CoDeSys runtime for CrossFire FX1

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to describe the usage of the CoDeSys runtime for CrossFire FX1.

1.2 References

http://www.**3s**-software.com

1.3 Revision history

Rev	Date	Author	Comments
1.0	2006-10-03	Fredrik Löwenhielm, CCS	Initial version

2 Background

The CoDeSys runtime for CrossFire FX1, is a piece of software running on a Infineon C167 processor. It contains all necessary device drivers and software for communicating with the CoDeSys development editor. The runtime communicates with CoDeSys development editor by using RS232.

3 Installation

The runtime is preinstalled at the CrossFire FX1 when delivered. In order to start using the runtime with CoDeSys it is necessary to install a “target” for FX1 in CoDeSys. This is done with the program ‘InstallTarget’ which is installed automatically when CoDeSys is installed.

Follow these instructions:

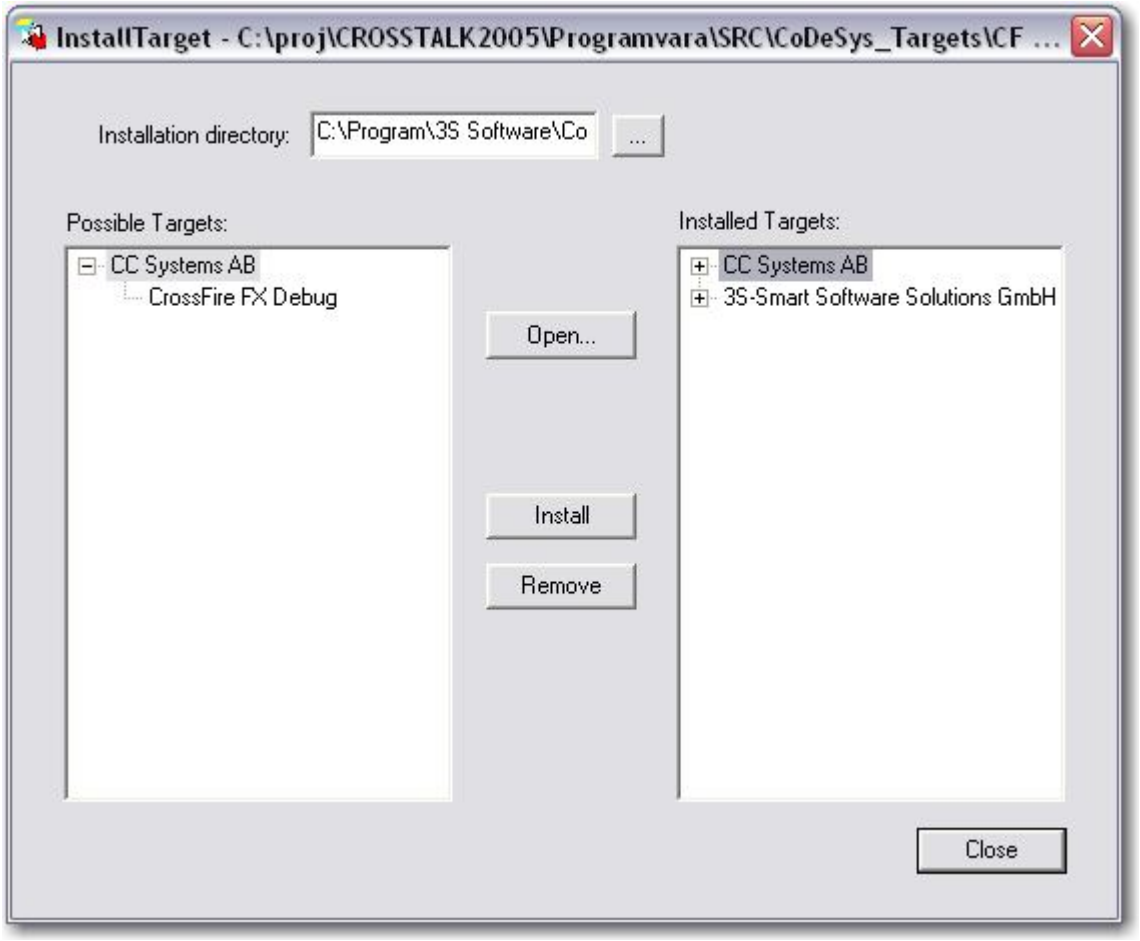
1. Intall CoDeSys. It can be downloaded from 3S web site. It can also be sent to you from CC Systems at request.

2. Start the program InstallTarget. It is found in the start menu of Windows at Start->Program->3S Software->CoDeSys V2.3->InstallTarget

3. In InstallTarget, choose ”Open”. Choose the install folder for CrossFire FX1 that has been delivered to you. If you havn’t received the install folder, you can download it from CC Systems web site.

4. Choose the *.tnf file and open it.

5. The screen should look like this:



6. You can change the installation folder, where all files will be put on your computer. This is done with the button “Installation directory”. However, it is recommended that you don’t change the default settings.

7. Click Install.

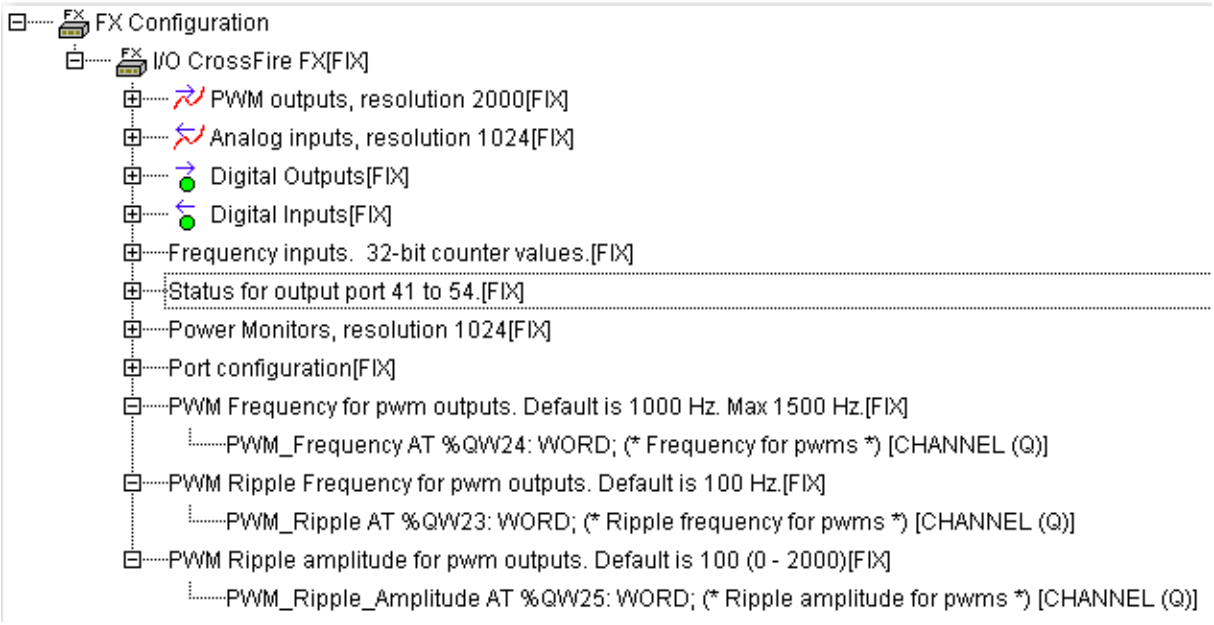
8. Your target for CrossFire FX1 is now installed.

4 Creating a new project in CoDeSys

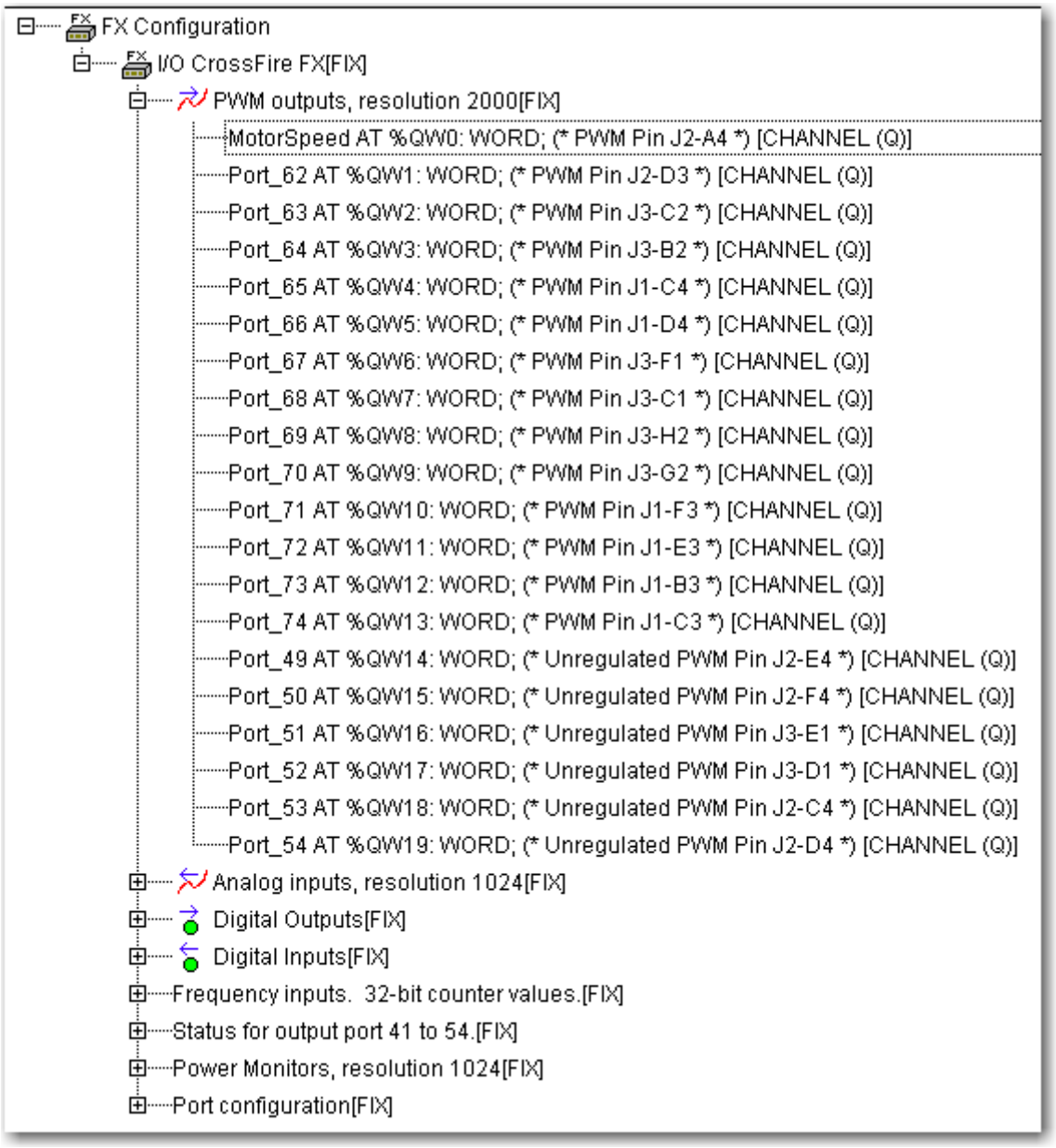
Start CoDeSys. In the first dialog that is shown you can choose your target. If you have followed the instruction for installing your target you should now be able to choose the CrossFire FX1 target. Normally you don’t have to change any of the target settings for CrossFire FX1. Just click OK after choosing CrossFire FX1 as your target.

5 Accessing the IO on your CrossFire FX1

The IO on CrossFire FX1 can be found under PLC Configuration on the Resources tab in CoDeSys. It should look like this:



Under the I/O tree structure it is possible to access all I/O through variables. The variables are global and can thereby be accessed in all your code modules. It is also possible to change the names of the variables by clicking on them and writing a new name. For example, below, Port 61 which is a PWM output has been given the name MotorSpeed:



Below all the accessible I/Os and configurations are desribed:

5.1 PWM outputs

Port 61 - 74 and port 49 – 54 on CrossFire FX1 are pwm outputs. The value range to set these outputs are 0 – 2000. Ports 49 – 54 are pwms without current control.

5.2 Analog inputs

Port 7 – 15 and 31 – 40 and 81 – 94 are analog inputs. The value that can be read from the I/O variables is in the range 0 – 1023. The ports 81 – 94 measures the current flowing through the pins.

In order for port 81 to 94 to work the corresponding low side driver for the ports have to be activated. The low side driver are found under digital output. Port 81 to 94 are internally connected to ground when low side driver is activated. If driver is not activated, the port pins have high-impedance. This means that it is only possible to have a current through the ports when the low side driver is activated, and therefore the analog inputs of the ports are only valid when low side driver is activated. These ports can hereby be used both as digital outputs and analog inputs.

5.3 Digital outputs

Ports 41 – 48 are digital outputs. Also the low side drivers port 81 – 94 are controlled here. Writing TRUE to the variable will turn on the output, and FALSE will turn off the output. Note. In order to use port 81 – 94 as analog input (current measurement) the corresponding low side driver has to be on.

5.4 Digital inputs

Ports 1 – 6 are digital inputs. Also, it is possible to use port 17 – 25 as digital inputs. However, ports 17 – 25 are configured as frequency inputs by default. They can be configured as digital inputs by changing the configuration under Port Configuration. See chapter Port configuration for more information.

5.5 Frequency inputs

These variables are used if ports 17 - 25 are configured as frequency inputs. By default they are configured as frequency inputs. The variables counts up for every pulse that reaches the port pins. The counter value can of course be converted to a frequency value by writing the proper software in CoDeSys user application.

If any of the ports 17 – 25 is configured as digital input instead of frequency input, the corresponding counter value should be ignored.

5.6 Status for port 41 - 54

For every pair of outputs of port 41 – 54 there is one status bit giving information about the status of the output drivers. A low signal on the status bits means that there is a present error. By using the truth table below and switching on and off the faulty ports it is possible to determine which of the two ports the error belong to.

	Out X Ctrl	Out Y Ctrl	Out X Real	Out Y Real	Status bit
Normal operation	L	L	L	L	H
	L	H	L	H	H
	H	L	H	L	H
	H	H	H	H	H
Open load on Out X	L	L	Z	L	H
	L	H	Z	H	H
	H	X	H	X	L
Open load on Out Y	L	L	L	Z	H
	H	L	H	Z	H
	X	H	X	H	L
Short to source voltage on Out X	L	L	L	H	L
	L	H	H	H	H
	H	X	H	X	H

Short to source voltage on Out Y	L	L	L	H	L
	H	L	H	H	H
	X	H	X	H	H
Over temperature on both Out X and Out Y	L	L	L	L	H
	X	H	L	L	L
	H	X	L	L	L
Over Temperature on Out X	L	X	L	X	H
	H	X	L	X	L
Over Temperature on Out Y	X	L	X	L	H
	X	H	X	L	L
Undervoltage / Overvoltage	X	X	L	L	H

X and Y stands for the two ports that belongs to the status bits. For example, if X is port 41, Y is port 42. Ctrl stands for the output the user sets in CoDeSys. Real is the actual signal that comes out from the hardware. L stands for Low signal. H for High signal. Z for high impedance. X for don't care.

5.7 Power monitors

These variables contains analog values of different voltage levels in the hardware. The voltage levels measurable are V+, Vbat, VP1, VP2, VP3, VP4 and the two output 5 volt reference voltages +5 V Ref1 and +5 V Ref2.

5.8 Port configuration

These variables decides if port 17 to 25 should be configured as either frequency input or digital input. Value FALSE means that the port is configured as frequency input, and the corresponding pulse counter of that port can be read under the frequency variables. Value TRUE means that the port is configured as digital in, and the value of the port can be read under the digital in variables.

5.9 Pwm frequency

The PWM frequency is the base frequency of the pwm outputs. The maximum value is 1500 Hz. This frequency should be greater than the ripple frequency. Default value is 1000 Hz.

5.10 Pwm ripple frequency

The pwm ripple frequency is used for adding a ripple to the pwm outputs. By using a ripple the coil is always in movement, and that eliminates the "start energy" that otherwise is needed when changing the value of the pwms. This makes the controlling of the coil more linear. Default value is 100 Hz. Maximum is 500 Hz.

5.11 Pwm ripple amplitude

This is the amplitude of the ripple for the pwm outputs. Possible values are 0 – 2000, and the these values are on the same scale as the output values of the pwms. For example, if the user sets the ripple amplitude to 100, and sets the output of a pwm to 1000, the output will ripple between 900 – 1100. If instead the ripple amplitude is set to 200, the ripple would be 800 – 1200. Default value is 100.

.

6 Accessing the EEProm

In order to access the eeprom of the unit, the library EEPROM_DRV.lib has to be loaded. This is done under the Resource Tab in CoDeSys. Choose Library manager. Chose Insert->Additional Library from the menu in CoDeSys. Now, a dialog should appear with all the libraries that was installed with the target. Choose the EEPROM_DRV.lib library.

Now the library is loaded and the functions EEProm_Init, EEProm_Write, EEProm_Read and EEProm_GetSize is ready to be used.

The explanation of parameter and return values of the functions can be found in the library manager.

Note. If power to the unit is lost during a write operation to the EEprom, the content of the EEprom may be corrupted. However, it is possible to write mechanisms in the CoDeSys software to recover from corrupted eeprom. This can for example be done by always writing to two different areas. At startup the areas can be compared, and thereby corruption of the eeprom be detected and corrected.

7 Accessing the CAN bus

The CAN bus is accessed by using the library 3S_CanDrv.lib. This library is described in the CoDeSys documentation. Normally the user does not access the CAN bus directly with this library. Instead a CANopen master is added in the PLC configuration. This is also described in the CoDeSys documentation.

8 Writing an application in CoDeSys

How to write an application in CoDeSys, is not described in this document. The reader should instead use the manuals that was installed together with CoDeSys. 3S is responsible for the manuals of CoDeSys.

9 Connecting to the target

When you have your CoDeSys project ready you need to download it to your hardware. First you have to set your communication parameters. This is done in the menu Online->Communication parameters in CoDeSys.

In the dialog, click New and the choose Serial (RS232). Click Ok. Now you are ready to download your project.

Choose Online->Login. CoDeSys will ask you if you want to download your application. Click Yes. Now your application should start downloading. When it is downloaded you can start and debug your program. For more information, consult your CoDeSys documentation.

10 Creating a boot project

When you download your project to your hardware it is only downloaded to the RAM memory. If you turn off power to your hardware the project will be lost, and you have to download it again.

In order to save the project in the flash memory of the hardware you have to create a boot project. When you're logged in to the hardware choose Online->Create boot project.

The project will now be copied from the RAM memory to the flash memory. The next time the unit starts, it will automatically load the program from the flash memory to the RAM memory and start executing the code.