21.



Figuur 9-20

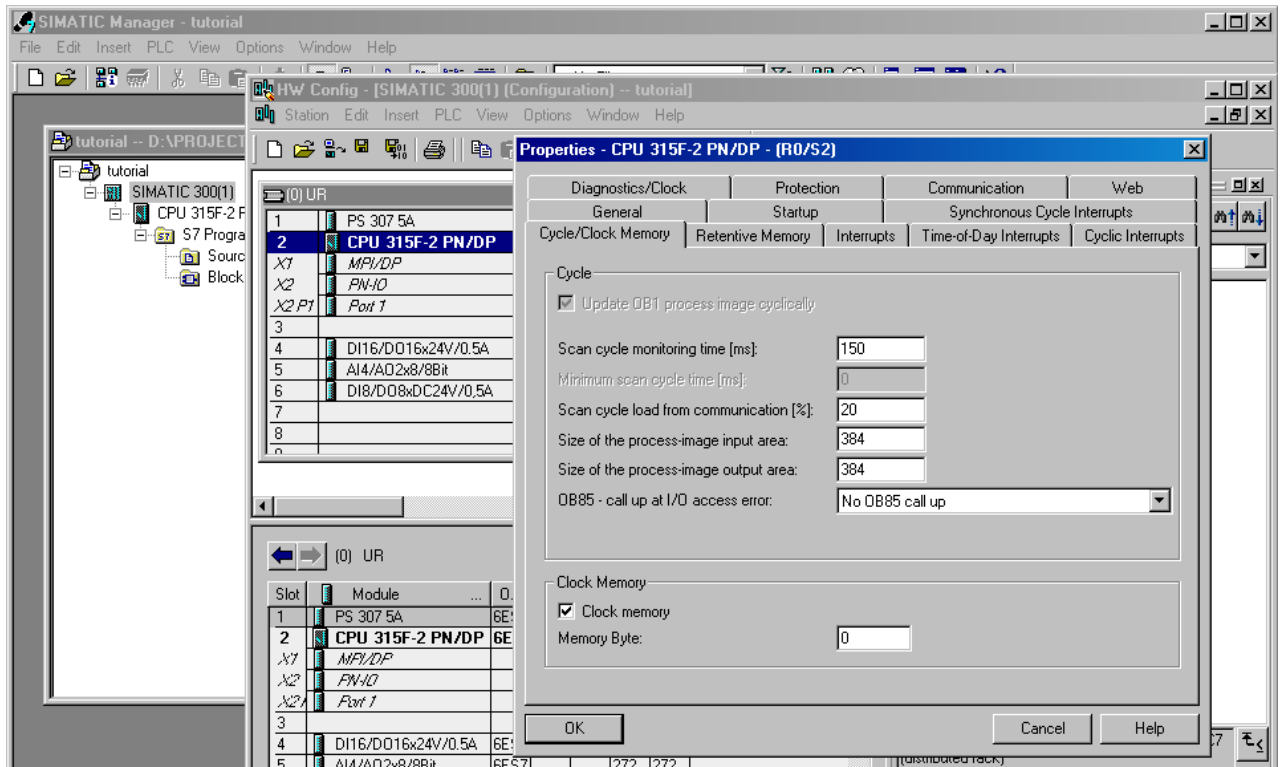


Figuur 9-21

9.6 Clock memory byte

In de PLC is het mogelijk om één byte te reserveren als *clock memory*. De PLC genereert in elke bit een kloksignaal van verschillende frequenties. Alle kloksignalen hebben een duty cycle van 50%. Het gebruik van de clock memory wordt tijdens de configuratie opgegeven.

Start HWconfig en selecteer de CPU-module. Dubbelklik om de eigenschappenvenster te openen. Selecteer daarna het tabblad *Cycle/Clock Memory*. Vink de optie *clock memory* aan en pas eventueel het adres aan. Standaard wordt MB0 gebruikt. Zie figuur 9-22.



Figuur 9-22

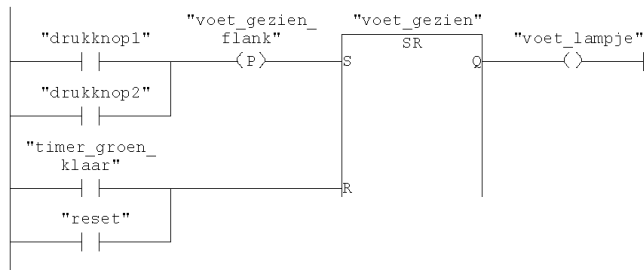
Hieronder staat een tabel met alle frequenties en periodetijden. Om bijvoorbeeld een knipperlamp voor alarmering te maken met een frequentie van 1 Hz moet bit M0.5 gebruikt worden als MB0 het clock memory byte is.

Bit	7	6	5	4	3	2	1	0
Frequentie (Hz)	0,5	0,625	1,0	1,25	2,0	2,5	5,0	10,0
Periode (s)	2,0	1,6	1,0	0,8	0,5	0,4	0,2	0,1

Tabel 9-1

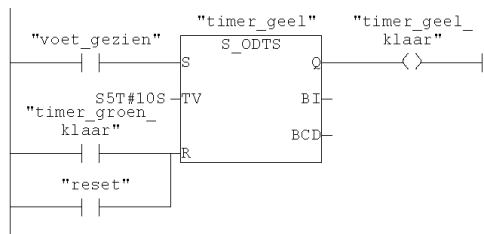
Network: 1 Voetganger gezien

Wil een voetganger oversteken? Vang de actie in. Flankdetectie is nodig omdat voet_gezien naar 1 gaat als RESET van 1 naar 0 gaat en SET-actie nog steeds 1. Nu moet SET-actie eerst 0 worden.



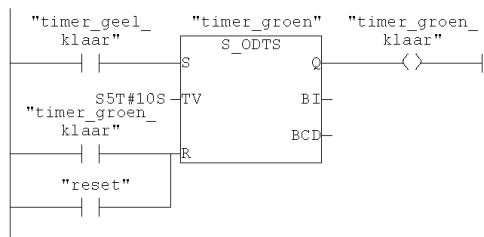
Network: 2 Timer voor auto geel

Eerste tijdmeting. Meet de tijd dat het gele autoverkeerslicht actief is en houdt voetgangerslicht op rood.



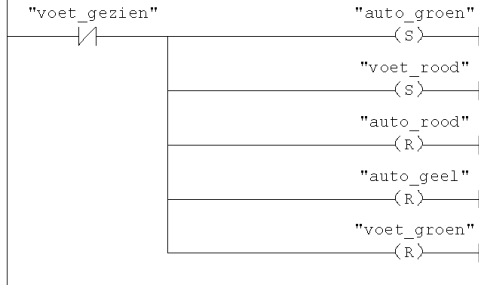
Network: 3 Timer voor auto groen

Tweede tijdmeting. Meet de tijd dat het auto verkeerslicht rood is en het voetgangerslicht op groen. Hier zit een gevaarlijke constructie in. Timer_groen stuurt timer_groen_klaar aan als het klaar is, maar reset zichzelf dan ook na 1 scan cycle. Netter zou zijn om een extra timer te nemen, die bv. 100 ms te laten draaien en die de andere timers te laten resetten.



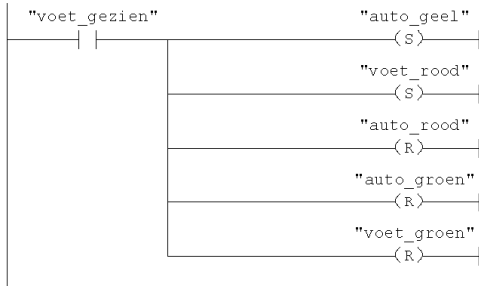
Network: 4 Verkeerslicht auto groen

Is voet gezien gelijk aan 0? Dan hebben we autolicht groen en voetgangerslicht rood. Wederzijds uitgesloten met volgend netwerk.



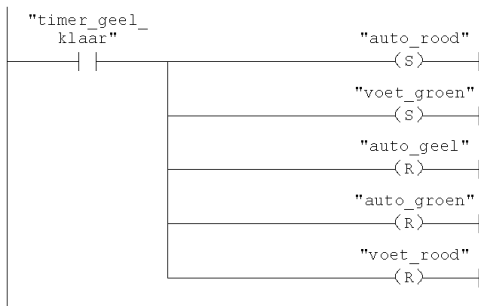
Network: 5 Verkeerslicht auto geel

Is voet gezien gelijk aan 1? Dan hebben we autolicht geel en voetgangerslicht rood. Wederzijds uitgesloten met vorig netwerk.



Network: 6 Verkeerslicht auto rood

Hebben we timer_geel_klaar is 1? Dan hebben we autolicht rood en voetgangerlicht groen. Als timer_geel_klaar is 1, dan worden de acties van de vorige twee netwerken ongedaan gemaakt. Dat komt omdat de PLC z'n outputs pas aan het eind van de huidige scan cycle aanpast (bij Siemens eigenlijk aan het begin, maar goed).



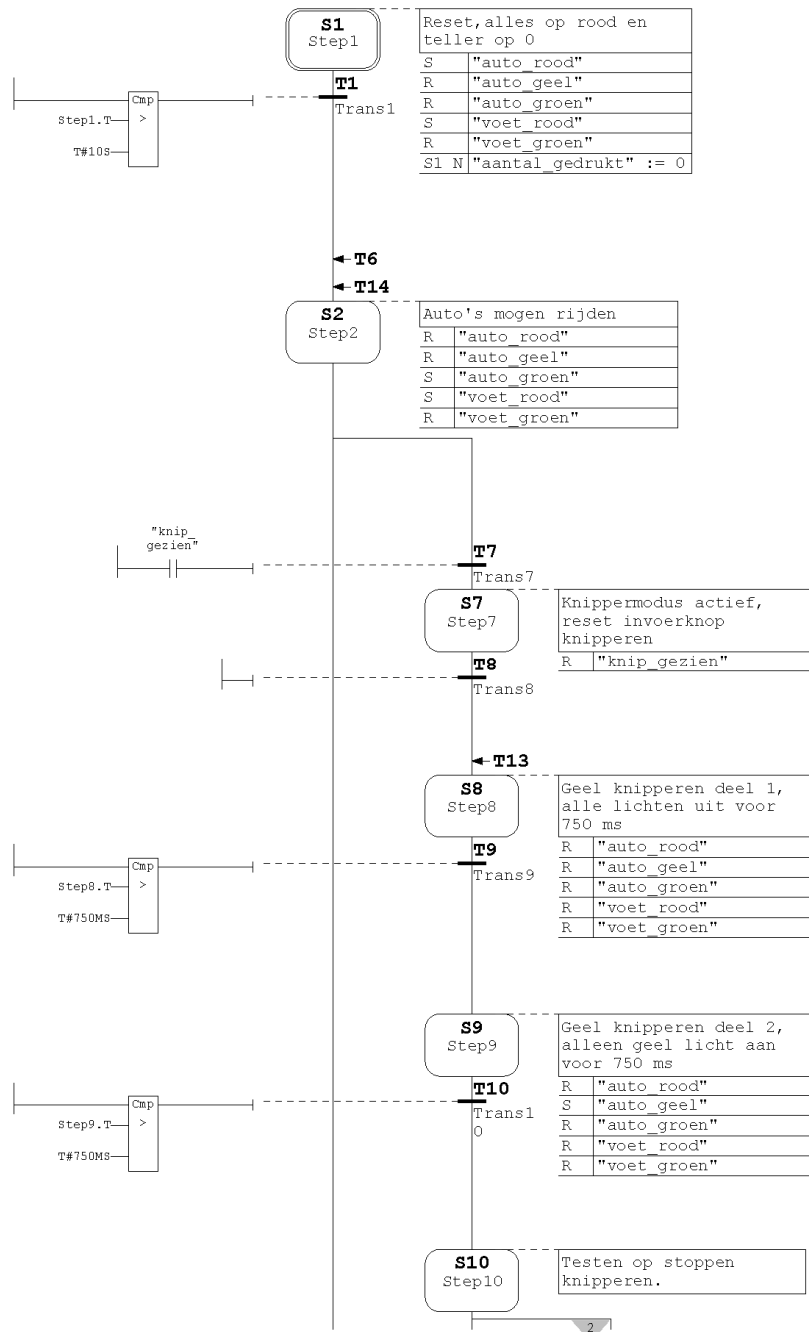
Bijlage 2 – Graph programma

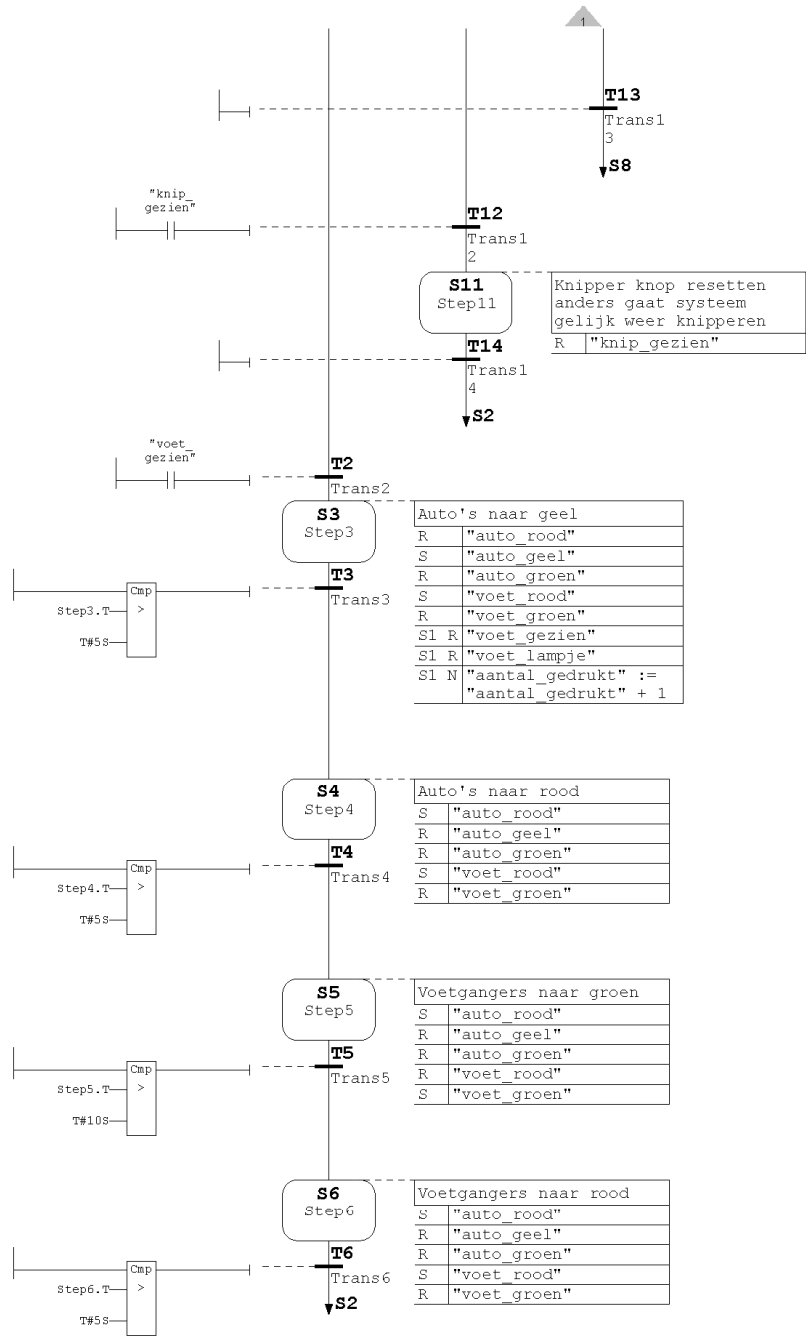
SIMATIC

tut_graph_extended\SIMATIC_300_TRAF_EXT\
CPU 315F-2 PN/DP\...\FB2, DB2 - <Offline>

01/06/2008 12:26:57 PM

Block comment





Bijlage 3 – PLC Blocks, input- en outputadressen, memory

De PLC beschikt over een aantal blocks, waarin programma en data gestructureerd kan worden.

Organization Blocks (OB)

OB's zijn de interfaces tussen het OS en het gebruikersprogramma. De belangrijke OB is OB1: *cyclic operation block*, voer het programma "steeds" uit. Daarnaast zijn er tal van andere OB's.

Dus: om een programma draaiend te krijgen moet ALTIJD OB1 gebruikt worden, ook al doet deze niets anders dan FB's of FC's aanroepen

Function Blocks (FB)

Dit zijn blokken met parameteroverdrachtmogelijkheden en eigen geheugen. Dit geheugen voor dataopslag ligt in een DB en wordt persistent genoemd, dat wil zeggen dat een FB zijn eigen data kan bewaren over aanroepen heen. Voor C-programmeurs is het te vergelijken met static variabelen.

Functions (FC)

Dit zijn blokken die ook parameteroverdrachtmogelijkheden hebben, maar geen eigen geheugen. Nadat een functie klaar is, wordt alle data gewist. Typische voorbeelden zijn het berekenen van de wortel van een getal, grootste gemene deler. Functions kunnen waarden retourneren.

Data Blocks (DB)

Dit zijn de blokken waarin de data wordt opgeslagen. Ze kunnen maximaal 16420 bytes groot zijn.

System Function (SFC), System Function Blocks (SFB)

Deze zijn als FC en FB, maar nu met voorgeprogrammeerde functies, bijvoorbeeld communicatie over de bus.

Input- en outputadressen

De digitale ingangen lopen van I0.0 t/m I1.7. Hierin zijn 0 en 1 het input adresbyte en .0 en .7 het bitnummer binnen een adres. Er zijn in totaal 16 ingangen.

De digitale uitgangen lopen van Q0.0 t/m Q1.7. Hierin zijn 0 en 1 het output adresbyte en .0 en .7 het bitnummer binnen een adres. Er zijn in totaal 16 uitgangen.

De analoge ingangen zijn PIW272 t/m PIW278. De uitgangen zijn te vinden op PQW272 en PQIW274. Merk op dat het om PI en PQ gaat (buiten de process image table om) en dat het words zijn. Bv. PIW278 bestaat uit byte 278 en 279.

De ingangen van de simulatiemodule lopen van I8.0 t/m I8.7, maar dit kan veranderd worden. De uitgangen lopen van Q8.0 t/m Q8.7

Memory

Het geheugen loopt van M0.0 tot M2047.7. Het geheugen is ook byte-, word- en long word-adresseerbaar. Het is een algemeen geheugen dat gedeeld wordt met andere OB's, FB's en FC's.

Bijlage 4 – Enige formules voor temperatuurmeting

Een NTC-weerstand heeft een weerstandswaarde die zeer goed benaderd wordt met de formule:

$$R_{NTC} = A \cdot e^{\frac{B}{T}} \quad (1)$$

A en B zijn hierin parameters van de NTC.

De overdracht van de spanningsdeler uit figuur 6-1 is als volgt te berekenen

$$Z = \frac{V_{NTC}}{V_{POWER}} = \frac{R_{NTC} // R_{IN}}{R_S + R_{NTC} // R_{IN}} \quad (2)$$

Hierin is V_{POWER} de spanning van de voeding, V_{NTC} de spanning over de NTC, R_{NTC} de weerstand van de NTC, R_{IN} de ingangsweerstand van de analoge ingang en R_S de serieweerstand.

Invullen en uitwerken levert de functie

$$Z = \frac{R_{IN} \cdot A \cdot e^{\frac{B}{T}}}{R_S \cdot R_{IN} + R_S \cdot A \cdot e^{\frac{B}{T}} + R_{IN} \cdot A \cdot e^{\frac{B}{T}}} \quad (3)$$

Hieruit is T te herleiden:

$$T = \frac{B}{\ln\left(-\frac{Z \cdot R_S \cdot R_{IN}}{A(-R_{IN} + Z \cdot R_S + Z \cdot R_{IN})}\right)} \quad (T \text{ in K}) \quad (4)$$

V_{NTC} is echter ook gelijk aan:

$$V_{NTC} = \frac{ADC_{VALUE}}{ADC_{VALUE_MAX}} \cdot V_{ANALOG_MAX} \quad (5)$$

Hierin is ADC_{VALUE} de huidige waarde van de ADC, ADC_{VALUE_MAX} de ADC-waarde bij de maximale analoge ingangsspanning en V_{ANALOG_MAX} de maximale analoge ingangsspanning. Z is dan eenvoudig te bepalen uit het eerste deel van (2) en in te vullen in (4).

De volgende waarden voor diverse parameters zijn:

$$A = 0,0235664 \Omega$$

$$B = 3863,65 \text{ K}$$

$$R_S = 47.000 \Omega \text{ (nominaal)}$$

$$R_{IN} = 100.000 \Omega \text{ (nominaal)}$$

$$V_{POWER} = 24,0 \text{ V}$$

$$V_{ANALOG_MAX} = 10,0 \text{ V}$$

$$ADC_{VALUE_MAX} = 27648$$

$$^{\circ}\text{C} = \text{K} - 273,15$$

Bijlage 5 – SCL-code temperatuurconversie en min-max testen

```
FUNCTION FC3 : REAL // Function 3

TITLE = 'Convert ADC To Temp'
VERSION: '1.0'
AUTHOR: THR
NAME: TCONV
FAMILY: CONV

//
// This routine converts the raw temperature data from the self
// made NTC temp sensor to the temp in degree C.
//
//              +24 V          The transfer function for Vout is:
//              |
//              +--+          Vout = Z * 24 V
//              |            where
//      Rs       |            Z = -----
//              +--+          Rs + Rntcp
//              |
// ---+-----+----- Vout
// |
// +--+ +--+          Rntcp = {A * exp (B/T)} // Rin
// |    |    |          with A and B being parameters of Rntcp,
// Rin |    |    |          and Rin being the input resistance of
// |    +--+ +--+          the analog input.
// |    |    |
// |    +-----+
// |              |
// |              ===
//

// The tranfer function has to be rewritten to a function with
// T explicit. Note that T is in Kelvin.
// The raw ADC value has a maximum of C600 hex at 10.0 V and
// a minimum of 0000 hex at 0.0 V.
// Please note that with a high power supply voltage, the self
// heating effect of the NTC is considerable.
//

CONST
    VPOWER := 24.0; // Power Supply
    VANALOG_MAX := 10.0; // Max analog voltage
    ADCVALUE_AT_VANALOG_MAX := 27648; // ADC value code @ ...
    A := 0.0235664; // A parameter in Ohm
    B := 3863.65; // B parameter in K
    RS := 47000.0; // Series resistor of 47 KOhm
    RIN := 100000.0; // Input resistance of analog input
    K_TO_C := 273.15; // Convert to Celsius
END_CONST

VAR_INPUT
    ADC_VALUE : WORD; // The input from the analog channel
END_VAR

VAR_OUTPUT
    VMEAS : REAL; // Voltage measured
```

```

    RNTC : REAL;                // Calculated resistance of NTC
END_VAR

VAR_TEMP
    Z : REAL;                  // Conversions
    ZI : INT;
END_VAR

BEGIN
    // ADC value to integer
    ZI := WORD_TO_INT(ADC_VALUE);
    // Voltage measured
    VMEAS := (1.0*ZI)/ADCVALUE_AT_VANALOG_MAX * VANALOG_MAX;
    // Calculate Z as function of ...
    Z := VMEAS/VPOWER;

    // Rntc in Ohms. Function found with the aid of Maple 10.
    RNTC := -(VMEAS*RS*RIN/(-VPOWER*RIN+VMEAS*RIN+VMEAS*RS));

    // Temperature in degree C. Function found with the aid of Maple 10.
    // This is the return value of the function.
    FC3 := B / LN( -1.0*Z*RS*RIN / (A*(-RIN+Z*RS+Z*RIN)) ) - K_TO_C;

END_FUNCTION

FUNCTION FC4 : VOID

TITLE = 'Check Minimum and Maximum Temps'
VERSION: '1.0'
AUTHOR: BRW
NAME: CMM
FAMILY: CHK

CONST
    C_TEMPTOOHIGH := 30.0;    // Higher bound temperature
    C_TEMPTOOLOW := 15.0;    // Lower bound temperature
END_CONST

VAR_INPUT
    TEMPERATURE : REAL;     // The current temperature
END_VAR

VAR_OUTPUT
    TEMPTOOHIGH : BOOL;     // Signals temperature too high
    TEMPTOOLOW : BOOL;     // Signals temperature too high
END_VAR

BEGIN
    IF TEMPERATURE >= C_TEMPTOOHIGH THEN // If current temp too high...
        TEMPTOOHIGH := true;           // signal it.
    ELSE
        TEMPTOOHIGH := false;
    END_IF;
    IF TEMPERATURE <= C_TEMPTOOLOW THEN // If current temp too low...
        TEMPTOOLOW := true;           // signal it.
    ELSE
        TEMPTOOLOW := false;
    END_IF;
END_FUNCTION

```

```

FUNCTION FC8 : REAL

TITLE = 'Record temperature and calculate average'
VERSION: '1.0'
AUTHOR: BRW
NAME: RECCALC
FAMILY: TEMP

VAR_INPUT
    record_temp : BOOL;           // Should we record?
    current_temp : REAL;         // The current temperature
END_VAR

VAR_TEMP
    loop_index : INT;           // Used as loop index
    cumu_temp : REAL;          // Cumulative temp so far
END_VAR

BEGIN
    // If we must record a new temperature
    IF record_temp THEN
        // Do record it
        DB1.temperatuur[DB1.index] := current_temp;
        // Advance the index
        DB1.index := DB1.index + 1;
        // But if it is past the end, flip it to the begin
        IF DB1.index > 10 THEN
            DB1.index := 1;
        END_IF;
    END_IF;

    // Make sure we start at 0.0
    cumu_temp := 0.0;

    // Visit all array items and sum them
    FOR loop_index:= 1 TO 10 BY 1 DO
        cumu_temp := cumu_temp + DB1.temperatuur[loop_index];
    END_FOR;

    // Calculate average
    FC8 := cumu_temp / 10.0;
END_FUNCTION

```

```

FUNCTION_BLOCK FB8

TITLE = 'Record temperature and calculate average'
VERSION: '1.0'
AUTHOR: BRW
NAME: RECCALC
FAMILY: TEMP

VAR_INPUT
    record_temp : BOOL;           // Should we record?
    current_temp : REAL;         // The current temperature
END_VAR

VAR_OUTPUT
    avg_temp : REAL;             // The average temp returns
END_VAR

VAR
    temperature : ARRAY[1..10] OF
        REAL := 10(0.0); // Special treat: 10 items of
temperature initialized to 0.0
    index : INT := 1;           // Index into the array, init. to 1
END_VAR

VAR_TEMP
    loop_index : INT;           // Used as loop index
    cumu_temp : REAL;          // Cumulative temp so far
END_VAR

BEGIN
    // If we must record a new temperature
    IF record_temp THEN
        // Do record it
        temperature[index] := current_temp;
        // Advance the index
        index := index + 1;
        // But if it is past the end, flip it to the begin
        IF index > 10 THEN
            index := 1;
        END_IF;
    END_IF;

    // Make sure we start at 0.0
    cumu_temp := 0.0;

    // Visit all array items and sum them
    FOR loop_index:= 1 TO 10 BY 1 DO
        cumu_temp := cumu_temp + temperature[loop_index];
    END_FOR;

    // Calculate average
    avg_temp := cumu_temp / 10.0;
END_FUNCTION_BLOCK

DATA_BLOCK DB8 FB8
// Empty data block
BEGIN

END_DATA_BLOCK

```

Bijlage 6 – referenties

Bij het samenstellen van deze tutorial is gebruik gemaakt van de volgende documentatie:

Simatic Manuals

- Programming with STEP 7, edition 03/2006, ordernummer A5E00706944-01
- S7-GRAPH V5.3 for S7-300/400: Programming Sequential Control Systems, edition 02/2004, ordernummer A5E00290656-01
- S7-GRAPH V5.3 for S7-300/400: Programming Sequential Control Systems Getting Started, edition 02/2004, ordernummer A5E00290666-01
- S7-SCL V5.3 for S7-300/400, Edition 01/2005, ordernummer A5E00324650-01
- Statement List (STL) for S7-300 and S7-400 Programming, edition 03/2003, ordernummer A5E00706960-01
- S7-300 Automation System: Module Data, edition 02/2007, ordernummer A5E00105505-05
- S7-300 CPU 31xC and CPU 31x, Technical Specifications, edition 12/2006, ordernummer A5E00105475-07
- Ladder Logic (LAD) for S7-300 and S7-400 Programming, edition 03/2006, ordernummer A5E00706949-01
- Configuring Hardware and Communication Connections STEP 7, edition 03/2006, ordernummer A5E00706939-01

Boeken

- Berger, Hans, Automating with STEP7 in STL and SCL, 3rd edition, ISBN 3-89578-243-2, ordernummer A19100-L531-B893-X-7600. Er is nu een vierde editie.
- Berger, Hans, Automating with SIMATIC, 3rd edition, ISBN 3895782769, ordernummer A19100-L531-B927-X-7600

Websites

- PLC open - <http://www.plcopen.org/>
- Siemens Automation Website - <http://support.automation.siemens.com/WW/livelink.exe?func=cslib.csinfo&lang=en&siteid=cseus&aktprim=0&extranet=standard&viewreg=WW&objid=10805055&treeLang=en>

Componenten

- Betatherm NTC - <http://nl.farnell.com/betatherm/10k3a542i/thermistor-ntc/dp/9707271> - let op: dit is een vervanger voor de originele NTC, met iets andere parameters.