CrossFire™ SX – CODESYS Developers Guide

Revision: 1.1
2020-01-14

CrossFire™ SX

CODESYS Developers Guide

crosscontrol
www.crosscontrol.com

Contents

1. Introduction

1.1. Background

CrossFire SX is a freely programmable 32-bit I/O controller, designed for use in Agricultural and Construction equipment. It offers 38 I/O channels, fully versatile and configurable in software. Crossfire SX, CODESYS is freely programmable through CODESYS with full support for CANopen Master, CANopen Slave and J1939.

The purpose of this document is to describe the usage of the CODESYS runtime for CrossFire SX.

Overview:

- 32-bit safety CPU, Texas Instruments TMS570 ARM Cortex R4, 180 MHz
- 4 4A outputs, 12 2A outputs, 16 inputs, 2 H-bridge, 4 node-id digital inputs
- 3 CAN with full support for CANopen Master, CANopen Slave and J1939
- CODESYS version: 3.5
- Debug interface: RS232, 115.2 kbps
- Retain memory: 640 bytes stored in FRAM
- Available RAM: 256kB divided between runtime and user application External SRAM available as option
- FLASH available to CODESYS application: 1.5MB
- Extensive supervision functions including external watchdog and feedback of current, frequency and duty cycle and measurement of internal voltages and temperatures for improved safety

1.2. References

- [1] CrossFire SX - Technical Manual.docx
- [2] CrossFire SX – Firmware upgrade instructions.docx
- [3] CODESYS V3, Installation and Start.pdf
- [4] CODESYS online help
- [5] https://www.codesys.com/
- [6] http://store.codesys.com
- [7] http://support.crosscontrol.com

1.3. History

Rev	Date	Author	Changes
0.1	2016-06-02	Fredrik Wahlström	First draft
0.2	2017-01-23	Carl-Magnus Moon	
0.3	2017-03-08	Mattias Lång	Update with new template

0.4	2017-08-08	Carl-Magnus Moon	Updates regarding H-bridge
1.0	2017-08-16	Carl-Magnus Moon/ Jens Rubensson	PWM does now support 2Khz Limitations in PWM frequency and duty cycle feedback
1.1	2020-01-14	Carl-Magnus Moon	Added documentation for new input modes (pulse counter and high resolution frequency input). Clarifications regarding the retain memory. Added chapter about memory configuration.

2. Installation

2.1. Software installation

Please read the CrossFire SX - Technical Manual.docx for information about mechanical and electrical installation of CrossFire SX as well as electrical characteristics.

CrossFire SX, CODESYS comes preinstalled with CODESYS runtime. To start developing with CODESYS it is necessary to install the correct version of the CODESYS IDE on your PC. The CODESYS IDE version should match the version of the CODESYS runtime on the CrossFire SX. As of writing this manual, the correct version is 3.5SP10. CODESYS IDE can be downloaded from 3S (http://store.codesys.com) or CrossControl support site (http://support.crosscontrol.com/).

It is also necessary to install the CrossFire SX device description file into CODESYS device repository. This is done from CODESYS IDE by selecting Tools/Device Repository and clicking “Install…”. Browse for the CF_SX_V3.5.10.0.devdesc.xml file. CF_SX_V3.5.10.0.devdesc.xml can be downloaded from CrossControl support site http://support.crosscontrol.com.

2.2. Very Important!

After downloading an application the outputs will be shut off due to the idle task watchdog will expire. It is necessary to perform a “reset warm” command from CODESYS IDE to activate them again.

It is not possible to change port settings after the first PLC cycle. In case this is needed, the soft_restart_node command must be used. Performing a “reset warm” or restarting the unit is also possible.

2.3. Software Upgrade

It is possible to upgrade the firmware of the CrossFire SX with the SX Config Tool. It is also possible to download a CODESYS application with this tool. More information about this is available in the CrossFire SX – Firmware upgrade instructions.docx.

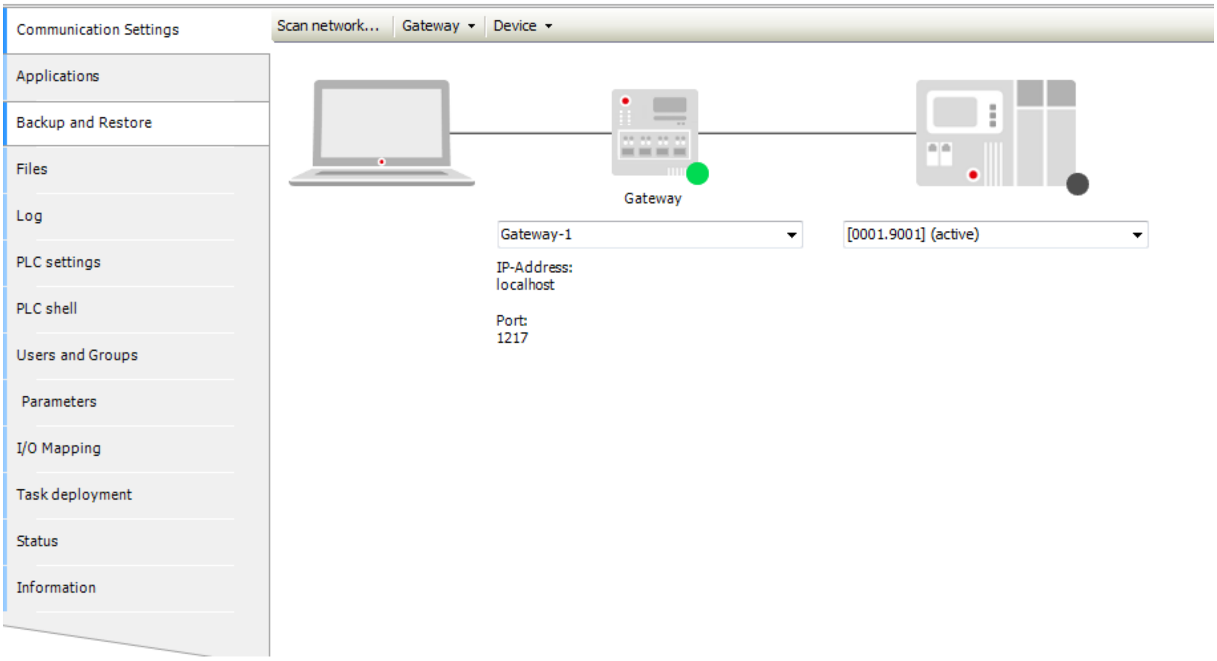
2.4. LED

The Red/Green LED is mainly controlled by the user application. During boot up the LED is used to indicate start-up errors. Please read the technical manual for full details.

Color	Meaning
Amber	No or incompatible application.
Blinking red	Startup tests failed.

2.5. Connecting to the target

CrossFire SX is connected to the PC with the RS232 connection. To be able to connect to the CrossFire SX though the serial connection, it is necessary to configure the local gateway. This is done by going to the Communication Settings tab in CODESYS IDE. Select “Configure the local Gateway” from the Gateway menu.



Gateway-1

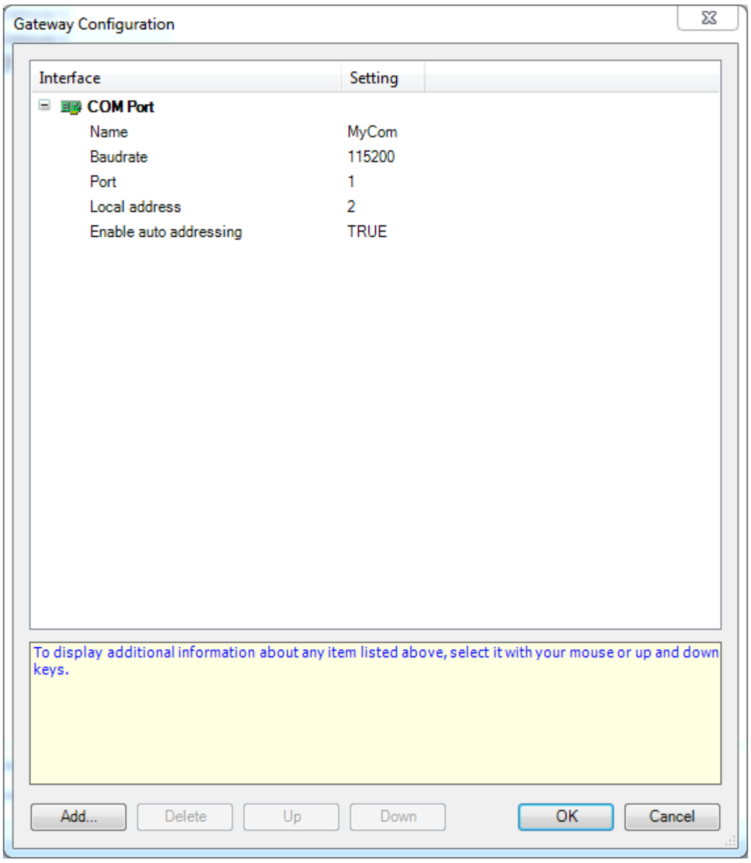
IP-Address:
localhost

Port:
1217

[0001.9001] (active)

Baudrate should be set to 115200. Port should be set up to the number of PC serial port that is used (COM1, COM2…). Auto Addressing should be enabled. Name can be set to anything desired.

The settings shall look like the following screenshot.



Interface	Setting
COM Port	
Name	MyCom
Baudrate	115200
Port	1
Local address	2
Enable auto addressing	TRUE

To display additional information about any item listed above, select it with your mouse or up and down keys.

If any of the configuration settings are missing click “Add…” and select “Add configuration setting..”. Make sure to restart the gateway after changing configuration. This can be done by right clicking on the gateway in the system tray and selecting “Stop Gateway” followed by “Start Gateway”.

When the gateway is configured, select the gateway and click “Scan Network” to scan for CrossFire SX units. When the unit is found make sure to select CrossFire SX as active.

In case of problems, check the cabling carefully and also the gateway settings. In case of using an external USB to serial cable, try another brand.

In case a non-working CODESYS application has been downloaded to the unit by mistake, it might be necessary to download a working CODESYS application through the boot loader using the CrossFire SX Config Tool.

After downloading an application the outputs will be shut off due to the idle task watchdog will expire. It is necessary to perform a “reset warm” command from CODESYS IDE to start them again.

It is also possible to download a CODEYS application to the unit using CrossFire SX Config Tool. The CrossFire SX Config Tool downloads the application over the CAN bus using the CANopen protocol. More information about this is available in the CrossFire SX – Firmware upgrade instructions.docx.

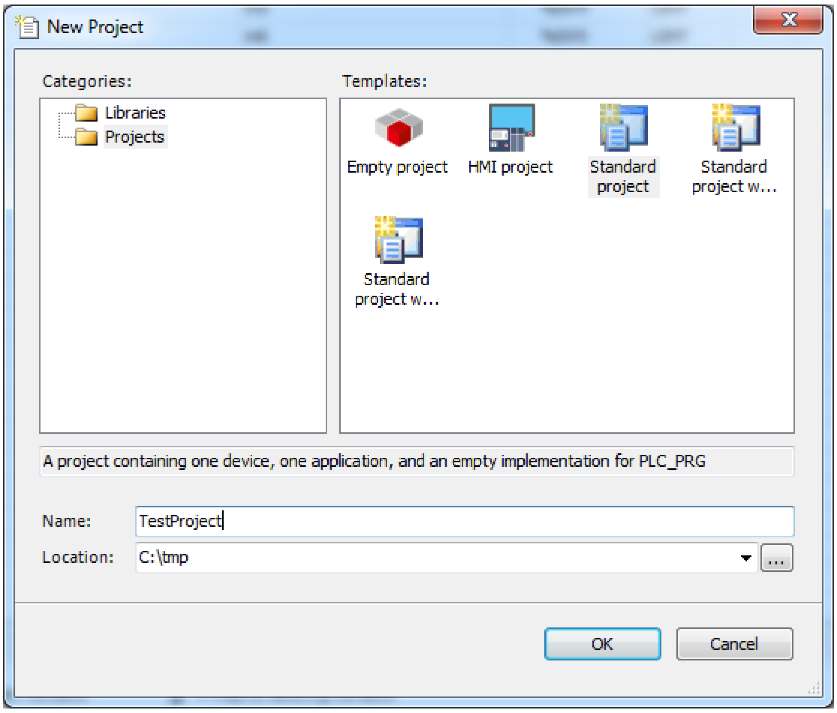
3. Programming CrossFire SX

3.1. Writing an application in CODESYS

CODESYS programming in general is not described in this manual. The reader should instead read the manuals and online-help that were installed together with CODESYS.

3.2. Creating a new project in CODESYS

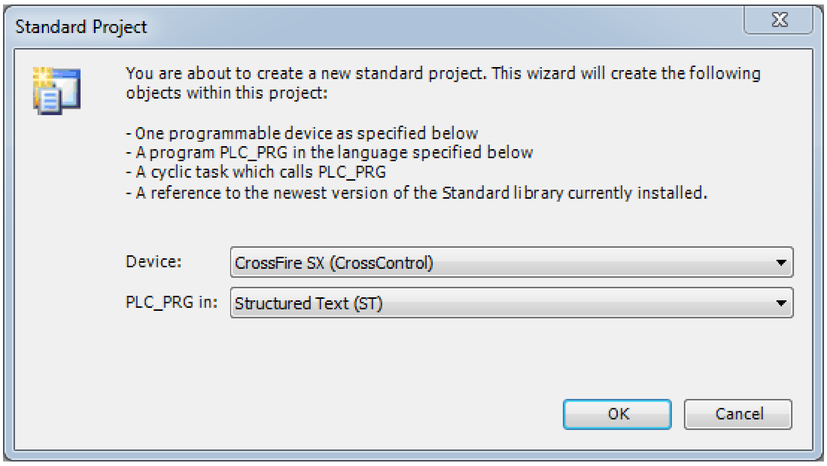
Start CODESYS IDE. Select “New Project…” from the File menu. Select “Standard Project”. Select “CrossFire SX” in the Device list. If CrossFire SX is not available in the list, the CrossFire SX device description file needs to be installed first. Select desired programming language for PLC_PRG. Click ok and a new empty project will be created.



A project containing one device, one application, and an empty implementation for PLC_PRG

Name: TestProject

Location: C:\tmp



You are about to create a new standard project. This wizard will create the following objects within this project:

- One programmable device as specified below
- A program PLC_PRG in the language specified below
- A cyclic task which calls PLC_PRG
- A reference to the newest version of the Standard library currently installed.

Device: CrossFire SX (CrossControl)

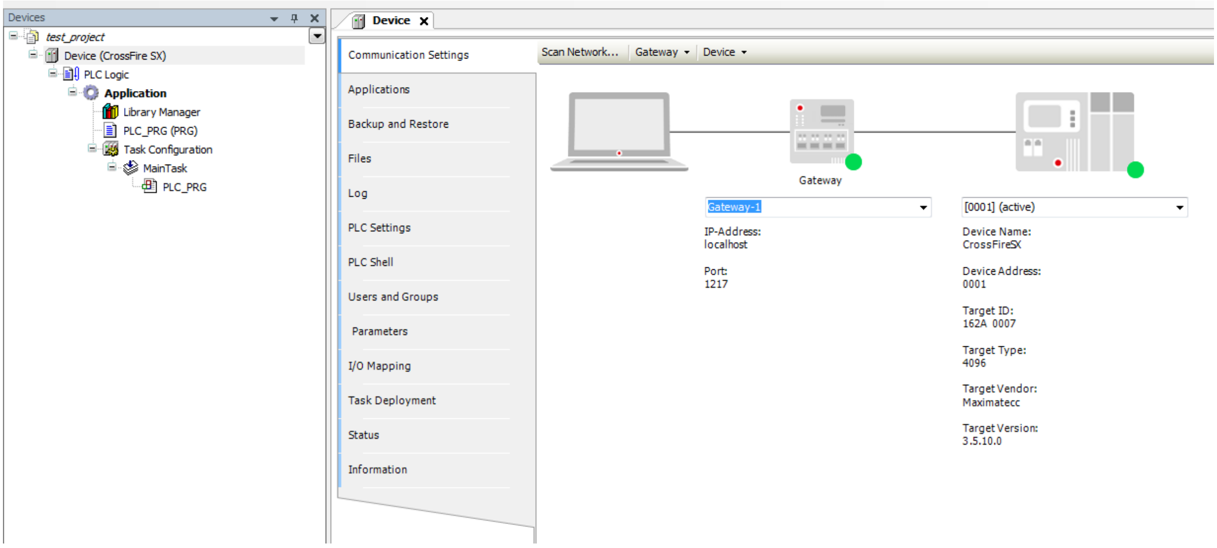
PLC_PRG in: Structured Text (ST)

Often it is easier to use the supplied template project instead of creating a completely empty project. In the template project all IO is already mapped to variables. Also a number of enums and structs are predefined to make programming easier. The template project can be downloaded from CrossControl support site (support.crosscontrol.com).

3.3. The Device Tree

The device tree is a very important part of a CODESYS project. To the device tree additional items such as PRGs, FBs, FUNs, tasks and can buses can be added.

When selecting the CrossFire SX device in the device tree, a large number of setting tabs will be available, usually to the right of the device tree.



Gateway-1

IP-Address:
localhost

Port:
1217

[0001] (active)

Device Name:
CrossFireSX

Device Address:
0001

Target ID:
162A 0007

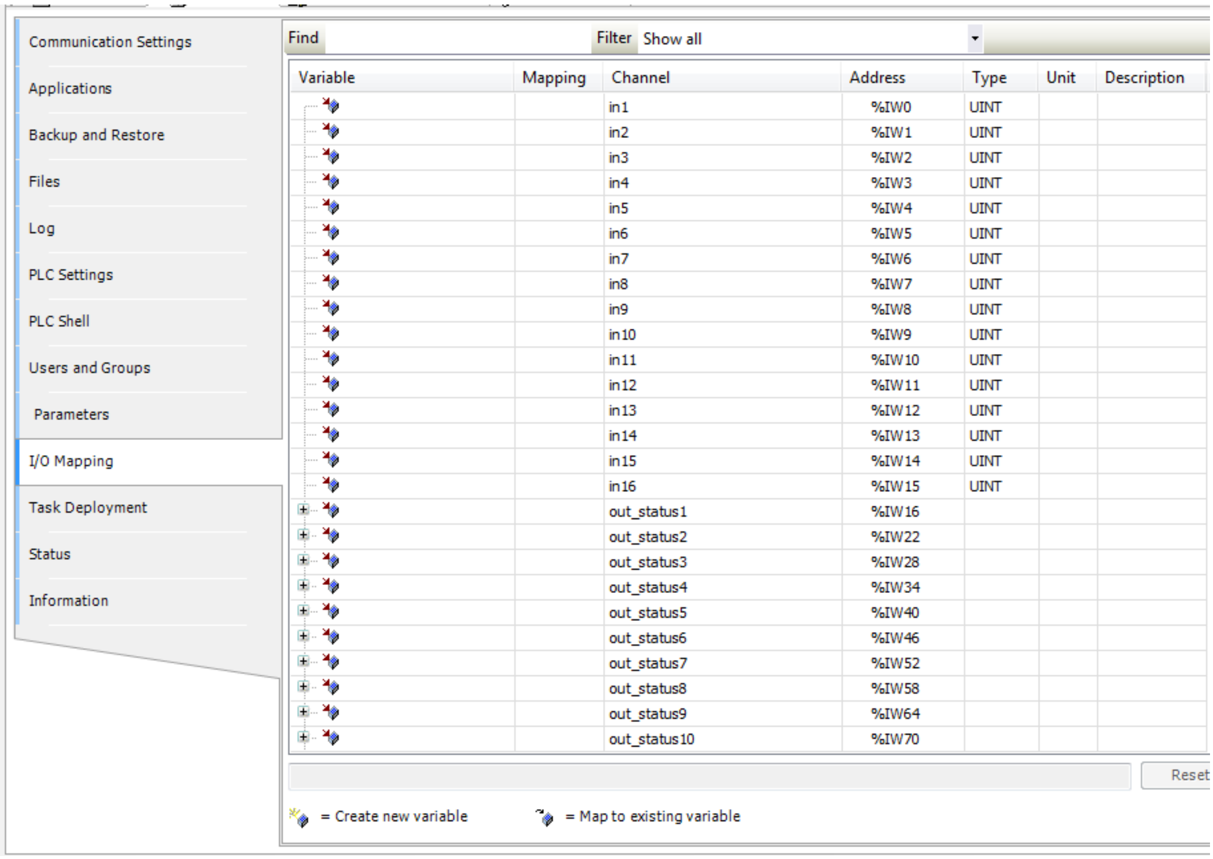
Target Type:
4096

Target Vendor:
Maximatecc

Target Version:
3.5.10.0

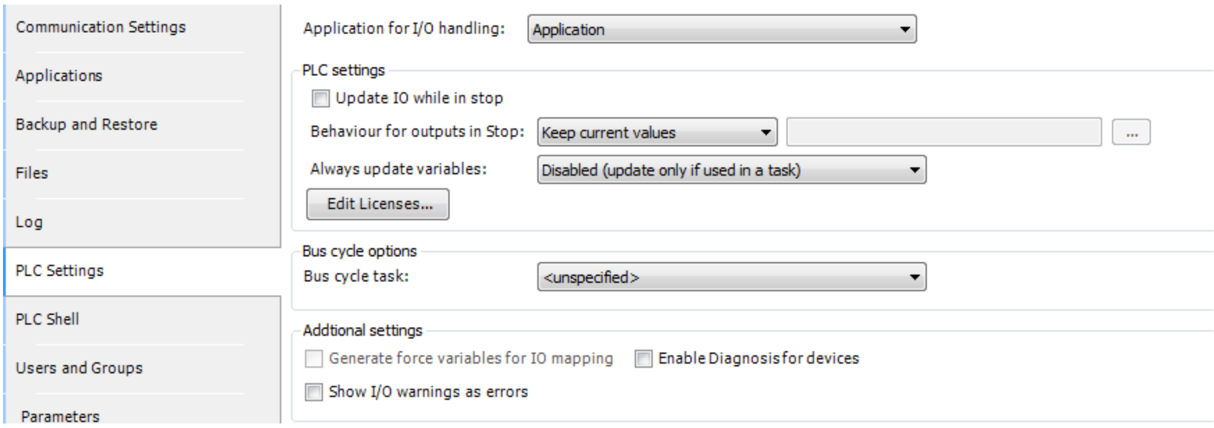
3.4. Accessing the I/O on CrossFire SX

The I/O on CrossFire SX can be found in the I/O Mapping tab of the CrossFire SX device. As the I/O is configurable, both configuration settings and actual I/O data is available. Also status information is available. Make sure to make use of the status information as this can make the complete system much more safe and tolerant against errors.



Variable	Mapping	Channel	Address	Type	Unit	Description
		in1	%IW0	UINT		
		in2	%IW1	UINT		
		in3	%IW2	UINT		
		in4	%IW3	UINT		
		in5	%IW4	UINT		
		in6	%IW5	UINT		
		in7	%IW6	UINT		
		in8	%IW7	UINT		
		in9	%IW8	UINT		
		in10	%IW9	UINT		
		in11	%IW10	UINT		
		in12	%IW11	UINT		
		in13	%IW12	UINT		
		in14	%IW13	UINT		
		in15	%IW14	UINT		
		in16	%IW15	UINT		
		out_status1	%IW16			
		out_status2	%IW22			
		out_status3	%IW28			
		out_status4	%IW34			
		out_status5	%IW40			
		out_status6	%IW46			
		out_status7	%IW52			
		out_status8	%IW58			
		out_status9	%IW64			
		out_status10	%IW70			

Note that only I/O that is mapped to a variable and is actually used will be updated during debugging. It is possible to change this behavior in the PLC Settings tab. Change the setting “Always update variables” to “Enabled1” or “Enabled2”. Changing this behavior will make debugging easier. However, the performance will be worse as data will be copied that is not used; therefore this setting is not recommended if high performance is needed.



Application for I/O handling: Application

PLC settings

□ Update IO while in stop

Behaviour for outputs in Stop: Keep current values

Always update variables: Disabled (update only if used in a task)

Bus cycle options

Bus cycle task: <unspecified>

Addtional settings

□ Generate force variables for IO mapping □ Enable Diagnosis for devices

□ Show I/O warnings as errors

3.5. CrossFire SX I/O list

Below, all I/O of CrossFire SX will be described

in1-in16

The 16 inputs of the CrossFire SX are accessible through the in1-in16 channels. The value available here depends on the input configuration.

Mode	Nominal Range	Comment
Not Used	0	
0-5V	0-5000mV	
0-10V	0-10000mV	
0-32V	0-32000mV	
4-20mA	0-20000uA	
Digital	0,1	
Encoder	0-16777215 steps (24-bit)	For encoder inputs the value must be read as the combined value from both inputs used. The value has 24-bit resolution. LSB will be located at the lowest port number, MSB at the highest port number.
Frequency	0-20000Hz	For frequencies above 1kHz,frequency is measured as pulses during 10ms which results in a resolution of 100Hz
Pulse Counter	0-16777215 steps (24-bit)	Available on port 13,14,15,16. For pulse counter inputs the value must be read as the combined value from both inputs used. The value has 24-bit resolution. LSB will be located at the lowest port number, MSB at the highest port number.
High Resolution Frequency	0-20000Hz	Available on port 11,12,13,14,15,16. Measures both pulses and exact time which give a better resolution than standard frequency measurement.

Note that the values read might be slightly above the nominal range.

Out_status1-out_status16

The outputs of CrossFire SX have a number of built in protection functions. It is possible to read the status of the 16 outputs of the CrossFire SX though the out_status1-out_status16 parameters. The following status is available:

Parameter	Nominal Range	Comment
Current Feedback	0-4000mA	The current sourced by the port. Mainly used for PWMi but can also be used in the other modes.
Current Feedback STM32	-	Only to be used by CrossControl
Status Bits	bit 0: Short to GND bit 1: Reserved bit 2: Reserved bit 3: Thermal warning bit 4: Charge pump under-voltage lockout bit 5: Over current bit 6: Reserved	Status bits indicating different errors on the port.

	bit 7: Reserved	
Duty Cycle	0-1000 (0-100% in 0.1% steps)	The duty cycle feedback if PWM mode is used
Frequency	0-20000Hz	The frequency feedback if PWM mode is used
Digital	0,1	Mainly used in digital mode but can also be used in other modes

The Short to GND, Change pump under voltage lockout and Over current errors will shut down the output. Use the retry_after_output_failure parameter to activate the output again. Thermal warning error is just an indication; port will not be shut down automatically. It is not recommended to use output for longer periods with thermal warning bit active.

Please note that all the above status is measured values and might differ slightly from the expected value. It is recommended to test the feedback functions in a real world application. The values read might be slightly above the nominal range.

For the PWM frequency and duty cycle feedback values to be reliable the time between any falling or rising edge of the PWM signal must not be shorter than 3 µs. For example at 1 kHz PWM frequency the 3 µs limit corresponds to a max duty cycle of 99.7% and a min duty cycle of 0.3%.

hbridge_status1-hbridge_status2

Note! LS_Current STM32 and HS_Current STM32 are not used.

Status is a bitmask and can be a combination of the following:

Bit 0: current limiting is active (motor is still running but motor current is reduced by the drive-circuit)

Bit 1: general fault indication from drive circuit

Bit 2: over current on high side

VOLTAGE_...

The supply voltages supervised on the CrossFire SX board

VOLTAGE_UOGA	The voltage on output group A
VOLTAGE_UOGB	The voltage on output group B
VOLTAGE_VCOMP	The voltage supply for the comparators used by digital inputs.
VOLTAGE_3V0	3.0V reference voltage used for current regulation.
VOLTAGE_2V8	2.8V supply used as digital threshold level
VOLTAGE_6V5	6.5V supply used to power sensor supply voltage tracking regulator.
VOLTAGE_SENSORSUPPLY5V	Sensor supply output voltage 0-42.5V range.
VOLTAGE_5V14	5.14V supply for TMS570 AD-converter analog rail (this is not VREF)
VOLTAGE_ADC_SENSOR_HIRES	Sensor supply output voltage 0-8.33V range.

To make it easier for the programmer, a bit mask indicating voltage errors are available in the voltage_error parameter. The exact voltages can be interesting for debug or logging purposes but is not needed to detect voltage errors.

HBRIDGE_A_HS_CURRENT (OGA_HB_HS), HBRIDGE_A_LS_CURRENT (OGA_HB_LS), HBRIDGE_B_HS_CURRENT(OGB_HB_HS), HBRIDGE_B_LS_CURRENT(OGB_HB_LS)

These parameters contain the current (in mA) going through the two H-bridges. Note that this value is only accurate at 100% duty cycle so this value is mainly intended to see if a connected motor is running or not. There are two values available, one for low side (GND) and one for high side (VBAT).

HBRIDGE_A_HS_CURRENT (OGA_HB_HS)	Current for H-bridge 1 on high side
HBRIDGE_A_LS_CURRENT (OGA_HB_LS)	Current for H-bridge 1 on low side
HBRIDGE_B_HS_CURRENT (OGB_HB_HS)	Current for H-bridge 2 on high side
HBRIDGE_B_LS_CURRENT (OGB_HB_LS)	Current for H-bridge 2 on low side

TEMP_...

There are four temperature sensors on the CrossFire SX board. They are available through these feedback signals. Values read are in degrees Celsius. Values might be negative.

TEMP_OGA	Temperature in output group A
TEMP_OGB	Temperature in output group B
TEMP_POWERIN	Temperature in the power stage
TEMP_STM32	Temperature on the current control processors

The over_temp bitmask is also available to make it easier to detect temperature errors. The actual temperatures are mainly interesting for debug or logging purposes.

node_id_pins

Returns the node id from the four node-id pins. These digital inputs are intended to set the CANopen node id but can also be used for other purposes.

These pins are also used by the CANopen bootloader to decide which nod id will be used during software upgrade. The node id from the node id pins can be overridden in software. If writing a non-zero value to FRAM, the value from FRAM will be used instead.

CANopen Node ID	A3	A2	A1	A0
1	0	0	0	0
2	0	0	0	1
3	0	0	1	0
4	0	0	1	1
….				
16	1	1	1	1

- State 0 is reached by wiring corresponding ID pin to GND.
- State 1 is reached by leaving corresponding ID pin disconnected.
- Note, if no pins have been connected to GND then the Node ID will be set to 16.

watchdog_status

This parameter returns the status of the external watchdog.

Bit 0-3 contains the following:

Value	Status
0	WATCHDOG_STAND_BY_STATE
3	WATCHDOG_RESET_STATE
4	WATCHDOG_SAFE_STATE
5	WATCHDOG_ACTIVE_STATE
7	WATCHDOG_DIAG_STATE

During normal operation the watchdog will be in WATCHDOG_ACTIVE_STATE state.

Bit 7: contains a flag telling if the idle task watchdog has expired. The idle task watchdog will expire if the idle task is not run as expected. This usually means the system is over loaded or the idle task watchdog time is too low.

fw_upgrade_pin

This parameter returns the status of the firmware upgrade pin. This pin is mainly used by the boot loader to decide if it should stay in boot mode but can also be read from application. 0 means pin is set to GND, 1 means pin is set to VBAT.

system_versions

This parameter returns the versions of the different software parts of the system. The version numbers are read as 32-bit hex values. For instance 1.0.1.0 is encoded as 0x01000100

Boot_Version	Version of boot loader
CODESYS_RT_Version	Version of CODESYS runtime
STM32_OGA_Version	Version in current control processor for output group A
STM32_OGB_Version	Version in current control processor for output group B

voltage_error

This parameter indicates a problem with the supply voltages on the CrossFire SX. There are five voltages that are supervised. In case of an error on any of them, the corresponding bit will be set in the bitmask. Note that also voltages not supervised are available in the bitmask. It is also possible to read the exact voltage using the VOLTAGE_... channels.

Bit	Voltage	Supervised
0	ADC_3V_AD	Yes
1	ADC_UOGA	No
2	ADC_VCOMP	Yes
3	ADC_REF_TEMP	No
4	ADC_6V5	No
5	ADC_UOGB	No
6	ADC_SENSOR	No
7	ADC_2V8_REF	Yes
8	ADC_5V14_AD	Yes
9	ADC_SENSOR_HIRES	Yes

The CrossFire SX will NOT be automatically shut down in case of errors of any of these. It is highly recommended that the application programmer reads this status bits and takes appropriate action in case of error.

over_current_4_20

The inputs are switched off in case of over current in 4-20mA mode. Over current is detected at roughly 25mA. This bit mask contains which inputs have been switched off. A switched off input is indicated with a '1' and a working input is indicated by '0'. The inputs will be switched on again automatically after 300ms. If there is still over current they will be switched off again.

unit_status

This parameter returns generic unit status information

- Bit 0: Over temperature has been detected and outputs has been shut off.
- Bit1: Error reading temp sensor 1
- Bit2: Error reading temp sensor 2
- Bit3: Error reading temp sensor 3
- Bit4: Error reading temp sensor 4
- Bit5: Error when setting input configuration. All inputs have been shut off.

Bit 0 is set if over temperature has been detected. Over temperature is calculated as a combination of all temperature sensors. If this happens, outputs will be shut off automatically. Outputs can only be enabled again by sending the restart command or restarting the unit.

Bit 1-4 is set if the corresponding temperature sensor cannot be read due to some hardware or software error. The user must decide if it is better to continue running or shut down in this case. No automatic shutdown will be performed.

Bit 5 indicates an error in the circuits controlling the input modes. If this is detected all inputs are automatically shut off and will return 0.

node_id_can0_read

At start up the node-id read from FRAM can be read here. It is recommended to copy this value to an internal variable the first loop turn. This new variable can be mapped to the corresponding output parameter and can be changed when desired causing the new value to be stored into FRAM.

baudrate_can0_read

At start up the baudrate read from FRAM can be read here. It is recommended to copy this value to an internal variable the first loop turn. This new variable can be mapped to the corresponding output parameter and can be changed when desired causing the new value to be stored into FRAM.

production_test_result

Only to be used by CrossControl.

production_test_done

Only to be used by CrossControl.

out1-out16

These channels are used to control the outputs of the CrossFire SX. How the value is used depends on the mode of the output.

Mode	Range
Not used	-
PWM	0-1000 (0-100% in 0.1% steps)
PWMi	0-4000mA for 4A outputs, 0-2000mA for 2A outputs
Digital	0,1

hbridge1-hbridge2

These channels control the two H-bridges. The H-bridge always operates at 20 kHz PWM frequency.

Parameter	Range
Direction	0: stop 1: forward 2: backward 3: brake
Speed	0-1000 (0-100% in 0.1% steps)

status_led_control

Controls the red/green status LED.

Value	Color
0	Off
1	Green
2	Red
3	Amber (red + green)

retry_after_output_failure

This is a bitmask where each of the 16 outputs uses one bit. Setting a bit to 1 clears the errors of the corresponding output. It is necessary to set the bit back to 0 to be able to clear the error again.

supply_shutoff

For safety reasons it is possible to shut of the supply voltages to Output Group A (OGA), Output Group B (OGB) and sensor supply. Setting a bit to ‘1’ shuts off the output group. This can be done if the user application detects a severe error and tries to shut of an output but through the feedback channels can see that this fails. Then the complete group can be switched off instead.

bit 0: Shut off Group A

bit 1: Shut off Group B

bit 2: Shut of Sensor Supply

reset_encoder

This is a bitmask used to reset the encoder value. Bit 0 corresponds to port 1 and 2, bit 1 corresponds to port 3 and 4 and so on. This parameter must be set back to 0 to be able to reset encoder again.

baudrate_can0_write

The value supplied here will be stored to FRAM. This value will be used by the boot loader in case of a software upgrade. It is possible to read the value stored the next boot.

node_id_can0_write

The value supplied here will be stored to FRAM. This value will be used by the boot loader in case of a software upgrade. It is possible to read the value stored the next boot.

idletask_watchdogtime

This parameter sets the watchdog time for supervising the idle task. Default value is 80ms. This value can be increased if the idle task is expected to run very slow or decreased if faster switch off time is required.

calibration

Only to be used by CrossControl.

soft_restart_node

Setting this parameter to 0x1234 performs a soft restart of the CrossFire SX. All port setting changes will take effect and output errors are cleared. It is necessary to set the value back to 0 before a new soft restart can be performed.

production_test_command

Only to be used by CrossControl.

in_config1…in_config16

Configuration of input 1-16. The following settings are available:

Setting	Values
Mode	0 = Not Used 1 = 0-5V 2 = 0-10V 3 = 0-32V 4 = 4-20mA 5 = Digital 6 = Encoder 7 = Frequency 8=Pulse Counter (port 13,14,15,16) 9=High Resolution Frequency (port 11,12,13,14,15,16)

Bias	0 = No Bias 1 = Pull Up 2 = Pull Down	

For encoder and pulse counter mode two inputs must be combined. Only adjacent ports may be combined, for instance port 13 and port 14 or port 15 and port 16.

It is not possible to change port settings after the first PLC cycle. In case this is needed, the soft_restart_node command must be used.

out_config1…out_config16

Configuration of output 1-16. The following settings are available:

Setting	Values	Comment
Mode	0 = Not used 1 = PWM 2 = PWMi 3 = Digital	
PWM Frequency	50-2000Hz (max 400Hz prior to release 7)	
Dither Frequency	25-400Hz in predefined steps	
Dither Amplitude	0-400mA	
P	0-5000	
I	0-100	
D	-	Not used - Reserved
A	0-255	Soft start parameter A, read system manual for more information
B	0-255	Soft start parameter B, read system manual for more information

PID parameters can be changed to optimize current control (PWMi) for specific valves. In most cases this is not needed.

Output 1,2,9,10 are rated for 4A. The other outputs are rated for 2A. The 4A outputs are designed to be able to drive capacitive load like LED lamps. The 2A outputs can drive some capacitive load but with limitations. Over current detection level is roughly 2200mA for 2A outputs and 5000mA for 4A outputs. Read the system manual for more information.

Note that it is not possible to change port settings after the first PLC cycle. In case this is needed, the soft_restart_node command must be used.

hbridge_config1…hbridge_config2

Configuration of H-bridge 1 and 2.

The H-bridge has two outputs. Depending on if forward or backward command is given, output A is connected to low side (LS) and output B to high side (HS) or vice versa. In case of the brake command, both outputs are connected to LS.

It is possible to set two current limits for the H-bridge. One is measured on LS and one on HS. If the LS current limit is exceeded, the current is limited but not shut off. If the current limit on HS is exceeded the output is turned off. It is recommended to set the current limit on HS higher than the LS limit. The HS limit is mainly intended to protect against short circuit between HS and GND. The LS limit is mainly intended to reduce inrush current.

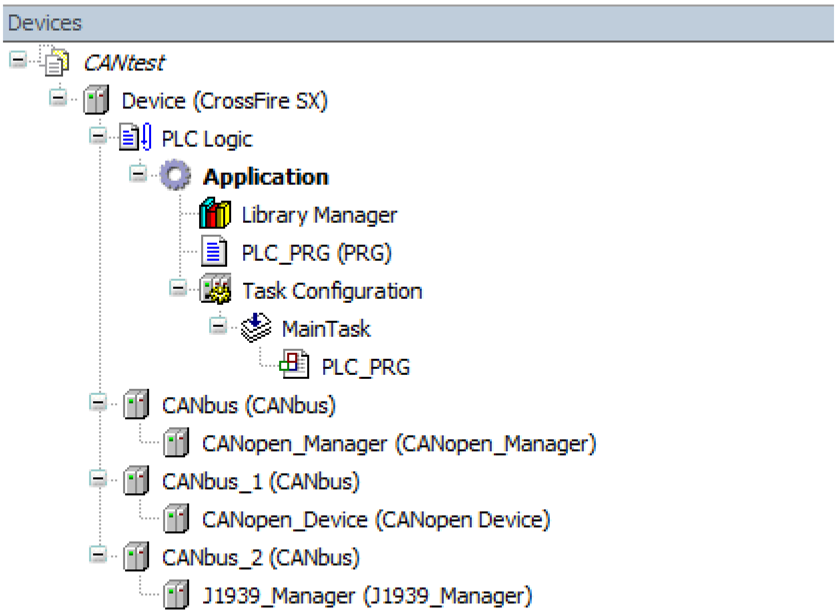
It is recommended to use both outputs from the H-bridge (not connecting directly to GND) to get full protection and current limitation.

Current Limit LS	0-20000	mA
Current Limit HS	0-25000	mA

3.6. Accessing the CAN bus

CrossFire SX supports three CAN buses. The user can add up to three CAN bus devices to the CODESYS device tree. All three CAN buses are individually configurable regarding baud rate, node-id (if applicable) and bus protocol.

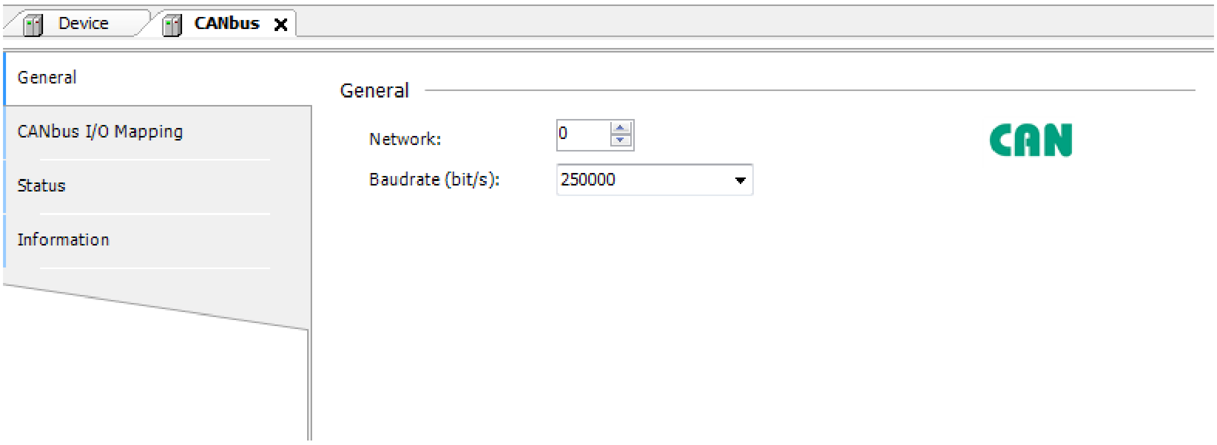
The CAN0 bus is also used by the boot-loader. The baud rate and nod-id used by the bootloader is programmable through the IO device.



Devices

- *CANtest*
  - Device (CrossFire SX)
    - PLC Logic
      - **Application**
        - Library Manager
        - PLC_PRG (PRG)
        - Task Configuration
          - MainTask
            - PLC_PRG
    - CANbus (CANbus)
      - CANopen_Manager (CANopen_Manager)
    - CANbus_1 (CANbus)
      - CANopen_Device (CANopen Device)
    - CANbus_2 (CANbus)
      - J1939_Manager (J1939_Manager)

The Network and Baudrate can be set in the CANbus settings dialog. Network 0 corresponds to CAN1, network 1 to CAN2 and network 2 to CAN3. The Baudrate can be hard coded by selecting the Baudrate in the menu. For CAN1 (network 0) It is also possible to select “Use current setting”. In case “Use current setting” is used the Baudrate is read from FRAM at start up. For the other CAN buses this setting will have no effect. To write the desired value to FRAM, the baudrate_can0 IO parameter in the IO driver is used. The reason for having this special handling for CAN0 is that CAN0 is also used by the boot loader.



General

Network: 0

Baudrate (bit/s): 250000

3.7. Debugging in CODESYS

To make it easier to debug an application, the built in CODEYS debugger can be used. From the debug menu it is possible to set breakpoints and step in the code. It is also possible to right click in the left edge of the code editor to set breakpoints.

It is also possible to force variable values from the I/O mapping dialog.

3.8. Creating a boot project

When downloading an application to the CrossFire SX, it is automatically stored to flash memory. There is no need to create a boot project as on for instance Linux based systems.

3.9. Retain memory

CrossFire SX has a built in retain memory of 640 bytes. Retains will be automatically stored to FRAM by a low priority background task, so it is necessary to not overload the system to make this work. Writing to the FRAM is never done more often than every 10ms, independent on how often they are changed.

To make sure that variables are located in the retain segment, they should be declared with the VAR RETAIN keywords.

To make sure that the retain memory is not made corrupt at a sudden power loss; a double buffering algorithm is used. This means that if a sudden power loss happens, only the latest changes will be lost.

The retain memory is protected by a crc checksum checked at start up. If the checksum is not correct, the retain memory will be cleared. It is recommended that the user application uses a magic byte or checksum to be able to detect a cleared persistent memory and to set default values for all variables stored in retain memory.

Writing retains is a relatively slow operation (roughly 12ms for 256 bytes). Only values that have been changed will be written. It is therefore recommended to not change variables located in retain memory continuously if not necessary. It is also recommended to not put variables in retain memory that does not need to be there. The FRAM can handle almost unlimited writes so the risk of corrupting the FRAM by writing to much is minimal.

3.10. Task configuration

CrossFire SX supports two cyclic tasks and three freewheeling tasks up to a total of four tasks. Cyclic tasks are executed by a timer at a very precise interval. Cyclic tasks also have a configurable priority and might abort each other if a higher priority cyclic task needs to run. Freewheeling tasks execute when there is no cyclic task to execute.

It is very important to configure the tasks in a correct way to get good performance and not to overload the processor. Make sure to not load the processor to 100% by your own tasks as there is also background job that needs to be done. Background job mainly consists of writing persistent data to FRAM but also to perform internal diagnostics of the unit. The CODESYS task monitor is very good to be able to see how the tasks execute on the target.

To make sure that your tasks perform well, it is recommended to use the CODEYS watchdog to supervise your tasks. By doing this you can take appropriate action if your tasks are blocked or is not running as fast as expected. Appropriate action might be to send an emergency CAN message or shut down the outputs.

There is also a watchdog supervising that the background task gets to run (idle task watchdog). This watchdog is also adjustable by the user through the IO driver.

The min cycle time that is possible to set for a cyclic task is 10ms.

3.11. Memory configuration

3.11.1. FLASH

The CrossFire SX has 3MB FLASH MEMORY divided into two banks. The first bank is reserved for the interrupt vectors, the bootloader and the CODESYS runtime itself and the second bank is used for the CODESYS user application.

Bank	Size
Bank 0 (interrupt vectors, bootloader and CODESYS runtime)	1572864 bytes
Bank 1 (user application)	1572864 bytes
TOTAL	3145728 bytes

3.11.2. RAM

The CrossFire SX RAM memory is used according to the following

Section	Size
Stack	13824 bytes

RAM (HEAP and general readwrite data)	174592 bytes
IEC data area (inputs, outputs, memory, globals)	73088 bytes
Retains	640 bytes
TOTAL	262144 bytes

The division of the IEC data area is done in the CF_SX_V3.5.10.0.devdesc.xml file. The standard division is according to the following.

Block	Size
Input data	384
Output data	384
Memory	4096
Globals	

4. Technical support

Additional sources of information are available on the CrossControl support site:
https://crosscontrol.com/support/
You will need to register to the site in order to be able to access all available information

Contact your reseller or supplier for help with possible problems with your device. In order to get the best help, you should have access to your device and be prepared with the following information before you contact support.

- The part number and serial number of the device, which you can find on the brand label.
- Date of purchase, which can be found on the invoice.
- The conditions and circumstances under which the problem arises.
- Status indicator patterns (i.e. LED blink pattern).
- Prepare a system report on the device, using CCSettingsConsole (if possible).
- Detailed description of all external equipment connected to the unit (when relevant to the problem).

5. Trademarks and terms of use

All trademarks sighted in this document are the property of their respective owners.

CrossFire™ SX is a trademark which is the property of CrossControl AB. CiA is a registered trademark which is the property of CAN in Automation.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

CC Linux is an officical Linux distributuition pursuant to the terms of the Linux Sublicense Agreement

Microsoft® and Windows® are registered trademarks which belong to Microsoft Corporation in the USA and/or other countries.

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Qt is a registered trademark of The Qt Company Ltd. and its subsidiaries.

CrossControl is not responsible for editing errors, technical errors or for material which has been omitted in this document. CrossControl is not responsible for unintentional damage or for damage which occurs as a result of supplying, handling or using of this material including the devices and software referred to herein. The information in this handbook is supplied without any guarantees and can change without prior notification.

For CrossControl licensed software, CrossControl grants you a license, to under CrossControl's intellectual property rights, to use, reproduce, distribute, market and sell the software, only as a part of or integrated within, the devices for which this documentation concerns. Any other usage, such as, but not limited to, reproduction, distribution, marketing, sales and reverse engineering of this documentation, licensed software source code or any other affiliated material may not be performed without the written consent of CrossControl.

CrossControl respects the intellectual property of others, and we ask our users to do the same. Where software based on CrossControl software or products is distributed, the software may only be distributed in accordance with the terms and conditions provided by the reproduced licensors.

For end-user license agreements (EULAs), copyright notices, conditions, and disclaimers, regarding certain third-party components used in the device, refer to the copyright notices documentation.