



# GETTING STARTED



**HCS Building Automation**  
Kompasstraat 7, 2901 AM Capelle a/d IJssel  
Postbus 182, 2900 AD Capelle a/d IJssel  
Tel. +31(0)10 - 458 00 00  
Fax +31(0)10 - 450 52 11  
E-mail: [info@hcs-ba.nl](mailto:info@hcs-ba.nl)

Versie 2 2 apr 2014

Onderdeel van de Costerm Groep

## Index

1. 1. Eerste stappen met CoDeSys.....	4
1.1. Installeren van CoDeSys.....	4
1.2. Installeren van de target.....	4
1.3. Opstarten van CoDeSys.....	4
1.4. Starten met een nieuw project (target-onafhankelijk).....	4
1.4.1. Tabblad POUs.....	7
1.4.2. Tabblad Datatypes.....	7
1.4.3. Tabblad Visualisations.....	8
1.4.4. Tabblad Resources.....	8
1.5. Maken van een klein programma.....	8
1.5.1. Werken met functies.....	9
1.5.2. Werken met functieblokken.....	15
2. 2. Eerste stappen met de HCS3000.....	18
2.1. Het maken van een project (met gebruik van de Template).....	18
2.1.1. Declareren van in- en uitgangen (met declaratielijst).....	19
2.1.2. Toevoegen van library aan project.....	20
2.1.3. Declareren van een regelgroep KLOK.....	21
2.1.4. Declareren van een regelgroep RAD.....	22

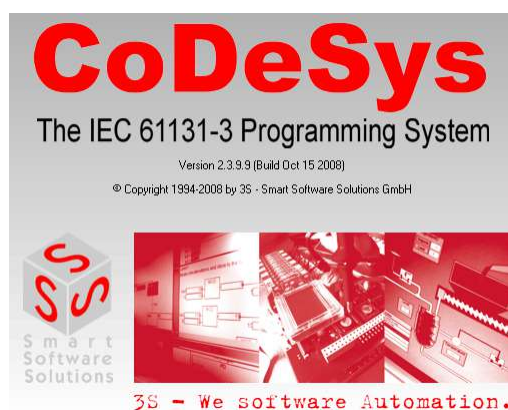
## Inleiding

Deze handleiding is bedoeld voor gebruik met de regelaars van het type HCS3200 en beschrijft de handelingen, welke uitgevoerd moeten worden voor het opstarten van een nieuw project met deze hardware.

Om de HCS3200 te programmeren wordt gebruik gemaakt van het programma CoDeSys, welke werkt met het IEC61131-3 programmeer systeem.

In deze handleiding is ervan uitgegaan dat de meest recente versie van het programma CoDeSys (hoofdversie 2.3), dat gebruikt kan worden voor het programmeren van deze HCS-regelapparatuur, al op de computer geïnstalleerd is.

De meest recente versie van CoDeSys V2.3, welke te gebruiken is in combinatie met de HCS3200 (V1.10) is V2.3.9.35.



In het eerste hoofdstuk wordt in hoofdlijnen de functionaliteit van CoDeSys uitgelegd, waarna in hoofdstuk 2 ingegaan wordt op het maken van een project met de HCS3200.

## 1. Eerste stappen met CoDeSys

### 1.1. Installeren van CoDeSys en target

Voor het installeren van CoDeSys is er verschil tussen 32- en 64-bits besturingssystemen.

Bij het installeren van CoDeSys op een 64-bits besturingssysteem mag CoDeSys niet in de standaard-directory C:\Program Files (x86) geïnstalleerd worden, maar **altijd** in de directory **C:\Program Files**.

Dit om problemen met compatibiliteit met 32-bits systemen te voorkomen.

Voor een gedetailleerde omschrijving van de installatieprocedure verwijzen wij u naar de handleiding 'Installatie CoDeSys en Target.pdf'.

### 1.2. Installeren van de target

Het programma CoDeSys is een programmeeromgeving, welke geschikt is voor gebruik met hardware van diverse fabrikaten.

Om het programmeren van hardware van een bepaald type of fabrikaat met CoDeSys mogelijk te maken, moet hiervoor een bijbehorende target geïnstalleerd worden.

Indien een target niet geïnstalleerd is, kan geen communicatie met de regelapparatuur tot stand gebracht worden.

De handelwijze voor het installeren van de target voor de HCS3200 wordt beschreven in handleiding 'Installatie CoDeSys en Target.pdf'.

### 1.3. Opstarten van CoDeSys

Het programma CoDeSys is op te starten via het startmenu van Windows XP via:

– Start → Programma's → 3S Software → CoDeSys V2.3 → CoDeSys V2.3.

### 1.4. Starten met een nieuw project (target-onafhankelijk)

Als CoDeSys opgestart is, kan met een programma gestart worden door de volgende keuze te maken:

- File → New.

Het dialoogscherm voor het kiezen van de target wordt zichtbaar (zie afbeelding).



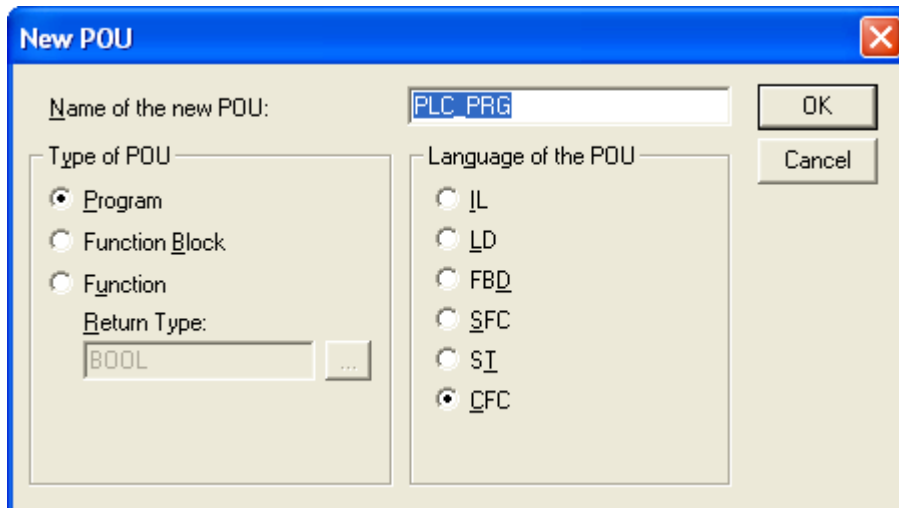
We kiezen hier target 'None' door direct op 'OK' te klikken.

Hierdoor verschijnt het dialoogscherm 'new POU' (zie afbeelding hieronder).

De linker zijde van het dialoogscherm geeft de keuze POU (Program Organization Unit) weer.

In CoDeSys zijn een drietal typen POU's te definiëren, te weten 'Program', 'Function Block' en 'Function'.

Het gebruik van deze POU's wordt verder in de deze handleiding in het kort beschreven.



Voor een volledige omschrijving van bovengenoemde items verwijzen wij naar 'CoDeSys User Manual V2.3'.

In dit geval wordt gekozen voor 'Program'.

In het rechter deel van het dialoogscherm kan een naam voor het programma ingegeven worden en ook de programmeertaal waarin dit betreffende programma gemaakt moet worden.

In een project in CoDeSys moet altijd een startprogramma aanwezig zijn. Indien gebruik gemaakt wordt van een programma-taak, is de naam van het programma vrij (mits deze naam voldoet aan de regels voor naamgeving binnen CoDeSys).

Indien geen programma-taak aanwezig is, zal getracht worden een programma met de naam PLC\_PRG te starten.

Als standaard-naam wordt hier dus PLC\_PRG aangegeven, indien dit programma niet aanwezig is in het project.

CoDeSys geeft de mogelijkheid om de volgende programmeertalen te gebruiken:

- IL (Instruction List)
- LD (Ladder Diagram)
- FBD (Function Block Diagram)
- SFC (Sequential Function Chart)

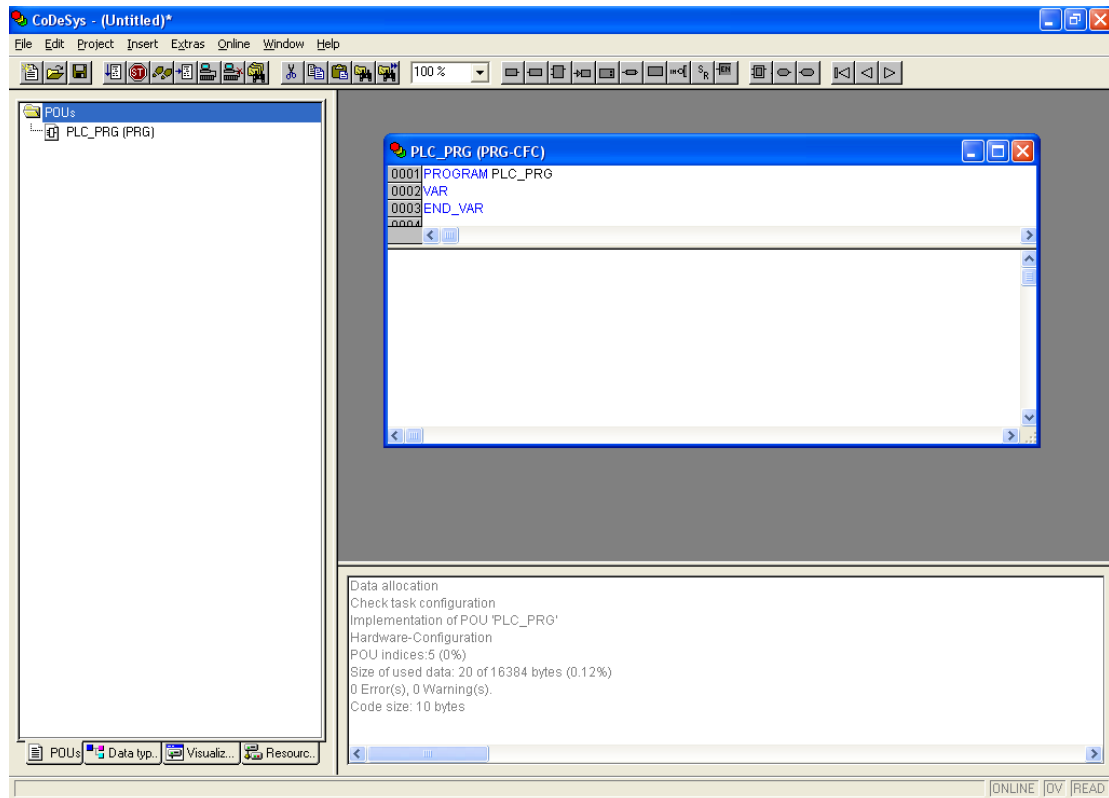
## HCS3000 Getting Started v1.10

- ST (Structured Text)
- CFC (Continuous Function Chart)

Ieder van bovengenoemde programmeertalen heeft specifieke eigenschappen.

In ons geval maken we de keuze CFC, met de naam PLC\_PRG en klikken vervolgens op 'OK'.

Het project ziet er hierna als volgt uit:



De linkerzijde van het beeldscherm geeft de project-structuur weer, welke is ingedeeld in 4 tabbladen, te weten:

- POU's
- Datatypes
- Visualisations
- Resources

De rechterzijde van het scherm geeft de geselecteerde POU weer (in ons geval PLC\_PRG), waarbij dit schermgedeelte opgedeeld is in twee delen.

Het bovenste deel geeft de onderdelen weer, welke in de POU declareerd zijn, zoals variabelen en instanties van functieblokken.

Het onderste deel geeft de daadwerkelijke programmering weer.

### **1.4.1. Tabblad POUs**

Zoals reeds gezegd bestaan er drie soorten POUs (Program Organization Units).

- Programs

Programs (programma's PRG) zijn zelfstandige schakelingen binnen een project, welke door een programma-taak of een aanroep vanuit een ander programma geactiveerd kunnen worden.

Een programma kan ingangen en uitgangen hebben.

De variabelen in een programma gebruiken een eigen geheugenlocatie binnen het project en kunnen hierdoor waarden onthouden.

Binnen een project hoort een programma slechts één keer aangeroepen te worden.

Indien een programma meerdere keren in een project aangeroepen wordt, zal het programma slechts de waarden van de laatste aanroep onthouden.

- Functions

Een function (functie FUN) is een schakeling, welke meerdere ingangen kan hebben, maar slechts één uitgang.

Functies gebruiken geen eigen geheugenlocatie en hebben dus ook geen mogelijkheid om waarden te onthouden.

Functies worden onder andere rekenkundige of logische schakelingen.

- Function Blocks

Een functieblok (functionblock FB) is een schakeling, welke meerdere ingangen en meerdere uitgangen kan hebben.

In tegenstelling tot programma's kunnen functieblokken meerdere malen gebruikt worden in een project.

Om dit te bewerkstelligen dient van het functieblok een instantie (= kopie) gedeclareerd te worden.

Deze instantie (kopie) moet voorzien worden van een eigen (unieke) naam binnen het project en krijgt hierdoor een eigen geheugenlocatie.

Het voordeel van het gebruik van functieblokken is, dat indien in een functieblok een wijziging aangebracht wordt, deze wijziging direct

doorgevoerd wordt in alle instanties van dit functieblok.

### **1.4.2. Tabblad Datatypes**

- Variabelen

Om waarden in een programma te kunnen bewerken, wordt gebruik gemaakt van variabelen.

Deze variabelen kunnen gedeclareerd worden in POUs, zodat deze alleen bestaan binnen de betreffende POU.

Indien een waarde gedeeld moet worden tussen verschillende POUs, bestaat ook de mogelijkheid om een variabele als globale variabele te declareren.

Variabelen kunnen bestaan in veel typen, waarvan er al een groot aantal als standaard aanwezig zijn.

Afhankelijk van de te verwachten waarde, waarvoor de variabele gebruikt gaat worden, moet gekozen worden voor een bepaald type.

Indien bijvoorbeeld de waarde ligt tussen 0 en 255, volstaat het om het

datatype BYTE te gebruiken.

Voor waarden tussen 0 en 65535 kan gekozen worden voor het datatype WORD.

Het is ook mogelijk om eigen datatypes te maken, bijvoorbeeld om de leesbaarheid van het programma te verbeteren, of om complexe datastructuren (structs) te kunnen gebruiken.

Voor details over het gebruik van datatypes en -structuren verwijzen wij naar 'CoDeSys User Manual V2.3'.

### **1.4.3. *Tabblad Visualisations***

Er is de mogelijkheid om waarden van variabelen zichtbaar te maken in grafische vorm.

Hiervoor worden visualisaties gebruikt.

In deze visualisaties kunnen de waarden van de variabelen in het project weergegeven worden.

Het wijzigen van de waarden door middel van bedienings-objecten, zoals sliders en keyboards is ook mogelijk.

Voor details over het gebruik van visualisaties verwijzen wij naar 'CoDeSys User Manual V2.3'.

### **1.4.4. *Tabblad Resources***

De inhoud van het tabblad resources geeft alles weer wat te maken heeft met instellingen en onderdelen van het project, zowel als het programma.

Een deel van de resources wordt in deze handleiding behandeld.

Voor de rest van de details over het gebruik van resources verwijzen wij naar 'CoDeSys User Manual V2.3'.

## **1.5. *Maken van een klein programma***

Het gemaakte project heeft nu een programma PLC\_PRG.

Dit programma heeft echter geen inhoud.

Het is mogelijk om binnen dit programma een schakeling te maken.

Om aan te geven wat de verschillen tussen de diverse programmeertalen binnen IEC61131-3 zijn, wordt in de volgende hoofdstukken een eenvoudige schakeling gemaakt, echter steeds weergegeven in een andere taal.

Het wordt een programma, waarbij een variabele met de naam OUT aangestuurd wordt, als één van de variabelen met de naam IN1 of IN2 actief wordt.

### **1.5.1. *Werken met functies***

CFC is een programmeertype, waarbij de schakelingen in grafische vorm weergegeven worden.

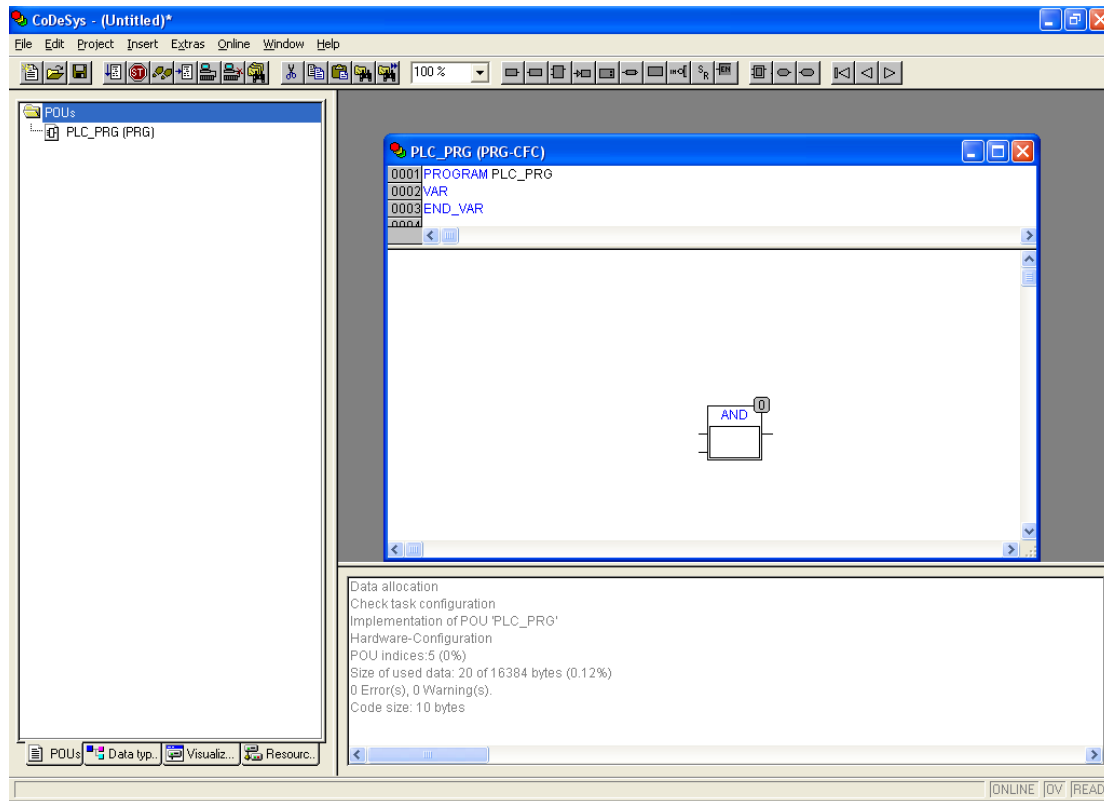
Voor het voorbeeldproject wordt vanwege de overzichtelijkheid de programmeervorm CFC gebruikt.



## HCS3000 Getting Started v1.10

Een functie is toe te voegen aan het programma door middel van 'Insert' → 'Box', of door de toetscombinatie Ctrl-B.

Vervolgens kan met de muis het geselecteerde onderdeel ergens in het onderste deel van het programma PLC\_PRG neergezet worden door op de linker muistoets te klikken.



De functie die verschijnt is de AND-functie.

In ons voorbeeldproject moet de functie gewijzigd worden in een OR-functie.

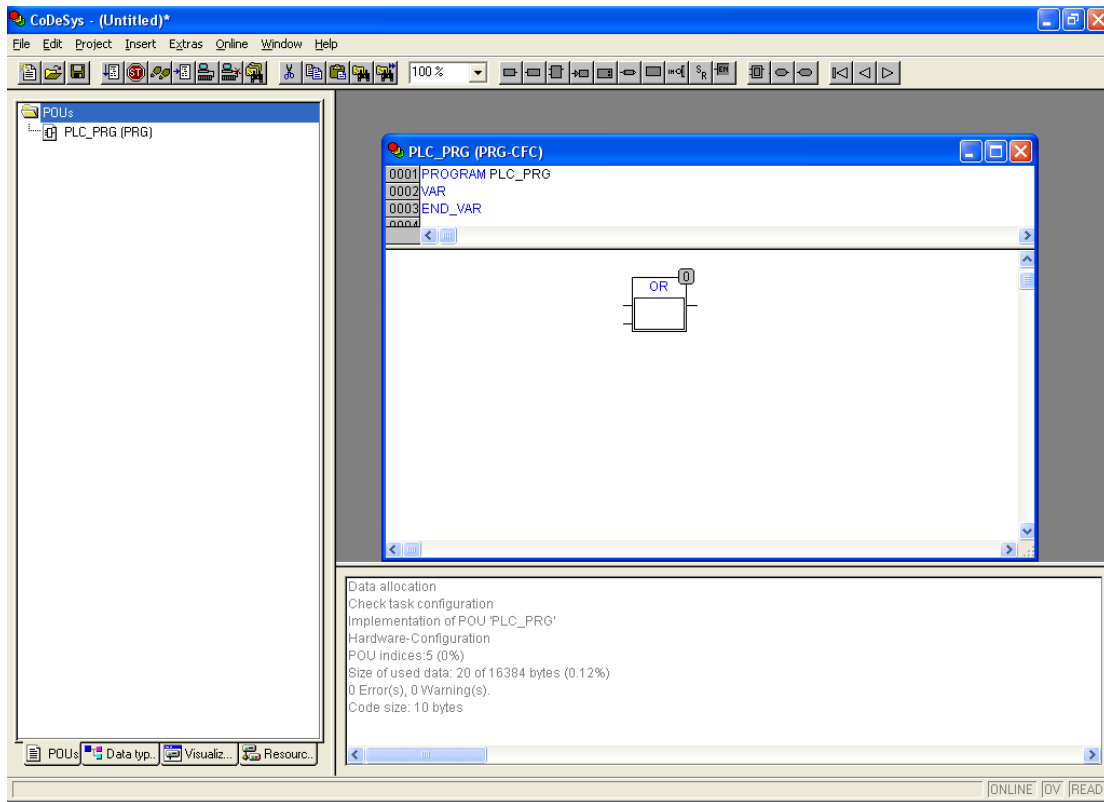
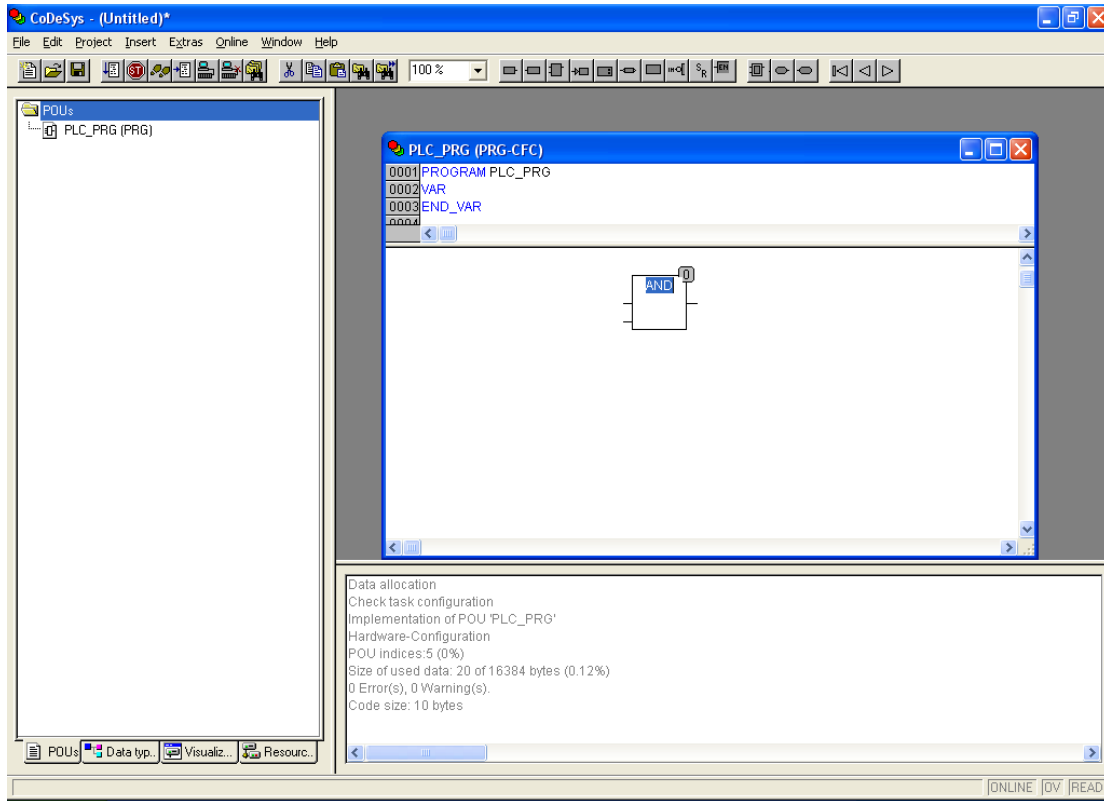
Dit gaat als volgt:

Klik met de linker muistoets op de tekst AND, waardoor deze geselecteerd wordt.

Vervolgens kan met het toetsenbord de tekst gewijzigd worden in OR.

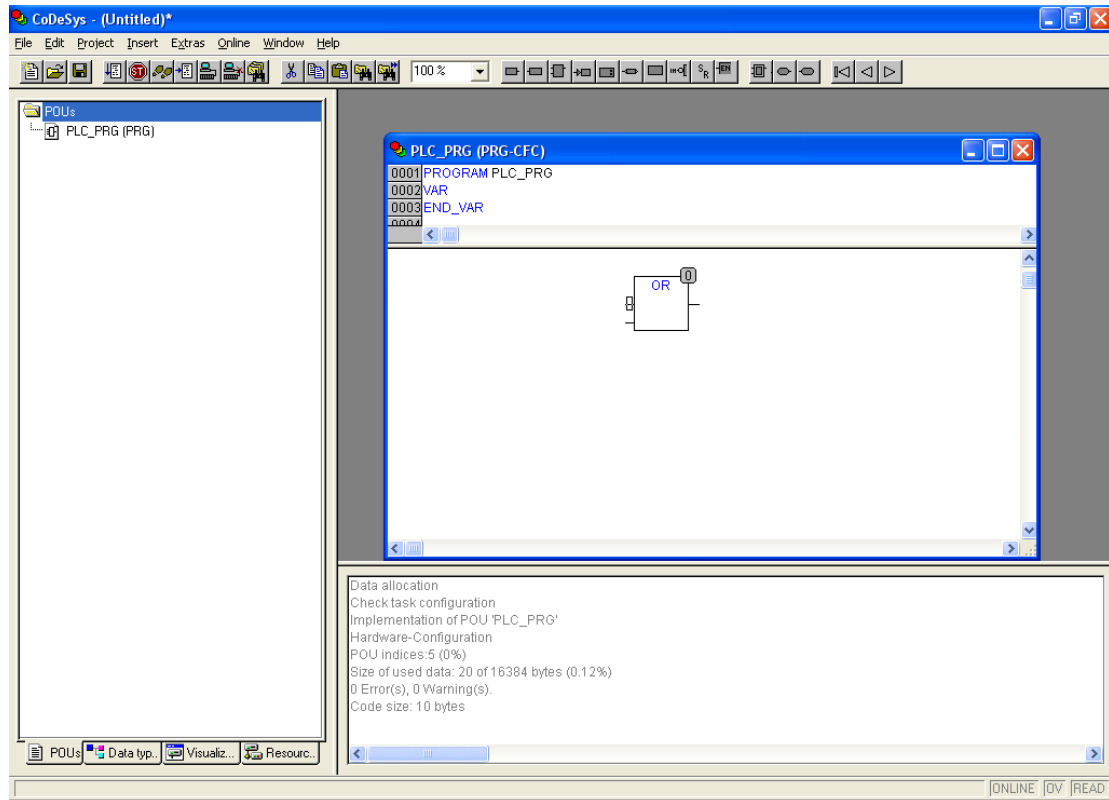
Afsluiten met Enter, voert de wijziging door.

# HCS3000 Getting Started v1.10



Om de variabelen te koppelen aan de ingangen van de OR-poort, kan de betreffende ingang geselecteerd worden, door deze met een enkele klik van

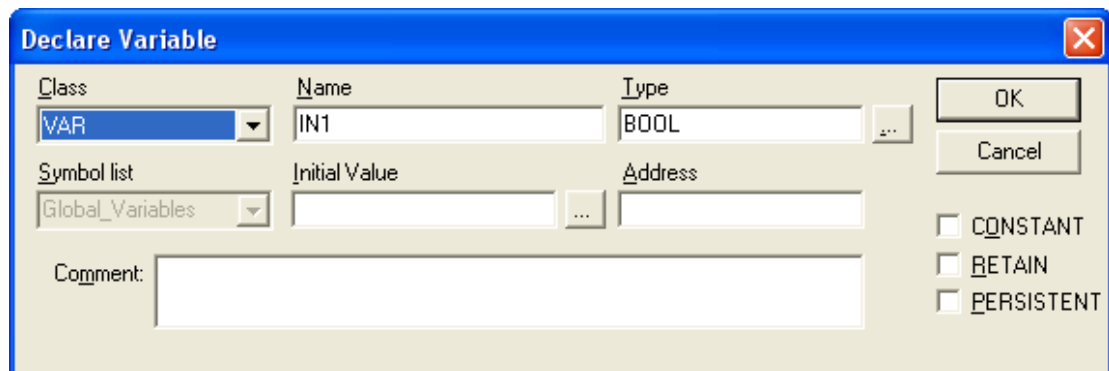
de linker muistoets.



Vervolgens kan de naam van de ingangsvariabele met het toetsenbord ingevoerd worden.

In ons geval heeft de variabele de naam IN1.

Na het indrukken van Enter verschijnt het volgende dialoogscherf:



Het programma CoDeSys is voorzien van auto-declaratie waardoor, bij het invoeren van een variabele-naam, gecontroleerd wordt of de variabele al bestaat in het project.

Indien een naam niet bestaat, wordt automatisch de dialoog geopend voor het declareren van de variabele.

- Class

Met de pull-down 'Class' kan het type variabele geselecteerd worden. De mogelijke keuzen zijn VAR (= interne variabele in het programma), VAR\_INPUT (= ingangsvariabele), VAR\_OUTPUT (= uitgangsvariabele), VAR\_IN\_OUT (= variabele, welke die als ingang en als uitgang gebruikt kan worden) en VAR\_GLOBAL (= globale variabele).

Bij de keuzes VAR\_INPUT en VAR\_IN\_OUT wordt de waarde van de variabele van buiten de POU aangeleverd.

Bij de keuze VAR\_OUTPUT wordt de variabele gebruikt om een waarde uit de POU naar buiten uit te voeren.

- Name

In het veld 'Name' kan de naam van de variabele ingevuld worden.

Dit veld is in ons geval reeds ingevuld door de auto-declaratie.

- Type

In het veld 'Type' kan het datatype aangegeven worden.

Omdat ons voorbeeld gebruikt wordt voor een logische schakeling

(BOOLEAN waarde 0 of 1) wordt hier gekozen voor datatype BOOL.

Indien de button rechts van het veld 'Type' aangeklikt wordt, wordt een overzicht getoond van de mogelijke datatypen.

- Symbol List

In geval dat de class VAR\_GLOBAL gekozen is, kan in de pull-down 'Symbol List' de lijst met globale variabelen gekozen worden, waarin de nieuwe variabele gedeclareerd moet worden.

- Initial Value

In het veld 'Initial Value' kan de waarde ingevoerd worden welke de variabele bij de start van het programma moet hebben.

- Address

In het veld 'Address' kan het adres van de geheugenlocatie ingevuld worden welke aan deze variabele toegewezen wordt.

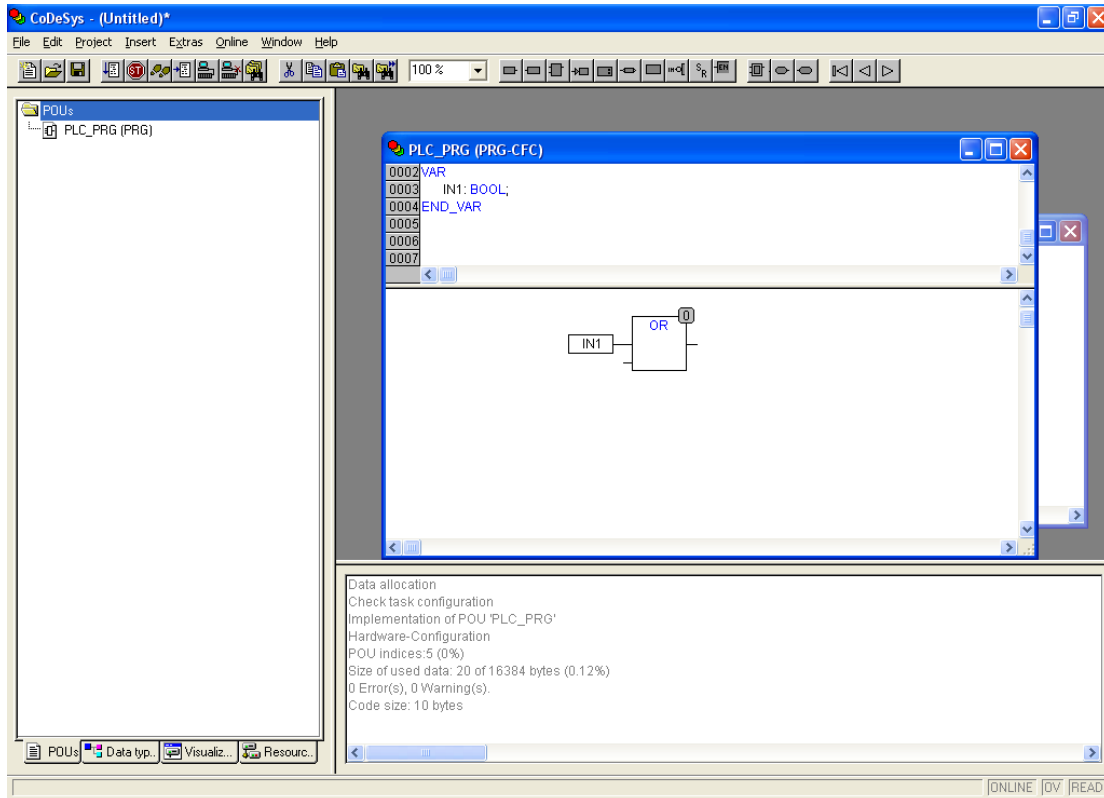
Indien hier niets ingevuld wordt, wordt het adres automatisch bepaald.

**Let op! Foutief invullen van een geheugenadres kan onvoorspelbaar gedrag van het programma veroorzaken.**

In ons voorbeeld wordt de variabele IN1 gedeclareerd als VAR, met als datatype BOOL.

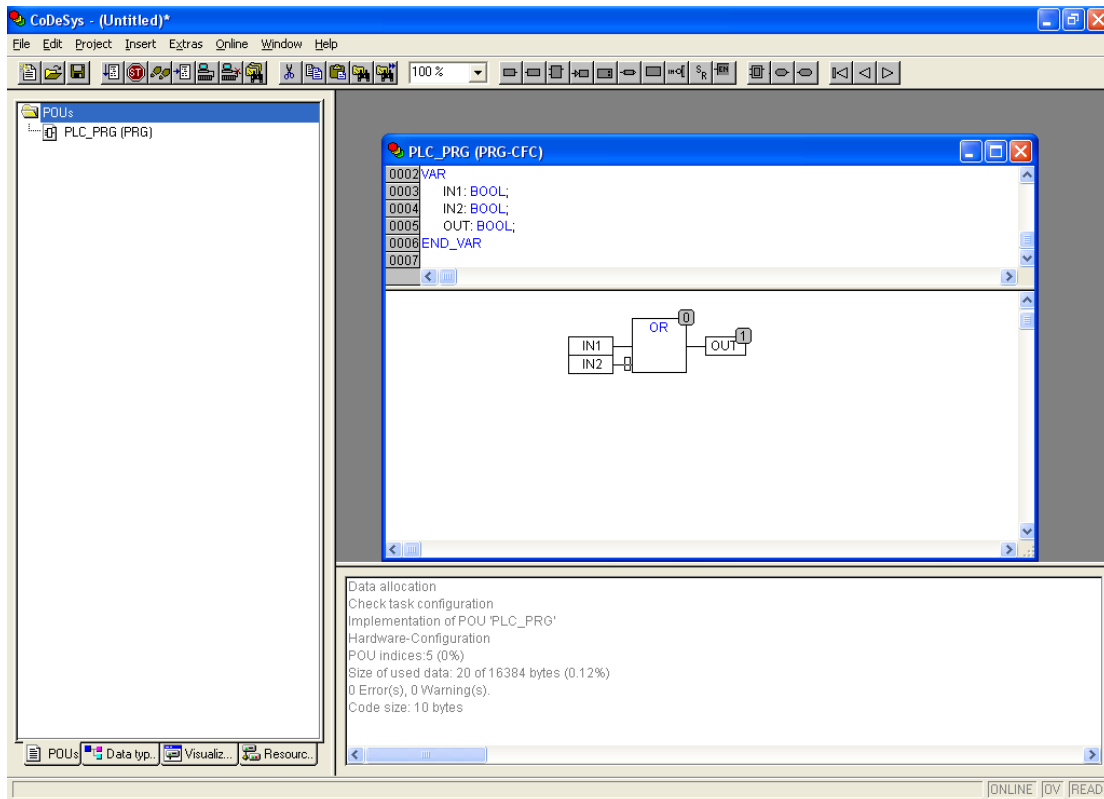
Na het succesvol declareren van de ingangsvariabele IN1 ziet het programma er als volgt uit:

# HCS3000 Getting Started v1.10



Doe vervolgens hetzelfde met ingangsvariabele IN2 en uitgangsvaariabele OUT.

Het programma ziet er dan als volgt uit:



## HCS3000 Getting Started v1.10

Indien er geen target geselecteerd is, wordt in CoDeSys automatisch de simulatie-mode actief gemaakt.

Hierdoor is het mogelijk om ook zonder de hardware aangesloten te hebben een stuk van het programma te maken en te testen.

Het testen van dit voorbeeldprogramma kan uitgevoerd worden door 'Online' → 'Login' (sneltoets Alt-F8) in te voeren.

Hierdoor wordt het programma gecontroleerd op juistheid en vervolgens wordt 'online' gegaan met het programma.

In geval van simulatie, wordt gesimuleerd dat er regelapparatuur aangesloten is.

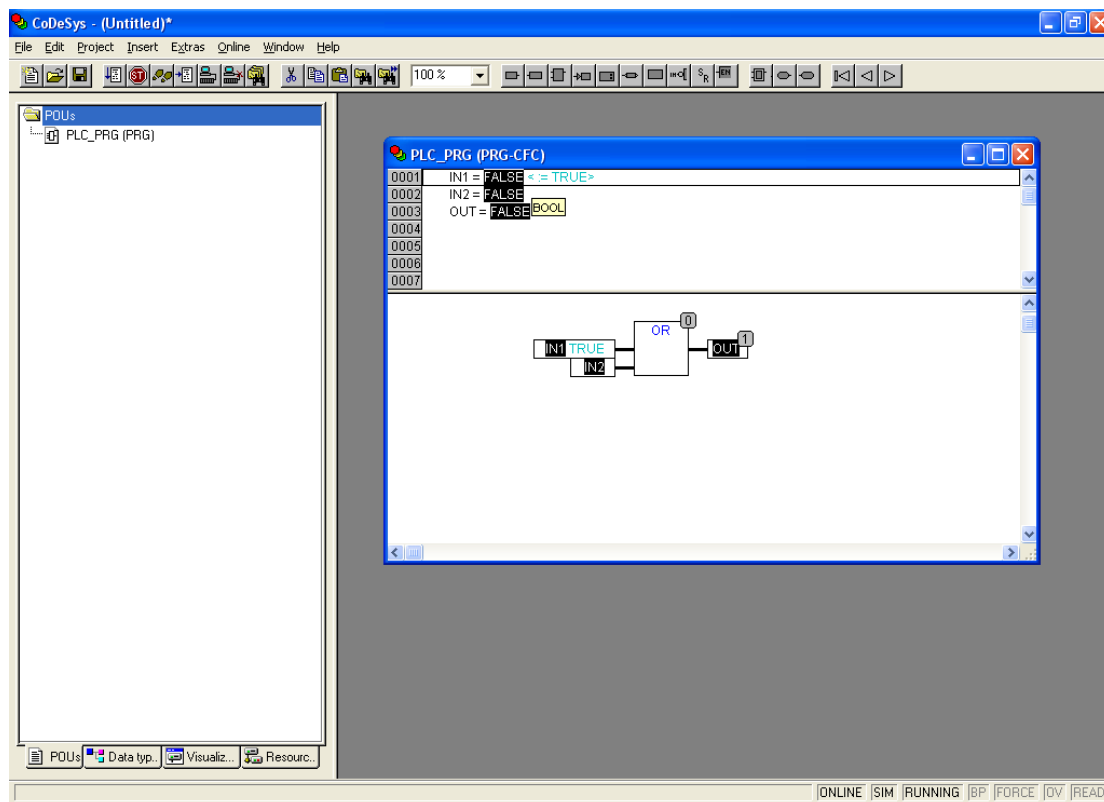
In de statusbalk onderin het scherm is te zien wat de huidige status van het project is, waarbij 'ONLINE' aangeeft dat er verbinding is met een programma (of project).

'SIM' geeft aan dat gewerkt wordt in de simulatie-mode.

De tekst 'RUNNING' is echter nog grijs, omdat het project nog niet gestart is. Starten van het project kan gedaan worden door 'Online' → 'Run' (sneltoets F5) in te drukken.

Om te controleren of het programma correct werkt, kan de waarde van de variabelen handmatig gewijzigd worden, door op de betreffende variabele te dubbelklikken met de linker muistoets.

Hierdoor verschijnt aan de rechterzijde van de variabelen (in lichtblauw) de waarde welke ingesteld moet gaan worden.

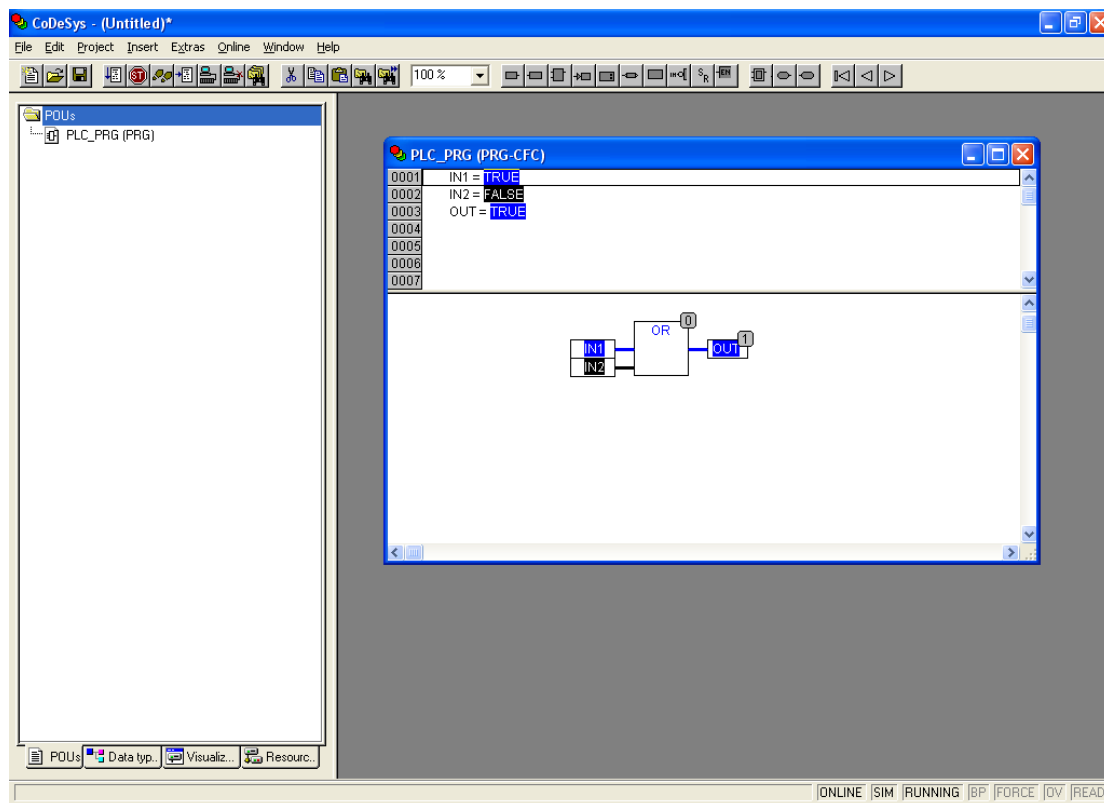


Het doorvoeren van de waarde in de variabele kan gedaan worden met 'Online' → 'Write Values' (sneltoets Ctrl-F7).

In de afbeelding is nu ook te zien dat de waarde daadwerkelijk gewijzigd is.

Omdat het data-type BOOL is, wordt een 0 (FALSE) in zwart weergegeven, terwijl een 1 (TRUE) in donker blauw weergegeven wordt.

Ook de uitgangsvariabele OUT is in donker blauw weergegeven, wat aanduidt dat deze waarde ook meeveranderd is.



Op deze manier is eenvoudig een programma te testen.

### 1.5.2. Werken met functieblokken

Naast functies, bestaan er ook functieblokken.

Waar functies slechts één uitgang hebben, hebben functieblokken veelal meer dan één uitgang.

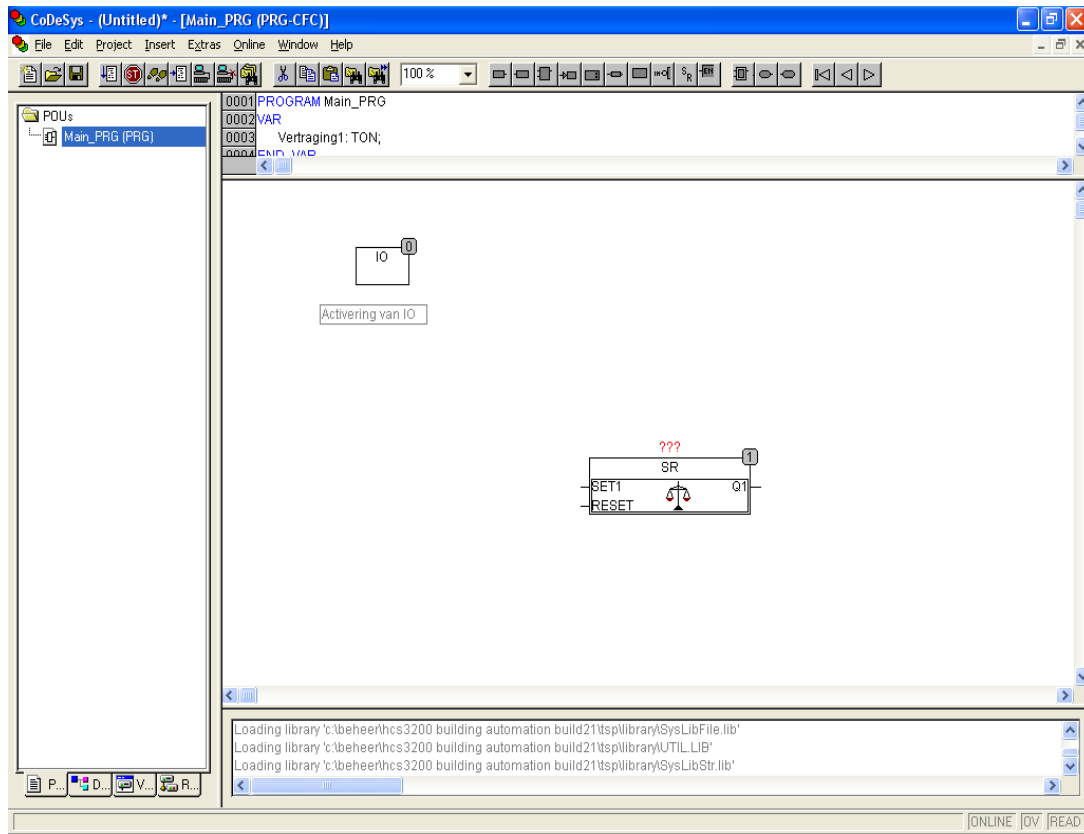
In ons volgende programeervoorbeeld wordt gebruik gemaakt van een functieblok.

Met Ctrl-B wordt een box geplaatst in het werkgebied (wederom een AND-poort).

Verander de tekst AND in SR en druk op Enter.  
De AND-poort is nu veranderd in een SR-flipflop.

Deze (en andere functies) zijn als standaard aanwezig in de ingevoegde libraries van de template.

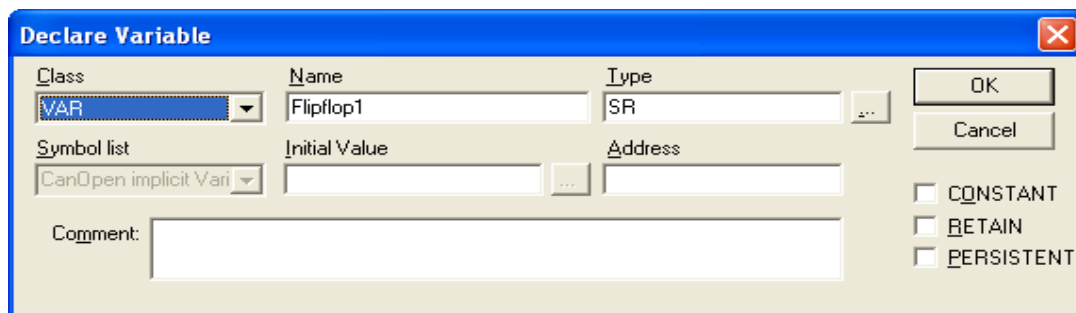
Het programma ziet er nu als volgt uit:



Boven het geplaatste blok staan een drietal vraagtekens (???).

Om een instantie van het functieblok te declareren, kan op de plaats van de vraagtekens een naam ingevoerd worden.

In ons geval wordt gekozen voor 'Flipflop1', waarbij na het indrukken van Enter het dialoogscherm voor de declaratie verschijnt, zoals we al eerder gezien hebben bij het declareren van variabelen.



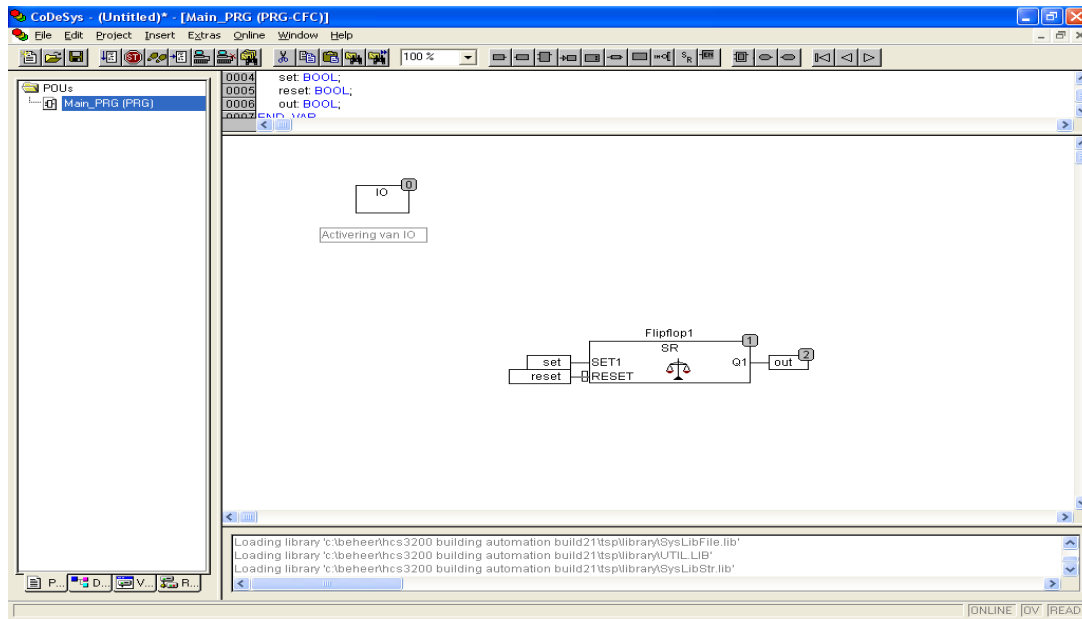
De werking is gelijk aan de werking bij variabelen, echter hier wordt het datatype SR gekozen.

Aan de in- en uitgangen van de SR moeten vervolgens variabelen aangesloten worden.

Hierbij kiezen we voor de ingangen (datatype BOOL) 'set' en 'reset' en de



uitgang (datatype BOOL) 'out'.  
Het programma ziet er als volgt uit:



## 2. Eerste stappen met de HCS3000

### 2.1. Het maken van een project (met gebruik van de Template)

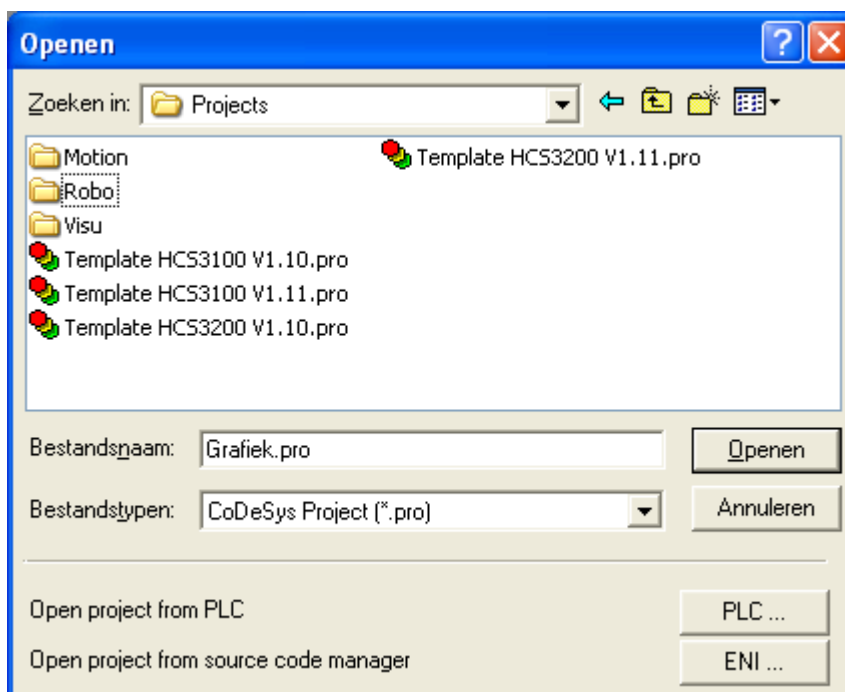
Ieder met CoDeSys gemaakt project kan als grondlegger (template) gebruikt worden voor het starten van een nieuw project.

Voor gebruik van de HCS3100 en de HCS3200 zijn templates gemaakt, waarin de voor een project benodigde hoofdonderdelen al aanwezig zijn.

Als CoDeSys opgestart is moet, om een nieuw project te starten vanaf de template, de volgende keuze gemaakt worden:

- 'File' → 'New from template...'

Het dialoogscherm 'openen' wordt zichtbaar (zie afbeelding hieronder).



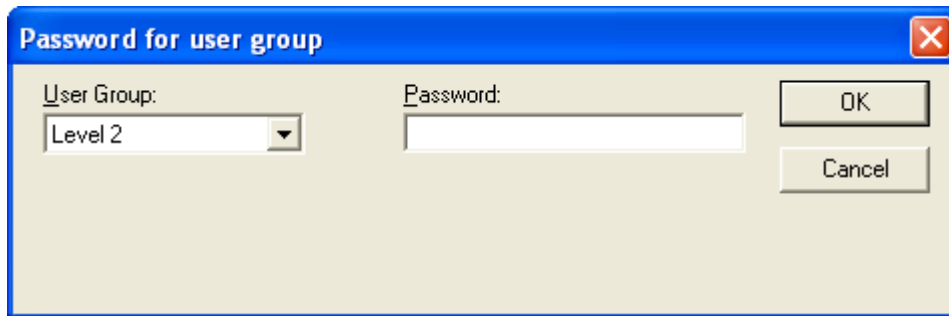
De template is te vinden in C:\Program Files\3S Software\CoDeSys V2.3\Projects, onder de naam 'Template HCS3200 V1.10.pro' of 'Template HCS3100 V1.10.pro', afhankelijk van de gekozen hardware.

Klik op 'Openen' om het nieuwe project te openen.

Het dialoogscherm 'Password for user group' verschijnt.

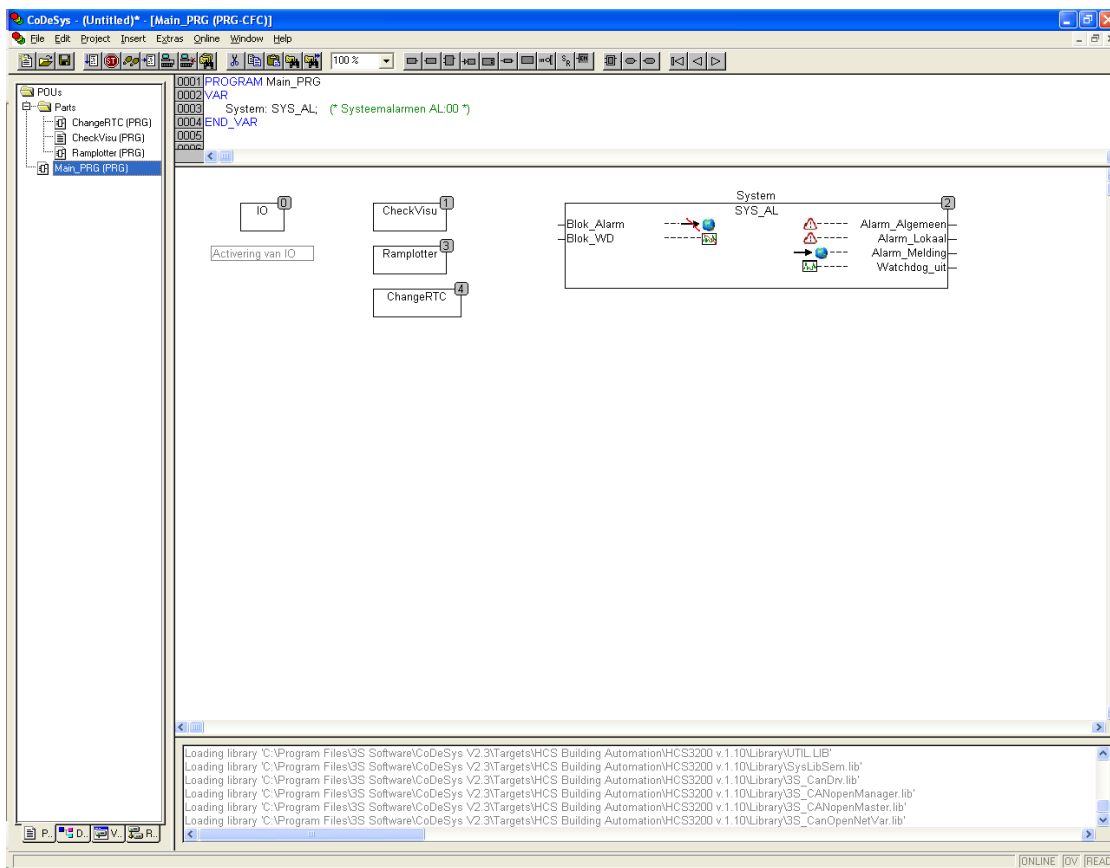
Het programmeren van het project gebeurt onder gebruikersgroep niveau 2, welke niet beveiligd is met een wachtwoord.

Na het klikken van 'OK' wordt het nieuwe project geopend.



In dit nieuwe project is al één POU (Main\_PRG) aanwezig, welke het startpunt voor het project vormt.

Iedere POU, welke in het project opgenomen moet worden, dient in de POU Main\_PRG aangeroepen te worden.



In Main\_PRG zijn al een programma-aanroepen aanwezig van enkele programmas, welke de koppeling naar de eigen IO van de regelaar en andere zaken verzorgen.

Het project kan onder een eigen naam opgeslagen worden door het commando 'File' → 'Save as...!.

### 2.1.1. Declareren van in- en uitgangen (met declaratielijst)

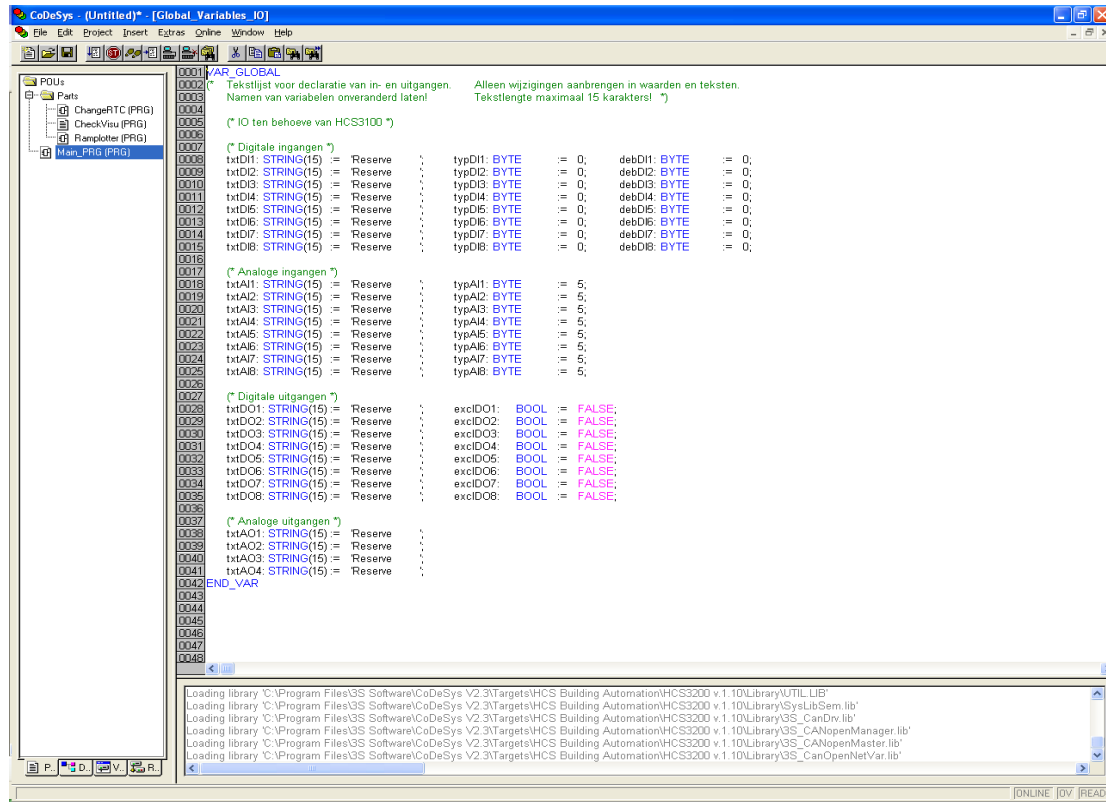
De hardware-matige in- en uitgangen van de HS3200 zijn in de template al

gedeclareerd.

De namen en typen, kunnen echter per project verschillen en zijn dus ook door in het CoDeSys-project te wijzigen.

Hiervoor is in de globale variabelen een declaratielijst gemaakt, met de naam 'Global\_Variables\_IO'.

Deze lijst is te vinden in tabblad 'Reources', onder 'Global Variables'.



De teksten en typen van alle aanwezige in- en uitgangen zijn in de lijst aanwezig en kunnen gewijzigd worden.

De koppeling van deze gegevens met de hardware van de regelaar wordt uitgevoerd door het programma IO, welke standaard in programma Main\_PRG aangeroepen wordt.

De het aantal karakters voor de IO-namen is beperkt tot 15 letters.

### 2.1.2. Toevoegen van library aan project

In de template van de HCS3200 zijn alle basisfuncties en -functieblokken opgenomen, waaronder de ketel-PID-regeling, koppeling van alarmen en dergelijke.

Voor gedetailleerde informatie over deze functies verwijzen wij u naar de handleiding van de betreffende HVAC-library.

Voor gebruik van specifieke groeptypen wordt gebruik gemaakt van separate

## HCS3000 Getting Started v1.10

libraries, welke functies bevatten die niet in de standaard-library aanwezig zijn.

Het betreft hier de volgende libraries:

HCS3000 v1.10 BO.lib	–	Boilers
HCS3000 v1.10 EG.lib	–	Eigen groepen
HCS3000 v1.10 KE.lib	–	Ketels
HCS3000 v1.10 KL.lib	–	Schakelklokken
HCS3000 v1.10 LU.lib	–	Luchtgroepen
HCS3000 v1.10 RD.lib	–	Radiatorgroepen
HCS3000 v1.10 VB.lib	–	Veldbusmodulen

Om een aanvullende library te laden in het project gaat u naar 'Window' → 'Library Manager'.

Hierdoor wordt de library-manager geopend, waarin alle in het project geladen libraries getoond worden.

Vervolgens gaat u naar 'Insert' → 'Additional library'.

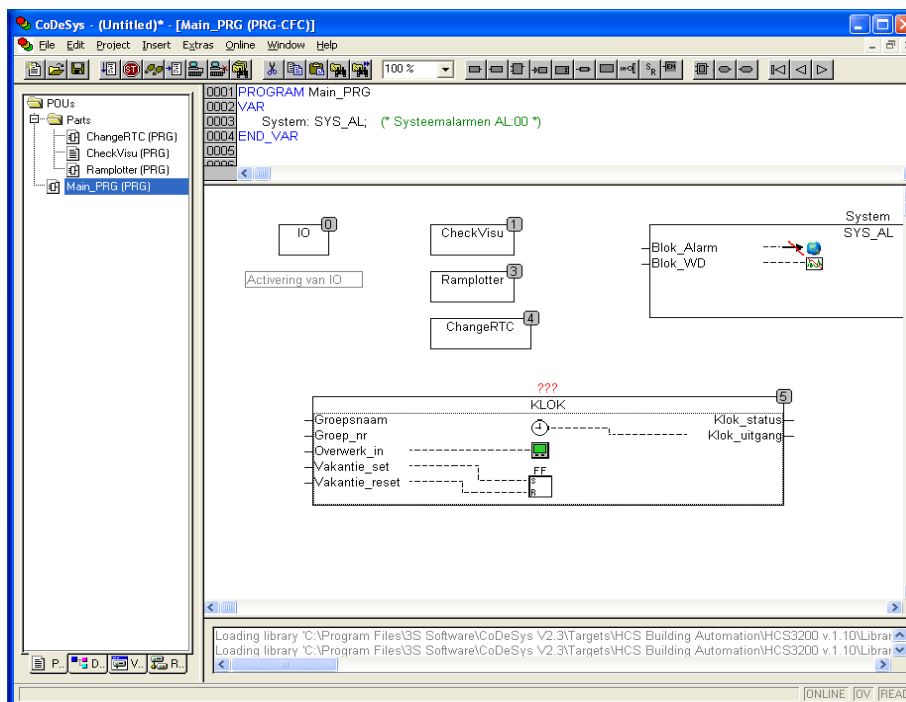
De hierboven genoemde libraries zijn te vinden in de target-directory van Codesys. (C:\Program Files\3S Software\CoDeSys V2.3\Targets\HCS Building Automation\HCS3200 v.1.10\Library).

### 2.1.3. Declareren van een regelgroep KLOK

Het declareren van een van deze functieblokken is gelijk aan het declareren van iedere andere functie of functieblok.

Met Ctrl-B wordt een blok geplaatst.

Selecteer vervolgens de tekst AND en verander deze in KLOK (sluit af met Enter).



## HCS3000 Getting Started v1.10

Het beeldscherm ziet er dan als volgt uit:

Geef de instantie van het functieblok KLOK een eigen naam, door deze in te voeren in de plaats van de vraagtekens.

Wij kiezen voor de naam kl1.

Aan de linkerzijde zijn een vijftal ingangsvariabelen, om de klok nader te definiëren.

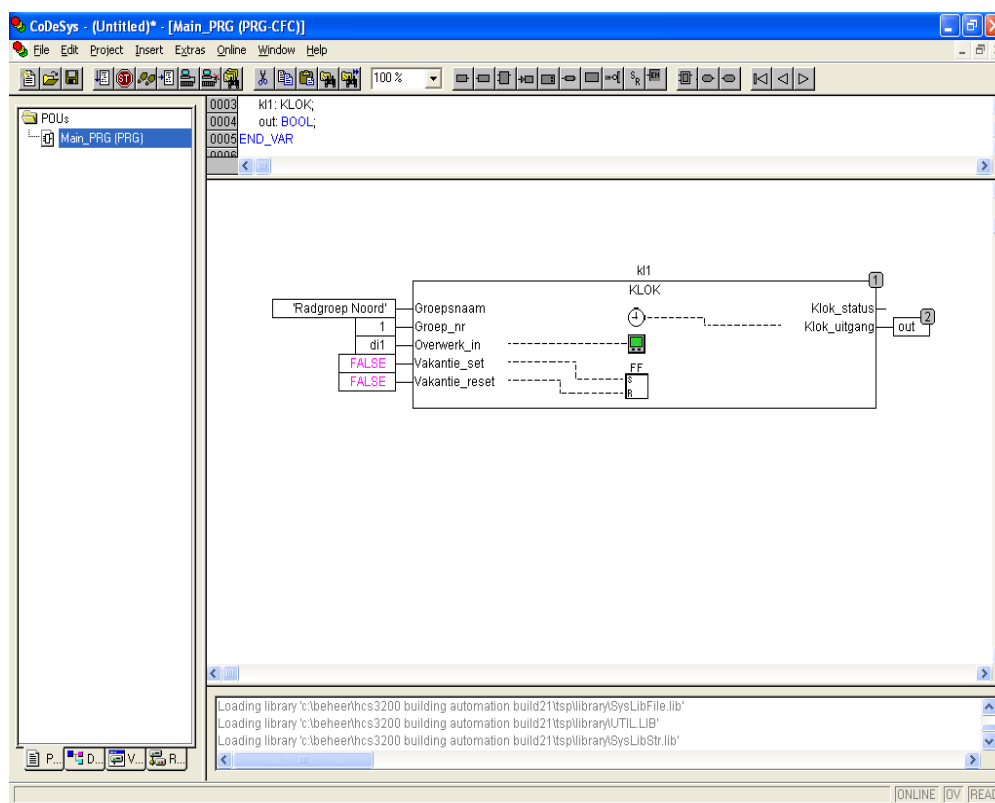
Bij 'Groepsnaam' kan de naam van de groep (15 letters) ingevoerd worden. Deze ingang is van het datatype STRING, waarbij de tekst moet ingevoerd zijn tussen enkele aanhalingstekens.

De ingang 'groep\_nr' geeft het nummer van de groep in de regelaar aan. Bijvoorbeeld '1'.

Het datatype is BYTE, zodat het getal zonder aanvullende tekens ingevoerd kan worden.

De ingang 'Overwerk\_in' is om de overwerk-ingang te koppelen aan (bijvoorbeeld) de betreffende ingangsvariabele.

In ons geval kiezen we voor digitale ingang DI1, door deze tekst in te voeren.



### 2.1.4. Declareren van een regelgroep RAD

De radiatorgroep is aanwezig in de library 'HCS3000 V1.10 RD.lib'.

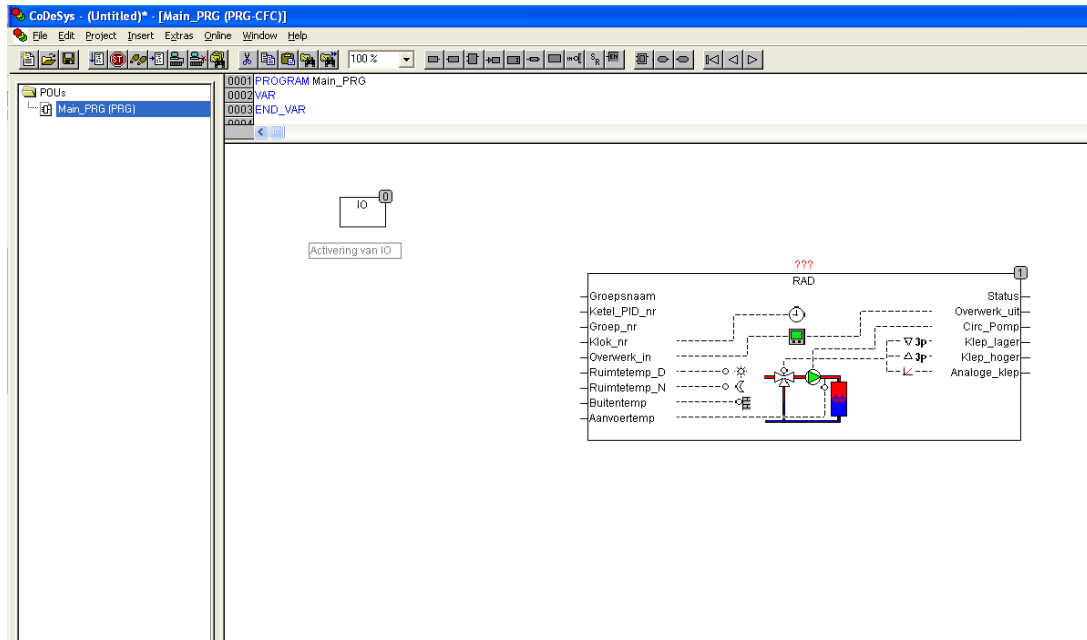
Voeg deze library in het project in, zoals in hoofdstuk 2.1.2 omschreven is.

Met Ctrl-B wordt een blok geplaatst.

## HCS3000 Getting Started v1.10

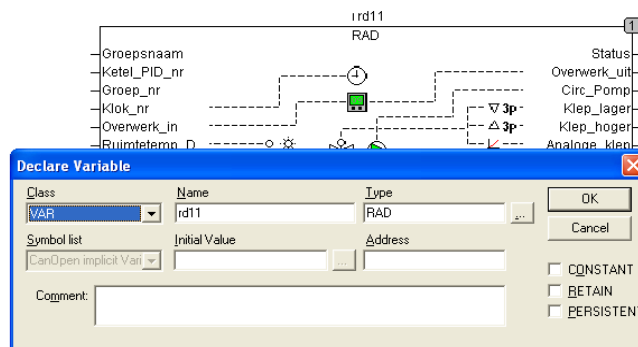
Selecteer vervolgens de tekst AND en verander deze in RAD (sluit af met Enter).

Het beeldscherm ziet er dan als volgt uit:



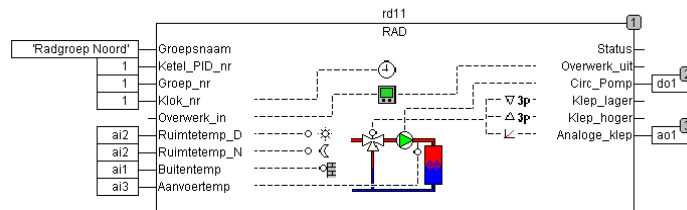
Voorzie het functieblok RAD van een instantie-naam (welke uniek is binnen het project), bijvoorbeeld 'rd11'.

CoDeSys toont nu een pop-up, voor het definiëren van de instantie.



Naam en type zijn al ingevuld.  
Klik op OK om dit te bevestigen.

*Let op! Als CoDeSys niet de pop-up toont, betekent dit dat de naam welke voor de instantie gekozen is, al in het project bestaat.  
Kies in dit geval een andere naam.*



Koppel vervolgens de in- en uitgangen van de radiatorgroep.

De variabele 'Groepsnaam' bepaalt welke naam getoond wordt in het display van de regelaar, de visualisatie en via telecommunicatie.

Het groepnummer en ketel-PID-nummer zijn van belang en bepalen de identiteit van deze groep binnen de regelaar.

In de regelaar zijn maximaal 2 ketel-PID-regelingen actief, waaraan de diverse groepen hun warmtevraag doorgeven.

De koppeling tussen de radiatorgroep en de bijbehorende ketel gebeurt automatisch.

Hiervoor zijn geen handmatige koppelingen nodig.

Dit geldt ook voor de koppelingen van de ketelgroepen naar boiler- en luchtgroepen.

Om de ingang van de ruimtetemperaturopnemer te koppelen kan eenvoudig de naam van de ingang ingetypt worden (in het voorbeeld ai2).

Evenzo voor de koppeling naar digitale of analoge uitgangen voor aansturen van kleppen en pompen.