

CC Pilot™ XS

CoDeSys developers guide



Table of Contents

Introduction.....	3
Purpose	3
References	3
History.....	3
How to - descriptions	4
Use Target installation	4
Create a simple 'hello world' application	5
Create an application with visualization	6
Download application to target.....	12
Save the project in non volatile memory	14
Only load current bitmaps in Visu	14
Start programs from CoDeSys.....	14
Start PLCWinCE minimized.....	16
Autostart the PLC runtime.....	16
Set up a CAN network	18
Configure CAN settings	20
Set bus identifier	21
Use retain variables	22
Use network variables.....	26
Configure a CanDevice (CANopen Slave)	30
Configure CC Pilot XS CoDeSys RunTime settings	32
Use simulated target.....	40
Use CC Systems simulation platform - CCSimTech	40

Introduction

Purpose

This document contains tips and how to's for developing applications for the CoDeSys runtime on the CC Pilot™ XS.

References

-

History

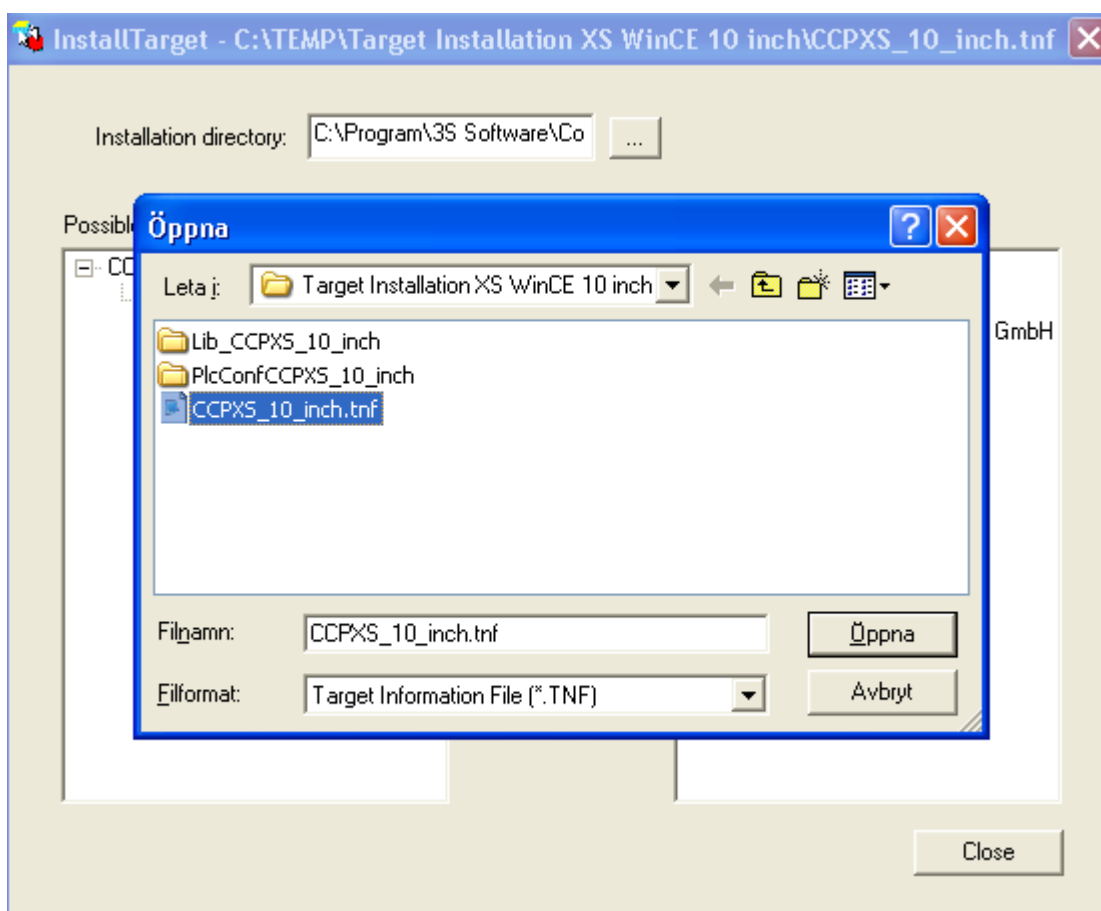
Rev	Date
1.0	2008-04-22
1.1	2008-08-22

How to - descriptions

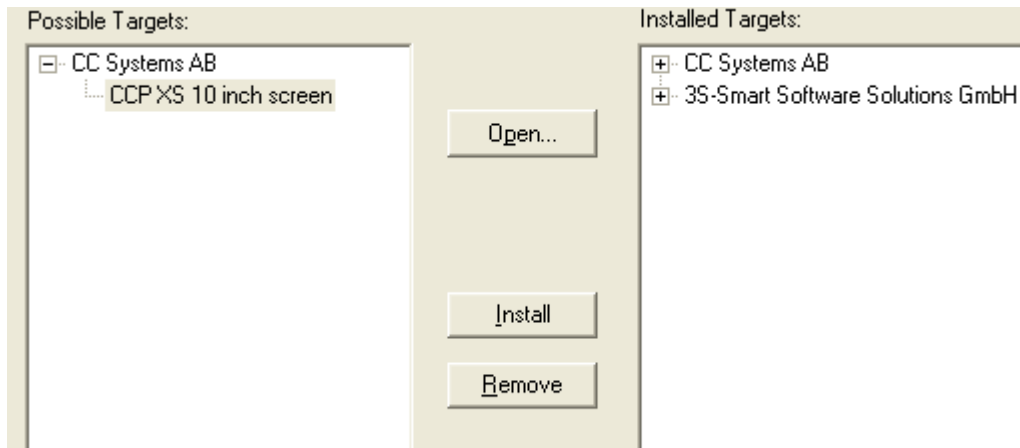
Use Target installation

For every target that is used in CoDeSys a prior target installation is required. These target installations can be found on www.cc-systems.com. The installation package comes in the form of a zip file, so first of all unzip the folder to some temporary location.

- Start the Install target software, which is located in the start menu under the CoDeSys folder
- When started, click Open and browse for the .tnf file in the unzipped folder



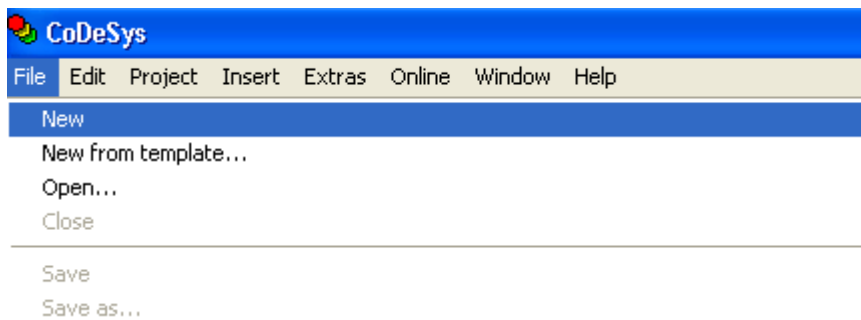
- Choose the target to in the left and hit Install



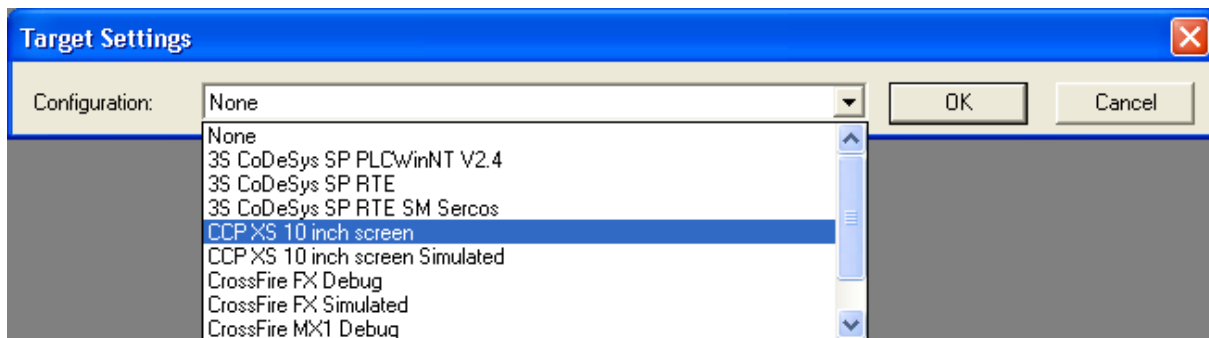
The target is now installed on the development PC, along with the required libraries. The next time a new project is created it will be possible to choose the newly installed target.

Create a simple 'hello world' application

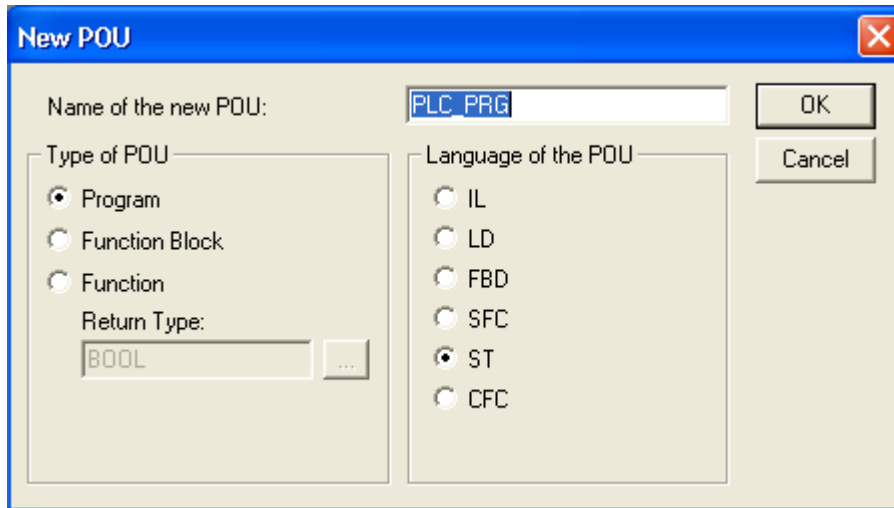
- Choose File→New



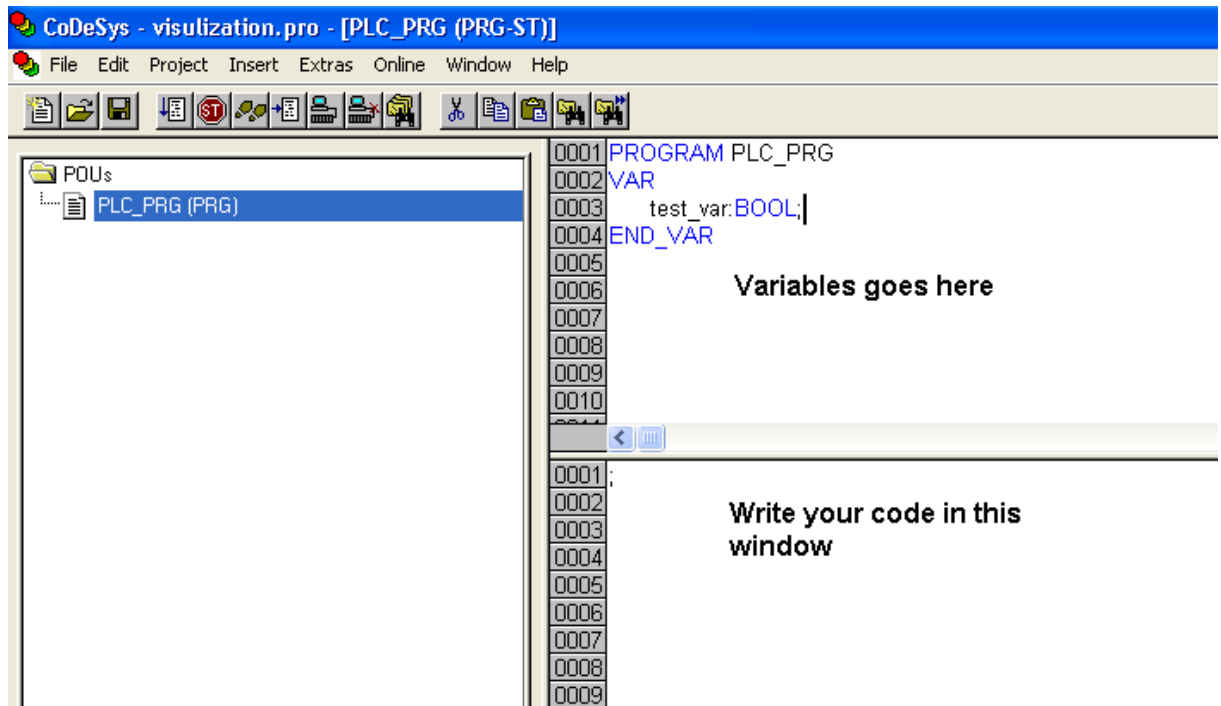
- In the Target Settings dialog under Configuration choose e.g. CCP XS 10 inch screen, this can be changed later to run the application on another target



- Accept the default configuration in the New POU dialog



PLC_PRG is the main default program. There are two separate windows for every POU (Program Organization Unit), the top window is for variables and the bottom is for the code. Just type a semicolon in the code section if you want to go ahead and with other things and still be able to compile.

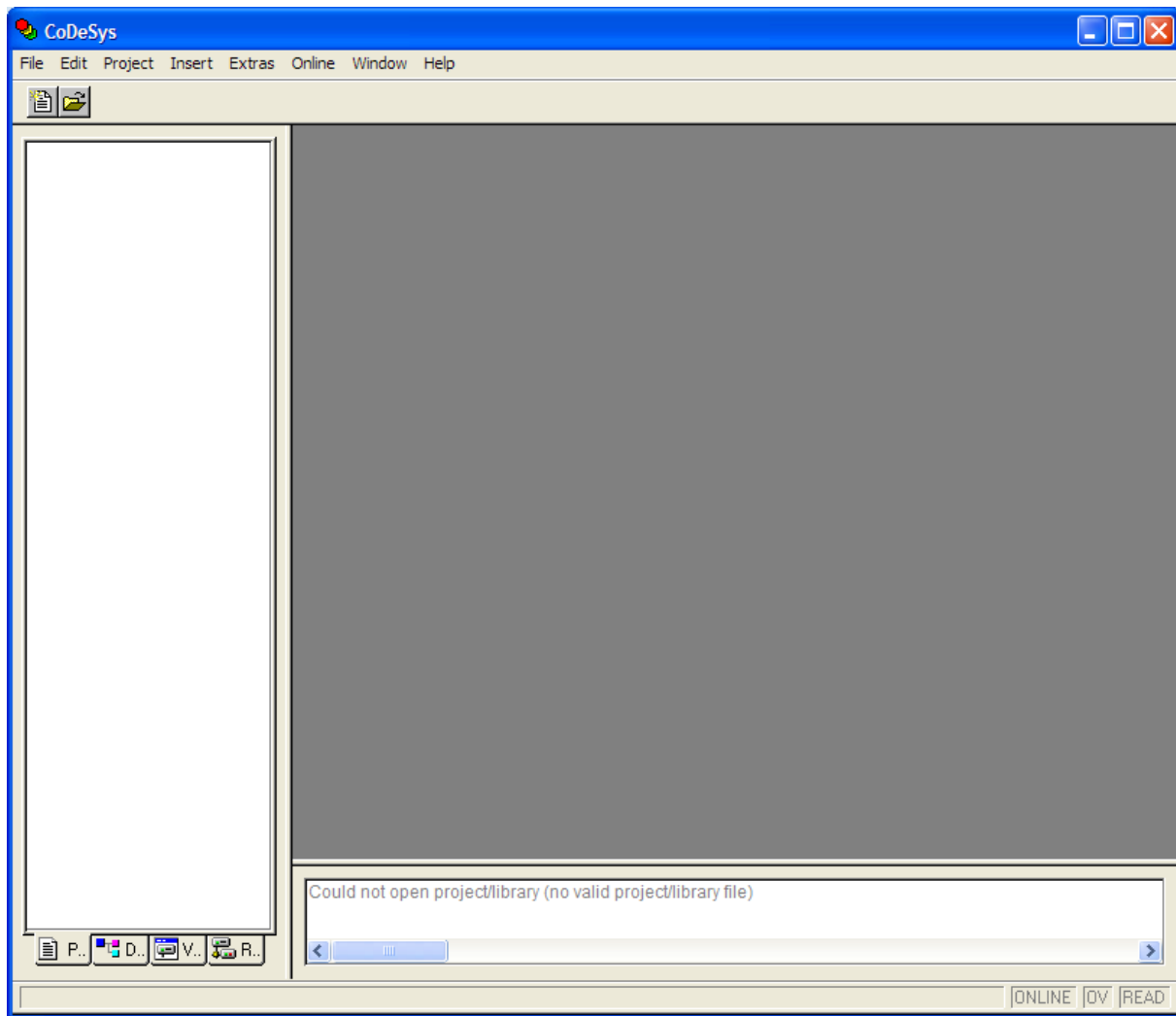


Since the CC Pilot™ XS has a screen it might be useful to add a visualization.

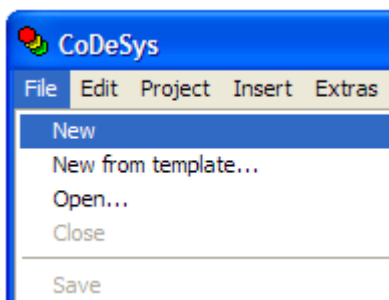
Create an application with visualization

To create a simple hello world GUI application with Visu, follow these steps:

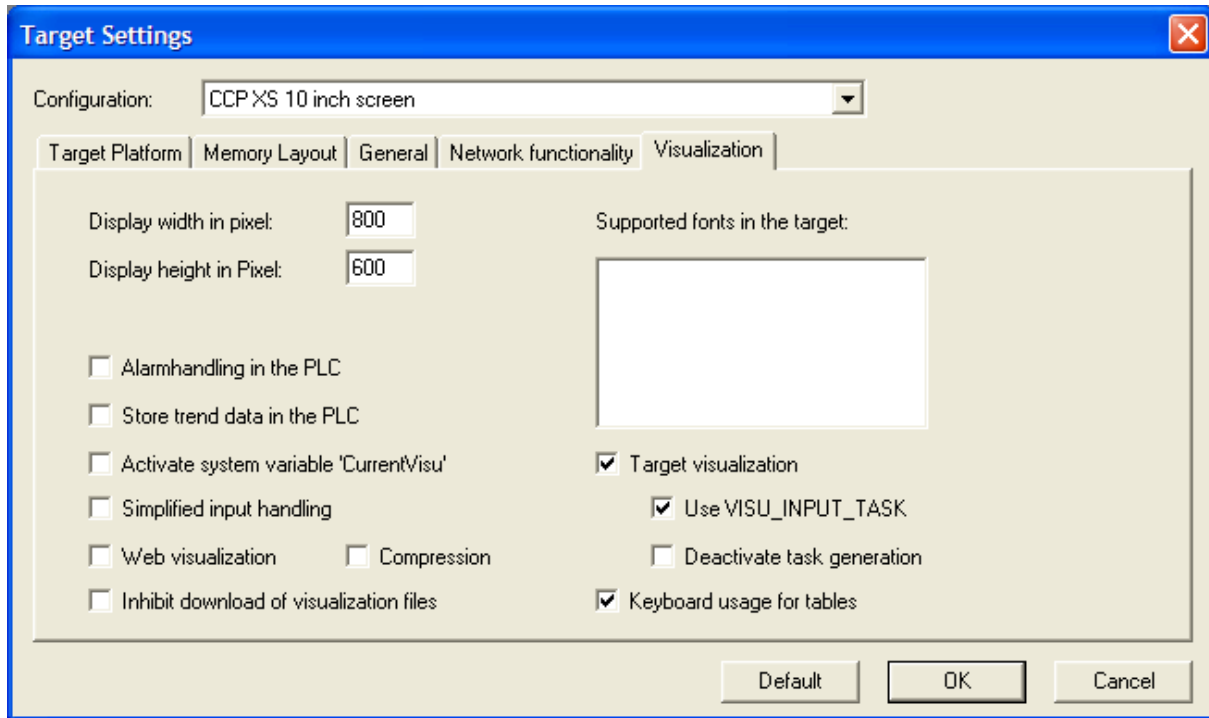
- Open the CoDeSys application development tool



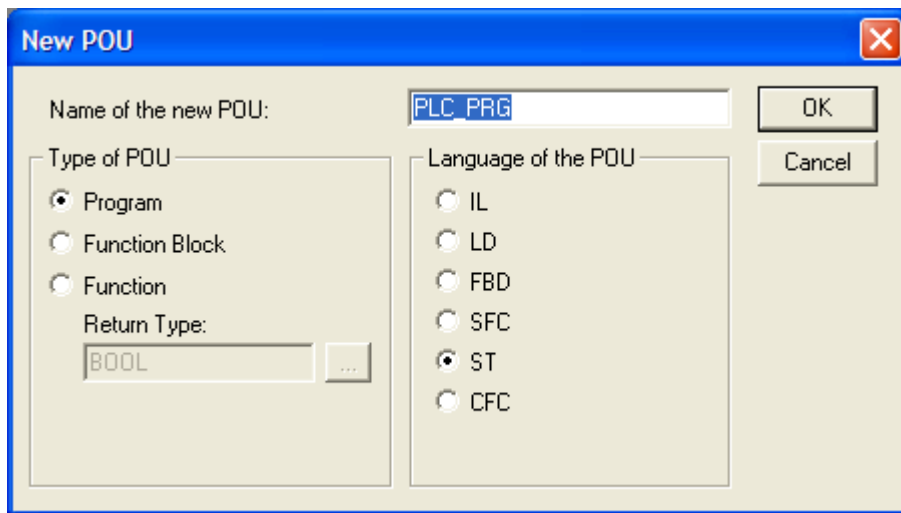
- Click File→New to create a new project



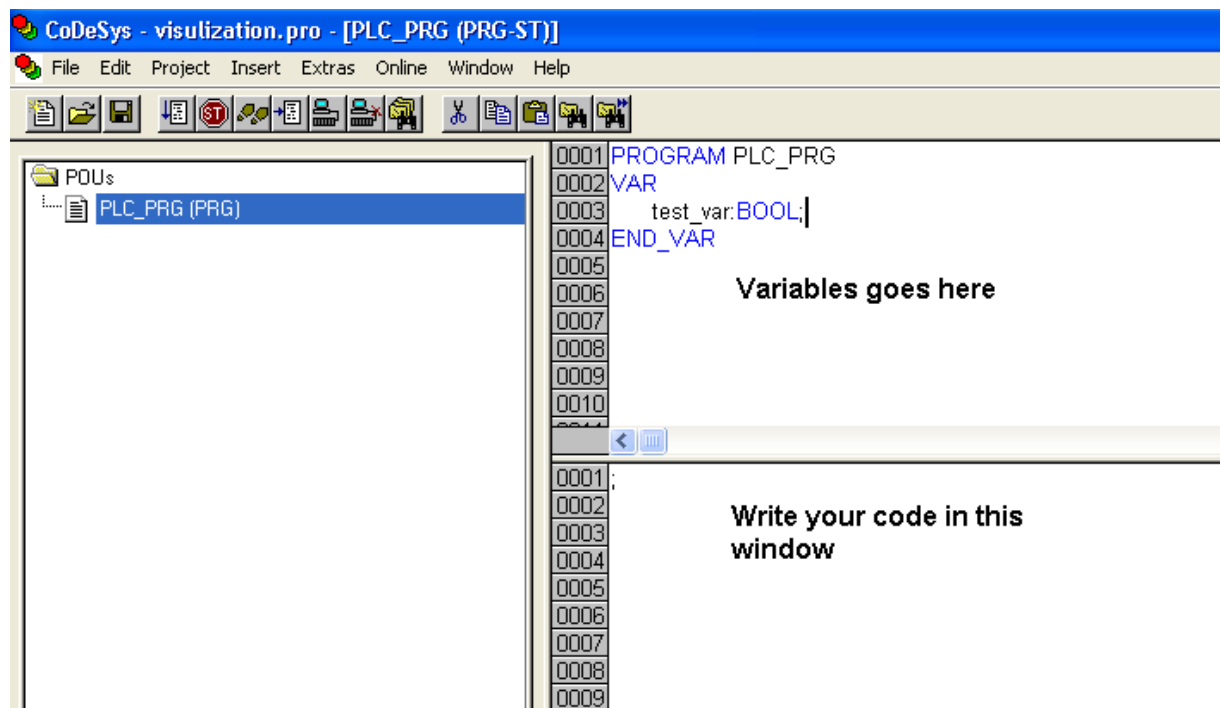
- In the target selection dialog, choose a target
- In the target settings dialog that expands, go to the visualization tab



- Check the 'Target visualization' box
- Check the 'Use VISU_INPUT_TASK' box
- Click ok
- In the following dialog click ok for adding a new POU with default settings

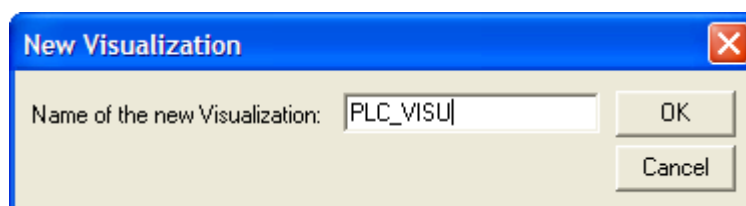


PLC_PRG is the main default program. There are two separate windows for every POU (Program Organization Unit), the top window is for variables and the bottom is for the code. Just type a semicolon in the code section if you want to go ahead and with other things and still be able to compile.



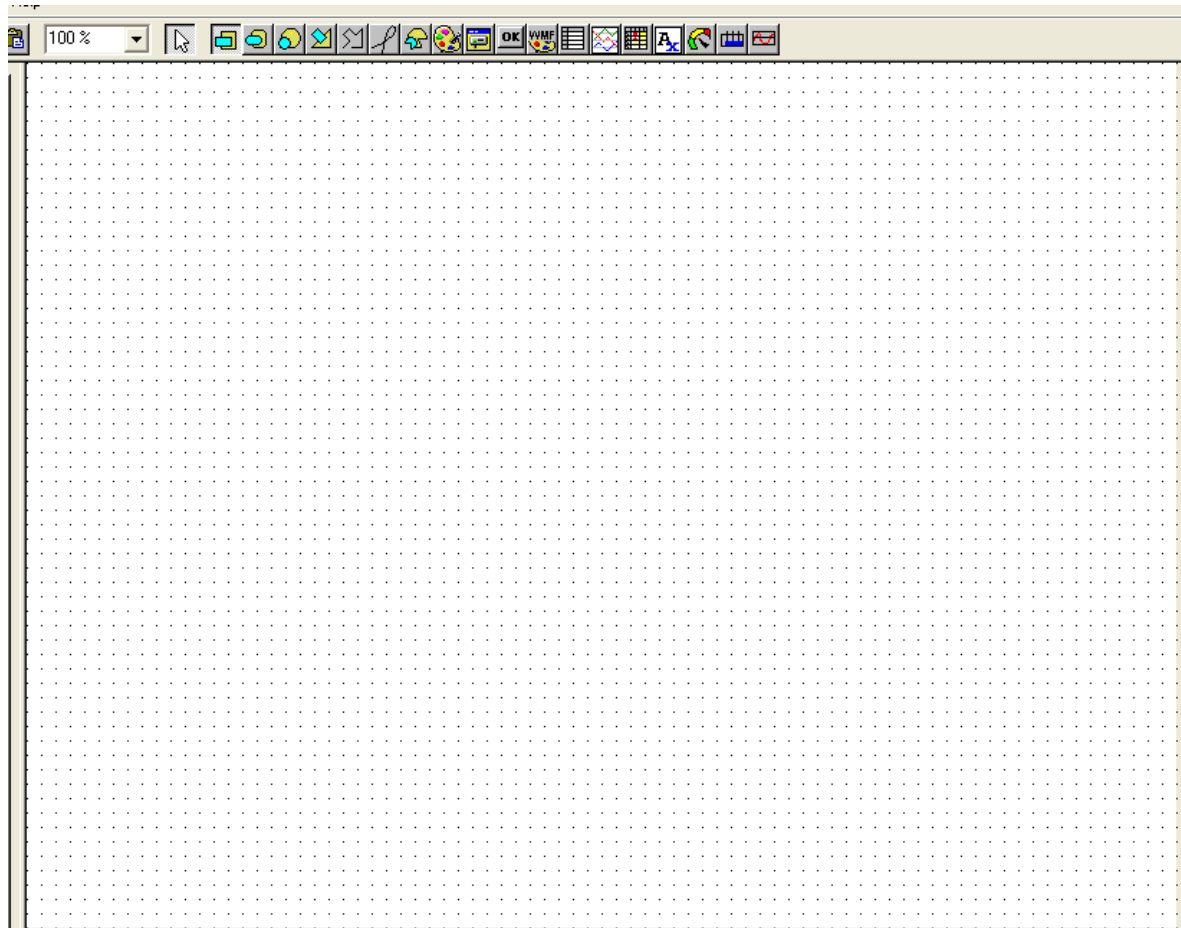
Since the CC Pilot™ XS has a screen it might be useful to add a visualization.

- Inside the visualization view, right click and select 'Add object'
- Name the new visualization 'PLC_VISU'



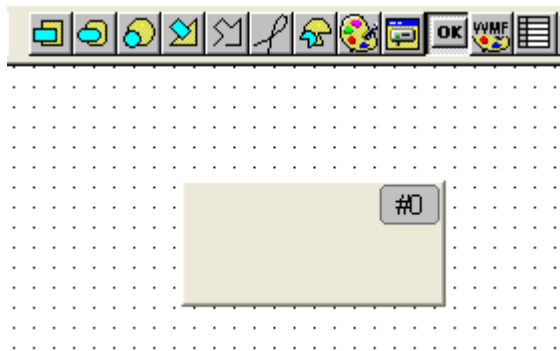
- Click Ok to create the visualization

The window that comes up gives you the possibility to design your user interface.

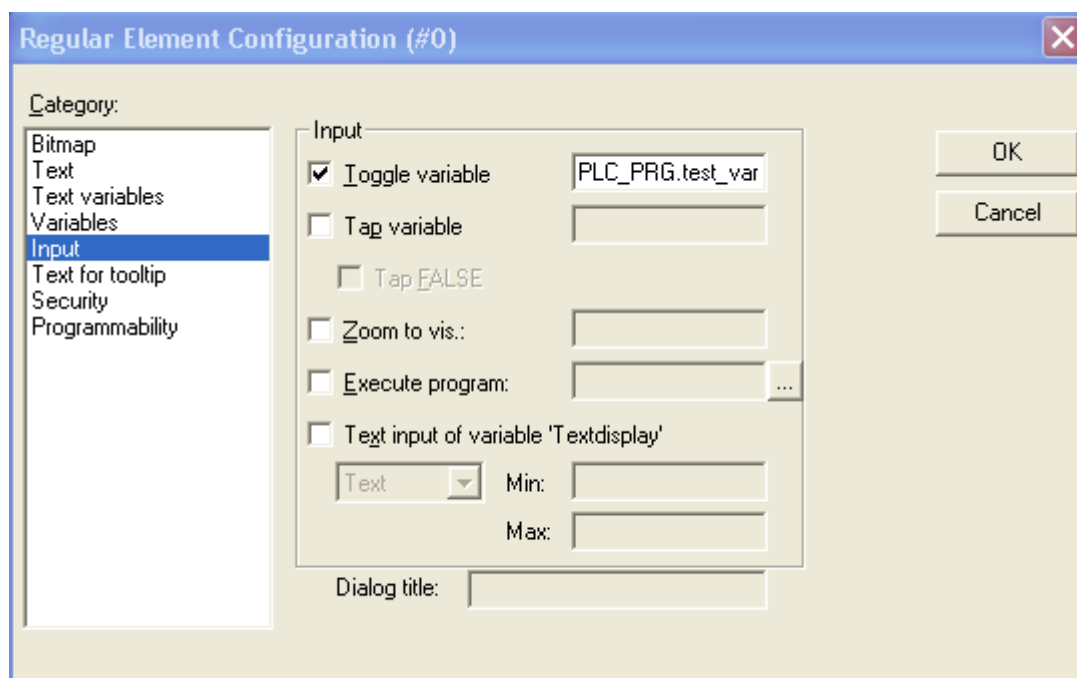


You can add buttons, alarms lists, bars and so on. Add a button for example as follows:

- Choose the button tool on the toolbar
- Click-drag on the screen to place and size the button

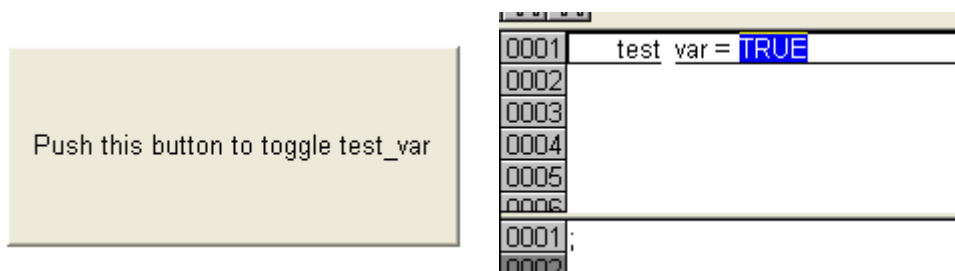


- Double click the button to bring up the configuration dialog
- For now just put in some text in the Content box under category Text
- Go to the Input category and mark the Toggle Variable checkbox
- In the input field write 'plc_prg.test_var' or press F2 and browse for the variable
- When finished click OK



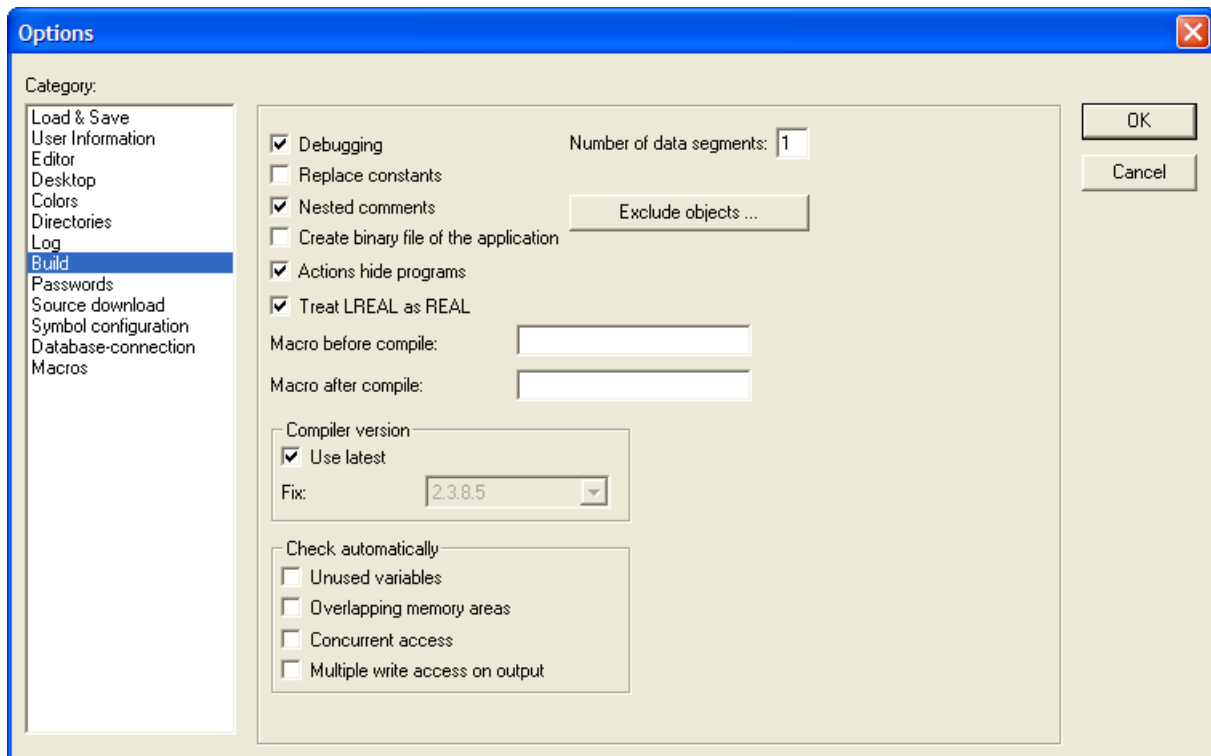
When running the application on the target, the visualization will show up both in CoDeSys on your development PC and on the CC Pilot™ XS. Pushing the button in the visualization will toggle **test_var**.

Variables such as **test_var** in this case can be monitored directly in the **PLC_PRG** view, for debugging. To view **PLC_PRG** go to the **POUs** tab and double click **PLC_PRG**.



There are some additional settings needed for the project to build. Make the following configurations:

- In the menu, click Project → Options
- Under 'Build' check the box, 'Treat LREAL as REAL'



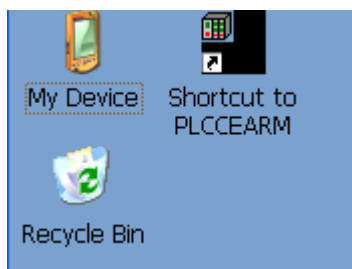
- Click ok

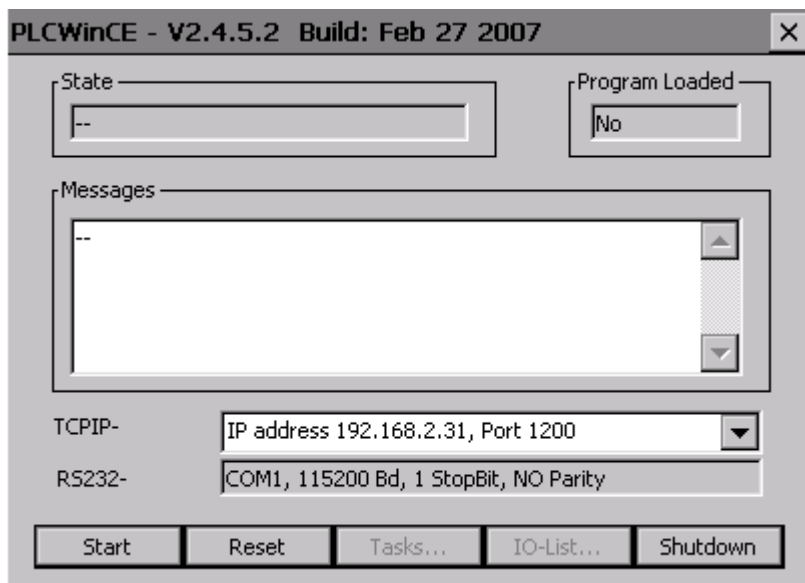
Your application is now ready to compile, build and download.

Download application to target

When your application is build you can download it to the target and run and debug the application. Follows these steps for downloading:

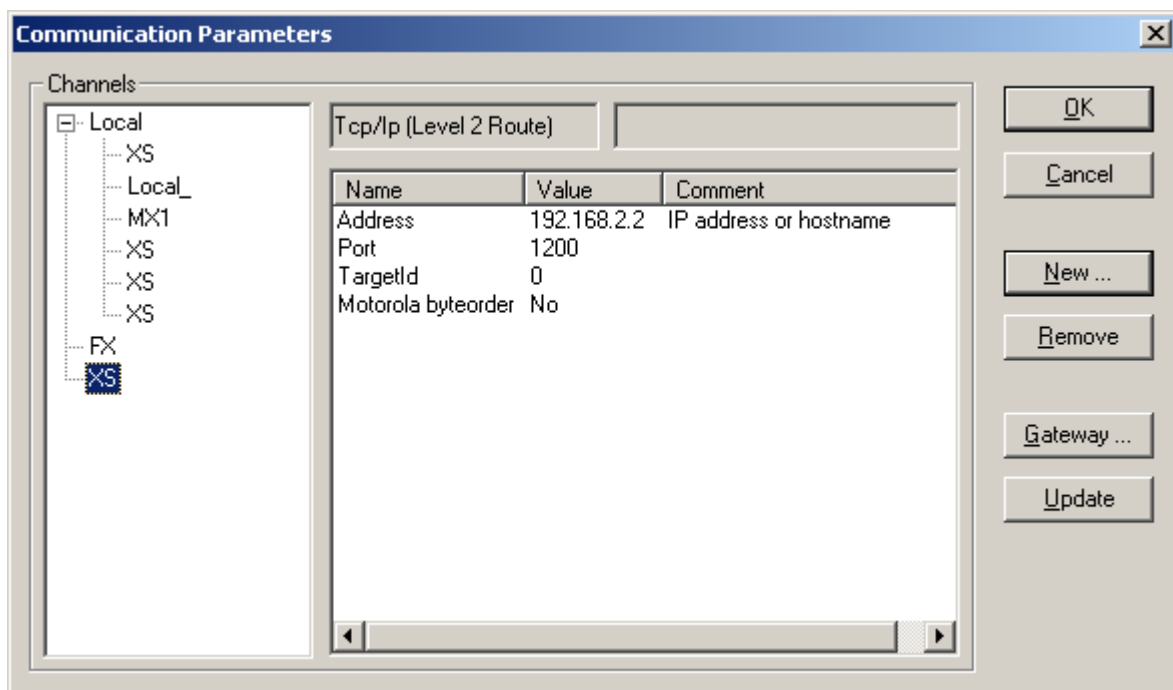
- Start the runtime by double clicking the executable file plccearm.exe or alternatively the desktop shortcut





The runtime control panel shows the communication parameters you need to know. To configure the connection in CoDeSys do the following:

- Press Online→Communication Parameters
- Press New
- Pick the communication method Tcp/Ip (Level 2 Route) and hit OK
- In the value column type the IP-address of your CC Pilot™ XS (e.g.192.168.2.2)



To download the project to the runtime choose Online→Login and accept to download the new program to the controller. Online→Run, F5 or Start on the runtime control pane will start the application and the visualization will show.

Save the project in non volatile memory

To get the unit to start preloaded with a CoDeSys application, the project must be set to save application in non volatile memory. To do so follow these steps:

- In the CoDeSys development environment, click Online→Create boot project

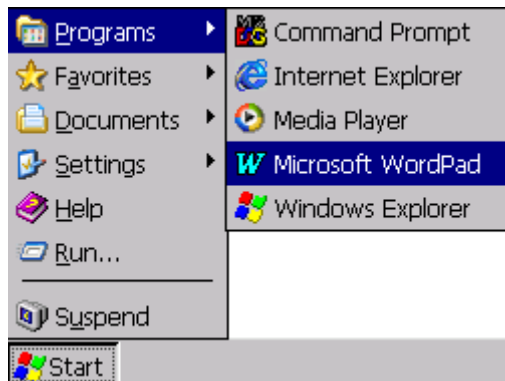
Now the project will be stored on the target even if the power is turned off. The next time the runtime starts the project application will load and run automatically.

Only load current bitmaps in Visu

The target visualization buffer in the CoDeSys runtime is about 5.5 MB in size. This means the visualization cannot contain anything that goes beyond the limits of this buffer (number of visualization function calls per cycle etc.). Jpeg images used in an application are extracted into BMP images in memory and then drawn as bitmaps, so this operation need more heap memory then the impression of the jpg-file size. To overcome this heap memory problem, the runtime can be configured to only load currently used images into memory.

To do this, the configuration file for the CoDeSys runtime called rts3s.cfg needs to be modified as follows:

- On the CC Pilot XS, open Microsoft WordPad by clicking Start→Programs→Microsoft WordPad



- Select and open the file \Program Files\SoftPLC\rts3s.cfg
- Under the [VISUALIZATION] section, add the following string:

TargetVisuLoadOnlyCurrentUsedBitmaps=Yes

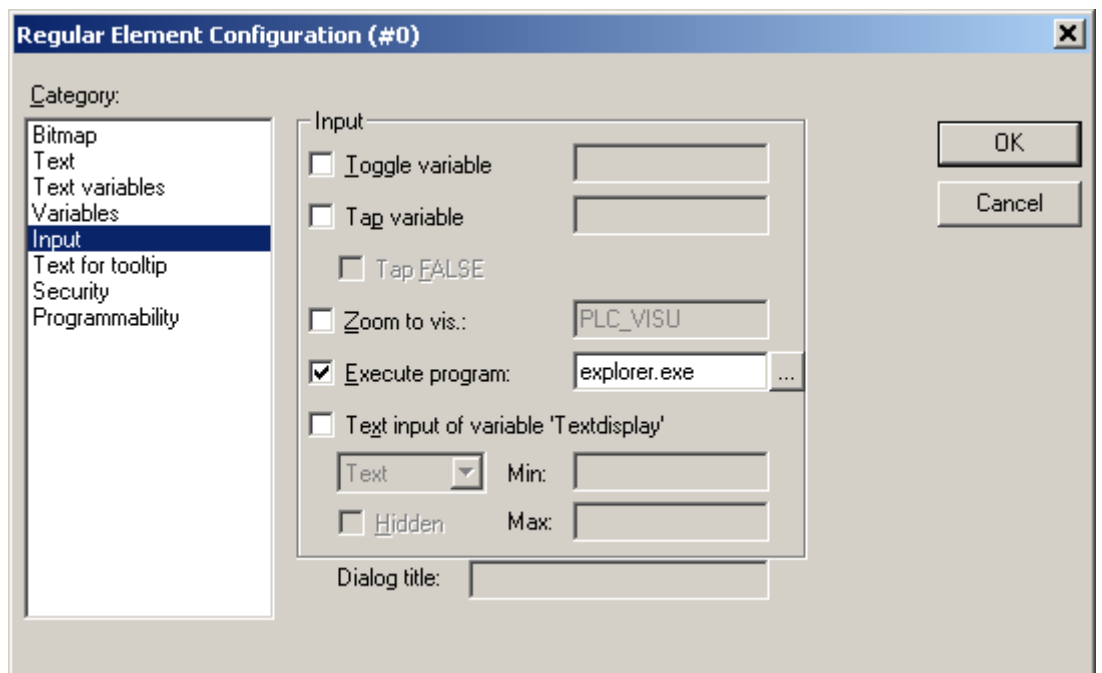
When this string is used, only the bitmap that is currently shown is loaded into the memory. Using this feature can make the time of switching bitmaps a little slower, but for applications with many bitmap pages, this is necessary.

Start programs from CoDeSys

This chapter describes how to start a program from a button click.

- Add a button or a shape to your visualization
- Double click on the visualization item you just created
- In the input view; Check the "Execute program:" box and enter the program call, for example explorer.exe

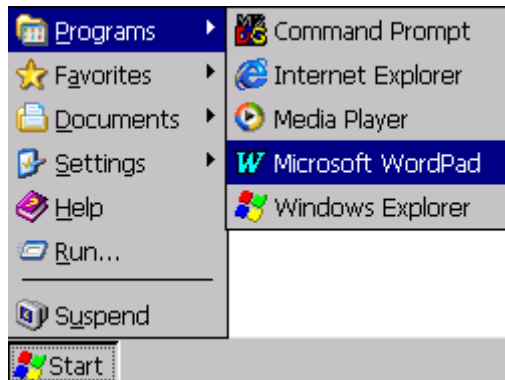
The dialog should look something like this:



Start PLCWinCE minimized

The PLC runtime application can be started minimized, in which case the graphical interface will be hidden. To activate this function the settings file rts3s.cfg has to be modified as follows:

- On the CC Pilot XS, open Microsoft WordPad by clicking Start→Programs→Microsoft WordPad

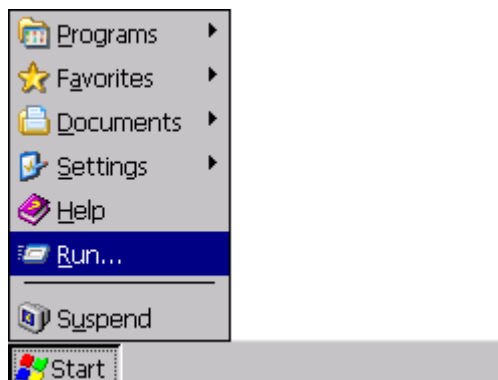


- Select and open the file \Program Files\SoftPLC\rts3s.cfg
- Under the [PLC] section, edit the string 'StartMinimized=No' to 'StartMinimized=Yes' to activate the function
- Save the settings file and restart the PLC application

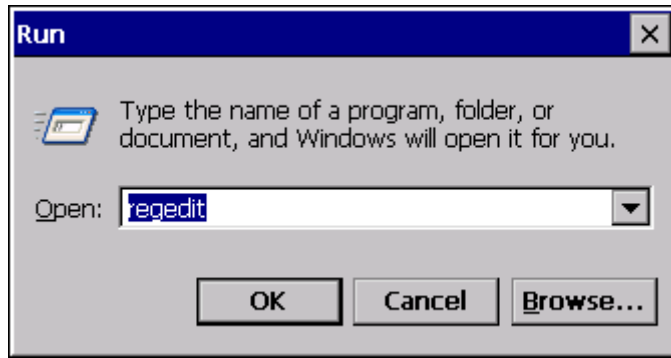
Autostart the PLC runtime

The PLC runtime application can be set to automatically start when the device is powered on. This feature is activated by modifying the registry as follows. Next time the device is started, the PLC runtime will start automatically.

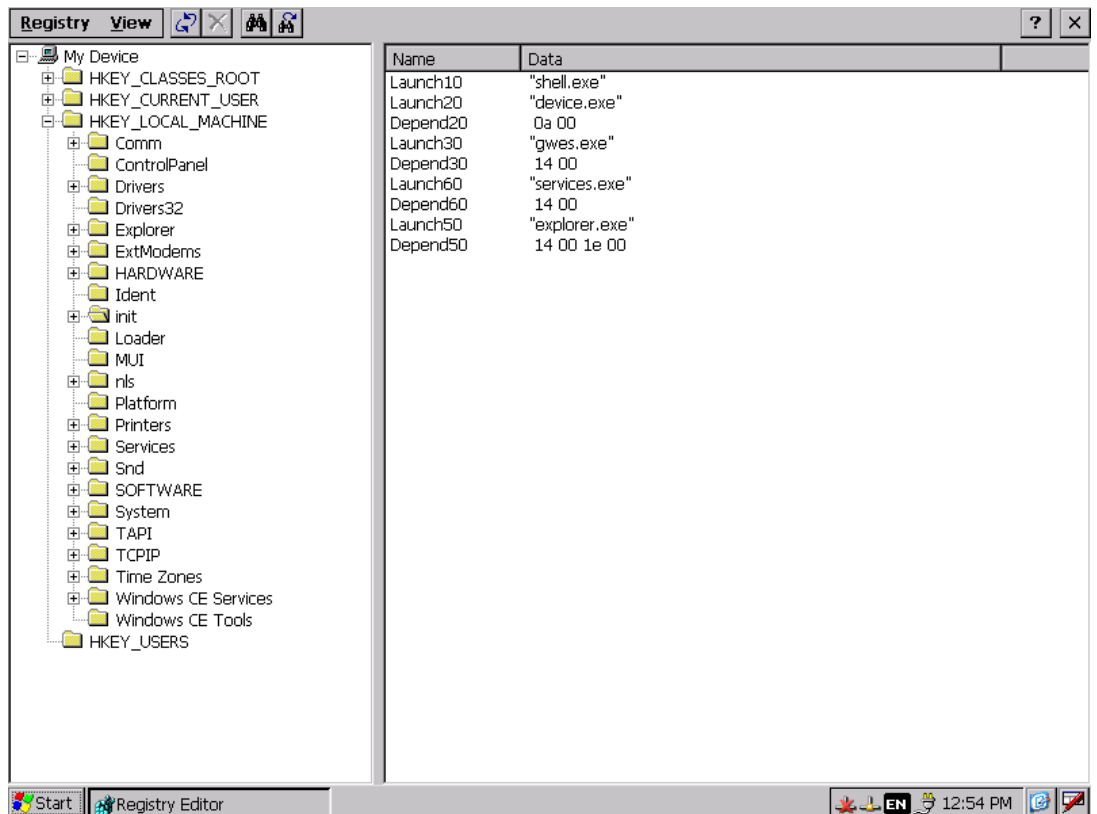
- On the CC Pilot XS, open the Registry editor by clicking Start→Run



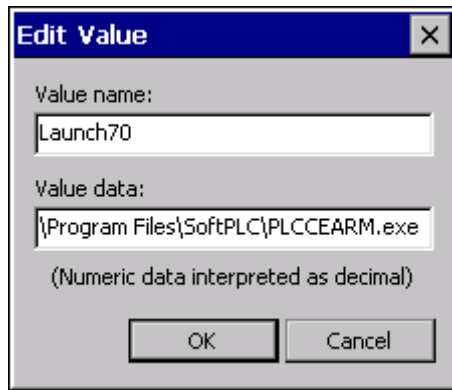
and enter 'regedit' in the Open: field



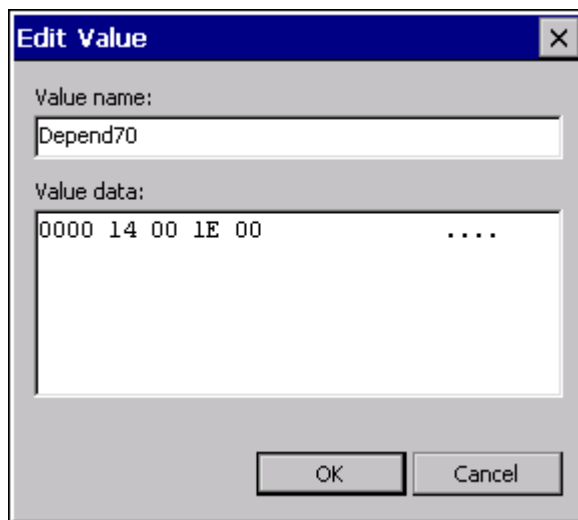
- In the registry editor, expand HKEY_LOCAL_MACHINE\init



- Add a new string from the menu, by clicking Registry→New→String value
- Set the value name and value data as follows:



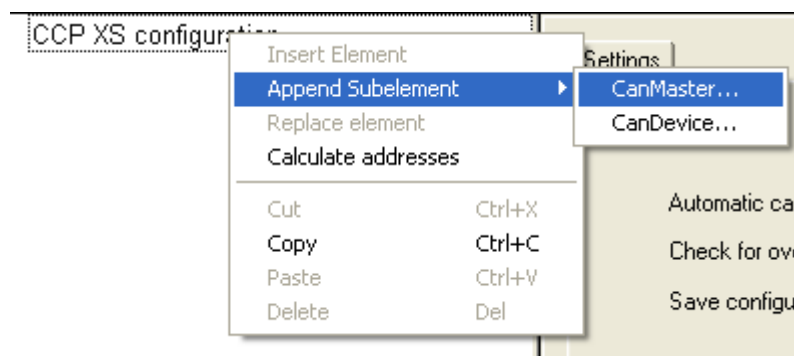
- Click ok
- Add a new string from the menu, by clicking Registry→New→Binary value
- Set the value name and value data as follows:



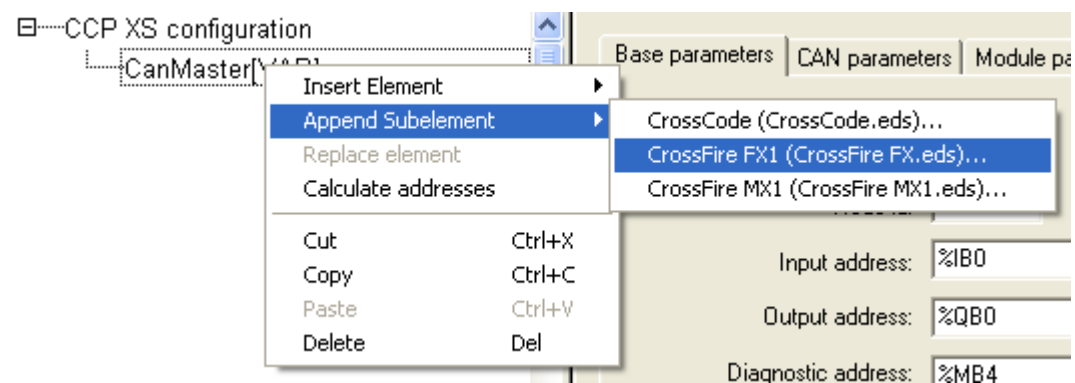
- Click ok
- Close the Registry editor
- Turn off the device by pressing the suspend button

Set up a CAN network

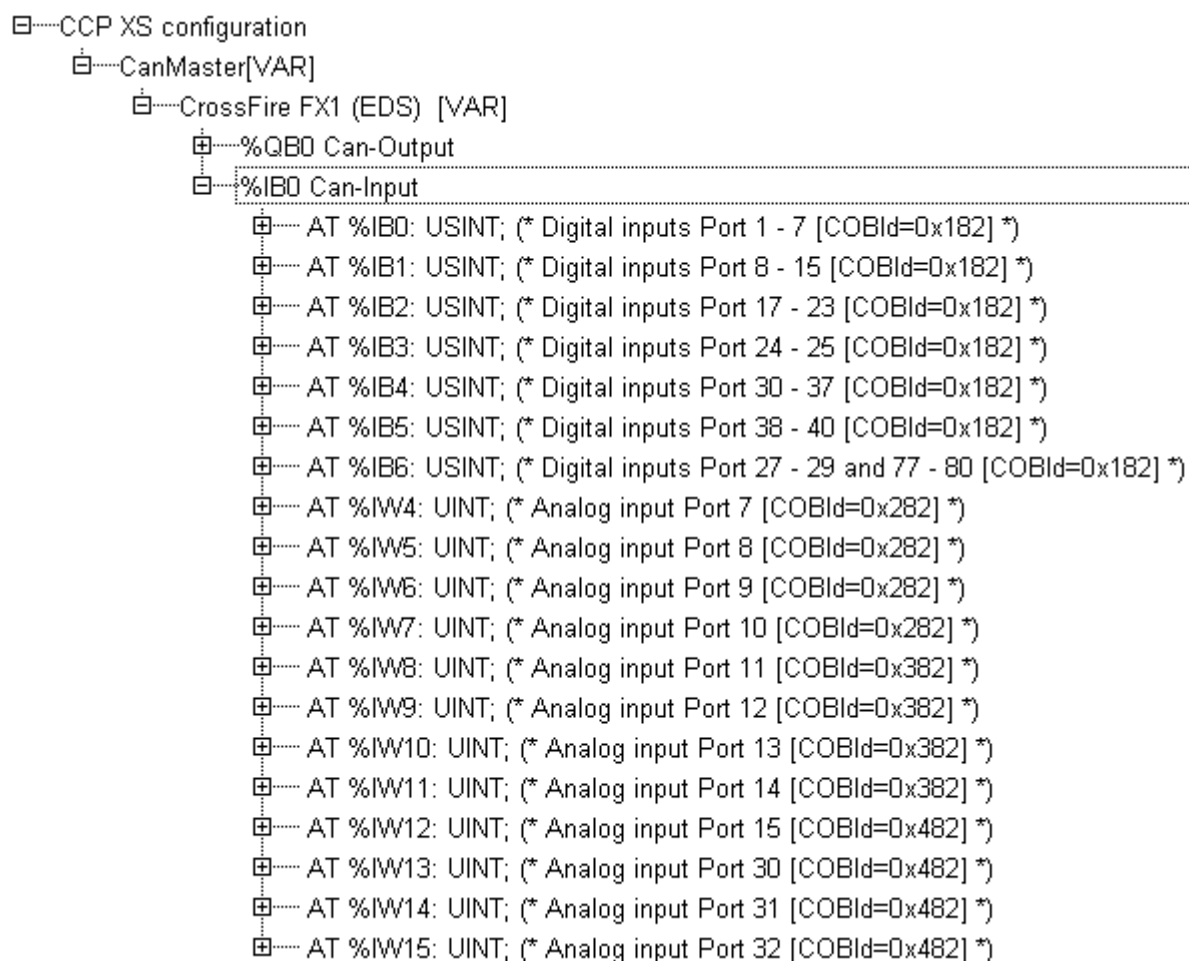
To set up a CAN network with a CC Pilot™ XS as a master and e.g. a CrossFire FX1 as a slave, start a new project choosing the CCP XS as a target. Under Resources → PLC Configuration, right click CCP XS configuration and choose 'Append Subelement' → CanMaster.



Then to add a slave, right click the master and choose 'Append Subelement' → 'CrossFire FX1' or the slave of your choice.



This will load the eds file for the CrossFire FX1 and its I/O will be available to the master and the application.

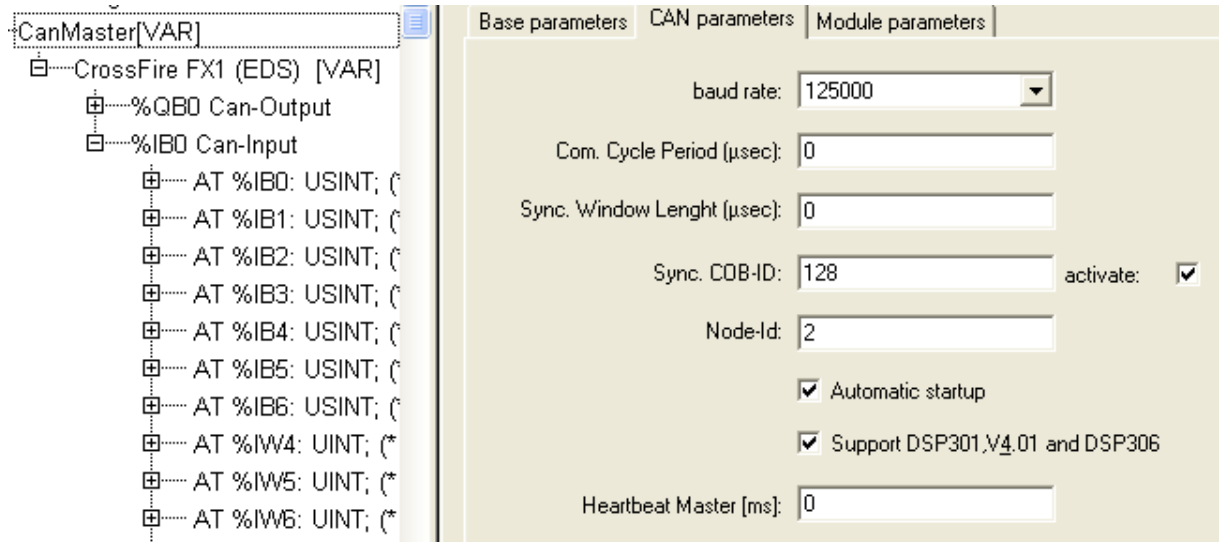


For example reading an analog input of the slave in the master software can be done by giving a variable the address of this particular I/O:

```
PROGRAM PLC_PRG
VAR
  analogIn1 AT %IW4:WORD;
END_VAR
```

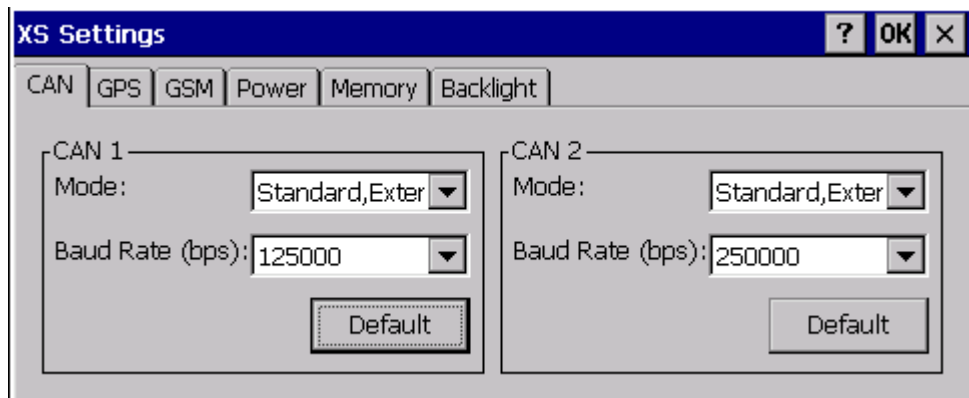
Configure CAN settings

For the CAN communication to work, the settings for CanMaster under PLC Configuration in the CoDeSys development environment, must match the CAN settings for the target, in this case the CC Pilot™ XS.



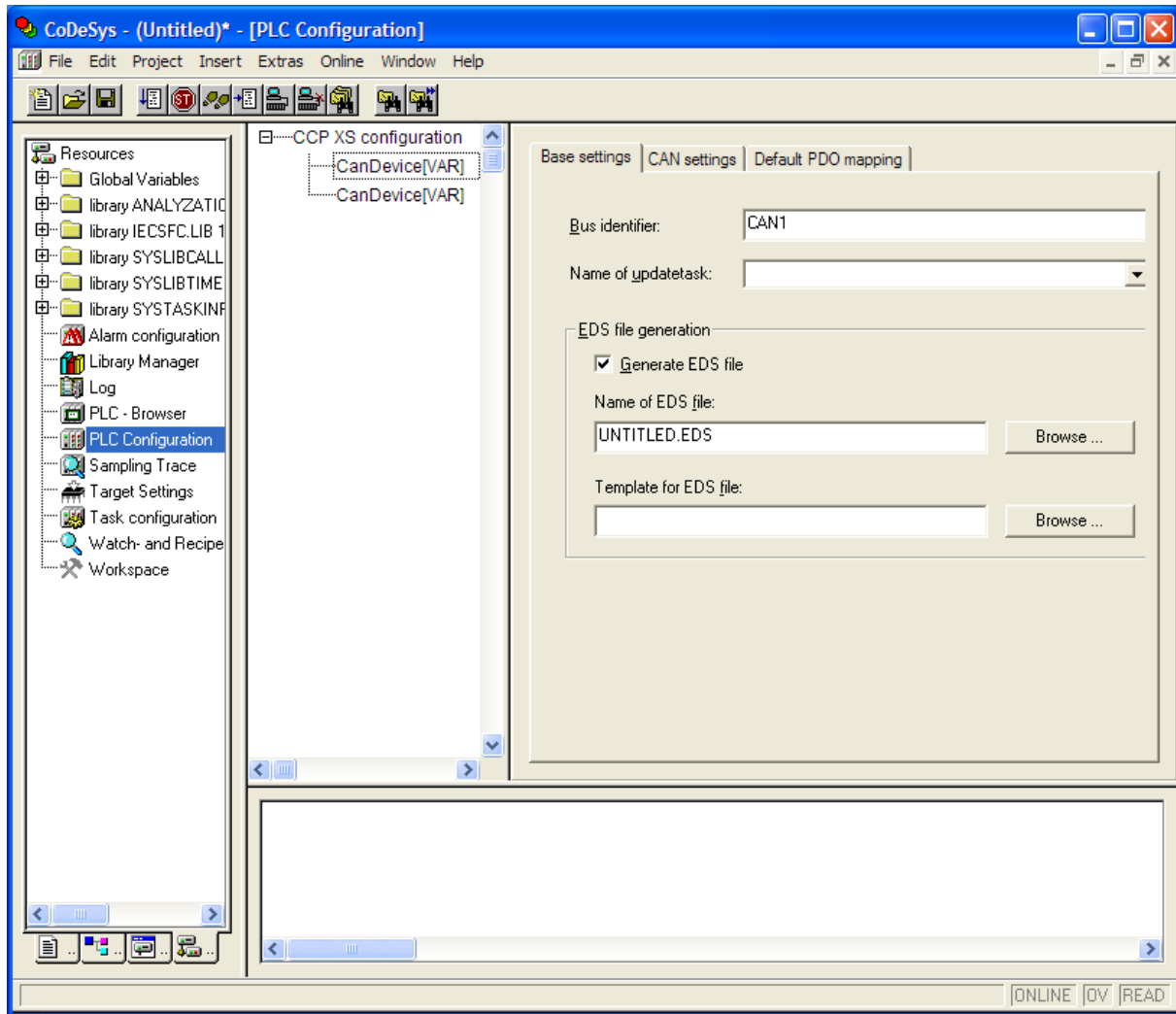
The settings for the CC Pilot™ XS can be found under Start → Settings → Control Panel → 'XS Settings'.

- Mode must be set to 'Standard, Extended' to be compatible with CANopen in CoDeSys
- Baud rate is set to match the value set in the CoDeSys development environment, in this example 125 kbps for CAN1 and 250 kbps for CAN2



Set bus identifier

In the PLC configuration dialog in the CoDeSys development environment, there is a parameter called 'Bus identifier'. This parameter is currently not used. Instead the order of the CAN devices sets the specific bus identifier. The first device in the list uses CAN1 and the second uses CAN2.



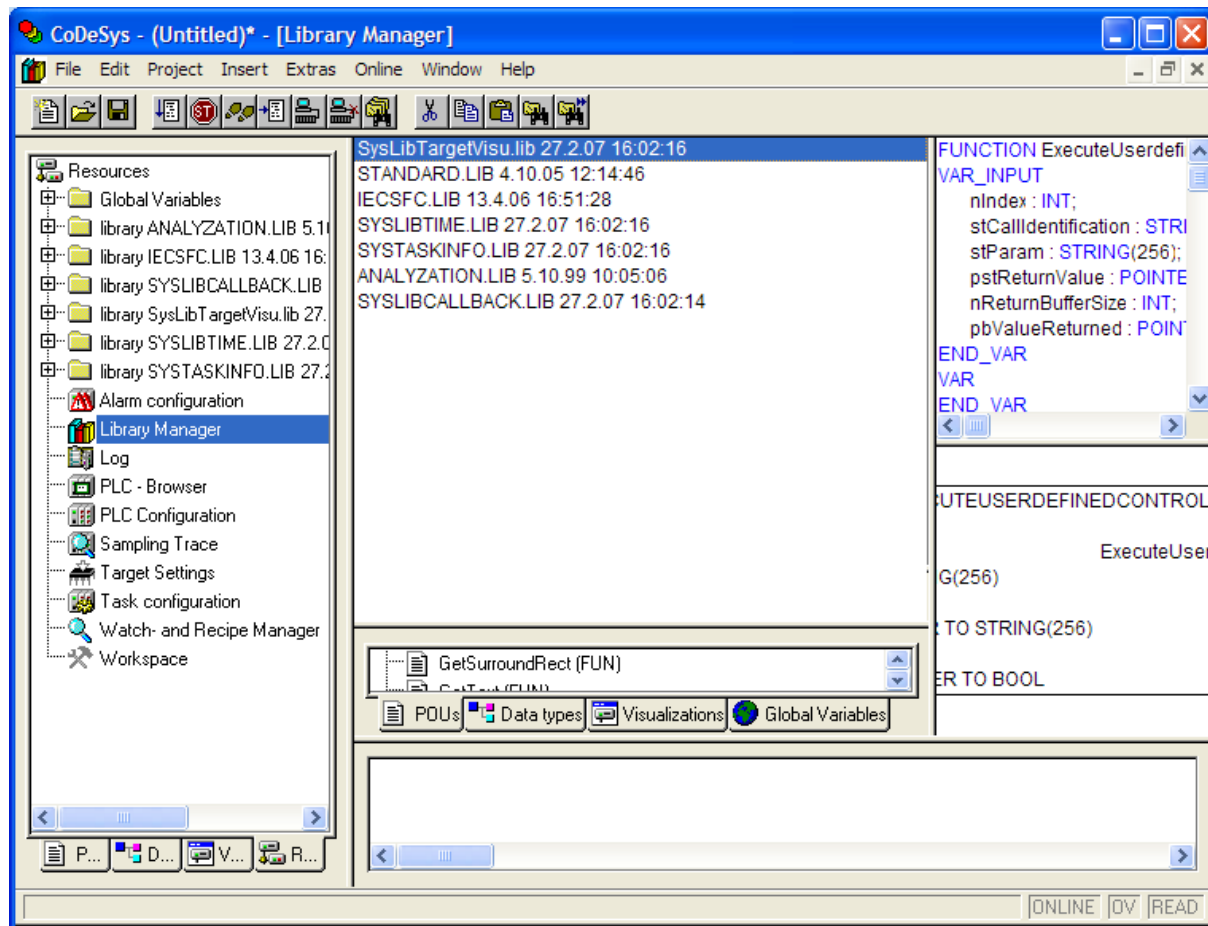
- To use only CAN1, add only one CAN device
- To use both CAN1 and CAN2, add two CAN devices. The first in the list will be assigned to CAN1 and the second in the list will be assigned to CAN2
- To use only CAN2, add two CAN devices and use only the second in the list, which is assigned to CAN2. The first in the list will serve as a dummy

Use retain variables

Using retain variables is a way of saving parameters to non volatile memory. You have to point out a file where the declared retain variables are saved to and restored from on command. Before you can use the retain variables function, you have to make sure the specific lib is included in your project.

First add the library:

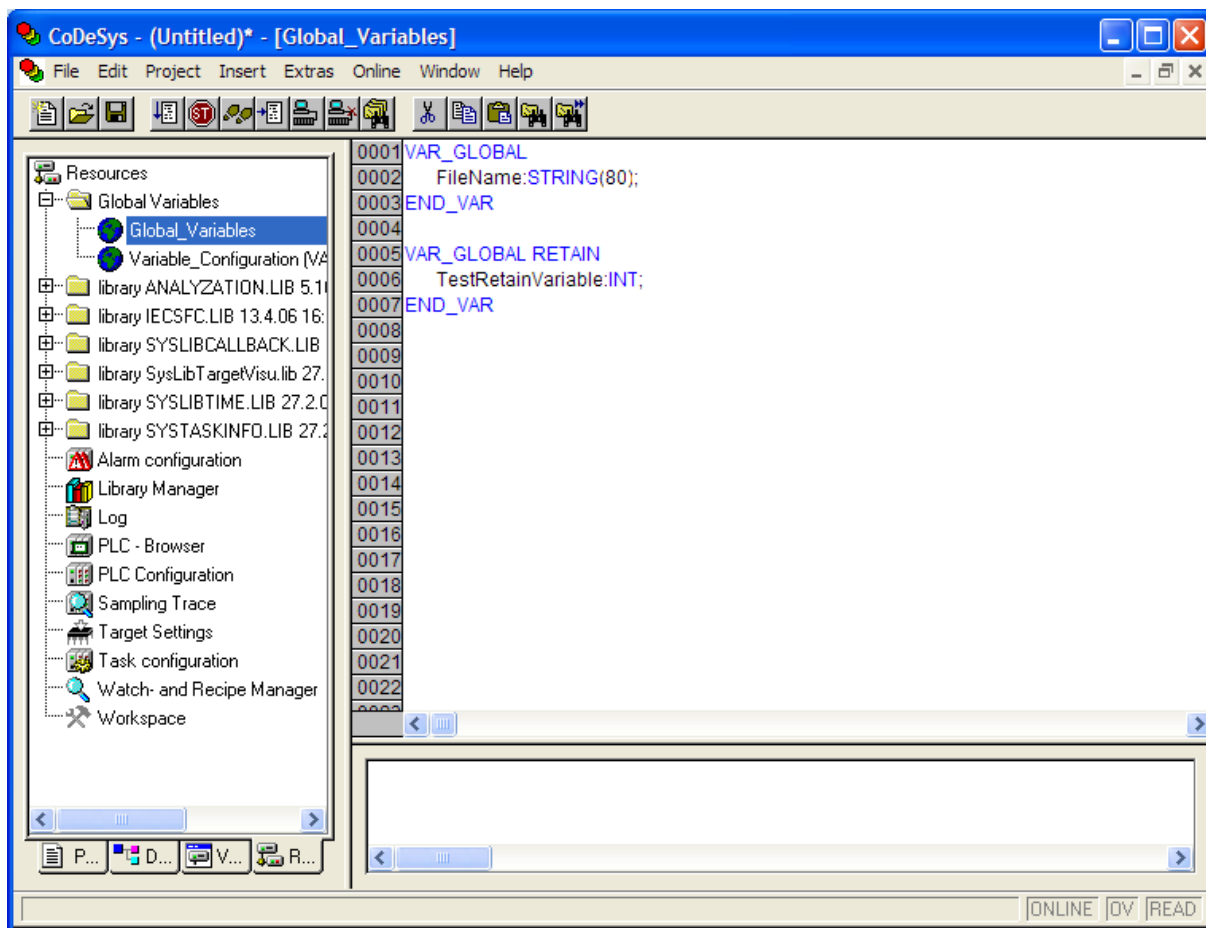
- Open the library manager in your project, by double clicking the icon in the resource view



- In the view with the libraries are listed, right click and select 'Additional library'
- Point out and select the library 'SysLibPlcCtrl.lib'

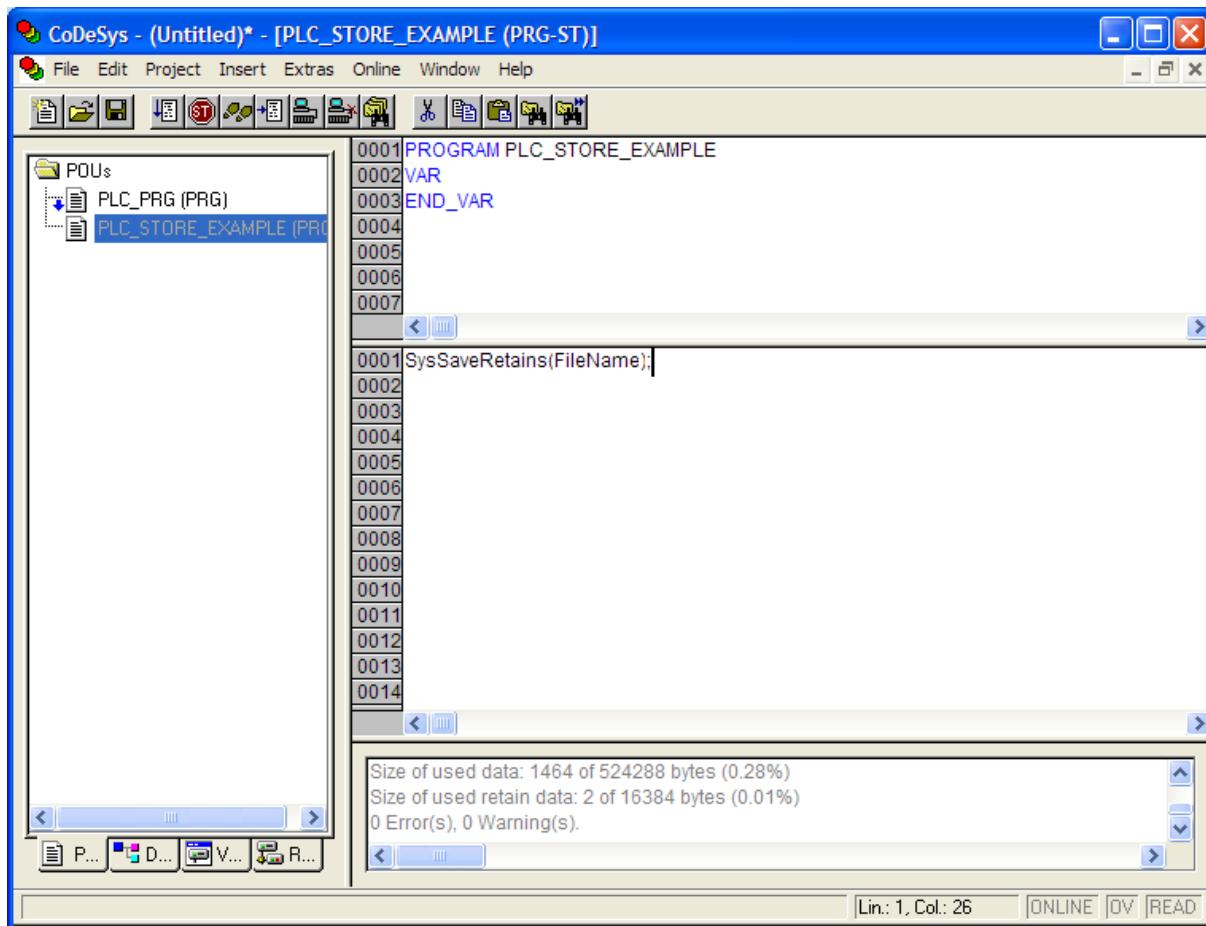
Now you have added the correct library for using retain variables and are ready to declare some retain variables to use.

- From the resource view, open the 'global variables' window
- Declare your retain variables under a new field called VAR_GLOBAL_RETAIN
- (A good thing is to also declare a global variable for the filename to use for the retain store/restore routine)

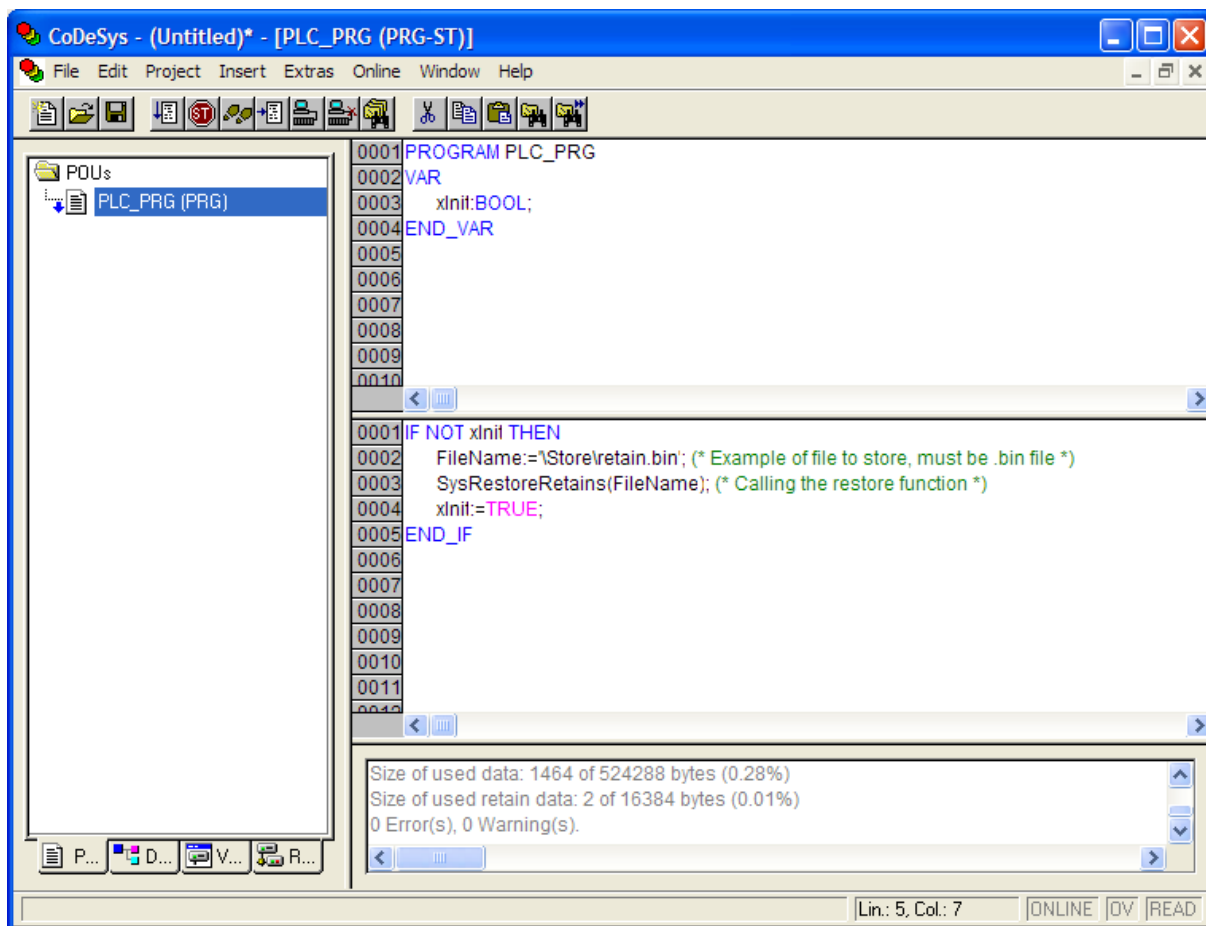


Now call the Save and Restore functions, preferably on shutdown or periodically and on startup respectively.

- Call the SysSaveRetains function to save the declared variables



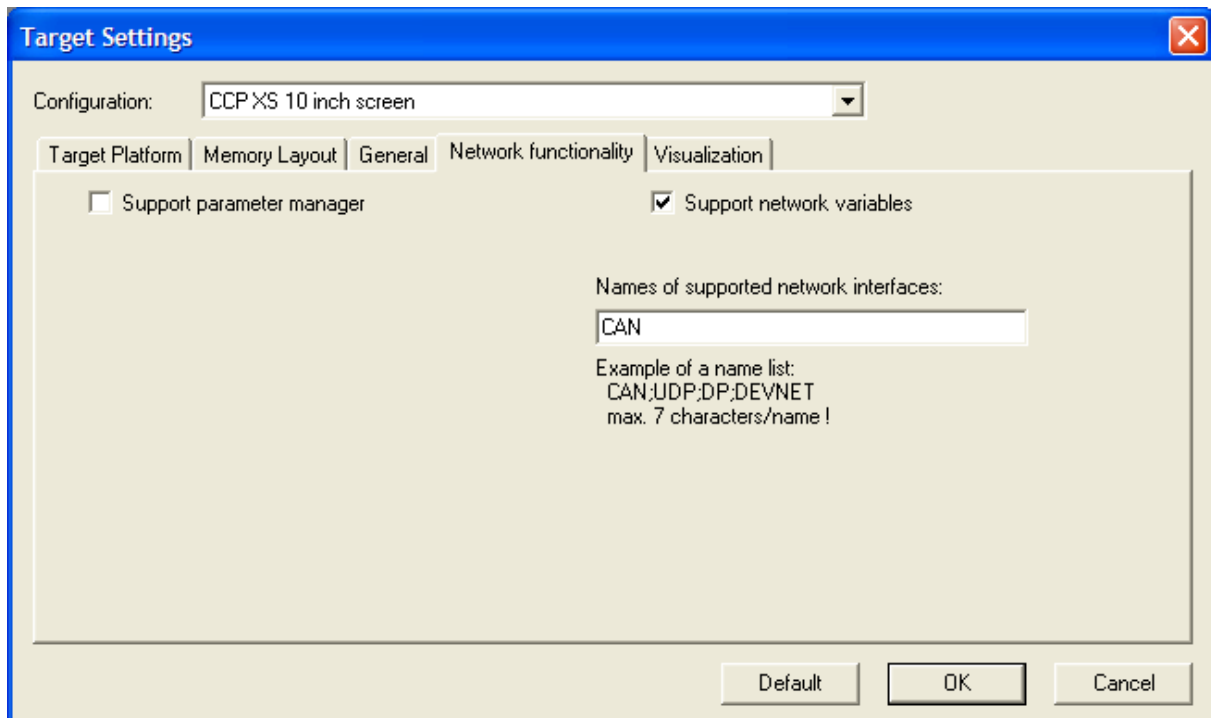
- Call the SysRestoreRetains to restore the declared variables



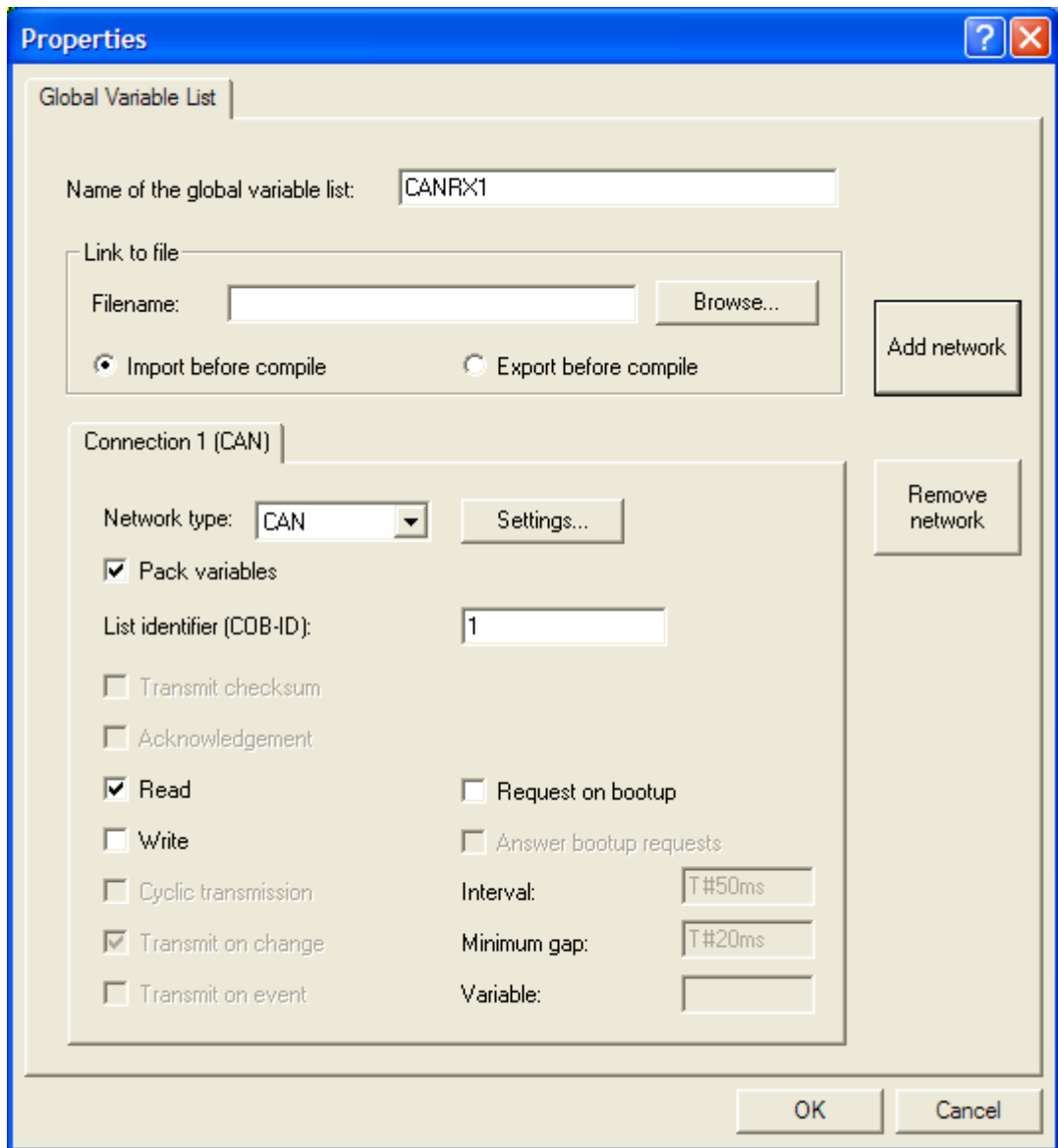
Use network variables

To use network variables in your project, follow these steps:

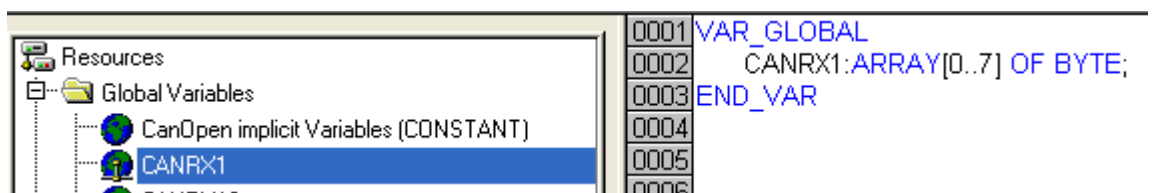
- When starting a new project, check the 'support network variables' box, under Network functionality in the Target Settings dialog. Enter 'CAN' in 'supported interfaces'



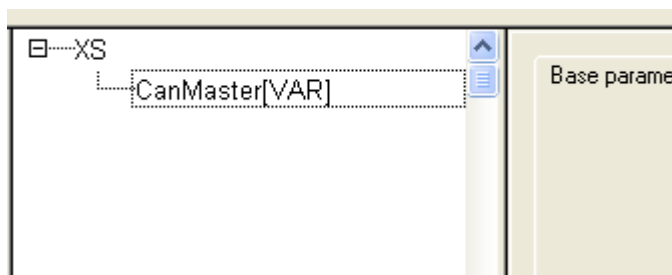
- When project is created, go to the resources view, right-click on GlobalVariables and choose 'Add Object'
- Name the variable, click 'Add network' and specify the COB-ID



- In the new global variable list for CANRX1 define an array of bytes to hold the data in the CAN-message corresponding to the COB-ID just specified

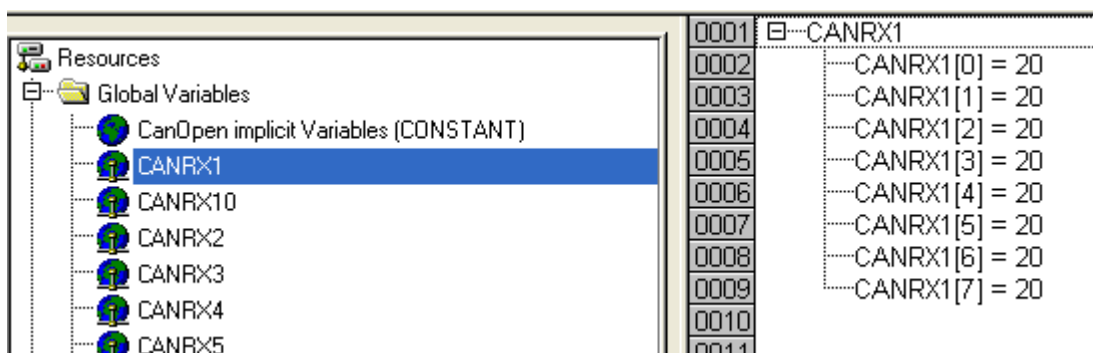


- Go to PLC Configuration and add a CANopen master, this loads the required libraries

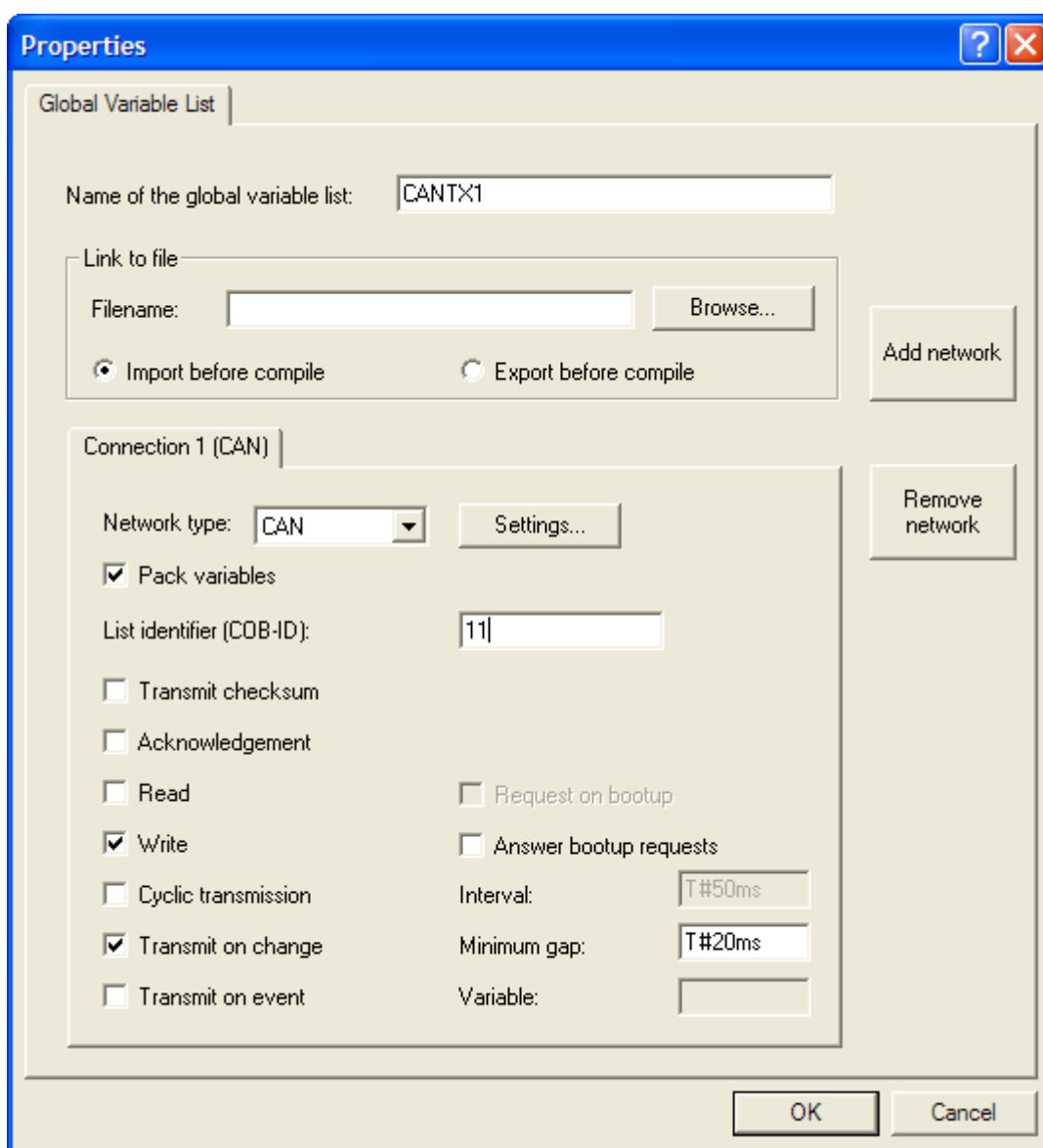


- Go on and add new network variables for every desired COB-ID.
- To call the CANopen libraries that handles the network traffic at least one of the network variables has to be read or written in a task that is scheduled, e.g. PLC_PRG

When a CAN-message with a COB-ID corresponding to a variable list specified for reading are sent on the bus, the variables in that list are updated.



To specify network variables for transmission, add a network variable object, check write instead of read and specify the transmission settings, and e.g. transmit on change.



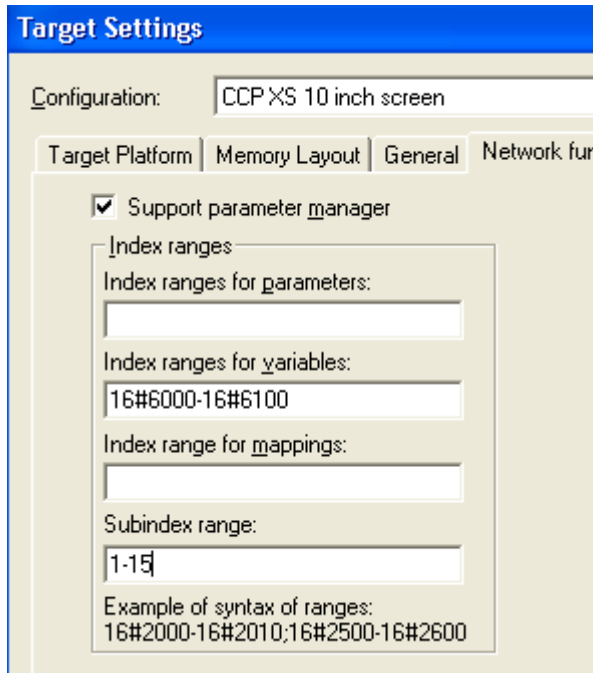
When the network variables are written the values are sent on the bus.

Example in CoDeSys:

```
FOR i:=0 TO 7 BY 1 DO
    CANTX1[i]:=send;
END_FOR;
```

Configure a CanDevice (CANopen Slave)

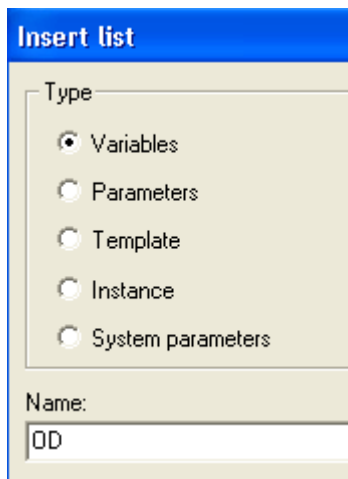
- Start a new project choosing 'CCP XS 10 inch screen' as target
- In the Target Settings dialog under 'Network functionality', check 'Support parameter manager'
- In 'Index ranges for variables' and 'Subindex range' choose a desired range, e.g.:



- When done hit OK, and hit OK again to accept the default POU, PLC_PRG
- Declare the following variables in the global variable list:

```
in:WORD;
out:WORD;
```

- Under resources, double-click Parameter Manager and right-click in the left pane that appears to add a new list. Name it something e.g. "OD" (for object dictionary)



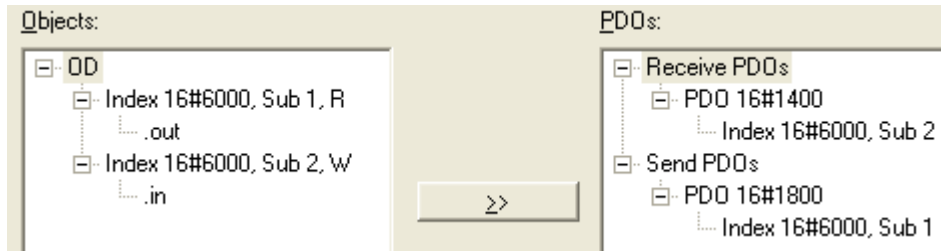
- Select the new list and in the right pane, right-click and insert two new lines

The new lines should look something like this:

	Name	Index	SubIndex	Accesslevel	Accessright	Variable
OD	in	16#6000	16#2	low	write only	.in
	out	16#6000	16#1	low	read only	.out

- Under PLC Configuration right-click on CCP XS configuration and choose 'Append Subelement' → CanDevice

- Under Default PDO mapping, click 'Insert send PDO'. Select the new PDO and the variable out and click >> to map the variable to that PDO. In the same way insert a receive PDO and map the variable in



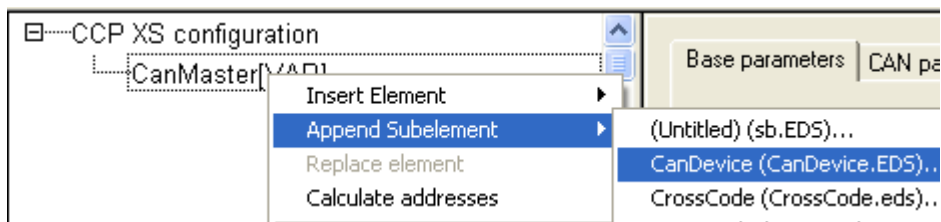
- Add three libraries into the project to make the CAN communication work by going to Window → Library Manager, right click and add:
 - 3S_CanOpenDevice.lib

These libraries will be added automatically:

- 3S_CanDrv.lib
- 3S_CanOpenManager.lib

- Compile and save the project

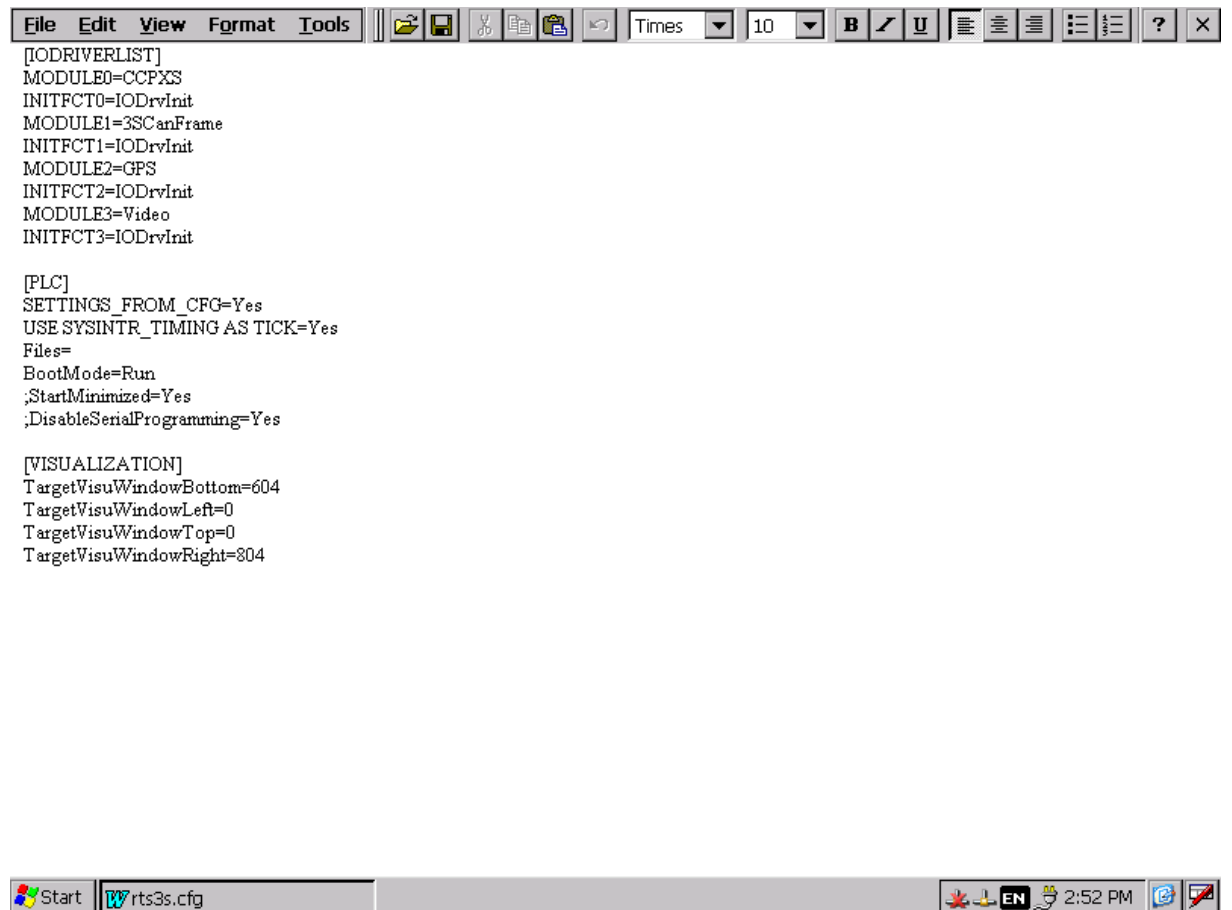
Under 'Base settings' → 'Name of EDS file' is the path to and name of the generated eds-file which describes this new node. This file can be used in a master to communicate with the node.



Configure CC Pilot XS CoDeSys RunTime settings

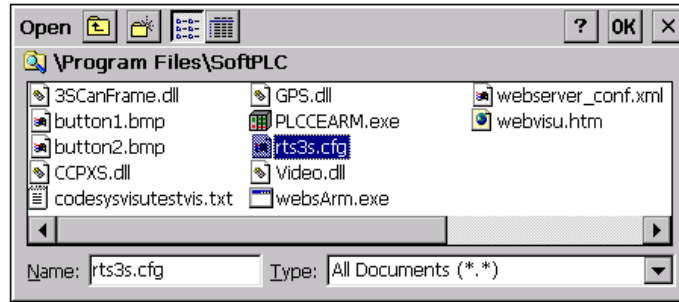
By the configuration settings it is possible to tweak the CoDeSys RunTime in certain ways. The settings described in the following is read by the runtime system kernel from the configuration file 'rts3s.cfg' which is located in the same path as the runtime system executable, i.e. \Program Files\SoftPLC\rts3s.cfg.

The rts3s.cfg file could look like this:



The “rts3s.cfg” is edited by the following steps:

- Open Microsoft Wordpad by pressing Start → Programs → ‘Microsoft Wordpad’
- Open the file \Program Files\SoftPLC\rts3s.cfg



- Edit and save the file
- Restart the computer

For information about the consequence of each sub-key, please read the document CoDeSys SP 32bit full_V.....pdf



Observe, the manual includes sub-keys that aren't supported in the CC Pilot XS and should not be used.

The following sub-keys are supported in the CC Pilot™ XS. The sub-keys are described as examples with correct section and an example value in red color, which can be altered. If the sub-keys should be inserted in the configuration file, then insert under correct section.

File path

```
[PLC]  
FILES="path"
```

[STRING value]

Path to where files are to be stored. This sun_key takes effect only if no IO driver or customization DLL responds TRUE to hook PH_GET_FILE_PATH.

Boot mode

```
[PLC]
BOOTMODE="Run"
```

[STRING value]

Possible values are "Run" and "Stop". This determines the PLC mode after loading a boot project successfully.

Disable serial programming

```
[PLC]
DISABLESERIALPROGRAMMING="Yes"
```

[STRING value]

Possible values are "Yes" and "No". If value "Yes" is given, the serial communication server will not be created by the runtime system.

Disable tcp/ip programming

```
[PLC]
DISABLETCP/IPPROGRAMMING="Yes"
```

[STRING value]

Possible values are "Yes" and "No". If value "Yes" is given, the Ethernet communication server will not be created by the runtime system.

Start minimized

```
[PLC]
STARTMINIMIZED="Yes"
```

[STRING value]

Possible values are "Yes" and "No". If the runtime system contains a window and the parameter value is "Yes", the windows will be minimized (run in background) after startup.

Disable online change

```
[PLC]
DISABLEONLINECHANGE="Yes"
```

[STRING value]

Possible values are "Yes" and "No". If this parameter has value "Yes", the feature online change will not be available and no second code buffer is allocated. The memory foot-print is reduced and bigger applications can be created.

Use system interrupt timing as tick

```
[PLC]
USE SYSINTR_TIMING AS TICK="Yes"
```

[STRING value]

Possible values are "Yes" and "No". If this parameter is "Yes", the 1 ms tick will be generated using the interrupt latency timing feature of CE.

Disable PLC load check

```
[PLC]
DISABLEPLCLOADCHECK = "Yes"
```

[STRING value]

Possible values are "Yes" and "No". If this setting is set to "Yes" (default is No), there will be no (implicit) low-priority task for measuring the CPU load caused by the PLC. The PLC browser command pload and the corresponding function in the system libraries will not return any meaningful results.

RS232 port

```
[COMMUNICATION]
[RS232]
PORT=0
```

[DWORD value]

Sets the COM port to be used for the serial communications server. Default 0 is used for 'no port'.

RS232 baud rate

```
[COMMUNICATION]
[RS232]
BAUDRATE=19200
```

[DWORD value]

Baud rate for the RS232 connection.

RS232 stop bits

```
[COMMUNICATION]
[RS232]
STOPBITS=0
```

[DWORD value]

Value sets stop bits according to table below.

Value used	0	1	2
Actual stop bits	1	1.5	2

RS232 parity

```
[COMMUNICATION]
[RS232]
PARITY=0
```

[DWORD value]

Value sets stop bits according to table below.

Value used	0	1	2
Actual parity	No	Odd	Even

RS232 init send string

```
[COMMUNICATION]
[RS232]
INITSENDSTRING="string to send.."
```

[STRING value]

String that is send immediately after the port is opened, e.g. for modem-initialization.

RS232 serial with routing

```
[COMMUNICATION]
[RS232]
SERIALWITHROUTING="Yes"
```

[STRING value]

Possible values are "Yes" or "No". If this is set to "Yes" then the Serial Level 2 Route-protocol will be used instead of the serial standard 3S-protocol.

TCP/IP port

```
[COMMUNICATION]
[TCPIP]
PORT=32001
```

[DWORD value]

Port to use for the TCP/IP communications server.

External communication DLL's

```
[COMMUNICATION]
[EXTERNALDLLS]
CHECKEXTREF="Yes"
```

[BINARY value]

Use "Yes" or "No". Default value is "Yes". Try to link or not functions from the following DLLs.

Enum Key (.0n) Containing String "Path" of name of Windows CE DLL containing external library functions.

Example: "Path"="PLCWINNT_EXTFUN"

Driver list

```
[IODRIVERLIST]
"MODULE0"="HILDPM3S"
"INITFCT0"="IODrvInit"
```

The values "MODULEx" and "INITFCTx" are entered here. "x" is a sequential number which begins with 0 and increases with each driver added. A maximum of 5 drivers may be registered; the maximum valid value of "x" is 4. Please note that the function name specified in "INITFCTx" is case sensitive.

No target Visu tooltips

```
[VISUALIZATION]
[NOTARGETVISUTOOLTIPS]
PORT="Yes"
```

[STRING value]

Default value is “No”. Set to “Yes” and no tool tips will be displayed in target visualizations.

No target Visu tooltips

```
[VISUALIZATION]
CEPRIOVISUINPUTTASK=140
```

[DWORD value]

This setting can be used to modify the priority of the target visualization task for inputs (mouse, keyboard, touch). The default for this priority is a realtime priority (as with normal IEC tasks). With some visualizations (big tables) this task has a long execution time; during this time lower-priority system tasks are not executed (e.g. CE device drivers for graphic display, flash access, serial port, ethernet). If an IEC task wants to perform such an access using system libraries, it will be slowed down by the higher-priority VISU_INPUT_TASK. In such cases the priority should be set to a value between 140 and 200.

Dynamic mouse cursor

```
[VISUALIZATION]
DYNAMICMOUSECURSOR="Yes"
```

[STRING value]

Default value is “Yes”; the mouse cursor in the target visualization can change its form over some visualization elements. If the performance penalty caused by this cannot be accepted, the option should be set to “No”.

Target Visu load only current used bitmaps

```
[VISUALIZATION]
TARGETVISULOADONLYCURRENTUSEDBITMAPS="No"
```

[STRING value]

String, Default “No”: The target visualization buffer in the runtime of CC Pilot XS is about 5.5 MB in size. This means the visualization cannot contain anything that goes beyond the limits of this buffer (number of visualization function calls per cycle etc.). The jpegs are extracted into BMPs in memory and then drawn as bitmaps, so this operation need more heap memory then the impression of the jpg-file size.

TargetVisuLoadOnlyCurrentUsedBitmaps = Yes → only load the required pictures when the screen is switched. Screen switching times can be poorer in this case, but the memory problems are gone. For cases with many pages and bitmaps this setup is necessarily.

TargetVisuLoadOnlyCurrentUsedBitmaps = No → default behaviour, all pictures are loaded to RAM at start-up. This setting is faster but will not be realizable in some cases.

Target Visu load window positions

```
[VISUALIZATION]
TARGETVISUWINDOWBOTTOM=0
TARGETVISUWINDOWLEFT=0
TARGETVISUWINDOWRIGHT=0
TARGETVISUWINDOWTOP=0
```

[DWORD values]

These values specifies the coordinates of the target Visu windows. By default, the target Visu window occupies the whole screen.

Use simulated target

It is possible to do development on a simulated target. The installation is done in the same way as for a real target. All installations are put in the folder 'C:\Program\3S Software\CoDeSys V2.3\Targets\CC Systems'. If you have installed a simulated 10 inch CCP XS there will be a sub folder here called RTS_CCPXS_10_inch_Sim which contains the executable CCPXS_10_inch_Sim.exe. This is the simulated runtime which need to be started. Most of what is explained above about developing for the hardware target is also true for the simulated target. PLC code can be executed and tested and visualizations can be viewed.

Use CC Systems simulation platform - CCSimTech

Simulation can be taken much further with the use of CCSimTech a product of CC System. CCSimTech is s hardware emulation platform that includes CAN and I/O emulation. It also includes tools for monitoring the I/O and the CAN bus. A whole control system can be emulated on the pc and enable debugging without the actual hardware. A mixed hardware and emulation setup is also possible, e.g. a CoDeSys runtime can be emulated on a PC which connects to physical nodes via e.g. a USB-to-CAN device.

More can be read about CCSimTech on www.cc-systems.com.