Xtra Combinations

Building Automation

Industrial Automation

Systems

Hardware, Engineering and Functional Description

Control system XControl
CPU module XC-CPU101...(-XV)

08/03 AWB2724-1453GB

MOELLER

Think future. Switch to green.

For Immediate Delivery call KMParts.com at (866) 595-9616

All brand and product names are trademarks or registered trademarks of the owner concerned.

1st published 2002, edition date 06/02
2nd edition 10/2002
3rd edition 04/2003
4th edition 08/2003
See revision protocol in the “About this manual” chapter

© Moeller GmbH, 53105 Bonn

Authors: Werner Albrecht
Editor: Thomas Kracht
Translator: David Long

All rights reserved, including those of the translation.

No part of this manual may be reproduced in any form (printed, photocopy, microfilm or any other process) or processed, duplicated or distributed by means of electronic systems without written permission of Moeller GmbH, Bonn.

Subject to alteration without notice.

For Immediate Delivery call KMParts.com at (866) 595-9616

Warning!
Dangerous electrical voltage!

Before commencing the installation

- Disconnect the power supply of the device.
- Ensure that devices cannot be accidentally restarted.
- Verify isolation from the supply.
- Earth and short circuit.
- Cover or enclose neighbouring units that are live.
- Follow the engineering instructions (AWA) of the device concerned.
- Only suitably qualified personnel in accordance with EN 50110-1/-2 (VDE 0105 Part 100) may work on this device/system.
- Before installation and before touching the device ensure that you are free of electrostatic charge.
- The functional earth (FE) must be connected to the protective earth (PE) or to the potential equalisation. The system installer is responsible for implementing this connection.
- Connecting cables and signal lines should be installed so that inductive or capacitive interference does not impair the automation functions.
- Install automation devices and related operating elements in such a way that they are well protected against unintentional operation.
- Suitable safety hardware and software measures should be implemented for the I/O interface so that a line or wire breakage on the signal side does not result in undefined states in the automation devices.
- Ensure a reliable electrical isolation of the low voltage for the 24 volt supply. Only use power supply units complying with IEC 60364-4-41 (VDE 0100 Part 410) or HD 384.4.41 S2.
- Deviations of the mains voltage from the rated value must not exceed the tolerance limits given in the specifications, otherwise this may cause malfunction and dangerous operation.
- Emergency stop devices complying with IEC/EN 60204-1 must be effective in all operating modes of the automation devices. Unlatching the emergency-stop devices must not cause restart.
- Devices that are designed for mounting in housings or control cabinets must only be operated and controlled after they have been installed with the housing closed. Desktop or portable units must only be operated and controlled in enclosed housings.
- Measures should be taken to ensure the proper restart of programs interrupted after a voltage dip or failure. This should not cause dangerous operating states even for a short time. If necessary, emergency-stop devices should be implemented.
- Wherever faults in the automation system may cause damage to persons or property, external measures must be implemented to ensure a safe operating state in the event of a fault or malfunction (for example, by means of separate limit switches, mechanical interlocks etc.).

For Immediate Delivery call KMParts.com at (866) 595-9616

For Immediate Delivery call KMParts.com at (866) 595-9616

Contents

About this manual 3
Modification index 3
Assembly 3
Abbreviations and Symbols 3

1 Layout of the XC100 5
CPU with PSU and local inputs/outputs 5
24 V PSU with local inputs/outputs 5
– Function 5
– Assembly 6
CPU 7
– Function 7
– Use of the CPU types 7
– Assembly 7
– Battery 11
CPU installation 11
Detaching the CPU 11

2 Engineering 13
Control panel layout 13
– Ventilation 13
– Layout of units 13
Preventing interference 13
– Suppressor circuitry for interference sources 13
– Shielding 13
Lighting protection 14
Wiring example for the supply section 14

3 CPU operation 15
Switch-on behaviour 15
Switch-off behaviour 15
Start behaviour 15
Stop behaviour 15
Cold start 15
Warm start 15
Test and commissioning 15
– Breakpoint/single-step mode 16
– Single-cycle mode 16
– Forcing 16
– State display of the XSoft 16
Programreset 16
– Warm reset 16
– Cold reset 16
– Full reset 16
Program parameterization 16
– Maximum program cycle time 16
– Start behaviour at Power-On 16
Program processing and system time 16
Cycle-time monitoring 16
Function blocks and functions 17
Direct peripheral access 18
– Functions 19
– Error code with “direct peripheral access” 22

For Immediate Delivery call KMParts.com at (866) 595-9616

Updating the operating system 23
Interrupt processing 24
– Creating and integrating an interrupt function 26
– Single Step operation with system events 28
Browser commands 29
File access on the file system of the Multi Media Card 29
Data remanence 29
Program transfer 30
Operating states 30

4 Establishing a PC – XC100 connection 31
Establishing a connection via the RS 232 interface (XC100) 31
– Programming cable 31
– XSoft software 31

5 Create sample project 33
Task 34
– Procedure 34
CANopen bus expansion 43
– Controller configuration 43
– Create program for decentralised inputs/outputs on the CANopen bus 49
Routing 50
– From programming PC to CANopen fieldbus slave 50
– Routing in a CANopen network 51
– RS 232 interface in transparent mode 52
– Automatic closing of the interface 57

Appendix 59
Technical data 59

Index 63

For Immediate Delivery call KMParts.com at (866) 595-9616

About this manual

Modification index

Edition date	Page	Subject	New	Modified	Omitted
10/02	60	"External filter: If required"			✓
04/03	15	"Warm start"		✓	
	18	"Direct peripheral access"	✓		
	23	"Updating the operating system"	✓		
	24	"Interrupt processing"	✓		
	29	"File access on the file system of the Multi Media Card"	✓		
	29	"Browser commands"	✓		
	43	"CANopen bus expansion"		✓	
	50	"Routing"	✓		
08/03	in general	baud rate has changed from 57600 to 38400		✓	
	16	"State display of the XSoft"	✓		
	24	"Communication error"	✓		
	28	"Single Step operation with system events"	✓		
	31	"Communication Error (#0): Logout Performed"	✓		
	59	"Battery (lifetime)"		✓	

Assembly

The XC-CPU101-... controllers – referred to below simply as XC100 – have been designed for application in machinery and plant control systems. These controllers are fitted with interfaces for connecting to a programming device (RS 232) and for linking to decentralized CANopen expansion units, so they can form the core of a comprehensive automation system.

The manual is laid out as follows:

The chapter "Hardware and installation" tells you about installing the controllers, project engineering, and the settings that you can make on the controllers.

Communication with a controller is established by connecting it to a PC, via the programming interface. Configuration of the XC100 is described in Chapter "Establishing a PC – XC100 connection".

This is followed by examples showing you how to use the XSoft software. You can learn about configuration and programming. After downloading it, you can test out the project → chapter "Create sample project".

Abbreviations and Symbols

The abbreviations and symbols used in this manual have the following meanings:

MWS	Menu selector switch
BAS	Operating mode switch
CPU	Central processing unit
CRC	Cyclic redundancy check
MMC	Multimedia card
I/O	Inputs/outputs

▶ Indicates instructions on what to do

Selecting ‹File → New› means: activate the command "New" in the "File" menu.

Note!
Warns of a hazardous situation that could result in damage to the product or components.

Caution!
Indicates the risk of major damage to property, or slight injury.

Warning!
Indicates the risk of major damage to property, or serious or fatal injury.

In order to provide a clear layout, the chapter title is shown in the header on left-hand pages, and the current section on right-hand pages. Exceptions are at the first pages of chapters and empty pages at the end of chapters.

For Immediate Delivery call KMParts.com at (866) 595-9616

For Immediate Delivery call KMParts.com at (866) 595-9616

1 Layout of the XC100

The XC100 controller has a compact design, and can be fitted with either local or decentralized expansion. The basic unit consists of:

- Module rack
- A CPU for control or visualization, with integral power supply and local inputs/outputs.
- XIOC signal modules.

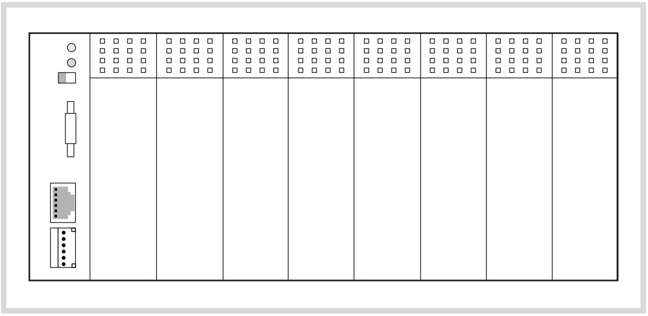


Figure 1: Layout of the XC-CPU101 with XIOC modules

→ Further details about the CPU can be found in the next section.

Detailed information about the module racks and XIOC modules can be found in the manual "Hardware and Engineering, XIOC Signal modules". This manual is provided as a PDF file (h1452g.pdf) on the CD.

The latest versions of specific manuals can be found at http://www.moeller.net/support:
Search item: AWB2725-1452GB)

CPU with PSU and local inputs/outputs

The CPU module of the XC100 has a compact design that is divided into two functional units:

- Processor unit with interfaces
- 24 V PSU with integral digital inputs (eight) and digital outputs (six).

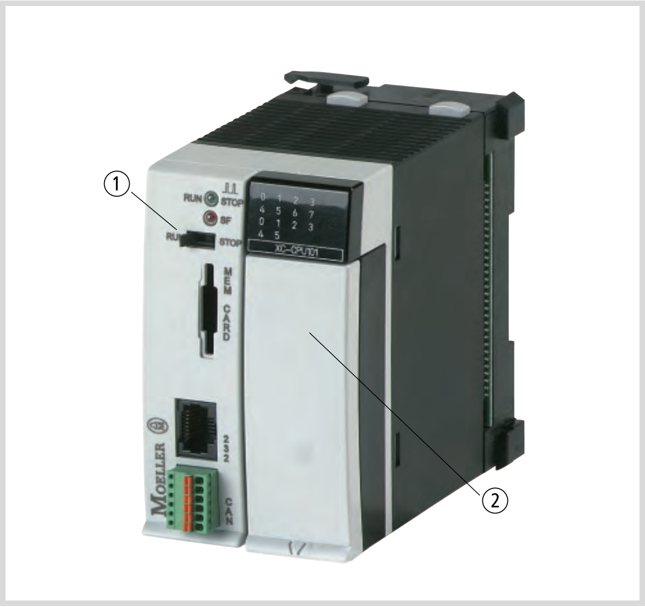


Figure 2: Assembly of the CPU module XC-CPU101

① Processor unit

② 24 V PSU with local inputs/outputs

24 V PSU with local inputs/outputs

The power supply unit provides the operating voltages required by the processor unit and the inputs/outputs (local and decentralized).

Function

The power supply unit converts the 24 V DC supply voltage into +5 V DC and 3.3 V DC. These voltages are fed to the bus on the basic rack unit and any expansion rack units that are present.

The special feature of connection to the 24 V supply voltage is that the processor unit and the local inputs/outputs can be fed separately. One 24 V connection is provided for the processor unit (labelled: 24V/0V) and another 24 V connection for the local inputs/outputs (labelled: 24VQ/0VQ).

For Immediate Delivery call KMParts.com at (866) 595-9616

Assembly

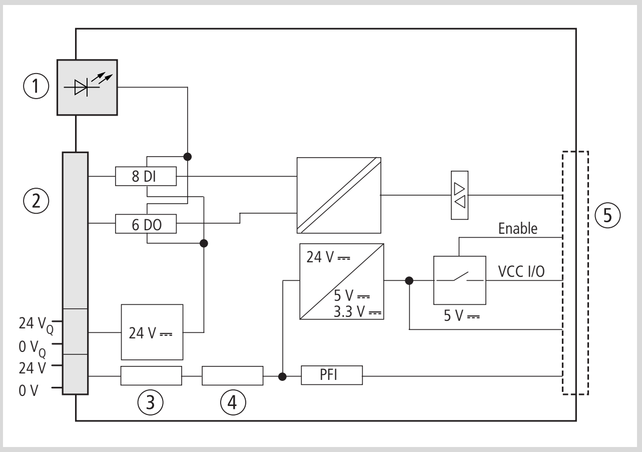


Figure 3: Block diagram: power supply unit

① Status indicator for I/Os
② Front connection terminals
③ Internal filter
④ Buffer
⑤ XIOC I/O-bus, module rack
PFI = Power Fail Interrupt

The voltage connection $0V_Q/24V_Q$ is only for the supply voltage to the integral local inputs (8) and outputs (6), and is electrically isolated from the bus.

The 0V/24V connection is internally filtered and buffered, and then fed to a voltage converter. This converter produces 5 V and 3.3 V outputs as system operating voltages. The internal power supply for the 5 V system voltage is so designed that it can provide (typically) 1 A for the processor as well as 3.2 A for the I/O section.

If there is a break or collapse of the 24 V supply (threshold is about 10 V) then a power-down logic switches of the 5 V supply to the signal modules (central I/O). The sequence is initiated by the PFI signal and leads to a power-down through the CPU.

Local digital inputs

The 18-pole terminal block for the voltage supply and for connection of the local inputs and outputs is located on the right half of the CPU behind the front cover. The eight inputs and six outputs are designed for 24 V DC digital signals.

Outputs 0.0 to 0.3 can be loaded by up to 500 mA and outputs 0.4 and 0.5 can be loaded by up to 1 A (each), with an ON ratio (duty factor) of 100 % and a utilization factor of 1.

The outputs are short-circuit proof. A short-circuit state should, however, not be permitted to exist over a longer period.

Terminal assignments

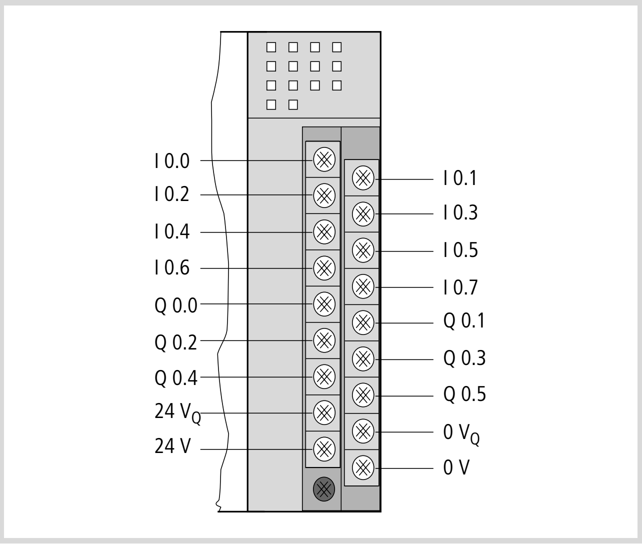


Figure 4: Connections for PSU and local I/O

I0.0 to I0.7: local digital inputs
Q0.0 to Q0.5: local digital outputs
$0V_Q/+24V_Q$: supply voltage for the local inputs/outputs
0V/+24V: supply voltage to the processor unit

LED indicators

The LEDs indicate the signal status for the inputs and outputs. An LED that is ON indicates a H-level signal on the corresponding terminal.

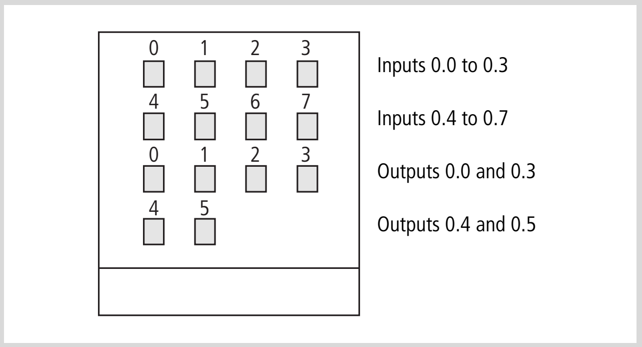


Figure 5: LEDs for the integral inputs/outputs

The two upper rows of LEDs show the signal status for the eight digital inputs of the CPU module (I0.0 to I0.7), and the two lower rows show the signal status for the six digital outputs (Q0.0 to Q0.5).

For Immediate Delivery call KMParts.com at (866) 595-9616

CPU

The XC-CPU101...(-XV) types of CPU are based on a C164 processor with an integrated CAN interface, and include battery-buffered flash and SRAM memories. The CAN fieldbus interface is electrically isolated. A battery is required for the operation of the data-saving function.

The monitoring of the system voltage ensures that the data-saving routine will be initiated if the voltage goes below a fixed preselected level. In order to ensure that the stored energy required for the data-saving routine is not used up by I/O activities, the 5 V system voltage for the I/O modules is switched off.

The internal real-time clock facilitates time and date dependent control functions.

The available operating and interface control devices are:

- LED display for RUN/Stop and general error
- Operating-mode selector switch RUN/Stop
- RS 232 interface, e. g. for connecting a programming device (XC-CPU101)
- CANopen interface, e. g. as a fieldbus interface
- Interface for a multimedia memory card (MMC).

The CPUs for XC100 controllers are available in various different versions:

- XC-CPU101-C64K-8DI-6DO (-XV)
- XC-CPU101-C128K-8DI-6DO (-XV)

C64K and C128K are designations for the size of the user memory.

"XV" designates a visualisation CPU, and permits the direct connection to and control of a text display (XV-101).

According to the size of the application program, the following memory values apply:

	XC-CPU101-....(-XV)	
	C64K-8DI-6DO	**C128K-8DI-6DO**
Program code	64 kByte	128kByte
Program data, of which:	64 kByte	128kByte
markers	4 kByte	8 kByte
retained data	4 kByte	8 kByte

The XC-CPU...-XV types have an additional 64 kByte flash memory for text.

Function

The task of the CPU is to generate output signals from the incoming local and central/decentralized signal, in accordance with the application program.

Input/output signal can be, for instance:

- digital or analog signals
- commands from the text display1)
- output to the text display1)
- connections to the programming system
- connections to the CANopen bus interface
- connections to fieldbus modules, if present
- connections to intelligent signal modules, if present.

1) Only with XC-CPU...-XV

Use of the CPU types

	XC100	**Text display XV-101-...**	
CPU types		**K42**	**K84**
XC-CPU101...	✓	–	–
XC-CPU101...(-XV)	✓	✓	✓

Assembly

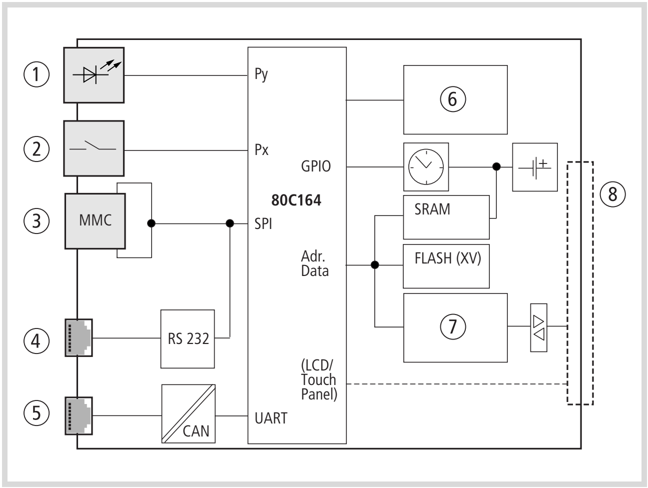


Figure 6: Block diagram of the XC-CPU101

① State indication RUN, Stop, SF
② Operating mode selector switch
③ Multimedia card
④ Programming device interface: RS 232 on XC-CPU101
⑤ CANopen fieldbus interface
⑥ Voltage monitoring
⑦ I/O bus interface
⑧ XIOC I/O bus (on module rack)

For Immediate Delivery call KMParts.com at (866) 595-9616

LED status indicator

→ chapter "Operating states" on Page 30.

Operating mode selector switch

The operating modes "Stop" and "Run" are selected by a rocker switch at the front of the CPU module. Please note that the position of the operating mode selector switch sets the behaviour of the CPU. The effectiveness of the software settings depends on the position of the operating mode selector switch. If the selector switch is changed to the "Stop" position while the equipment is in the "Run" mode, then the CPU will switch from the operating mode "Run" to the "Stop" state at the end of the cycle that is currently running.The position of the operating mode selector switch is polled at the end of each cycle, and the controller switches to the selected state, → chapter "CPU operation".

Memory Card/Multimedia Card (MMC)

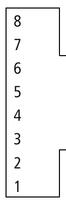
Important
The MMC may not be inserted or removed during operation.

The FLASH multimedia card is used as an optional backup medium for the application program and to save recipe data. The operating system supports memory capacities up to a maximum of 128 MByte. At present, Moeller offers MMCs in the sizes 16 and 32 MByte, with the type designations XT-MEM-MM16M and XT-MEM-MM32M. To write data to the multimedia card, just plug it into the corresponding card slot in the CPU. Use the command "create boot project" to transfer the program to the MMC.

Programming device interface

XC-CPU101: the CPU is fitted with an RS 232 interface. Communication between the controller and the programming device takes place through this interface. Physically, the interface is an RJ 45 socket. This means that normal commercial RJ 45 connectors or Ethernet patch cables can be used. This interface is not electrically isolated.

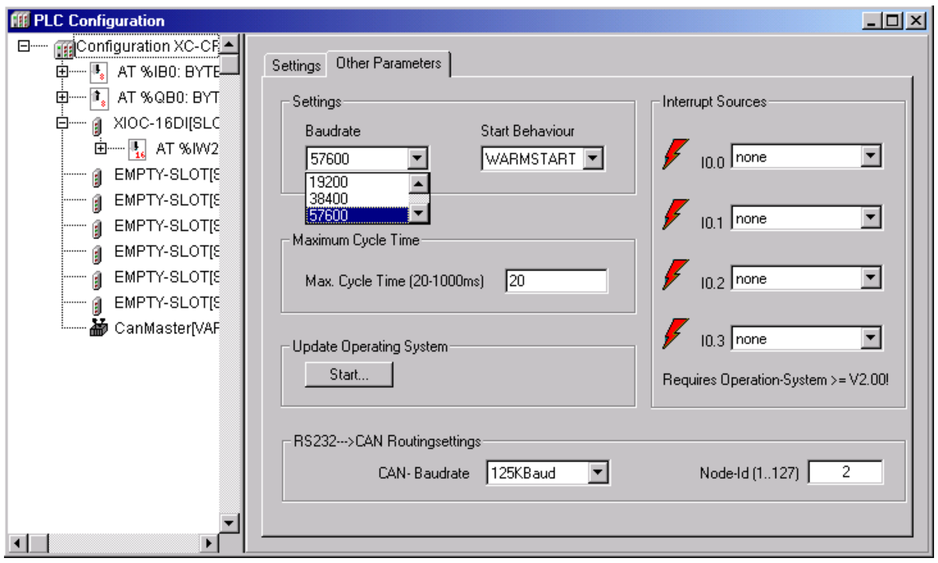
The interface assignments are as follows:



	RS 232
8	RxD
7	GND
6	–
5	TxD
4	GND
3	–
2	–
1	–

Programming device interface – altering the data transmission rate

▶ Open the ‹Resources → PLC Configuration› dialog field.
▶ Activate the "Other parameters" tab.
▶ Select the required data transfer rate in the "Baudrate" list field. In the example, this is 38400 kBit/s.



Settings
Baudrate: 57600 (19200, 38400, 57600)
Start Behaviour: WARMSTART

Maximum Cycle Time
Max. Cycle Time (20-1000ms): 20

Update Operating System

Interrupt Sources
I0.0: none
I0.1: none
I0.2: none
I0.3: none
Requires Operation-System >= V2.00!

RS232--->CAN Routingsettings
CAN- Baudrate: 125KBaud
Node-Id (1..127): 2

Figure 7: Controller configuration – "Other parameters"

▶ Close the "Other Parameters" window.
▶ Select the menu ‹Online → Login›.

For Immediate Delivery call KMParts.com at (866) 595-9616

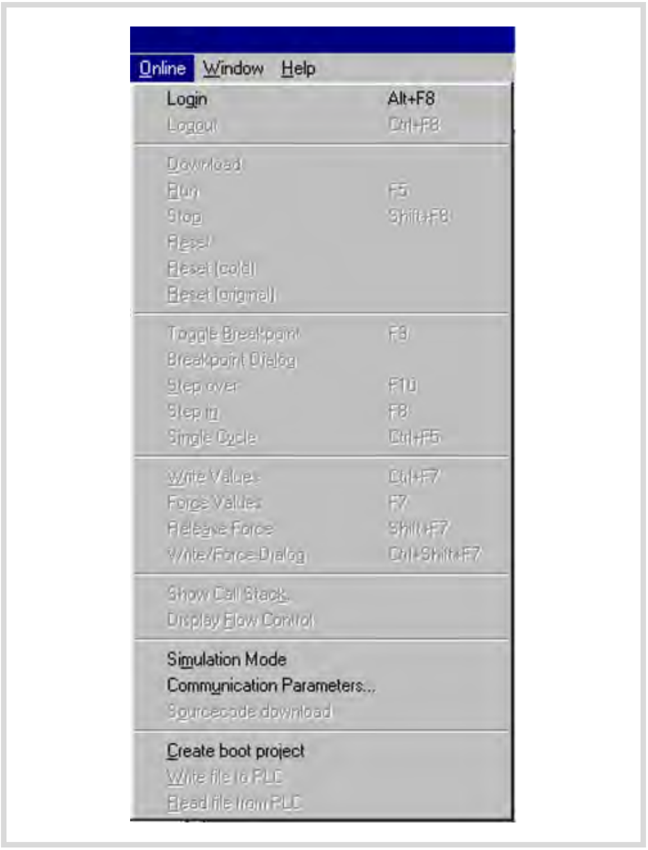
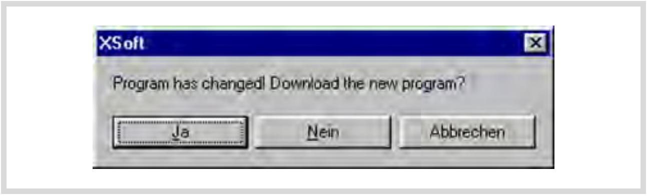


Figure 8: Menu “Online”

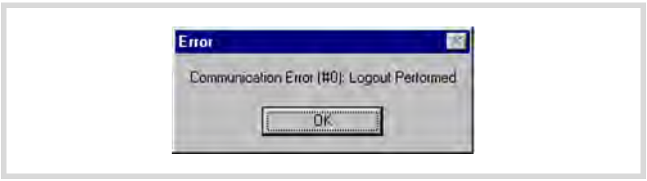
The query as illustrated in (→ figure 9) appears:



Program has changed! Download the new program?

Figure 9: Query concerning program change

- ▶ If you answer this query with “Yes”, you will see the error message shown below for a communication error, because the baud rates for XC100 and XSoft do not match. The next steps show you how to set the baud rate.

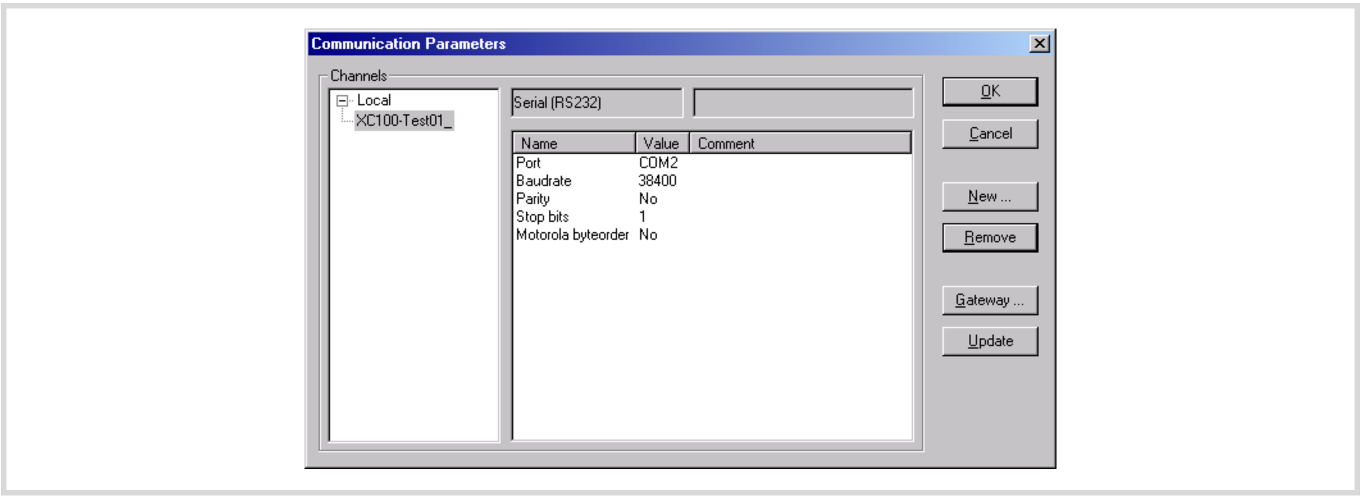


Communication Error (#0): Logout Performed

Figure 10: Communication error

- ▶ Acknowledge the error message, with “OK”.
- ▶ Select the menu ‹Online → Communication parameters› (→ figure 8).

Now you will see the “Communication window”, as shown in the next diagram.



Name	Value	Comment
Port	COM2	
Baudrate	38400	
Parity	No	
Stop bits	1	
Motorola byteorder	No	

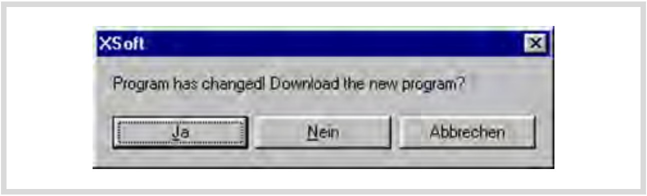
Figure 11: Communication parameters

- ▶ Use a double-click to select the field with the preset baud rate.

This field now has a grey background.

- ▶ Select the required baud rate, e. g. 38400 Bit/s, with further clicks in this field. Confirm with “OK”.
- ▶ Select the menu ‹Online → Log-in› again.

Once again, you will see the following message:



Program has changed! Download the new program?

Figure 12: Query concerning program change

- ▶ Again, answer this query with “Yes”.

For Immediate Delivery call KMParts.com at (866) 595-9616

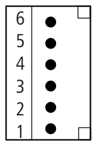
▶ Select the menu ‹Online → Start› (→ figure 8). This puts the controller into the RUN mode.

The subsequent communication between the XC100 and the PC (as the programming device) will be made at the selected transmission rate.

CANopen interface

The CPUs can be connected to the CANopen bus via the electrically isolated ISO-11898 interface.

The connector assignments are as follows:



Terminal	Signal
6	GND
5	CAN_L
4	CAN_H
3	GND
2	CAN_L
1	CAN_H

Connector type: 6-pole, plug-in spring-loaded terminal block, conductor cross-section up to 0.5 mm^2

The CPUs can be operated on the CAN bus either as the network (NMT) master or as the NMT slave.

If the CANopen link is broken, the inputs to the disconnected devices (receive date) will be interpreted as "0" level signals by the PLC, and the outputs controlled by the PLC will be reset. The devices which are still linked to the PLC will still be polled and/or controlled.

Power supply:
Arrange the supply power for the equipment so that the decentralized participants on the CANopen link are switched on at the same time or **before** the control unit is switched on. This avoids errors in starting up the CANopen link.

Start/Stop behaviour:
If you set the operating mode selector to the "Stop" position, all outputs of the decentralized devices will be set to the "0" level.

Bus termination resistors:
The ends of the network link must be terminated with 120 Ω bus termination resistors:

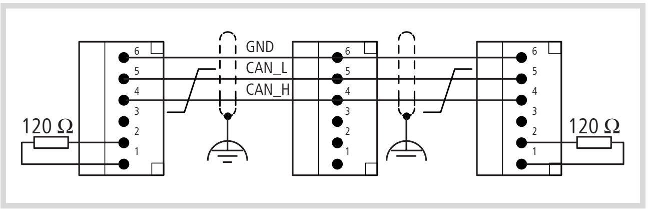


Figure 13: Possible configuration of a CANopen bus with bus termination resistors

Terminals 1 and 4 , 2 and 5 , 3 and 6 are internally connected.

Use only cable that is approved for CANopen application, with the following characteristics:

- Characteristic impedance 100 to 120 Ω
- Capacitance < 60 pF/m

Recommended cable:
from LAPPKABEL
CAN cable to ISO 11898
Type: 2170204T; 2 × 2 × 0.22
UNITRONIC-BUS LD

Baud rate [kbits/s]	F Length [m]	Core cross-section [mm^2]	Loop resistance
20	1000	0.75 to 0.80	16 Ω/km
125	500	0.50 to 0.60	40 Ω/km
250	250	0.50 to 0.60	40 Ω/km
500	100	0.34 to 0.60	60 Ω/km
1000	40	0.25 to 0.34	70 Ω/km

The assignment of the I/O address area (%IB, %QB) is laid down in the controller configuration.

Real-time clock

The XC100 is equipped with a real-time clock. This can be accessed via function blocks in the application program. The available clock functions are:

- GETREALTIMECLOCK (evaluation of the real-time clock)
- SETREALTIMECLOCK (set the real-time clock)

→ The function blocks are described in the separate manual "XSoft Function Blocks" (AWB2786-1456). This manual is provided as a PDF file (h1456g.pdf) on the CD.

The latest version of the manual can be found at http://www.moeller.net/support:
Search text: AWB2786-1456GB

XC-CPU101-...-XV

The XC-CPU101-...-8DI-6DO- XV units are equipped with an expandable operating system. The functionality of the system permits operation of these CPUs with text displays from the XV-101-.. series.

→ The text displays are described in the separate manual "Hardware and Engineering" (AWB2726-1461GB). The manual AWB2726-1461GB (description of the text displays) is also available on the CD, as a PDF file (h1461g.pdf).

The latest version of the manual can be found at http://www.moeller.net/support:
Search text: AWB2726-1461GB or AWB2726-1462GB)

For Immediate Delivery call KMParts.com at (866) 595-9616

Battery

A lithium battery, type 1/2 AA (3.6 V) is used for data-saving. The battery compartment can be found on the left side of the CPU unit, behind a cover plate. The charge level of the battery is monitored. If the battery voltage falls below a fixed preset level, then a general error message will be generated.

The battery buffer times are:

- Worst-case: 3 years continuous buffering
- Typical: 5 years of continuous buffering

▽ **Important**
To avoid loss of data, the battery must be changed when the low threshold level has been reached.

CPU installation

→ Detailed information about the installation of the module racks and XI/OC modules can be found in the manual "Hardware and Engineering XI/OC Signal Modules" (AWB2725-1452GB). This manual is provided as a PDF file (h1452g.pdf) on the CD. Here you can also find further information on the various types of module rack and the individual slot assignments for the CPU and the XI/OC signal modules.

The latest versions of specific manuals can be found at http://www.moeller.net/support:
Search item: AWB2725-1452GB

- ▶ Insert the loop on the bottom of the CPU module into the hole in the module rack $\boxed{1}$.
- ▶ Press the top of the CPU module onto the module rack, until you hear it click into position $\boxed{2}$.

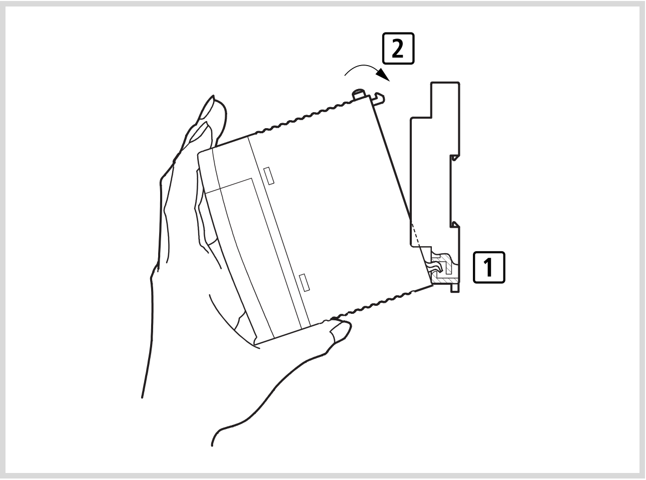


Figure 14: CPU installation

Detaching the CPU

- ▶ Press in the catch $\boxed{1}$.
- ▶ Keep the catch pressed in, and pull the top of the CPU module forwards $\boxed{2}$.
- ▶ Lift up the CPU module and remove it $\boxed{3}$.

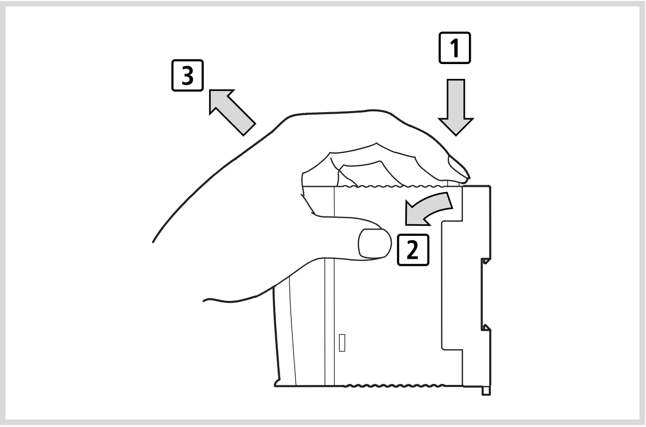


Figure 15: Detaching the modules

For Immediate Delivery call KMParts.com at (866) 595-9616

For Immediate Delivery call KMParts.com at (866) 595-9616

2 Engineering

Control panel layout

The layout of the components inside the switchgear cabinet is a major factor for achieving interference-free functioning of the plant or machinery. During the project planning and design phase, as well as its implementation, care must be taken that the power and control sections are separated. The power section includes:

- Contactors
- Coupling/interfacing components
- Transformers
- Frequency inverters
- Converters

In order to effectively exclude any electromagnetic contamination, it is a good idea to divide the system into sections, according to their power and interference levels. In small switchgear cabinets it is often enough to provided a sheet steel dividing wall, to reduce interference factors.

Ventilation

A clear space of at least 5 cm must be kept between passive components, to ensure adequate ventilation. If the neighbouring components are active elements, such as power supplies or transformers, then the minimum spacing should be 7.5 cm. The values that are given in the technical data must be observed.

Layout of units

Build the module racks and the controls into the switchgear cabinet in a horizontal position:

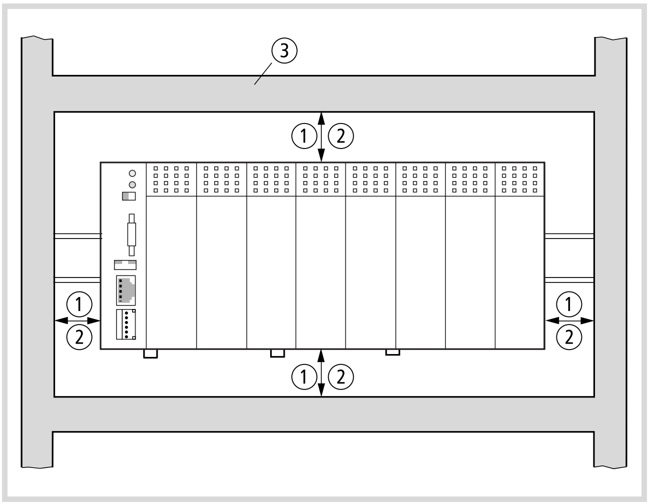


Figure 16: Cabinet layout

① Spacing > 50 mm

② Spacing > 75 mm to active elements

③ Cable duct

Preventing interference

Cable routing and wiring

Cables are divided into the following categories:

- Power cables (e. g. cables that carry high currents, or cables to converters, contactors, solenoids)
- Control and signal cables (e. g. for digital inputs)
- Measurement and signal cables (e. g. for fieldbus connections)

→ Always route power cables and control cable as far apart as is feasible. This avoids capacitive and inductive coupling. If separate routing is not possible, then the first priority must be to shield the cable responsible for the interference.

Take care to implement proper cable routing both inside and outside the switchgear cabinet, to keep interference as low as possible:

- ▶ Avoid parallel routing of sections of cable in different power categories.
- ▶ As a basis rule, keep AC cables separated from DC cables.
- ▶ Keep to the following minimum spacing:
  - – between power cables and signal cables – at least 10 cm;
  - – between power cables and data or analog cables – at least 30 cm.
  - – When routing cables, take care that both out and return leads of a circuit pair are kept together. The currents flowing in opposite directions thus cancel each other out as a summation, and the electromagnetic fields cancel each other out.

Suppressor circuitry for interference sources

- ▶ All suppressor circuitry should be wired in as close to the source of interference (contactors, relays, solenoids) as possible.

→ Switched inductors should always have suppressor circuitry fitted.

Shielding

- ▶ Use shielded cables for the connections to the data interfaces. The general rule is: the lower the coupling impedance, the better the shielding effect.

For Immediate Delivery call KMParts.com at (866) 595-9616

Lighting protection

External lightning protection

All cables that go outside buildings must be shielded. Metal conduit is the best solution to this problem. For signal cables, use overvoltage protection devices, such as varistors or other similar devices. Install the protection devices as close as possible to the cable entry into the building, at the latest there were the cable enters the switchgear cabinet.

Internal lightning protection

Internal lightning protection covers all those measures taken to reduce the effects of a lightning strike and the resulting electrical and magnetic fields on metallic installation and electrical plant. These measures are:

- equipotential bonding/earthing
- shielding
- using overvoltage protection devices.

Please consult the following manuals for advice on cable routing and shielding measures:

- AWB27-1287 “EMC Project Engineering for Automation Systems”.
- TB27-001-GB “Electromagnetic Compatibility (EMC) for Automation systems”.
- TB02-022-GB “Electromagnetic Compatibility (EMC) for Machinery and Plant”.

Wiring example for the supply section

→ You can find wiring examples for the XI/OC modules in the manual “Hardware and Engineering, XI/OC signal Modules” (AWB2725-1452GB). This manual is provided as a PDF file (h1452g.pdf) on the CD.

The latest version of the manual can be found at http://www.moeller.net/support:
Search text: AWB2725-1452GB.

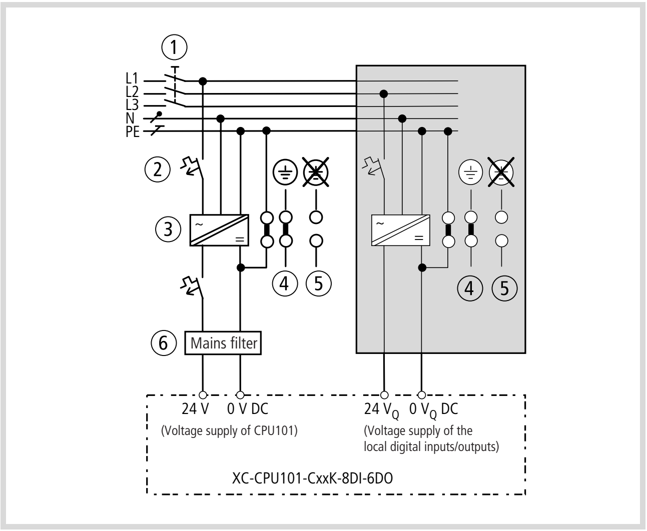

①
L1
L2
L3
N
PE
②
③
④
⑤
⑥
Mains filter
24 V
0 V DC
(Voltage supply of CPU101)
24 V_Q
0 V_Q DC
(Voltage supply of the local digital inputs/outputs)
XC-CPU101-CxxK-8DI-6DO

Figure 17: Wiring example for the supply section

① Main switch
② Protective cut-out
③ 24 V DC supply voltage
④ Earthed operation
⑤ In floating (i. e. unearthed) operation, an isolation monitor must be used (IEC 204-1, EN 60204-1, DIN EN 60204-1)
⑥ Line filter (Manufacturer: e. g. Schaffner); Observe power consumption!

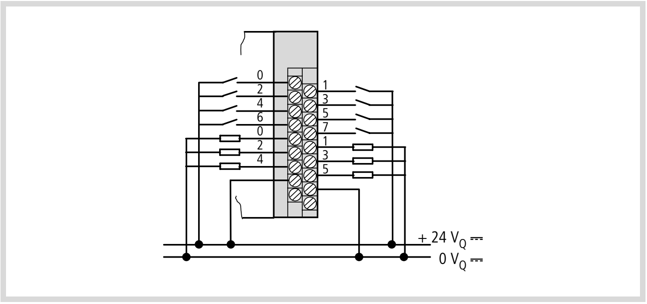


Figure 18: Example for wiring the terminal block

The wiring example indicates the wiring of a separate voltage supply for inputs/outputs.

For Immediate Delivery call KMParts.com at (866) 595-9616

3 CPU operation

Switch-on behaviour

After the supply voltage has been switched on, the CPU performs the necessary self-testing routine and CRC checks. When these have been successfully concluded, the run-time system is started. If the CRC check or one of the self-tests generates an error, then the CPU stops in the state "Switch-on not OK", → chapter "Operating states" on Page 30.

The run-time system manages the communication with the XSoft programming system, the execution of the application program, and its debugging. It can only support the execution of one application program. No application program can be run while the system update is being carried out.

After the run-time system has started, a check is made whether an application program is stored on the memory card. An application program on the memory card always has priority over one stored in the SRAM. If application programs are present on both the memory card and in the SRAM, then the CRC check investigates whether these two programs are identical. If they are different, then the program on the memory card is loaded into the controller.

If there is no memory card plugged in, then the system checks whether there is an executable program stored in the SRAM. If there is no such program, then the controller stops and waits in the "NOT READY" state, see → chapter "Operating states" on Page 30.

If an executable program is stored in the SRAM, then it will be started, depending on the position of the operating mode selector switch and taking the parameter settings for the start behaviour into consideration (see Section "Start behaviour", below).

Switch-off behaviour

Switch-off (operating mode selector switch: Run → Stop) leads to an interruption of the program run the end of the cycle. Running of the program is ended immediately with a voltage dip (switching via the PFI signal). The outputs are switched off at the same time. When the supply voltage returns, the controller carries out a restart (see Section "Switch-on behaviour").

Start behaviour

The start behaviour of the controller depends on:

- the position of the local operating mode selector switch
- the parameter settings for start behaviour that were set in the XSoft programming system.

(With the CPU version "XV" (visualization CPU), it is basically possible to operate the system from the system menu in the display).

The position of the operating mode switch determines whether the operating states changes from "Stop" to"Run". This change cant be forced by a corresponding choice within the programming system.

When a program starts, a check is made whether the configured inputs and outputs match those that are actually present. A check is also made whether the module that was parameterized is physically present, or a different module. A module that is not present will not have any effect on the start of the program, but if the module is a different type, the start will be prevented.

When the application program starts, a distinction is made between (see also the following sections):

- Cold start
- Warm start

Stop behaviour

The processing of the application program always halts at the end of a program cycle.

Cold start

A cold start is only made for the first start, after the program has been loaded into the controller. During this start, all the program variables are set to their initialisation values and the program is started.

Warm start

All subsequent starts of the program that has been loaded are warm starts. The variables that were declared with "RETAIN" retain their values, the other variables are set to their initialisation values.

Test and commissioning

The PLC supports the following test and commissioning features:

- Breakpoint/single-step mode
- Single-cycle mode
- Forcing
- Online modification
- Status indication (Power Flow).

For Immediate Delivery call KMParts.com at (866) 595-9616

Breakpoint/single-step mode

Breakpoints can be set within the application program. If an instruction has a breakpoint attached, then the program will halt at this point. The following instructions then be executed in single-step mode. The cycle-time monitoring is deactivated.

⚠ **Caution!**
The outputs that were already set when the breakpoint occurred remain set!

Single-cycle mode

In single-cycle operation, one program cycle is performed in real time. The outputs are enabled during the cycle. At the end of the cycle, the output states are cancelled and the outputs are switched off. The cycle-time monitoring is active.

Forcing

All the variables in an application program can be forced to a given setting. If variables for physical outputs of the local I/Os are forced, they will only be connected through to the peripherals in the "Run" state.

State display of the XSoft

- The signal state of the physical, Boolean inputs are displayed in the "Stop" and "Start" modes.
- The signal state of the physical, Boolean outputs are only displayed in the "Start" (RUN) mode.
- The display for a low signal is displayed with "FALSE" and has a black background.
- The display for a high signal is displayed with "TRUE" and has a blue background.
- All other variables are only represented in "Start" mode with the respective current variable value.

Programreset

The application program can be reset to one of the following levels:

- Warm reset
- Cold reset
- Full reset

Warm reset

This correspond to the initialisation during a warm start, see Section "Warm start"on Page 15.

Cold reset

This correspond to the initialisation during a cold start, see Section "Cold start"on Page 15.

Full reset

The application program in the controller is completely deleted. After this, the controller is in the "NOT READY" state. The Boot project on the PLC will also be deleted.

Program parameterization

An application program has various parameters that can be set or adjusted in the programming system:

- Maximum program cycle time
- Start behaviour at Power-On
- Parameters for CAN Routing.

Maximum program cycle time

The maximum cycle time for the application program can be set, in the range from 20 ms to 1000 ms. The default value is 20 ms.

Start behaviour at Power-On

This setting defines how the controller should respond after switch-on, if an application program is present and the operating mode selector switch is in the "Run" position.

The following settings are available:

- WARMSTART (default setting)
- COLDSTART
- STOP.

Program processing and system time

The application program is processed cyclically. The states of the inputs are read before the start of each program cycle, and the output states are written to the outputs at the end of the cycle. In addition, all system activities carried out before or after the processing cycle.

Among these are:

- Communication with XSoft
- Online alterations
- Processing of the CANopen protocol stack, etc.

Task configuration is not available for the XC100 at present.

As a result of the software architecture of the run-time system, timing jitter may occur between individual processing cycles.

Cycle-time monitoring

The cycle-time monitoring monitors the cyclic task of the application program using a hardware timer. If the time exceeds the parameterized time, the outputs of the controller will be disconnected and the XC100 is put into the "Stop" state.

For Immediate Delivery call KMParts.com at (866) 595-9616

Function blocks and functions

The controller supports the following function blocks and functions:

- Standard Library (StdLib): IEC function blocks:
  - CTD
  - CTU
  - CTUD
  - F_TRIG
  - R_TRIG
  - RS
  - SR
  - SEMA
  - RTC
  - TOF
  - TON
  - TP

- Functions
  - CONTACT
  - DELETE
  - FIND
  - INSERT
  - LEFT
  - LEN
  - MID
  - REPLACE
  - RIGHT

The description of the functions and function blocks from the standard library (StdLib) can be found in the manual "XSoft" (AWB2700-1437D). This manual is provided as a PDF file (h1437g.pdf) on the CD.

The latest version of the manual can be found at http://www.moeller.net/support: Search text: AWB2700-1437GB

A supplementary RTCLib.liblibrary is provided for access to the real-time clock in the controller:

- RTCLib.lib: clock function blocks
  - GETREALTIMECLOCK
  - SETREALTIMECLOCK

The function blocks of the Visu.Lib file are available for visualization.

- Visu-Lib: Visualization blocks (applies only for XC-CPU...-XV)
  - ClearLine
  - ClearScreen
  - EnableDisplay
  - GetDisplayInfo
  - InputValue
  - SetBacklight
  - SetContrast
  - SetCursor
  - WriteBargraph
  - WriteDate
  - WriteDay
  - WriteLine
  - WriteMultiString
  - WriteMultiValue
  - WriteString
  - WriteTime
  - WriteValue
  - WriteStringIndex

The following Sucosoft-S40 standard function blocks are available in the XS_40FB.Lib file:

XS_40FB.Lib: standard function blocks from the S40 software

- Communication
  - MI4netDP16
  - MI4netDP32
  - MV4netDP38
  - MV4netDP70

- Convert
  - DataScale
  - IEEE_To_Real
  - Real_To_IEEE

- DateAndTime
  - DATContactX
  - DateConcat
  - DateSplit
  - DATSplitX
  - TimeConcatX
  - TimeSplitX
  - TODConcat
  - TODSplit

- Timer
  - MS_TimeFalling
  - MS_TimeRising
  - S_TimeFalling
  - S_TimeRising
  - TimeGenerator
  - TimePulse

The function blocks from the Counter.lib file are provided for access to the counter module XIOC-1(2)CNT-100kHz.

- Counter.lib file:
  - CounterControl
  - CounterFlags
  - ReadCounter
  - WriteCounter
  - XIOC_INCEncoder

You can find the descriptions of the function blocks in the manual "Function Blocks for XSoft" (AWB2786-1456GB). This manual is provided as a PDF file (h1456g.pdf) on the CD.

The latest version of the manual can be found at http://www.moeller.net/support: Search text: AWB2786-1456GB

For Immediate Delivery call KMParts.com at (866) 595-9616

Direct peripheral access

The “Direct peripheral access” function enables access directly to the local and central input and output signals of the control. The I/O access does not occur via the input/output image. The local and central input and output signals you can find the input and output signals of the CPU and the centrally expanded XC200 control with the XIOC signal modules. XIOC signal modules which can be integrated via a bus system cannot be accessed via the “Direct peripheral access”.

Addressing is dependent on the slot number “0 to 7” of the signal modules. Further differentiation within the slot exists and relates to bit number “0 to max. 63” of the Inputs/Outputs.

Depending on the functionality of the XIOC signal modules, access occurs as a bit/word or read/write operation. The access parameters are indicated in Table 1.

The inputs/outputs which are required for “Direct peripheral access” are physically connected in the same manner as normal inputs/outputs.

Table 1: “Direct peripheral access” overview

Modules	I/O bit access			I/O word access			I/O slot
	Read	Write	Param./Module	Read	Write	Param./Module	Param.
XC-CPU101-C128K-8DI-6DO	✓	✓	DI: 0 to 7, DO: 0 to 5	✓	✓	0	0
XC-CPU101-C128K-8DI-6DO-XV	✓	✓	DI: 0 to 7, DO: 0 to 5	✓	✓	0	0
XC-CPU101-C64K-8DI-6DO	✓	✓	DI: 0 to 7, DO: 0 to 5	✓	✓	0	0
XC-CPU101-C64K-8DI-6DO-XV	✓	✓	DI: 0 to 7, DO: 0 to 5	✓	✓	0	0
XIOC-8DI	✓	–	0 to 7	✓	–	0	1 to 7
XIOC-16DI	✓	–	0 to 15	✓	–	0	1 to 7
XIOC-16DI-AC	✓	–	0 to 15	✓	–	0	1 to 7
XIOC-8DO	–	✓	0 to 7	–	✓	0	1 to 7
XIOC-16DO	–	✓	0 to 15	–	✓	0	1 to 7
XIOC-16DO-S	–	✓	0 to 15	–	✓	0	1 to 7
XIOC-12DO-R	–	✓	0 to 11	–	✓	0	1 to 7
XIOC-16DX	–	✓	0 to 15	✓	✓	0	1 to 7
XIOC-8AI-I2	–	–	–	✓	–	0 to 7	1 to 7
XIOC-8AI-U1	–	–	–	✓	–	0 to 7	1 to 7
XIOC-8AI-U2	–	–	–	✓	–	0 to 7	1 to 7
XIOC-4T-PT	–	–	–	✓	–	0 to 3	1 to 7
XIOC-2AO-U1-2AO-I2	–	–	–	–	✓	0 to 3	1 to 7
XIOC-4AO-U1	–	–	–	–	✓	0 to 3	1 to 7
XIOC-4AO-U2	–	–	–	–	✓	0 to 3	1 to 7
XIOC-2AO-U2	–	–	–	–	✓	0 to 1	1 to 7
XIOC-4AI-2AO-U1	–	–	–	✓	✓	AI: 0 to 3, AO: 0 to 1	1 to 7
XIOC-2AI-1AO-U1	–	–	–	✓	✓	AI: 0 to 1, AO: 0	1 to 7

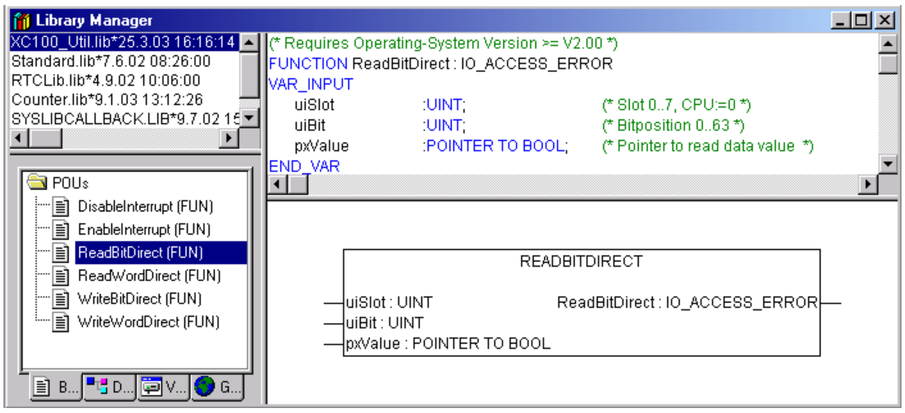
For Immediate Delivery call KMParts.com at (866) 595-9616

Modules	I/O bit access			I/O word access			I/O slot
	Read	Write	Param./Module	Read	Write	Param./Module	Param.
XIOC-1CNT-100KHZ	–	–	–	–	–	–	1 to 7
XIOC-2CNT-100KHZ	–	–	–	–	–	–	1 to 7
XIOC-2CNT-2AO-INC	–	–	–	✓	✓		1 to 7
XIOC-NET-DP-M	–	–	–	–	–	–	1 to 3
XIOC-QW-CAN-8DI-6DO	–	–	–	–	–	–	1 to 3
XIOC-QW-CAN-8DI-6DO-EXP	–	–	–	–	–	–	1 to 3

Functions

- **ReadBitDirect**

A bit of an input module can be read directly with this function. The state of an input bit is stored in the variables, which indicate to the parameterized pointer "ptr_xValue". The pointer variable will not be changed when a fault occurs during processing.



```
(* Requires Operating-System Version >= V2.00 *)
FUNCTION ReadBitDirect : IO_ACCESS_ERROR
VAR_INPUT
    uiSlot          :UINT;                 (* Slot 0..7, CPU:=0 *)
    uiBit           :UINT;                 (* Bitposition 0..63 *)
    pxValue         :POINTER TO BOOL;      (* Pointer to read data value  *)
END_VAR
```

```
                    READBITDIRECT
—uiSlot : UINT              ReadBitDirect : IO_ACCESS_ERROR—
—uiBit : UINT
—pxValue : POINTER TO BOOL
```

Figure 19: ReadBitDirect function

Parameters

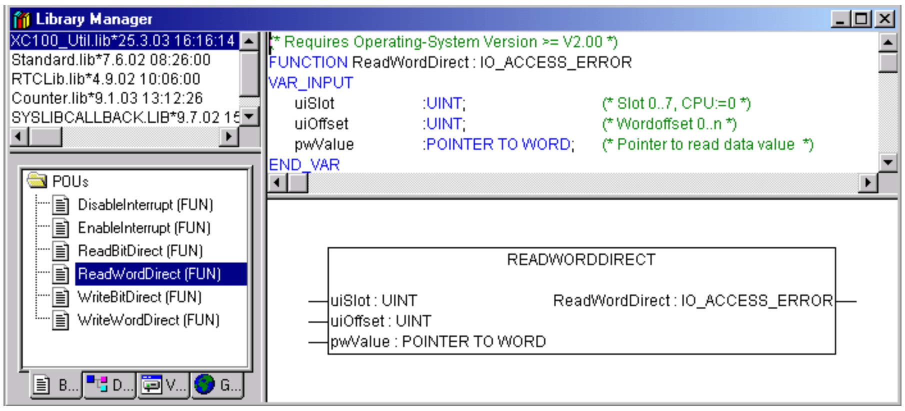
uiSlot:	Slot number of the signal module. For possible parameters see Table 1 on Page 18
uiBit:	Bit position within the input value of the signal module. For possible parameters see Table 1 on Page 18
ptr_xValue:	Pointer to the variable value
ReadBitDirect:	Display of the fault code, see Table 2 on Page 22

For Immediate Delivery call KMParts.com at (866) 595-9616

- **ReadWordDirect**

A word of an input module can be read directly with this function. The state of an input word is stored in the variables, which indicate to the parameterized pointer “ptr_wValue”.

The pointer variable will not be changed when a fault occurs during processing.



```
(* Requires Operating-System Version >= V2.00 *)
FUNCTION ReadWordDirect : IO_ACCESS_ERROR
VAR_INPUT
    uiSlot          :UINT;              (* Slot 0..7, CPU:=0 *)
    uiOffset        :UINT;              (* Wordoffset 0..n *)
    pwValue         :POINTER TO WORD;   (* Pointer to read data value  *)
END_VAR
```

Figure 20: ReadWordDirect function

Parameters

uiSlot:	Slot number of the signal module. For possible parameters see Table 1 on Page 18
uiOffset:	Word offset within a signal module. For possible parameters see Table 1 on Page 18
ptr_wValue:	Pointer to the variable value
ReadWordDirect:	Display of the fault code, see Table 2 on Page 22

For Immediate Delivery call KMParts.com at (866) 595-9616

- **WriteBitDirect**

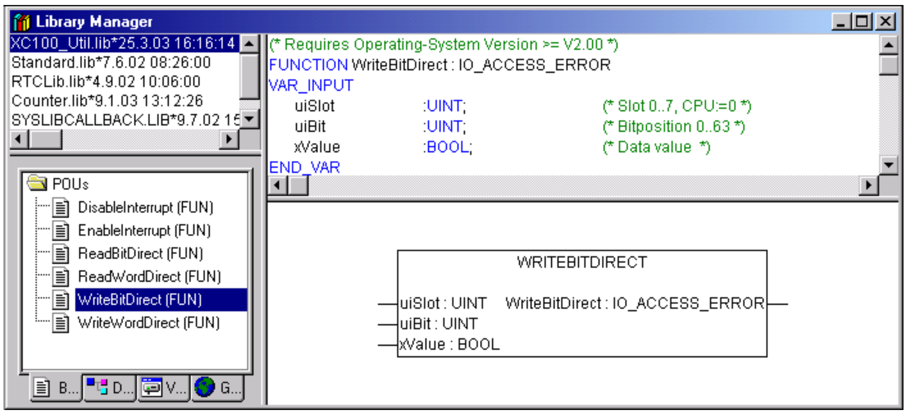
A bit of an output module can be controlled directly with this function. The respective output image is refreshed in addition to the physical output. Writing to the output is possible and not subject to limitation, for only the local 6 outputs of the XC100-CPU with slot "0".

The following limitations apply to slots "1" and above:

An individual output cannot be written with a "direct peripheral access". Writing of an entire output word is always the case. This means that the output image which is active at the time of the "direct access" will be output with the modified output bit. Thus, an output from other outputs within the output word occurs at the point of time when it is accessed and not at the end of the cycle.

For this reason, with a "direct peripheral access" to an output bit within a program cycle, the remaining outputs of the output word should not be used and evaluated.

A further refresh of the output word occurs at the end of the cycle.



```
(* Requires Operating-System Version >= V2.00 *)
FUNCTION WriteBitDirect : IO_ACCESS_ERROR
VAR_INPUT
    uiSlot          :UINT;              (* Slot 0..7, CPU:=0 *)
    uiBit           :UINT;              (* Bitposition 0..63 *)
    xValue          :BOOL;              (* Data value  *)
END_VAR
```

Figure 21: WriteBitDirect function

Parameters

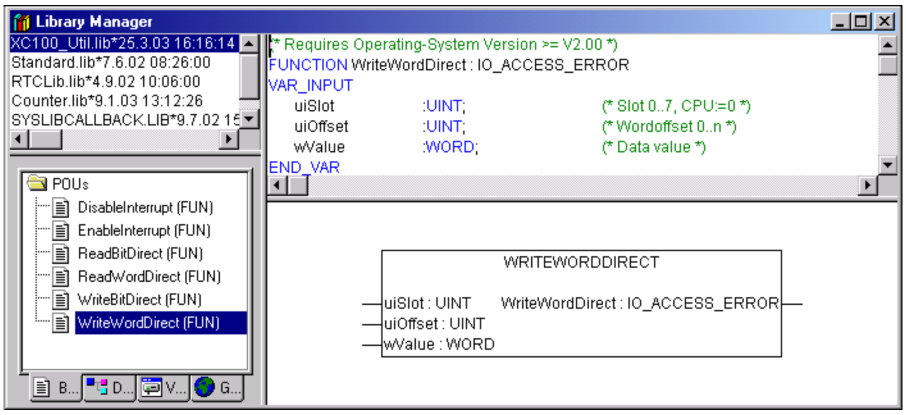
uiSlot:	Slot number of the signal module. For possible parameters see Table 1 on Page 18
uiBit:	Output bit within the signal module. For possible parameters see Table 1 on Page 18
xValue:	Input parameter from "Bit" type
WriteBitDirect	Display of the fault code, see Table 2 on Page 22

For Immediate Delivery call KMParts.com at (866) 595-9616

- **WriteWordDirect**

A word of an output module can be written directly with this function. At the time of access, the respective output image is also refreshed in addition to the physical output.

A further refresh of the output word occurs at the end of the cycle.



```
XC100_Util.lib*25.3.03 16:16:14
Standard.lib*7.6.02 08:26:00
RTCLib.lib*4.9.02 10:06:00
Counter.lib*9.1.03 13:12:26
SYSLIBCALLBACK.LIB*9.7.02 15
```

```
POUs
  DisableInterrupt (FUN)
  EnableInterrupt (FUN)
  ReadBitDirect (FUN)
  ReadWordDirect (FUN)
  WriteBitDirect (FUN)
  WriteWordDirect (FUN)
```

```
(* Requires Operating-System Version >= V2.00 *)
FUNCTION WriteWordDirect : IO_ACCESS_ERROR
VAR_INPUT
    uiSlot          :UINT;          (* Slot 0..7, CPU:=0 *)
    uiOffset        :UINT;          (* Wordoffset 0..n *)
    wValue          :WORD;          (* Data value *)
END_VAR
```

```
                 WRITEWORDDIRECT
uiSlot : UINT      WriteWordDirect : IO_ACCESS_ERROR
uiOffset : UINT
wValue : WORD
```

Figure 22: WriteWordDirect function

Parameters

uiSlot:	Slot number of the signal module. For possible parameters see Table 1 on Page 18
uiOffset:	Output word within a signal module. For possible parameters see Table 1 on Page 18
wValue:	Input parameter from “Word” type
WriteWordDirect	Display of the fault code, see Table 2 on Page 22

Error code with “direct peripheral access”

Verify all functions as far as possible for the validity of the call parameters. Verification is undertaken to determine if the access occurs in dependance on the parameterized signal module and the physical existence of the signal module. If a fault is determined, access is not undertaken and an error code is output → table 2 The data fields for the value transfer remain unchanged.

Table 2: Error code with direct peripheral access IO_ACCESS_NO_ERROR data type

IO_ACCESS_NO_ERROR:	0:	Function executed, a fault was not determined
IO_ACCESS_INVALID_ SLOTNUMBER	1:	Slot = 0 or greater than 7
IO_ACCESS_INVALID_OFFSET	2:	BitWord offset is too large
IO_ACCESS_DENIED	3:	Invalid access, e. g. write access to input module or access to non-available address range (offset too large)
IO_ACCESS_NO_MODULE	4:	No module available at the parameterized slot
IO_ACCESS_ INVALID _BUFFER	5:	No or incorrect pointer to the output variable with “Read....Direct” defined

For Immediate Delivery call KMParts.com at (866) 595-9616

Updating the operating system

With the XC100, you have the possibility of replacing the operating system (BTS) supplied with the PLC by a more recent one. Moeller offers the most recent operating system version for download on the Internet. It is an objective to design all future operating system versions to be downward compatible. It is not always possible to implement due to technical constraints. Furthermore, it is possible that the hardware version (e. g. an older version) that you have may not support all the new operating system features.

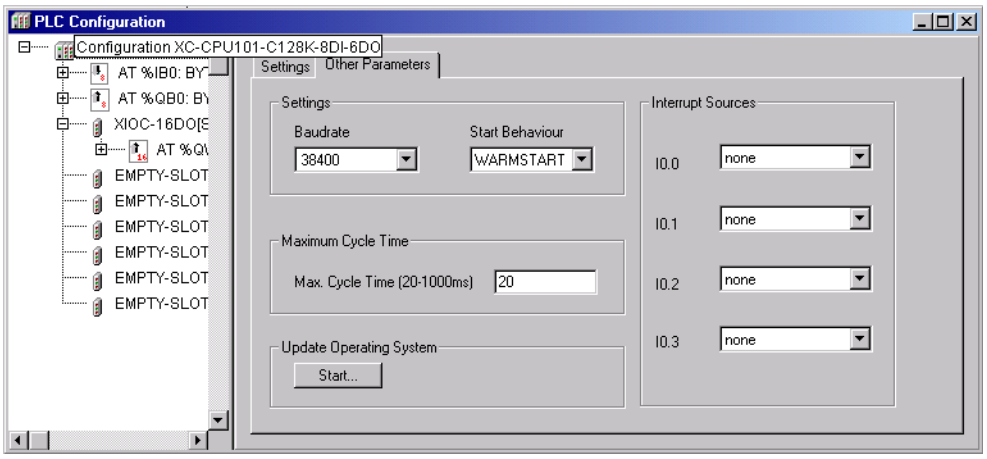
For this reason, please clarify the feasibility of a download with your personal Moeller sales contact before performing an update.

Procedure:

- An operating system download is only possible when logged on.
- All the data in the control will be erased. This applies to an operating system which is still present and to the existing user programs.

▶ Establish a serial connection via the RS 232 interface of the PC with the XC100, see Page 31.

▶ Activate the "Other Parameters" tap in the "PLC Configuration" window.



PLC Configuration

Configuration XC-CPU101-C128K-8DI-6DO
AT %IB0: BY
AT %QB0: BY
XIOC-16DO[S
AT %QW
EMPTY-SLOT
EMPTY-SLOT
EMPTY-SLOT
EMPTY-SLOT
EMPTY-SLOT
EMPTY-SLOT

Settings | Other Parameters

Settings
Baudrate: 38400
Start Behaviour: WARMSTART

Maximum Cycle Time
Max. Cycle Time (20-1000ms): 20

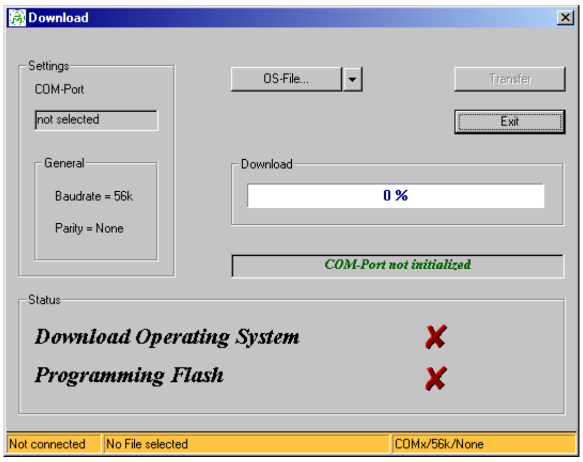
Update Operating System
Start...

Interrupt Sources
I0.0: none
I0.1: none
I0.2: none
I0.3: none

Figure 23: Start the download of the XC100 operating system

▶ Click on the "Start" button.

The "Download" window opens.



Download

Settings
COM-Port: not selected

OS-File... | Transfer | Exit

General
Baudrate = 56k
Parity = None

Download
0 %

COM-Port not initialized

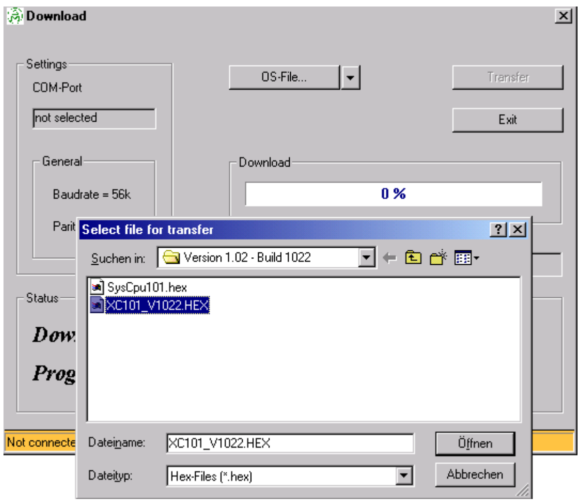
Status
Download Operating System ✗
Programming Flash ✗

Not connected | No File selected | COMx/56k/None

Figure 24: Download: Selection and information

▶ Press the "OS-File" button and select the required operating system file (*.hex). If an operating system file is not available or if you happen to have a newer operating system file, define the path where your BTS is saved.

→ If operating system files are available with the XSoft, you can open and select these files accordingly using the drop-down menu (beside the "OS-File" button).



Download

Select file for transfer

Suchen in: Version 1.02 - Build 1022
SysCpu101.hex
XC101_V1022.HEX

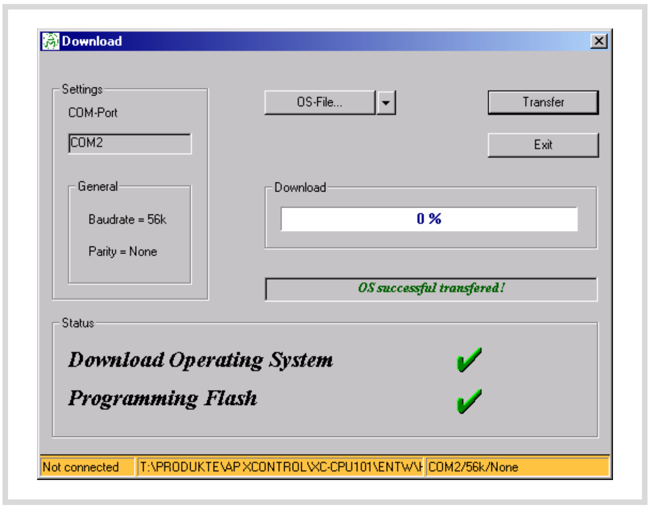
Dateiname: XC101_V1022.HEX
Dateityp: Hex-Files (*.hex)
Öffnen | Abbrechen

Figure 25: Operating system file selection

▶ If the required operating system file has been selected and opened, start the transfer with the "Transfer" button.

For Immediate Delivery call KMParts.com at (866) 595-9616

Do not attempt to interrupt or influence the automatic download routine until the following status is displayed:



OS successful transfered!

Status

Download Operating System ✓

Programming Flash ✓

Not connected | T:\PRODUKTE\AP XCONTROL\XC-CPU101\ENTW\ | COM2/56k/None

Figure 26: OS successful transferred!

▶ Click in this window on the “Exit” button.

You receive the “Reload application” message as the download will overwrite or delete the old operating system and the user program. You can start and test the user program.

After the operating system has been downloaded, the "Communication Error" appears. This message appears after each download of the operating system, and the PLC must be rebooted. A renewed "Logon" is required after each reboot.

Interrupt processing

In the XC100 it is possible to program and parameterize up to six interrupt events. Interrupts can be activated by:

- physical inputs I0.0 to I0.3 of the XC-CPU101
- XIOC signal modules with interrupt features.

If an interrupt occurs, the runtime module executes the program organizational unit (POU) which is linked to the interrupt source. Execution of the POU is time-monitored. The parameterized cycle time is used for this cycle time. The interrupts are enabled when changed to the RUN state and inhibited when changed to the STOP state. Interrupt sources which are not enabled in the configuration do not initiate an interrupt. If a POU is not assigned to an enabled interrupt source, the interrupt is recognized and executed but without running a POU.

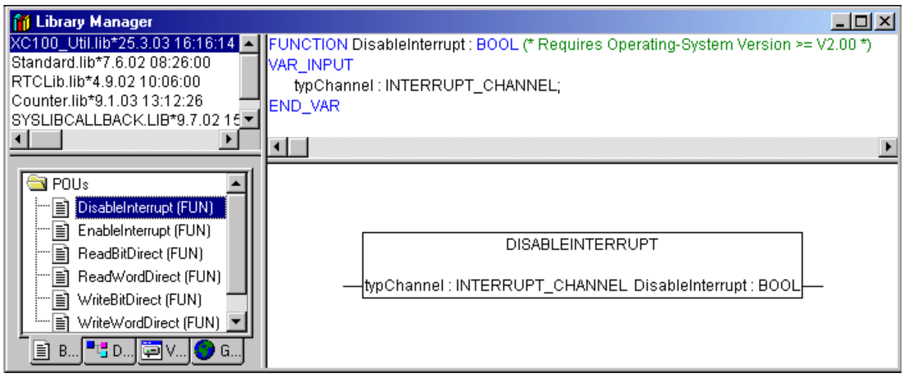
Frequent occurrence of an interrupt during a cycle can cause the cycle time to time-out and result in a reset being initiated by the Watchdog.

User interrupts can be inhibited and re-enabled from the program. The “DisableInterrupt” and “EnableInterrupt” functions are provided for this purpose. A call parameter in the XSoft determines if an individual interrupt or all interrupts are enabled or inhibited. Enabling of an inhibited interrupt must be performed with the same parameter used to inhibit it.

Both the “DisableInterrupt” and “EnableInterrupt” functions are components of the “XC100_Util.lib” library. This library must – if not already done so – be integrated into the library manager of the XSoft.

For Immediate Delivery call KMParts.com at (866) 595-9616

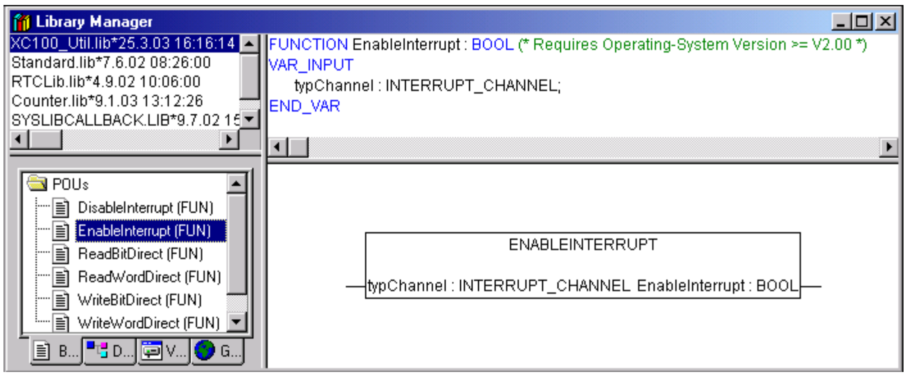
- **DisableInterrupt**
With this function, you disable (deactivate) a parameterized physical interrupt by accessing it from the user program.



```
FUNCTION DisableInterrupt : BOOL (* Requires Operating-System Version >= V2.00 *)
VAR_INPUT
    typChannel : INTERRUPT_CHANNEL;
END_VAR
```

Figure 27: DisableInterrupt function

- **EnableInterrupt**
With this function, the physical interrupt which was deactivated beforehand can now be re-enabled as an active interrupt.



```
FUNCTION EnableInterrupt : BOOL (* Requires Operating-System Version >= V2.00 *)
VAR_INPUT
    typChannel : INTERRUPT_CHANNEL;
END_VAR
```

Figure 28: EnableInterrupt function

For Immediate Delivery call KMParts.com at (866) 595-9616

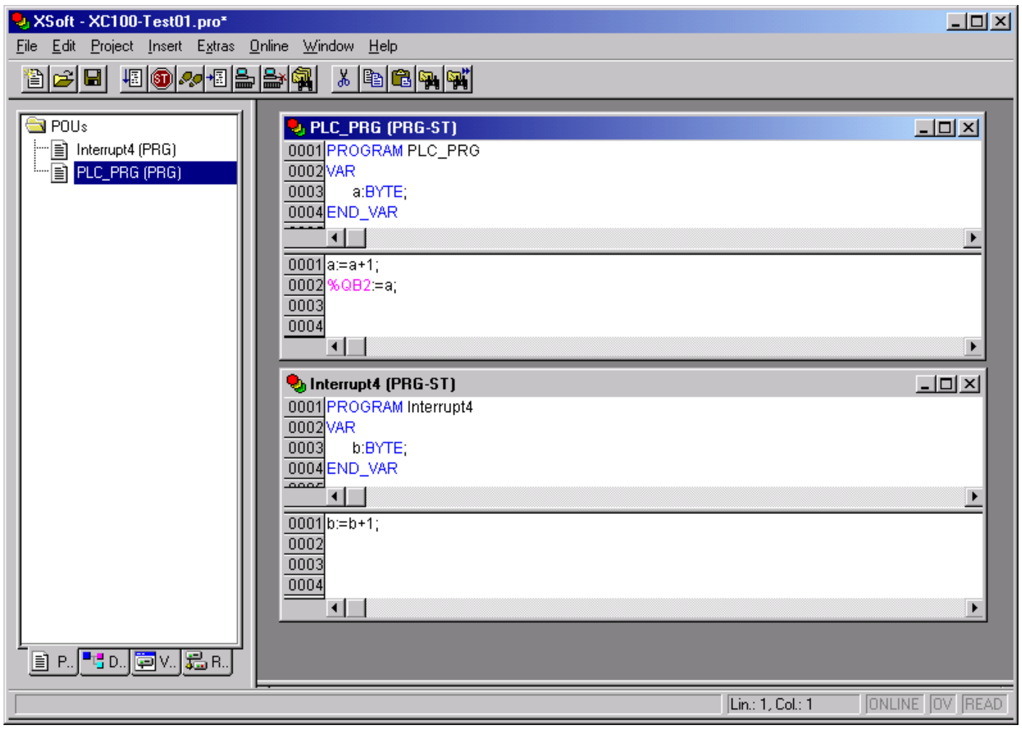
Creating and integrating an interrupt function

The formal procedure for the provision and integration of an interrupt function is described in individual steps in the following.

In the example, a H-signal on input I0.0 should branch into an interrupt module and execute it.

▶ Create a program module for the normal application ("PLC-PRG") for this purpose and a further module with the interrupt functionality "Interrupt4".

The following figure shows you both modules:



```
PROGRAM PLC_PRG
VAR
    a:BYTE;
END_VAR
```

```
a:=a+1;
%QB2:=a;
```

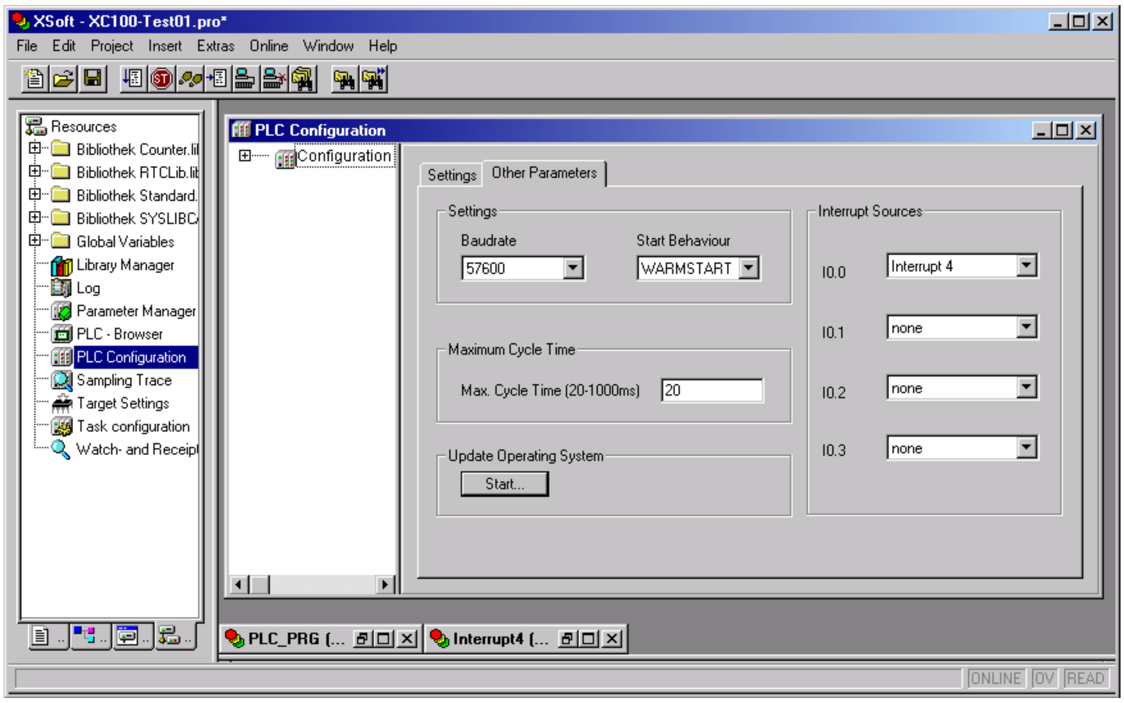
```
PROGRAM Interrupt4
VAR
    b:BYTE;
END_VAR
```

```
b:=b+1;
```

Figure 29: Modules "PLC_PRG" and "Interrupt4"

▶ Changeover to the PLC Configuration and assign Interrupt 4 from the list field to input I0.0.

For Immediate Delivery call KMParts.com at (866) 595-9616



Settings

Baudrate	Start Behaviour
57600	WARMSTART

Maximum Cycle Time

Max. Cycle Time (20-1000ms) 20

Update Operating System

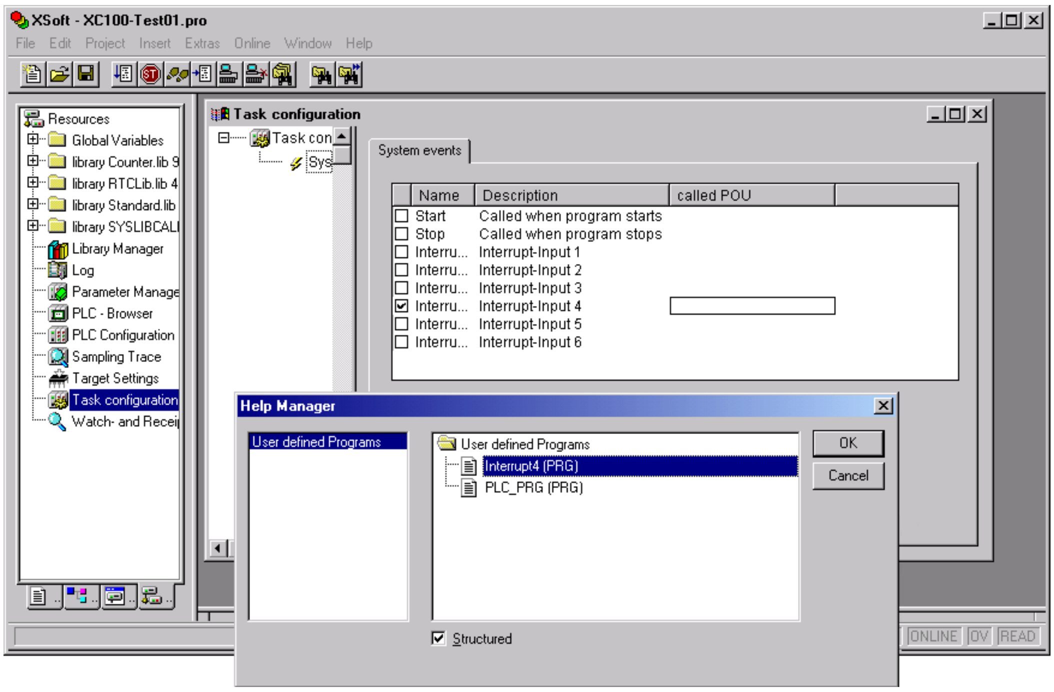
Interrupt Sources

I0.0	Interrupt 4
I0.1	none
I0.2	none
I0.3	none

Figure 30: Assign input I0.0 with interrupt 4

- ▶ Changeover to the “Task configuration” and tick the box in the “System events” input field for “Interrupt4”.
- ▶ Now stay on the same row and mark the “called POU” field with the left-hand mouse key and press function key “F2”.

The “Help Manager” window opens in which all predefined programs are listed:



	Name	Description	called POU
☐	Start	Called when program starts	
☐	Stop	Called when program stops	
☐	Interru...	Interrupt-Input 1	
☐	Interru...	Interrupt-Input 2	
☐	Interru...	Interrupt-Input 3	
☑	Interru...	Interrupt-Input 4	
☐	Interru...	Interrupt-Input 5	
☐	Interru...	Interrupt-Input 6	

User defined Programs
- Interrupt4 (PRG)
- PLC_PRG (PRG)

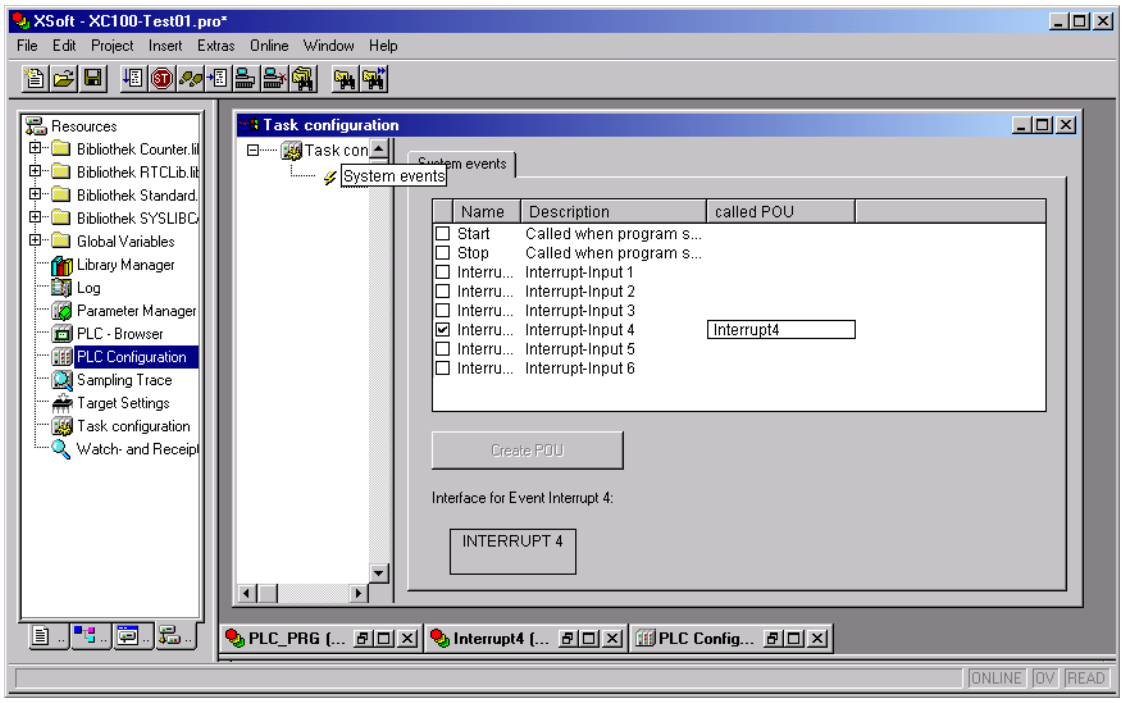
☑ Structured

Figure 31:Assignment of an interrupt module to an interrupt event

For Immediate Delivery call KMParts.com at (866) 595-9616

▶ Select the “Interrupt4” program and confirm with OK.

The following window appears:



	Name	Description	called POU
☐	Start	Called when program s...	
☐	Stop	Called when program s...	
☐	Interru...	Interrupt-Input 1	
☐	Interru...	Interrupt-Input 2	
☐	Interru...	Interrupt-Input 3	
☑	Interru...	Interrupt-Input 4	Interrupt4
☐	Interru...	Interrupt-Input 5	
☐	Interru...	Interrupt-Input 6	

Interface for Event Interrupt 4:

INTERRUPT 4

Figure 32:Interrupt module completed task configuration

▶ Save the program created, compile it and logon to the PLC and test the functions of the program modules which you have created.

Single Step operation with system events

Not only a task, but also a system event can call a project module for processing. The system events which can be used for this purpose depend on the target system. They are comprised of the standard system and the target system dependant events. Possible events, for example are Stop, Start, Interrupt, Online modification.

→ Single step operation is not possible with the "System event" programming modules.

For Immediate Delivery call KMParts.com at (866) 595-9616

Browser commands

You can directly access the states/events in the XC100 with the Browser commands. The Online description in the XSoft can be found at: ‹Resources → PLC-Browser›.

No.	Command	Description
1	?	Get a list of implemented commands.
2	reflect	Mirror current command line for test purposes.
3	mem	Memory-dump, Syntax: mem <start-addr> <end-addr>
4	memc	As mem, addresses are added to the start address of the code range.
5	memd	As mem, addresses are added to the start address of the data range.
6	pinf	Output project information
7	ppt	Output module pointer table
8	dpt	Output data pointer table
9	pid	Output project ID
10	cycle	Output cycle time
11	GetNodeId	Output CANopen Node ID
12	SetNodeId	Set CANopen Node ID
13	metrics	Output PLC information
14	reload	Load boot project from the MMC on the PLC
15	remove	Erase boot project from the MMC
16	format	Formatting of the MMC
17	getswitchpos	Output switch position
18	getbattery	Output battery status
19	getrtc	Read-out real-time clock [HH:MM:SS]
20	setrtc	Set real-time clock [HH:MM:SS]

File access on the file system of the Multi Media Card

The data access modules enable access to the file access system of the Multi Media Card (MMC) with the XC100. Up to four files can be opened simultaneously. Subdirectories are not supported.

The following modules are available:

- File Open
- File Close
- File Read
- File Write
- File Delete
- File Rename
- File Set Pos
- File Get Size

You can find the descriptions of the function blocks in the manual "Function Blocks for XSoft" (AWB2786-1456GB). This manual is provided as a PDF file (h1456.pdf) on the CD.

The latest version of the manual can be found at http://www.moeller.net/support: Search text: AWB2786-1456GB

Data remanence

The controller has a memory area for remanent data → page 7. Variables that have been declared with "VAR_RETAIN" are saved in this area, and are thus kept available (i. e. remanent) for a warm start of the application program. Data that are remanent for a cold start – "VAR_RETAIN Persistent" are not supported. Remanence (non-volatility) of the data is guaranteed when the PLC is switched off if a battery is inserted.

If it was not possible to finish a cycle that was being processed, because of a supply interruption, then the data will not be consistent, since the interruption could have occurred at any point of the cycle. When the supply returns, the residual-cycle will not be completed. The control starts according to the set start behaviour.

If this is to be prevented, appropriate measures must be taken as part of the engineering. One solution for this problem is an uninterruptible power supply with additional accumulator buffering.

For Immediate Delivery call KMParts.com at (866) 595-9616

Program transfer

The transfer of an application program always takes place via the battery-buffered SRAM area in the controller. Afterwards, a backup can be created on the multimedia card by using the "Create boot project" command. A program backup can only be created while the system is in the "HALT" state. The internal FLASH memory of the CPU cant be used as storage for a program backup.

Operating states

The following summary provides you with the state definitions for the XC100. The LED indications for the various states are also shown.

State	Display		Definition
	RUN/STOP	SF	
System test	OFF	OFF	System test in progress
System update	ON	ON	System update in progress
Switch-on OK	OFF	OFF	System test finished without error
Switch-on not OK	blinks	blinks	System test generated an error
NOT READY	OFF	ON	No application program present
STOP	blinks	OFF	Application program loaded, PLC in "STOP" state
RUN	ON	OFF	Application program loaded, PLC in "RUN" state
STOP/RUN with general error, diagnosis message	ON	ON	General error/diagnosis message available

For Immediate Delivery call KMParts.com at (866) 595-9616

4 Establishing a PC – XC100 connection

This section describes the measures that are required to link a PC to the XC100, so that the PC can be used as a programming device (hardware and software).

Establishing a connection via the RS 232 interface (XC100)

Communication is made through the RS 232 serial interface, which is not isolated via optocouplers. You can use either the COM1 or the COM2 port for the PC interface. Please use the programming cable XT-232-SUB-D/RJ 45 to make the physical connection.

Programming cable

The programming cable is made up as shown below:

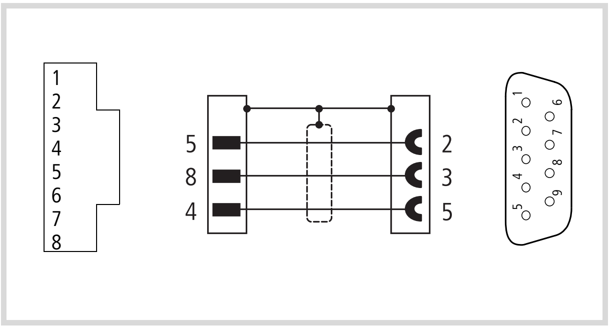


Figure 33: Core assignments for RS 232 programming cable

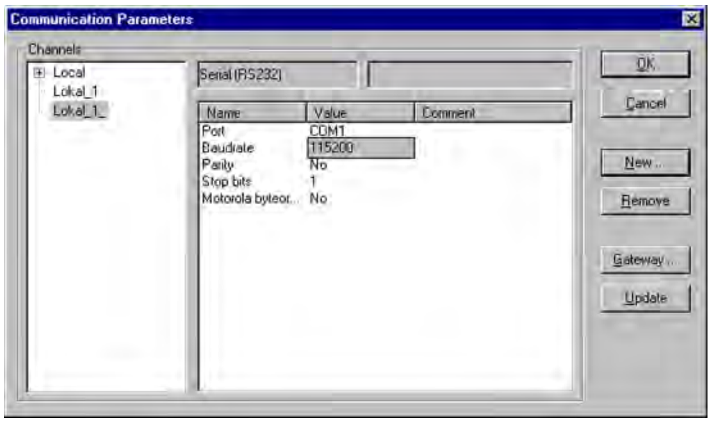
XSoft software

Use XSoft to define the communication parameters:

▶ Call up the menu item ‹Online → Communication Parameters› in XSoft, and select the COM1 or COM2 interface.
▶ Select the values that are shown in the diagram.

You can alter the default values by making a double-click on the entered value.

→ Further notes on the communication parameters can be found in the XSoft manual (AWB2700-1437GB). This manual is provided as a PDF file (h1437g.pdf) on the CD.



Name	Value	Comment
Port	COM1	
Baudrate	115200	
Parity	No	
Stop bits	1	
Motorola byteor...	No	

Figure 34: Select communication parameters

Communication Error (#0): Logout Performed
If it is not possible to establish communication between the programming PC and the XC100, the following steps must be implemented:

- inspect the physical connections
- inspect the baud rate of the communication parameters in the XSoft
- inspect the baud rate in the XC100 (default setting of 38400 kBit/s is set)
- inspect the baud rate of the CAN connection, if the CAN fieldbus is used.

The interface parameters must correspond in the XSoft and in the XC100!

For Immediate Delivery call KMParts.com at (866) 595-9616

For Immediate Delivery call KMParts.com at (866) 595-9616

5 Create sample project

In the following example you will learn how to use the XSoft software. You create a project by creating a configuration and a program. You can then test out the project after it has been downloaded.

The configuration is created in the XSoft configuration editor. A distinction is made between local and central I/Os:

The local (digital) inputs and outputs are integral parts of the CPU module, implemented on the power supply board. They are already pre-configured in the configuration editor.

The central inputs/outputs are implemented in the signal modules, which can be joined up via the module rack. Configure them to suit the requirements of the application. The available XI/OC signal modules can be used.

Connect appropriate CANopen fieldbus participants to the CANopen interface.

The basis for the configuration is the following hardware layout.

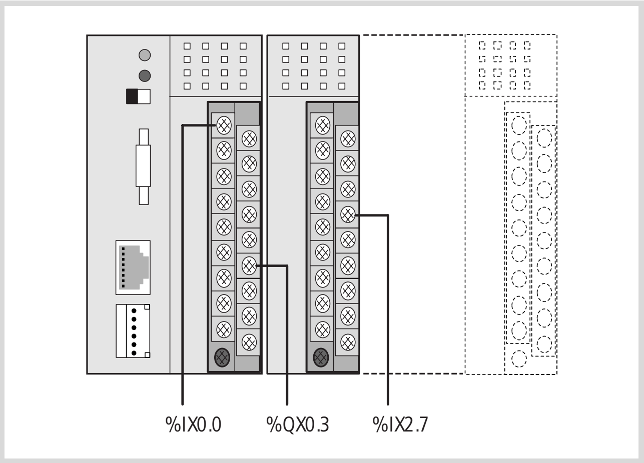


Figure 35: Hardware layout for the sample project

For Immediate Delivery call KMParts.com at (866) 595-9616

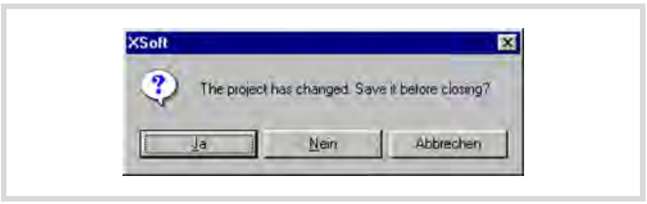
Task

Made a logical AND linkage of inputs I0.0 and I2.7 on the XC100. The result of this logical operation should be presented at output Q0.3.

The second step is to read in the inputs/generate the outputs, using a CAN Master.

Activate the appropriate CAN libraries before linking the CAN Master [VAR] module into the controller configuration. This sequence will not take place automatically, but must be explicitly carried out by the user.

▶ Select the menu item ‹File → New›.
▶ Answer the query about saving the old project.


XSoft

The project has changed. Save it before closing?

Ja | Nein | Abbrechen

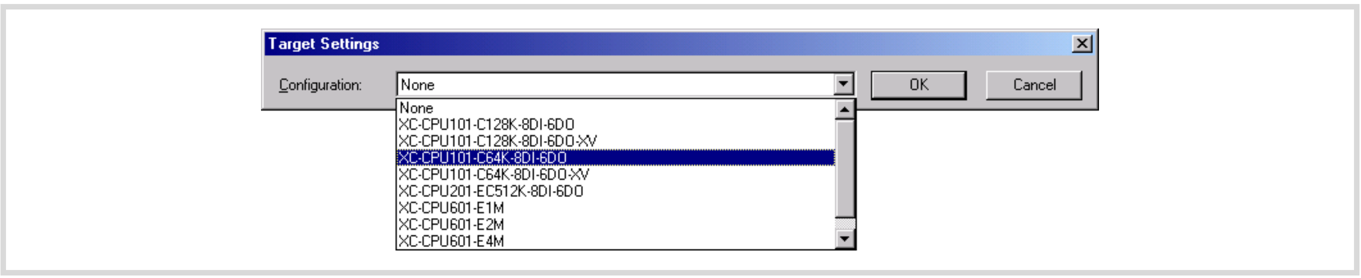
Figure 36: Save old project?

Procedure

Set up target system

After starting XSoft, create a new file:

Select the target system. In the example, the system XC-CPU101-C64K-8DI-6DO has been selected.


Target Settings

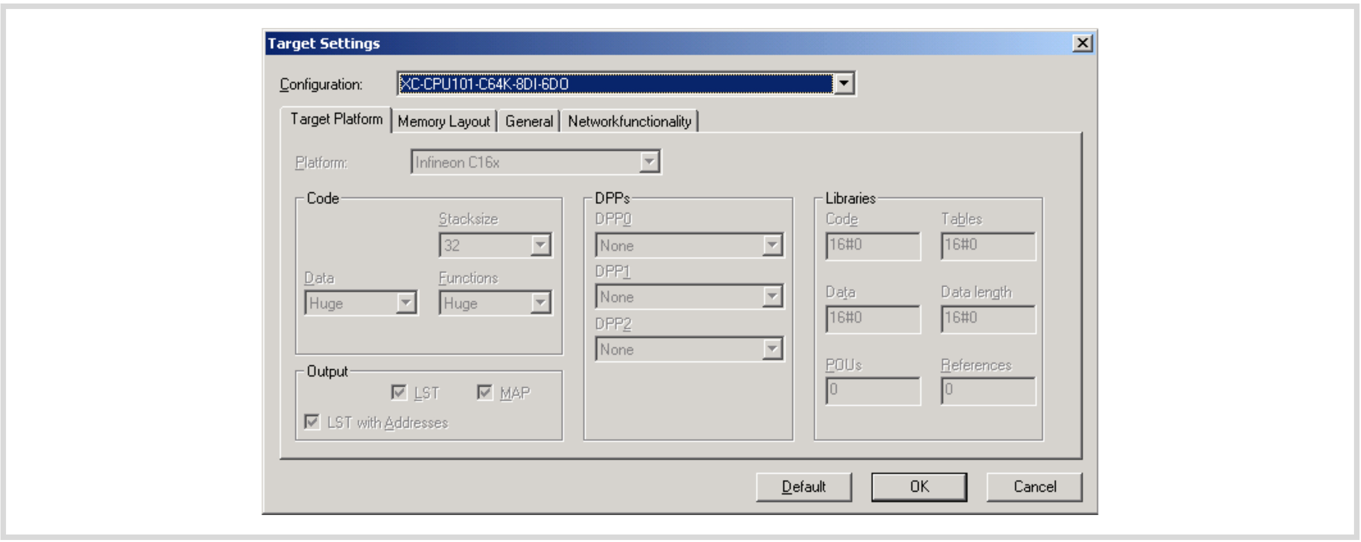
Configuration: None

OK | Cancel

None
XC-CPU101-C128K-8DI-6DO
XC-CPU101-C128K-8DI-6DO-XV
XC-CPU101-C64K-8DI-6DO
XC-CPU101-C64K-8DI-6DO-XV
XC-CPU201-EC512K-8DI-6DO
XC-CPU601-E1M
XC-CPU601-E2M
XC-CPU601-E4M

Figure 37: Select target system

A double-click on the target system produces the following diagrams, which you can then look at by selecting the individual register tabs. The register tabs "Target platform", Memory layout" and "General" just present information about the target system. No settings can be made in these tabbed screens.


Target Settings

Configuration: XC-CPU101-C64K-8DI-6DO

Target Platform | Memory Layout | General | Networkfunctionality

Platform: Infineon C16x

Code
Stacksize: 32
Data: Huge
Functions: Huge

Output
☑ LST ☑ MAP
☑ LST with Addresses

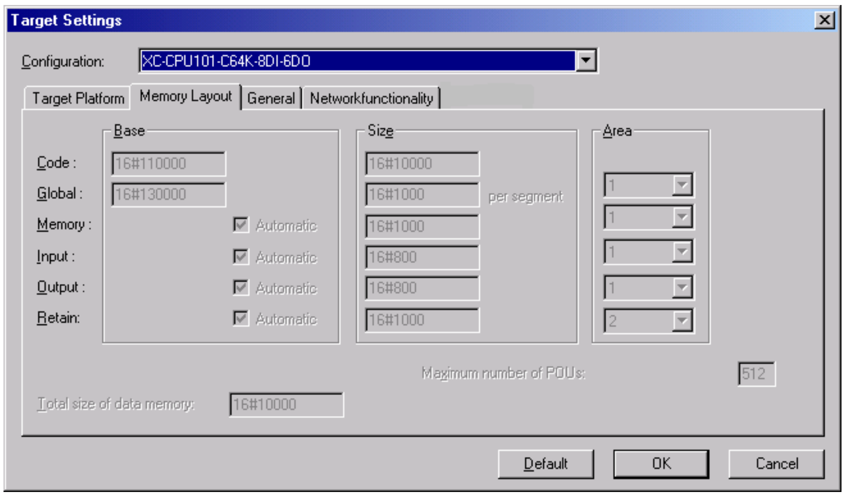
DPPs
DPP0: None
DPP1: None
DPP2: None

Libraries
Code: 16#0
Tables: 16#0
Data: 16#0
Data length: 16#0
POUs: 0
References: 0

Default | OK | Cancel

Figure 38: Target system settings – target platform

For Immediate Delivery call KMParts.com at (866) 595-9616


Target Settings

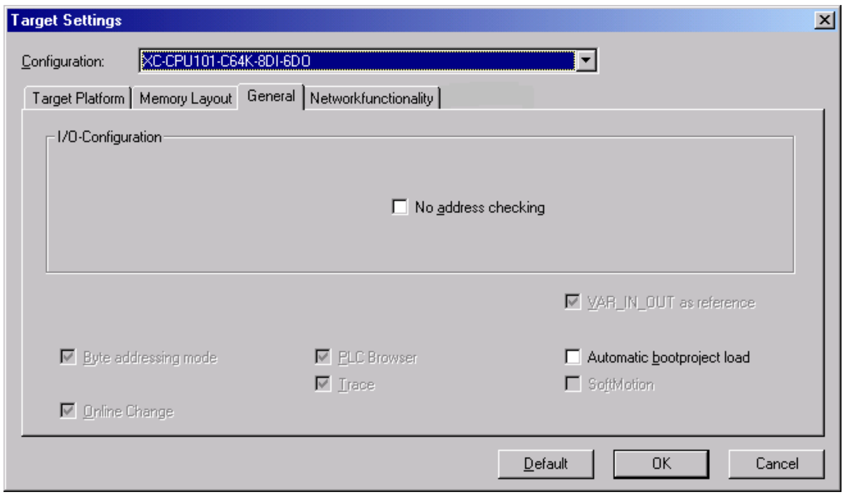
Configuration: XC-CPU101-C64K-8DI-6DO

	Base		Size		Area
Code :	16#110000		16#10000		1
Global :	16#130000		16#1000	per segment	1
Memory :		☑ Automatic	16#1000		1
Input :		☑ Automatic	16#800		1
Output :		☑ Automatic	16#800		1
Retain:		☑ Automatic	16#1000		2

Maximum number of POUs: 512

Total size of data memory: 16#10000

Figure 39: Target system settings – memory layout


Target Settings

Configuration: XC-CPU101-C64K-8DI-6DO

I/O-Configuration

☐ No address checking

☑ VAR_IN_OUT as reference

☑ Byte addressing mode

☑ PLC Browser

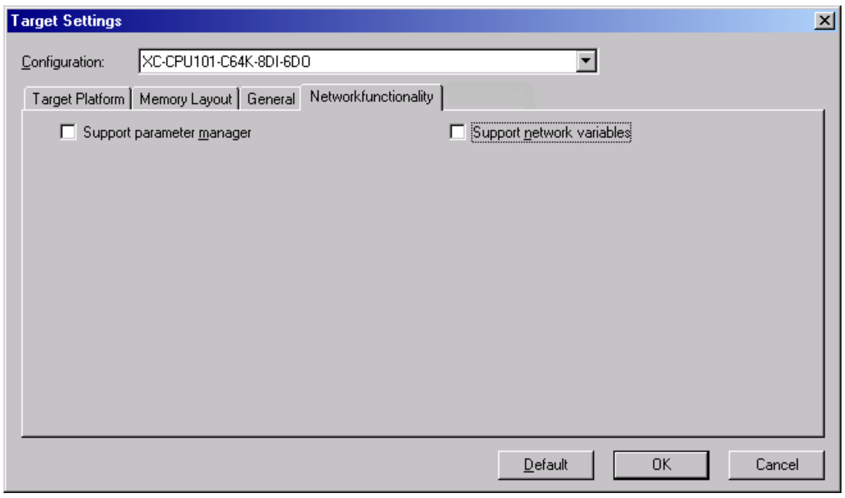
☐ Automatic bootproject load

☑ Trace

☐ SoftMotion

☑ Online Change

Figure 40: Target system settings – general


Target Settings

Configuration: XC-CPU101-C64K-8DI-6DO

☐ Support parameter manager

☐ Support network variables

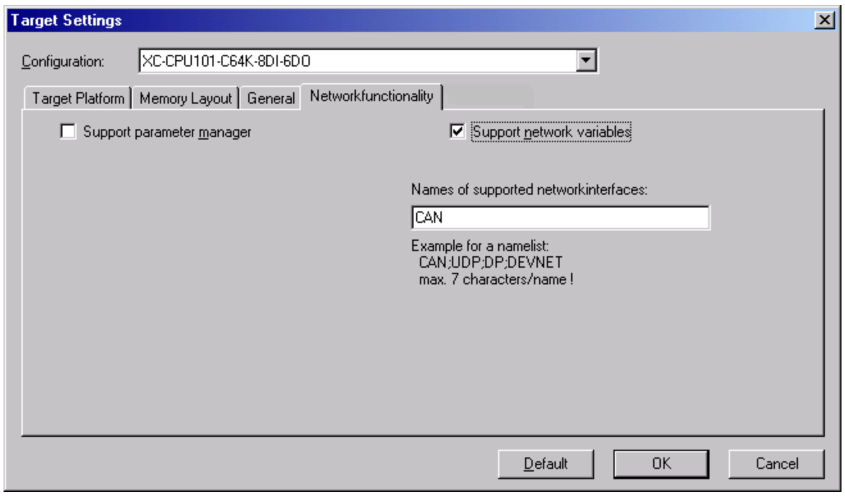
Figure 41: Target system settings – network functionality

For Immediate Delivery call KMParts.com at (866) 595-9616

▶ If a CANopen slave is to be integrated into the configuration, click the "Networkfunctionality" tab and tick the "Support network variables" check box.

You will get a message that this target system supports the CAN network.

The "Support network variables" control box must only be activated if you wish to work with network variables. When activated, the libraries required for operation of network variables are added automatically. This function is not required if you use a CAN Master/CAN Device.



Configuration: XC-CPU101-C64K-8DI-6DO

☐ Support parameter manager

☑ Support network variables

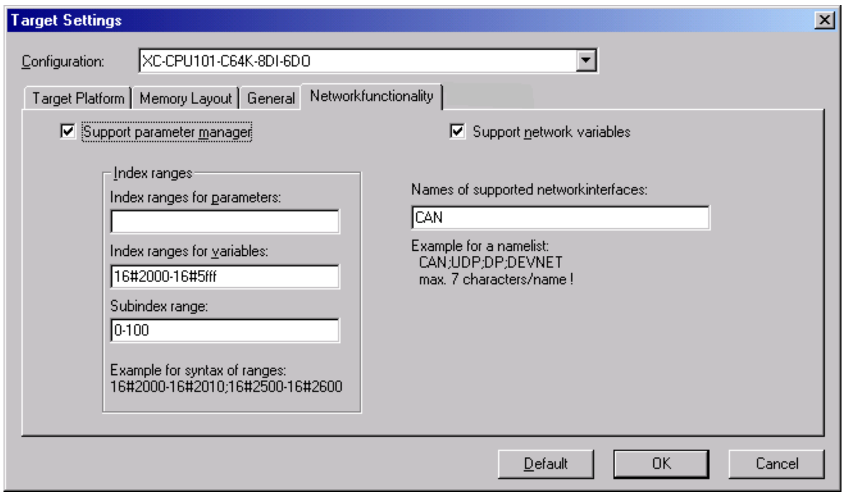
Names of supported networkinterfaces: CAN

Example for a namelist:
CAN;UDP;DP;DEVNET
max. 7 characters/name !

Figure 42: Target system settings – CAN network functions (1)

▶ Activate the "Support parameter manager" check box in order to view additional information concerning the index ranges.

The CAN relevant Parameter manager is only required for a CAN device. Standard settings are available for this purpose, which means that no modifications must be made.



Configuration: XC-CPU101-C64K-8DI-6DO

☑ Support parameter manager

☑ Support network variables

Index ranges

Index ranges for parameters:

Index ranges for variables: 16#2000-16#5fff

Subindex range: 0-100

Example for syntax of ranges:
16#2000-16#2010;16#2500-16#2600

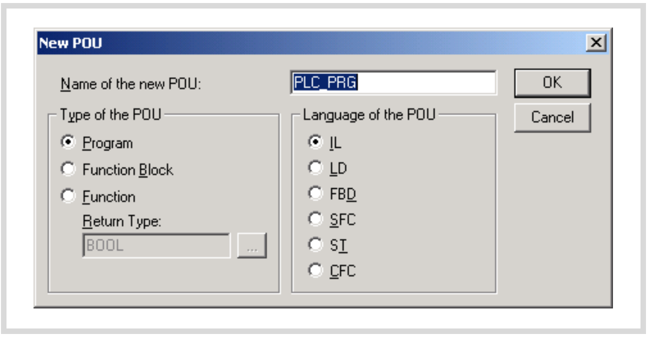
Names of supported networkinterfaces: CAN

Example for a namelist:
CAN;UDP;DP;DEVNET
max. 7 characters/name !

Figure 43: Target system settings – CAN network functions (2)

▶ Close this selection with "OK".
▶ Select the POU type "Program" and the programming language "IL":

For Immediate Delivery call KMParts.com at (866) 595-9616



New POU

Name of the new POU: PLC_PRG

Type of the POU: [✓ Program] Function Block Function

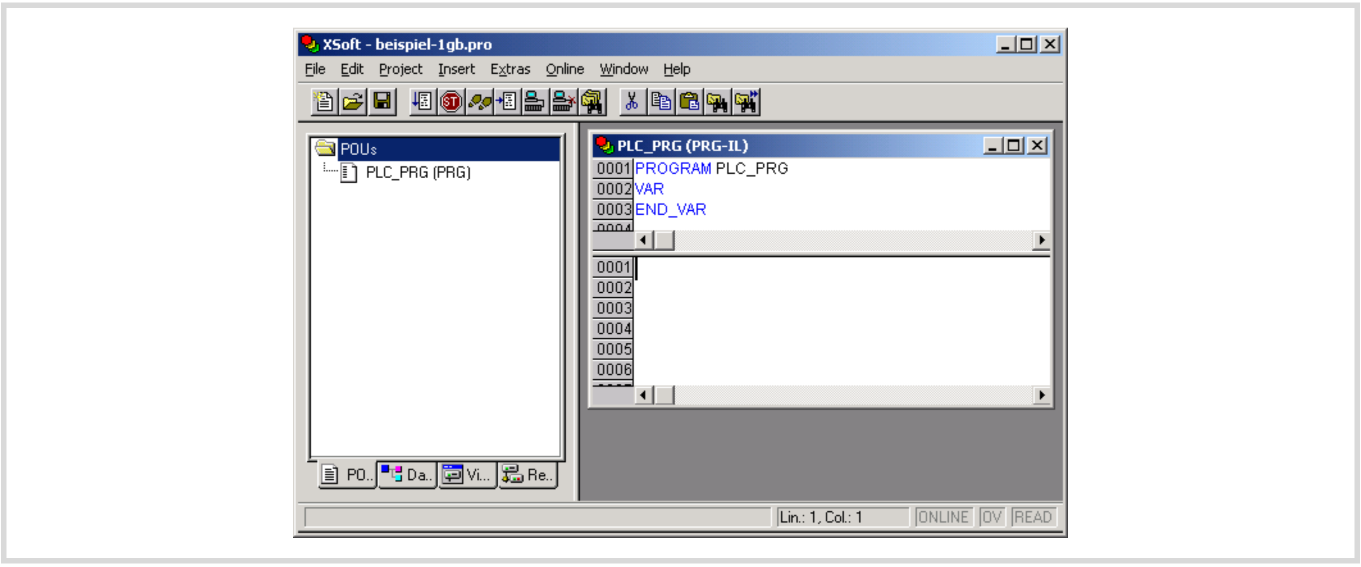
Return Type: BOOL

Language of the POU: [✓ IL] LD FBD SFC ST CFC

Figure 44: Select POU type

▶ Confirm with “OK” and save the file under “sample-1”.

A window will now appear, in which you can continue with the programming or configuration:



XSoft - beispiel-1gb.pro

```
PROGRAM PLC_PRG
VAR
END_VAR
```

Figure 45: POU type “Program” in IL representation

Including the CAN libraries

The following steps describe how to include the CAN libraries.

▶ In the menu ‹Window → Library Manager› make a double-click on “Library Manager” to call up the “Library Manager” window.

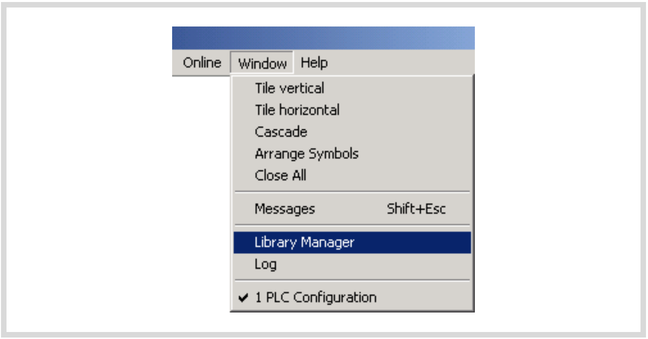
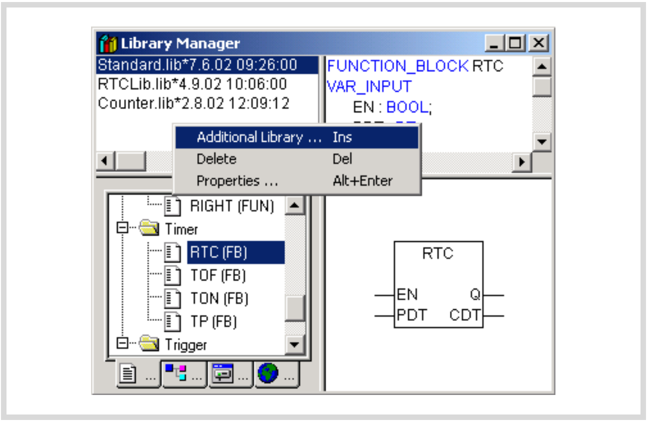


Figure 46: Window menu

▶ In the new window, make a right-button click in the white field at top left.



Library Manager

Standard.lib*7.6.02 09:26:00
RTCLib.lib*4.9.02 10:06:00
Counter.lib*2.8.02 12:09:12

```
FUNCTION_BLOCK RTC
VAR_INPUT
    EN : BOOL;
```

Figure 47: Library Manager

▶ Select the item “Additional library ... insert” in the context menu.

For Immediate Delivery call KMParts.com at (866) 595-9616

▶ Move the cursor into the list field “Search in:” and select the entry “Moeller”.

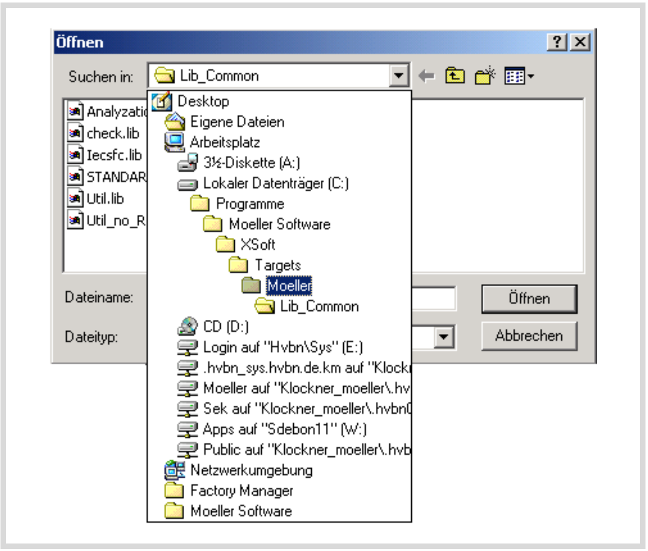

Öffnen
Suchen in: Lib_Common
Analyzati
check.lib
Iecsfc.lib
STANDAR
Util.lib
Util_no_R
Desktop
Eigene Dateien
Arbeitsplatz
3½-Diskette (A:)
Lokaler Datenträger (C:)
Programme
Moeller Software
XSoft
Targets
Moeller
Lib_Common
CD (D:)
Login auf "Hvbn\Sys" (E:)
.hvbn_sys.hvbn.de.km auf "Klock
Moeller auf "Klockner_moeller\.hv
Sek auf "Klockner_moeller\.hvbn
Apps auf "Sdebon11" (W:)
Public auf "Klockner_moeller\.hvb
Netzwerkumgebung
Factory Manager
Moeller Software
Dateiname:
Dateityp:
Öffnen
Abbrechen

Figure 48: Context menu “Search in:”

▶ After the “Moeller” entry has been selected, the following window will be opened:

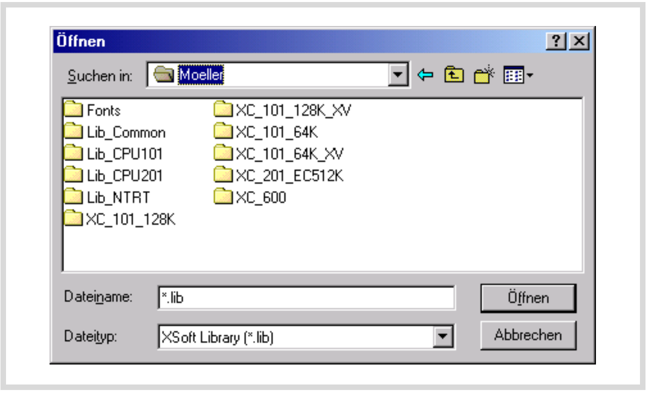

Öffnen
Suchen in: Moeller
Fonts
Lib_Common
Lib_CPU101
Lib_CPU201
Lib_NTRT
XC_101_128K
XC_101_128K_XV
XC_101_64K
XC_101_64K_XV
XC_201_EC512K
XC_600
Dateiname: *.lib
Dateityp: XSoft Library (*.lib)
Öffnen
Abbrechen

Figure 49: File list: Moeller

▶ Select the library “Lib_CPU101”.

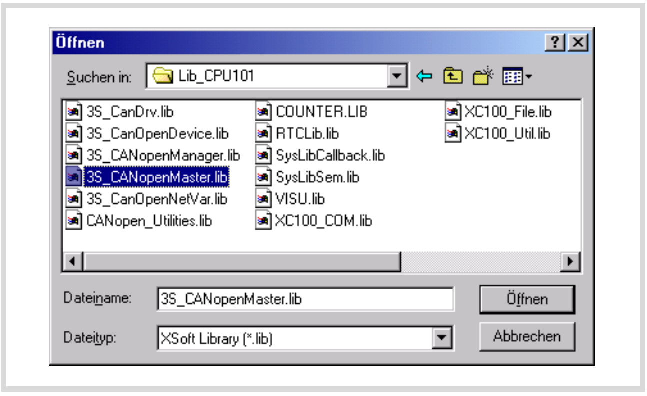

Öffnen
Suchen in: Lib_CPU101
3S_CanDrv.lib
3S_CanOpenDevice.lib
3S_CANopenManager.lib
3S_CANopenMaster.lib
3S_CanOpenNetVar.lib
CANopen_Utilities.lib
COUNTER.LIB
RTCLib.lib
SysLibCallback.lib
SysLibSem.lib
VISU.lib
XC100_COM.lib
XC100_File.lib
XC100_Util.lib
Dateiname: 3S_CANopenMaster.lib
Dateityp: XSoft Library (*.lib)
Öffnen
Abbrechen

Figure 50: Files: Lib_CPU101

After opening “Lib_CPU101”, select the “3S_CANopenMaster.lib” library and integrate it into the library manager with “Open”. All further CAN libraries which are required are then automatically added to the library manager.

→ Further steps and selection criteria which are necessary are described in a separate manual “Interfacing of an XI/ON slave to the XC100/XC200 via CANopen Fieldbus”.

This manual is provided as a PDF file (AN2700K18GB.PDF) on the CD.

The latest version of the manual can be found at http://www.moeller.net/support:
Search text: AN2700K18GB.

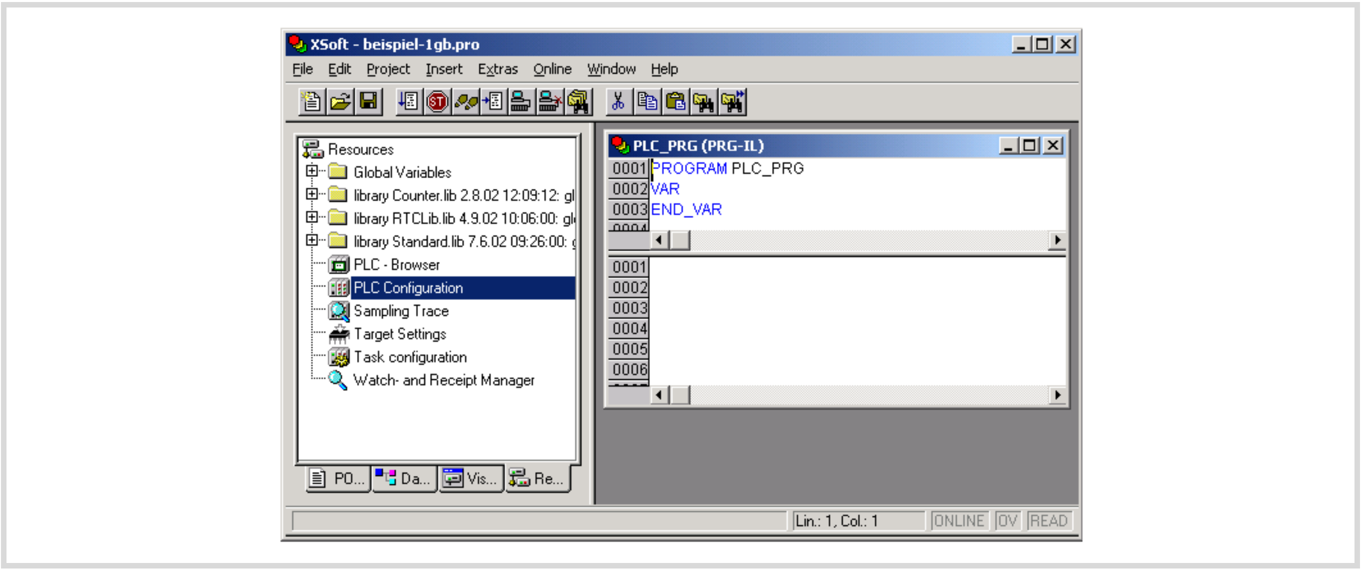
For Immediate Delivery call KMParts.com at (866) 595-9616

Configure the XC100

The example makes use of the "XC-CPU101-C64K-8DI-6DO".

▶ Select the "Resources" register (left half of window, at bottom), to configure the XC100 with the local and central inputs and outputs.

The following diagram will appear:

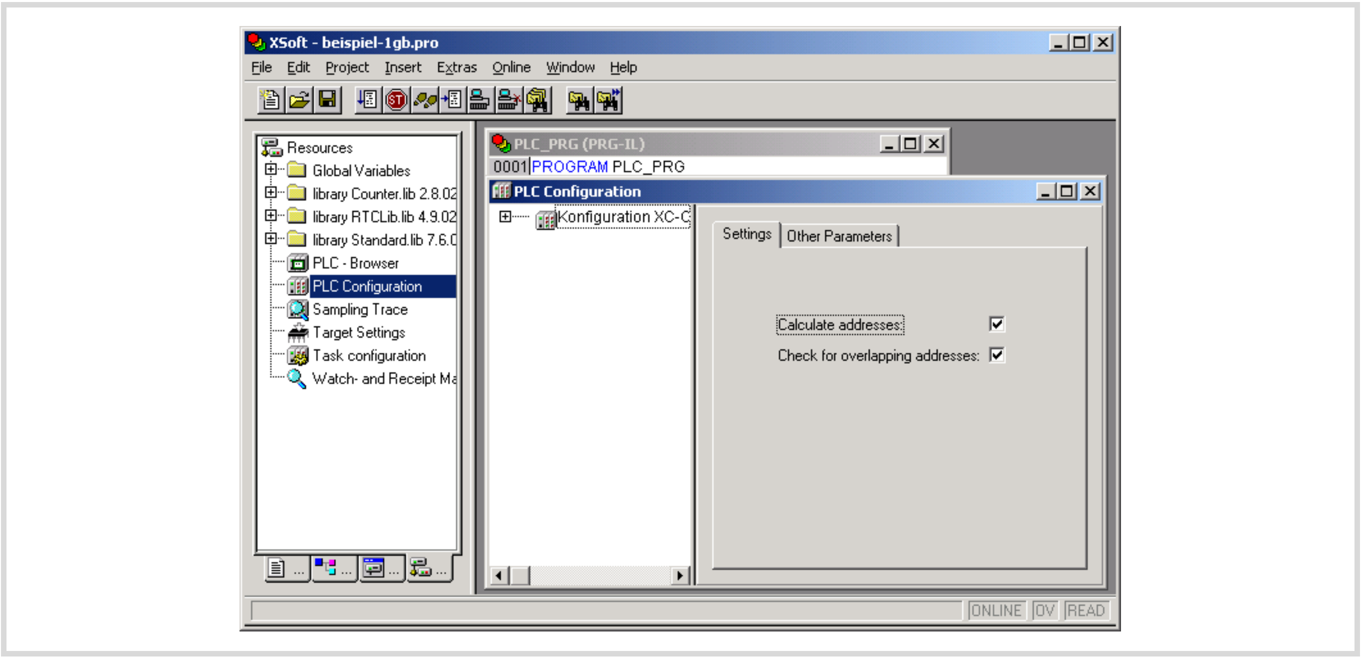


```
0001 PROGRAM PLC_PRG
0002 VAR
0003 END_VAR
```

Figure 51: Configure XC100 controller

▶ Double-click on the directory "Controller configuration".

Another window is opened: "PLC configuration":



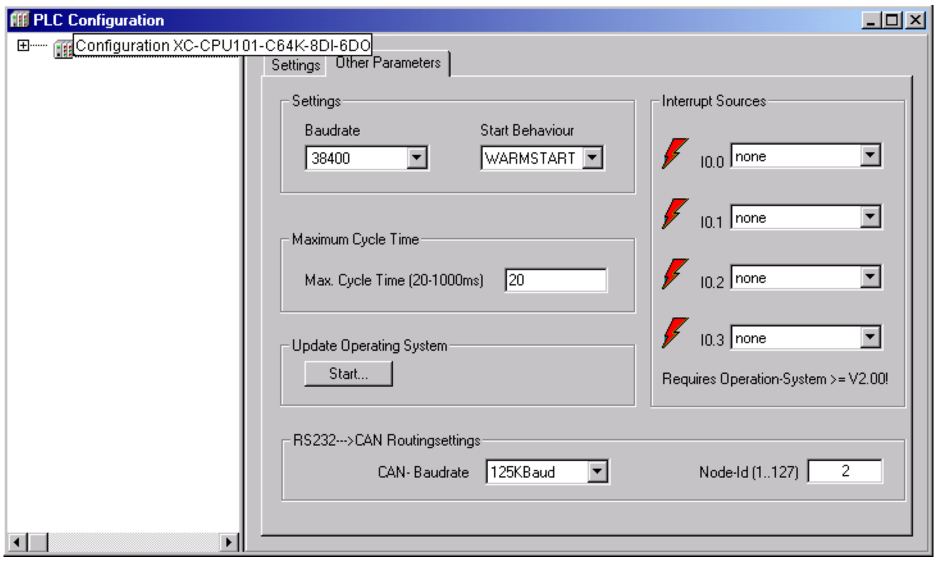
Calculate addresses: ☑
Check for overlapping addresses: ☑

Figure 52: Basic configuration of the XC100 – settings

▶ Click on the register card "Additional parameters".

A window appears with the default values for the "XC-CPU101-C64K-8DI-6DO".

For Immediate Delivery call KMParts.com at (866) 595-9616


PLC Configuration

Configuration XC-CPU101-C64K-8DI-6DO

Settings | Other Parameters

Settings
Baudrate: 38400
Start Behaviour: WARMSTART

Maximum Cycle Time
Max. Cycle Time (20-1000ms): 20

Update Operating System
Start...

Interrupt Sources
I0.0: none
I0.1: none
I0.2: none
I0.3: none
Requires Operation-System >= V2.00!

RS232-->CAN Routingsettings
CAN- Baudrate: 125KBaud
Node-Id (1..127): 2

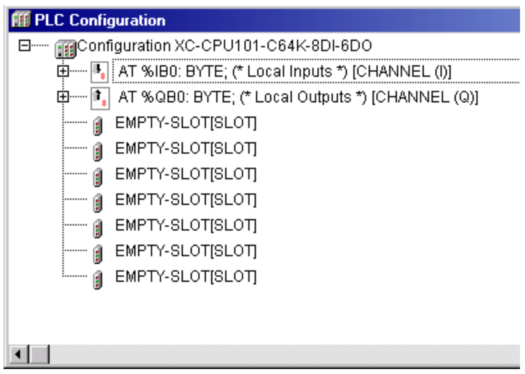
Figure 53: Basic configuration of the XC100 – additional parameters

▶ To display the I/O configuration, click on the plus sign in front of the directory “XC-CPU101-C64K-8DI-6DO”.

The local inputs and outputs (integral parts of the CPU) are already configured:

- “AT %IB0;Byte; (*Local Inputs*)”
- “AT %QB0;Byte; (*Local Outputs*)”

You can also set the parameters for up to seven central signal modules. The slots “EMPTY-SLOT” are wildcards for central expansion of the signal modules.


PLC Configuration

```
Configuration XC-CPU101-C64K-8DI-6DO
  AT %IB0: BYTE; (* Local Inputs *) [CHANNEL (I)]
  AT %QB0: BYTE; (* Local Outputs *) [CHANNEL (Q)]
  EMPTY-SLOT[SLOT]
  EMPTY-SLOT[SLOT]
  EMPTY-SLOT[SLOT]
  EMPTY-SLOT[SLOT]
  EMPTY-SLOT[SLOT]
  EMPTY-SLOT[SLOT]
  EMPTY-SLOT[SLOT]
```

Figure 54: Basic configuration of the XC100 – local I/Os

If you want to join up a central digital input module with 16 inputs, right next to the CPU, then carry out the following steps:

▶ Mark the first “EMPTY-SLOT” and then click the right mouse button.

A window is opened

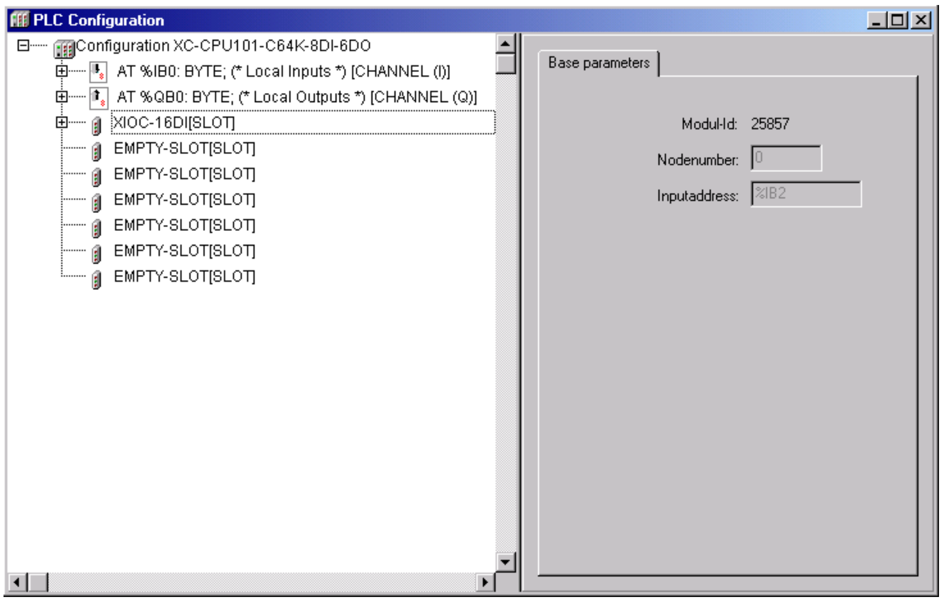
▶ Select the field “Replace element”.

The window that is now opened lists the signal modules which are available.

▶ Select the module “XIOC-16DI”.

The configuration now looks as follows:

For Immediate Delivery call KMParts.com at (866) 595-9616



PLC Configuration

```
Configuration XC-CPU101-C64K-8DI-6DO
    AT %IB0: BYTE; (* Local Inputs *) [CHANNEL (I)]
    AT %QB0: BYTE; (* Local Outputs *) [CHANNEL (Q)]
    XIOC-16DI[SLOT]
    EMPTY-SLOT[SLOT]
    EMPTY-SLOT[SLOT]
    EMPTY-SLOT[SLOT]
    EMPTY-SLOT[SLOT]
    EMPTY-SLOT[SLOT]
    EMPTY-SLOT[SLOT]
```

Base parameters

Modul-Id: 25857

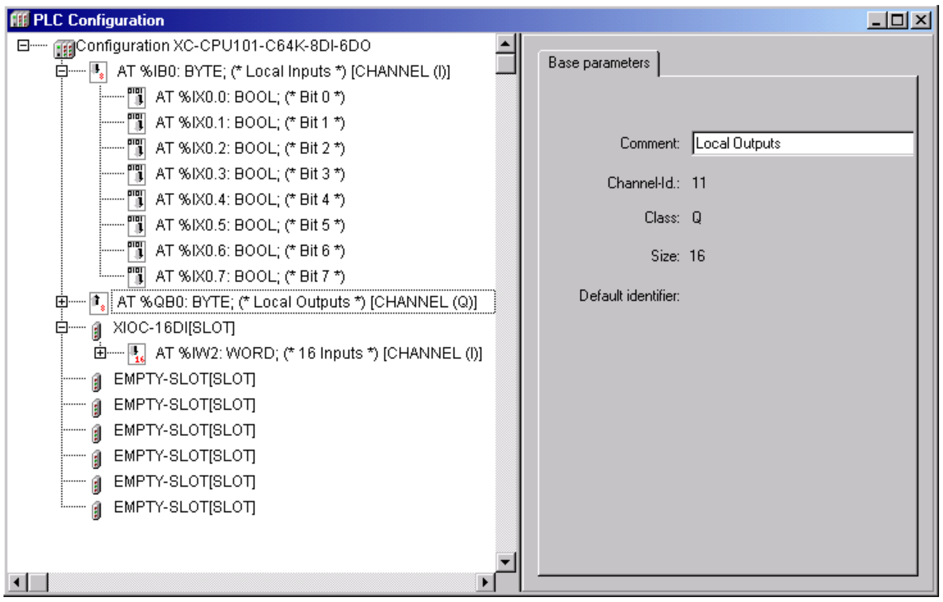
Nodenumber: 0

Inputaddress: %IB2

Figure 55: Configuration XC-CPU101-C64K-8DI-6DO

▶ In addition, click on the plus sign in front of the modules
- "AT %IB0;Byte; (*Local Inputs*)"
- "AT %QB0;Byte; (*Local Outputs*)"
- XIOC-16DI (SLOT).

You will now get detailed information, with the physical address area of the inputs and outputs.



PLC Configuration

```
Configuration XC-CPU101-C64K-8DI-6DO
    AT %IB0: BYTE; (* Local Inputs *) [CHANNEL (I)]
        AT %IX0.0: BOOL; (* Bit 0 *)
        AT %IX0.1: BOOL; (* Bit 1 *)
        AT %IX0.2: BOOL; (* Bit 2 *)
        AT %IX0.3: BOOL; (* Bit 3 *)
        AT %IX0.4: BOOL; (* Bit 4 *)
        AT %IX0.5: BOOL; (* Bit 5 *)
        AT %IX0.6: BOOL; (* Bit 6 *)
        AT %IX0.7: BOOL; (* Bit 7 *)
    AT %QB0: BYTE; (* Local Outputs *) [CHANNEL (Q)]
    XIOC-16DI[SLOT]
        AT %IW2: WORD; (* 16 Inputs *) [CHANNEL (I)]
    EMPTY-SLOT[SLOT]
    EMPTY-SLOT[SLOT]
    EMPTY-SLOT[SLOT]
    EMPTY-SLOT[SLOT]
    EMPTY-SLOT[SLOT]
    EMPTY-SLOT[SLOT]
```

Base parameters

Comment: Local Outputs

Channel-Id.: 11

Class: Q

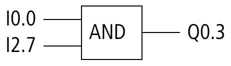
Size: 16

Default identifier:

Figure 56: Address area of the configuration

Create program

As described in the description of the task on Page 34, a logical AND combination is to be made between input I0.0 and input I2.7. The result of this logical operation is to be presented at output Q0.3.

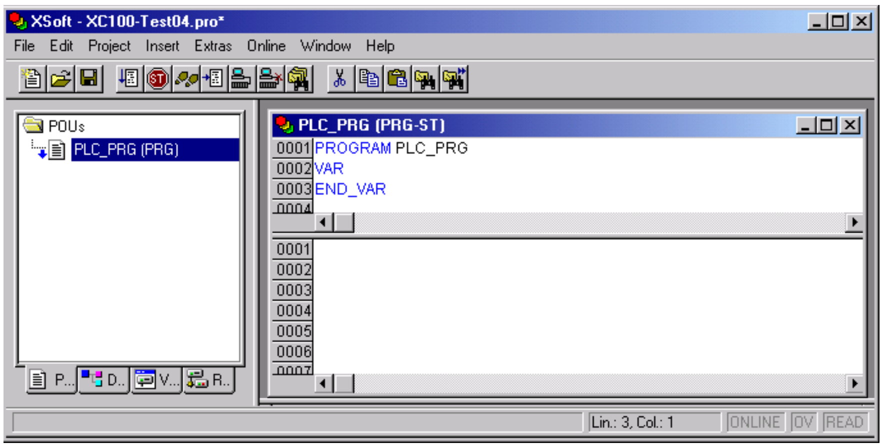


▶ Select the register "POUs"

▶ Double-click on the element "PLC-PRG"

The declaration and program window will be opened.

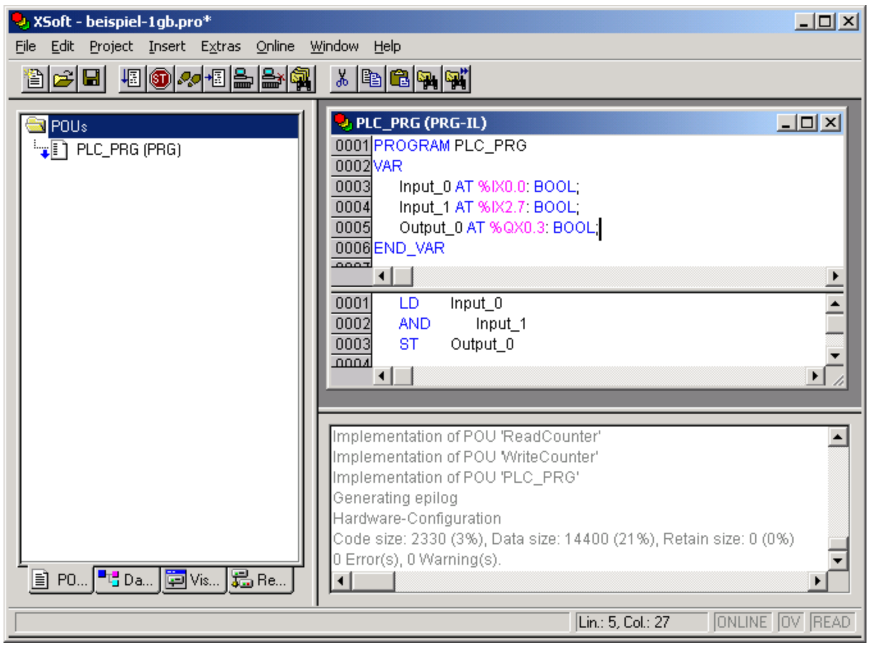
For Immediate Delivery call KMParts.com at (866) 595-9616



```
PROGRAM PLC_PRG
VAR
END_VAR
```

Figure 57: Program and declaration window

▶ Create the declaration and the program, as shown in the following diagram, and then compile the project.



```
PROGRAM PLC_PRG
VAR
    Input_0 AT %IX0.0: BOOL;
    Input_1 AT %IX2.7: BOOL;
    Output_0 AT %QX0.3: BOOL;
END_VAR
```

```
    LD      Input_0
    AND         Input_1
    ST      Output_0
```

```
Implementation of POU 'ReadCounter'
Implementation of POU 'WriteCounter'
Implementation of POU 'PLC_PRG'
Generating epilog
Hardware-Configuration
Code size: 2330 (3%), Data size: 14400 (21%), Retain size: 0 (0%)
0 Error(s), 0 Warning(s).
```

Figure 58: Compiled program

▶ Load the project into the controller.

▶ Test the project.

For Immediate Delivery call KMParts.com at (866) 595-9616

CANopen bus expansion

The description of the CANopen bus extension can be found in the application notes:

- XC...-XION (AN2700K18GB),
- XC...-XC using network variables (AN2700K19GB),
- Coupling multiple autonomous controls (CAN-Device) via CANopen (AN2700K-20GB).

In the next section, a decentral CANopen module (in this case: "WINbloc CAN 3AI UI 1AO UI") will be added to the sample project:

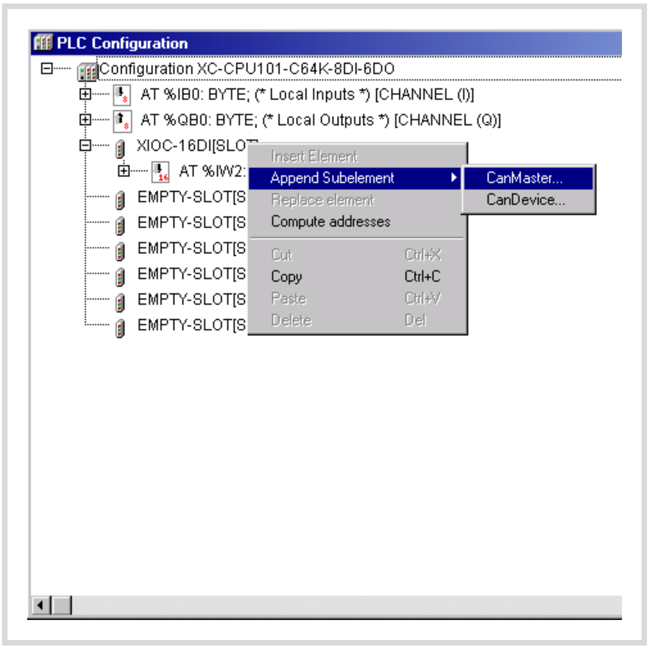
→ The CANopen interface is a standard on-board interface for the XC100, in addition to the RS 232 interface.

Controller configuration

If you want to incorporate a decentralised CANopen module, the controller configuration must be expanded accordingly:

▶ In the "Resources" window, mark the element "Controller configuration" and select the configuration "XC-CPU101-C64K-8DI-6DO".

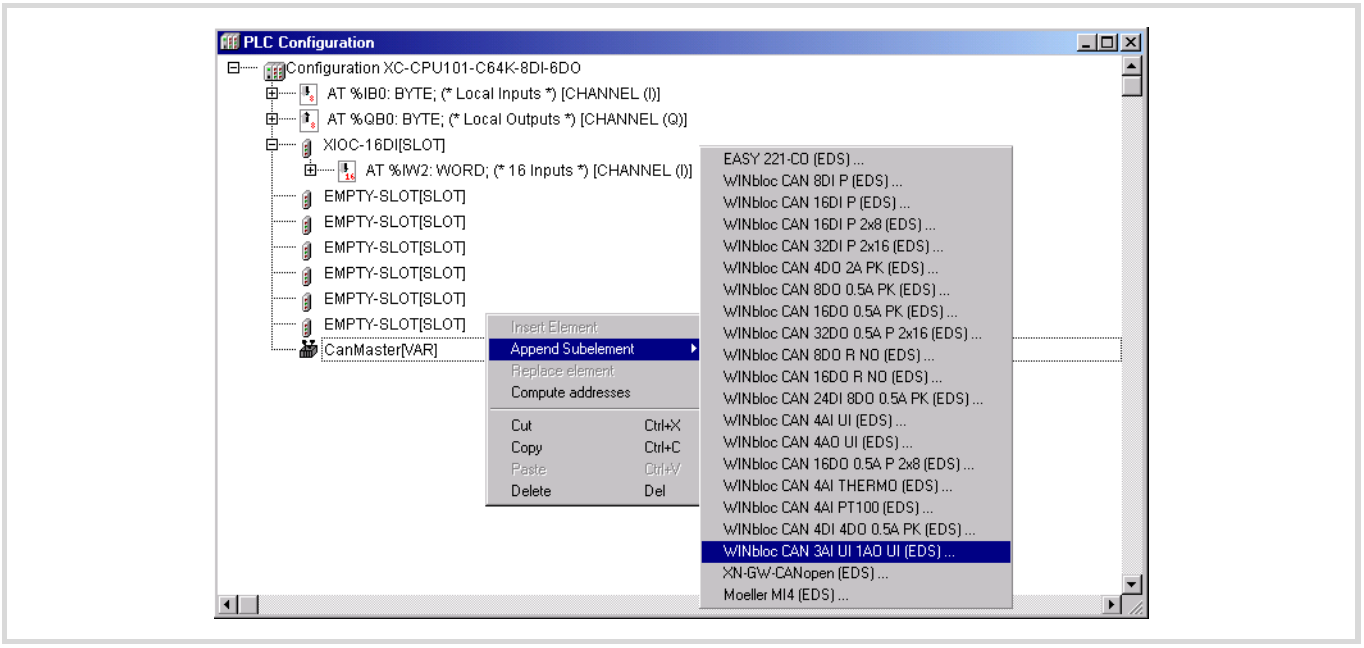
▶ Right click in the "Configuration XC-CPU101-..." field and select the context menu "Append Subelement → CanMaster" (→ figure 59).



```
Configuration XC-CPU101-C64K-8DI-6DO
  AT %IB0: BYTE; (* Local Inputs *) [CHANNEL (I)]
  AT %QB0: BYTE; (* Local Outputs *) [CHANNEL (Q)]
  XIOC-16DI[SLOT
    AT %IW2:
  EMPTY-SLOT[S
  EMPTY-SLOT[S
  EMPTY-SLOT[S
  EMPTY-SLOT[S
  EMPTY-SLOT[S
  EMPTY-SLOT[S
```

Figure 59: PLC configuration with a CAN Master

▶ Expand the configuration with the element "WINbloc CAN 3AI UI 1AO UI":
– Click in the field "CanMaster(VAR)", click on the right mouse button, and select the command "Append subelement".
– From the list of CAN modules, select the module "WINbloc CAN 3AI UI 1AO UI (EDS)":



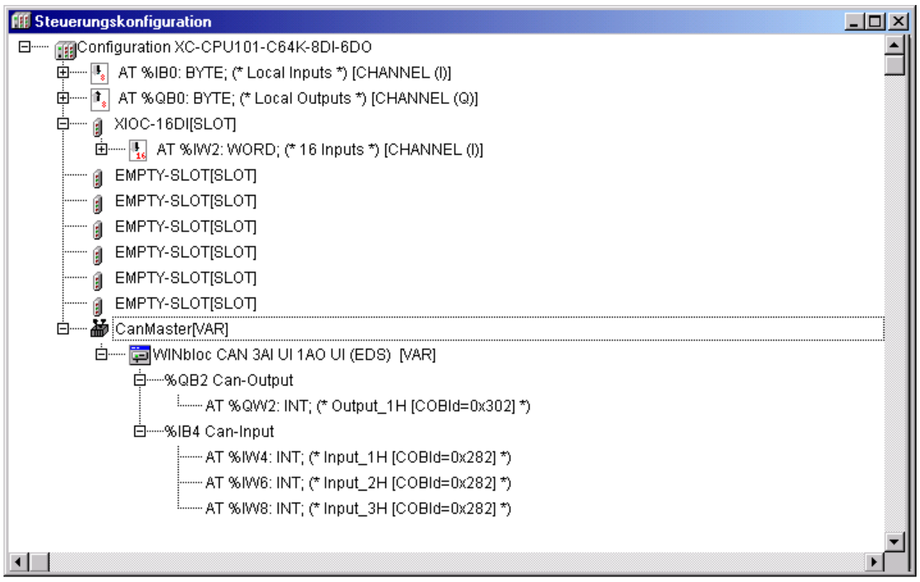
```
Configuration XC-CPU101-C64K-8DI-6DO
  AT %IB0: BYTE; (* Local Inputs *) [CHANNEL (I)]
  AT %QB0: BYTE; (* Local Outputs *) [CHANNEL (Q)]
  XIOC-16DI[SLOT]
    AT %IW2: WORD; (* 16 Inputs *) [CHANNEL (I)]
  EMPTY-SLOT[SLOT]
  EMPTY-SLOT[SLOT]
  EMPTY-SLOT[SLOT]
  EMPTY-SLOT[SLOT]
  EMPTY-SLOT[SLOT]
  EMPTY-SLOT[SLOT]
  CanMaster[VAR]
```

PLC Configuration
Insert Element
Append Subelement
Replace element
Compute addresses
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
EASY 221-CO (EDS) ...
WINbloc CAN 8DI P (EDS) ...
WINbloc CAN 16DI P (EDS) ...
WINbloc CAN 16DI P 2x8 (EDS) ...
WINbloc CAN 32DI P 2x16 (EDS) ...
WINbloc CAN 4DO 2A PK (EDS) ...
WINbloc CAN 8DO 0.5A PK (EDS) ...
WINbloc CAN 16DO 0.5A PK (EDS) ...
WINbloc CAN 32DO 0.5A P 2x16 (EDS) ...
WINbloc CAN 8DO R NO (EDS) ...
WINbloc CAN 16DO R NO (EDS) ...
WINbloc CAN 24DI 8DO 0.5A PK (EDS) ...
WINbloc CAN 4AI UI (EDS) ...
WINbloc CAN 4AO UI (EDS) ...
WINbloc CAN 16DO 0.5A P 2x8 (EDS) ...
WINbloc CAN 4AI THERMO (EDS) ...
WINbloc CAN 4AI PT100 (EDS) ...
WINbloc CAN 4DI 4DO 0.5A PK (EDS) ...
WINbloc CAN 3AI UI 1AO UI (EDS) ...
XN-GW-CANopen (EDS) ...
Moeller MI4 (EDS) ...

Figure 60: Selection of the CAN module

If you click on the plus sign in front of the individual entries for the CAN bus expansion, you will get the following display after selection of the CAN module.

For Immediate Delivery call KMParts.com at (866) 595-9616

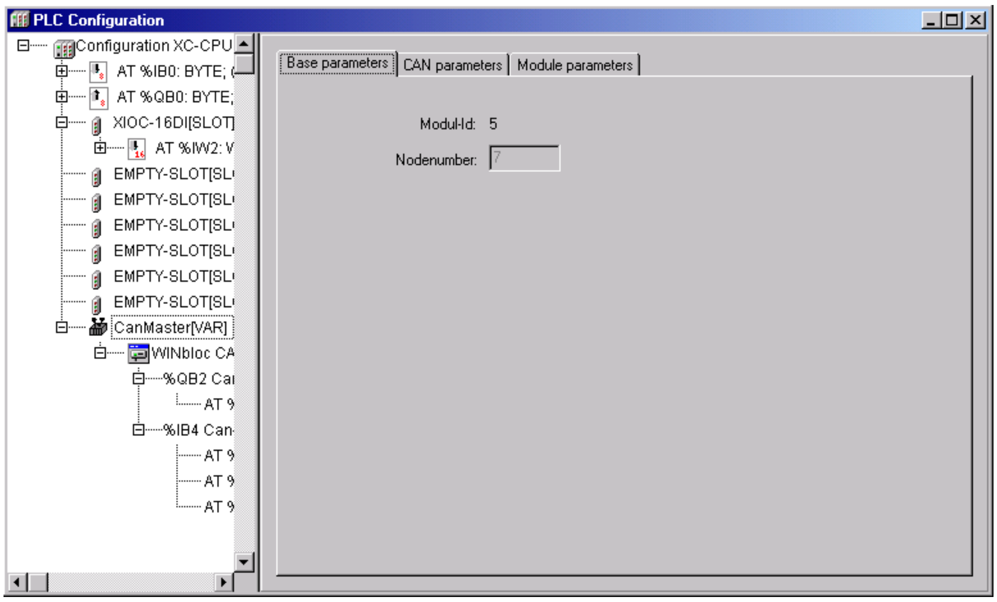


Steuerungskonfiguration

```
Configuration XC-CPU101-C64K-8DI-6DO
    AT %IB0: BYTE; (* Local Inputs *) [CHANNEL (I)]
    AT %QB0: BYTE; (* Local Outputs *) [CHANNEL (Q)]
    XIOC-16DI[SLOT]
        AT %IW2: WORD; (* 16 Inputs *) [CHANNEL (I)]
    EMPTY-SLOT[SLOT]
    EMPTY-SLOT[SLOT]
    EMPTY-SLOT[SLOT]
    EMPTY-SLOT[SLOT]
    EMPTY-SLOT[SLOT]
    EMPTY-SLOT[SLOT]
    CanMaster[VAR]
        WINbloc CAN 3AI UI 1AO UI (EDS)  [VAR]
            %QB2 Can-Output
                AT %QW2: INT; (* Output_1H [COBId=0x302] *)
            %IB4 Can-Input
                AT %IW4: INT; (* Input_1H [COBId=0x282] *)
                AT %IW6: INT; (* Input_2H [COBId=0x282] *)
                AT %IW8: INT; (* Input_3H [COBId=0x282] *)
```

Figure 61: Detailed display of the CAN slave configuration

The basic, CAN, and module parameters for CanMaster (VAR) can be seen by calling up the individual register cards. These are shown in the three following diagrams:



PLC Configuration

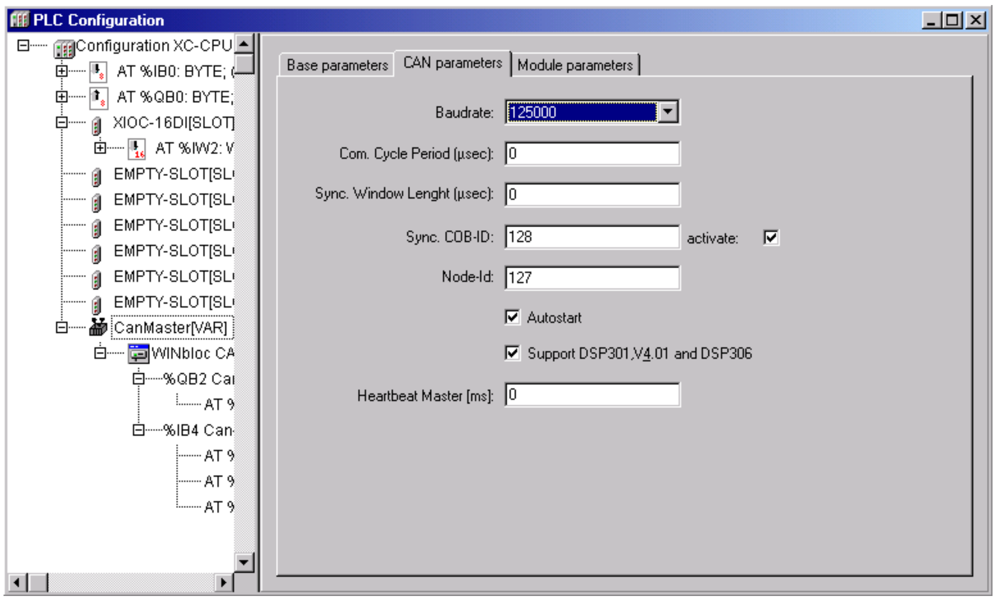
Base parameters | CAN parameters | Module parameters

Modul-Id: 5

Nodenumber: 7

Figure 62: CAN Master configuration – basic parameters

For Immediate Delivery call KMParts.com at (866) 595-9616



Baudrate: 125000

Com. Cycle Period (µsec): 0

Sync. Window Lenght (µsec): 0

Sync. COB-ID: 128 activate: ☑

Node-Id: 127

☑ Autostart

☑ Support DSP301,V4.01 and DSP306

Heartbeat Master [ms]: 0

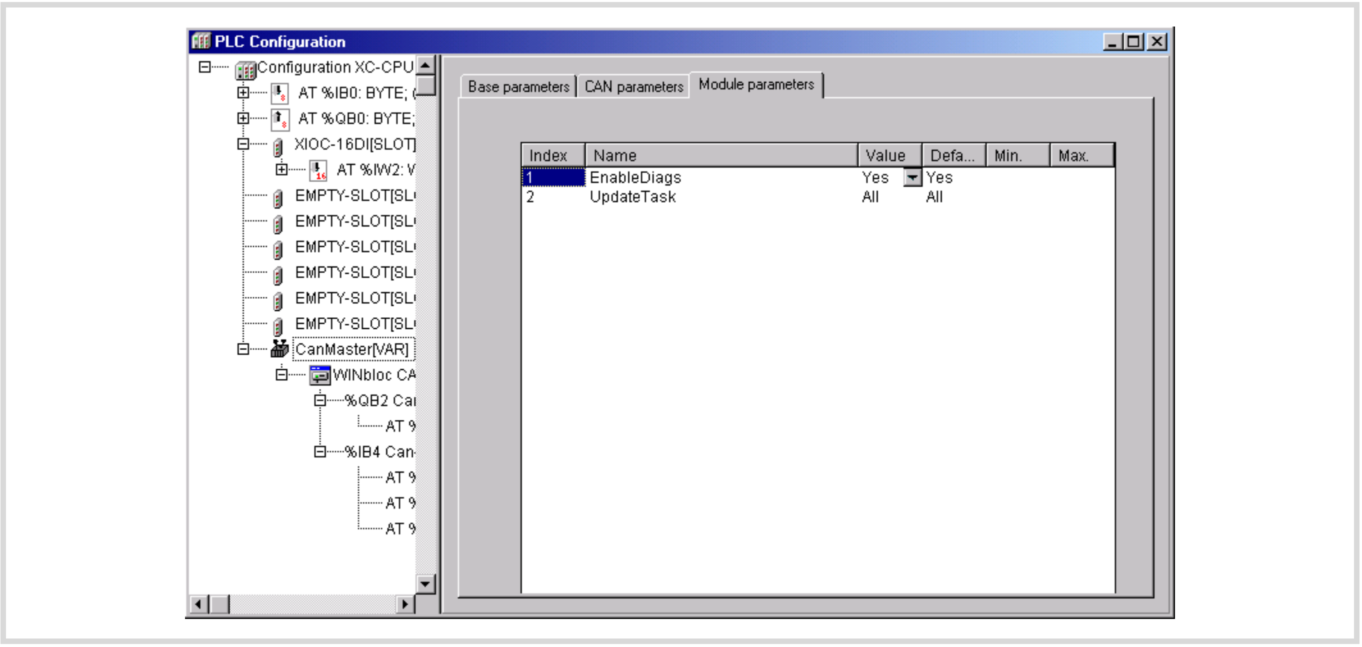
Figure 63: CAN Master configuration – CAN parameters

The usable baud rate for the CAN connection depends on the version of the operating system:

Table 3: Usable baud rates for CAN connection

	Operating system version	
Baud rate	**< V. 2.0**	**V 2.0**
50000	✓	✓
100000	✓	✓
125000	✓	✓
250000	✓	✓
500000	–	✓
1000000	–	✓

For Immediate Delivery call KMParts.com at (866) 595-9616



PLC Configuration

Base parameters | CAN parameters | Module parameters

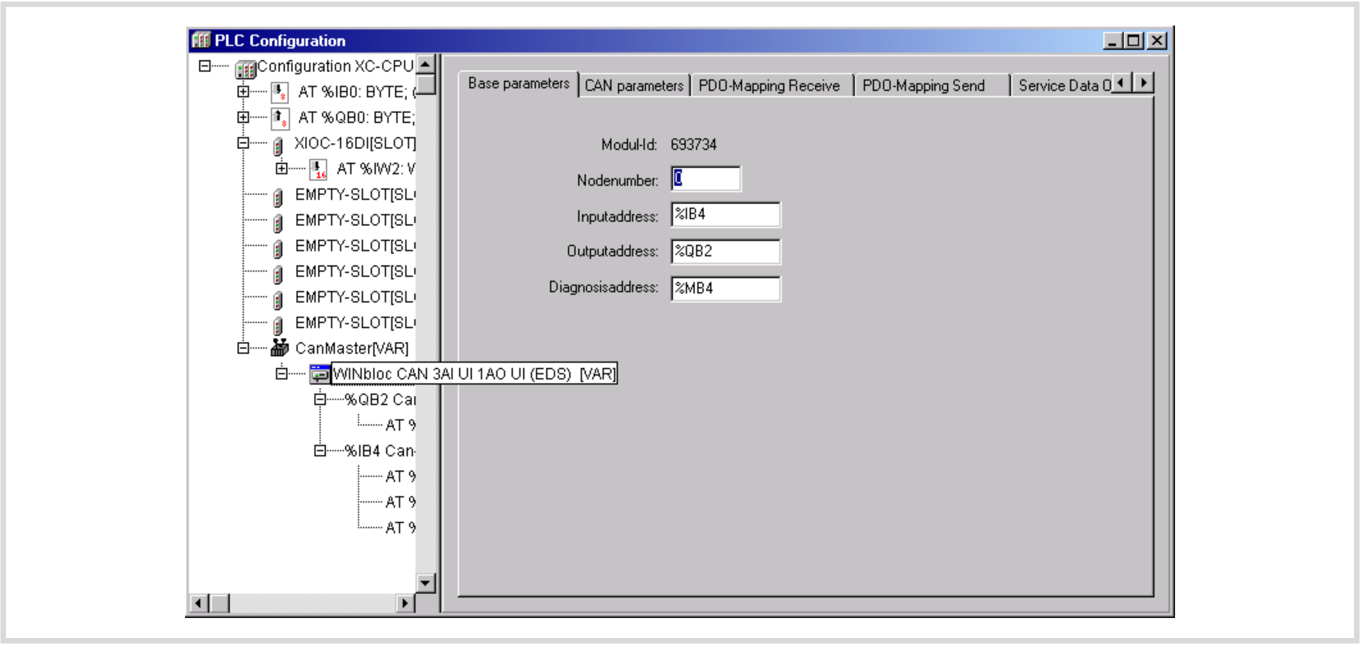
Index	Name	Value	Defa...	Min.	Max.
1	EnableDiags	Yes	Yes		
2	UpdateTask	All	All		

Figure 64: CAN Master configuration – module parameters

The individual parameter details for the CAN-WINbloc module are shown in the following diagrams.

▶ Select the module "Winbloc CAN 3AI UI 1AO UI (EDS) (VAR)" in the configuration field, and call up the individual register cards.

Further information on this CANopen module can be found in the manual "WINbloc CAN", AWB2700-1385GB.



PLC Configuration

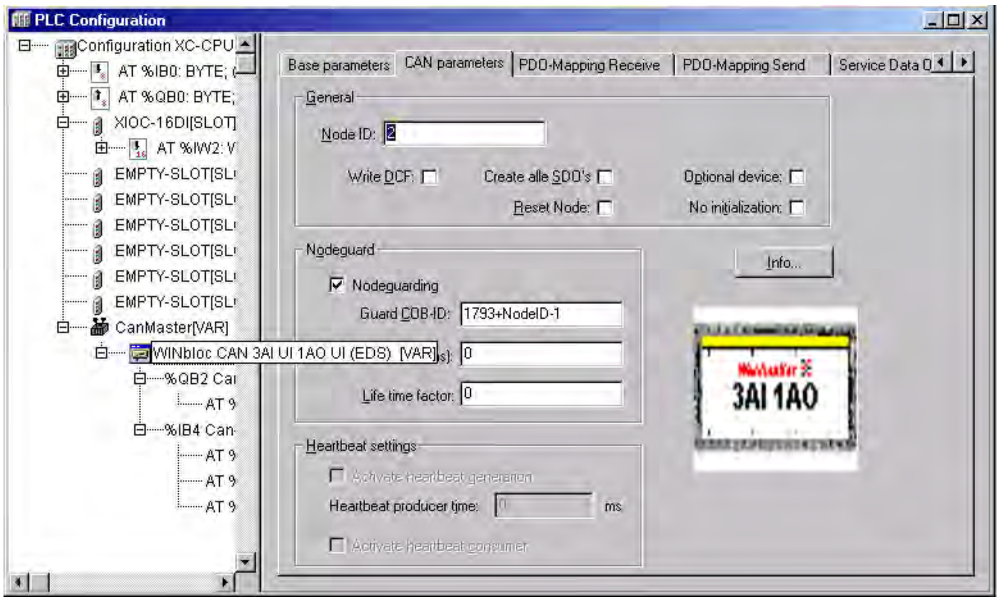
Base parameters | CAN parameters | PDO-Mapping Receive | PDO-Mapping Send | Service Data O

Modul-Id:	693734
Nodenumber:	0
Inputaddress:	%IB4
Outputaddress:	%QB2
Diagnosisaddress:	%MB4

WINbloc CAN 3AI UI 1AO UI (EDS) [VAR]

Figure 65: CAN-WINbloc configuration – basic parameters

For Immediate Delivery call KMParts.com at (866) 595-9616



PLC Configuration

```
Configuration XC-CPU
  AT %IB0: BYTE;
  AT %QB0: BYTE;
  XIOC-16DI[SLOT]
    AT %IW2: V
  EMPTY-SLOT[SL
  EMPTY-SLOT[SL
  EMPTY-SLOT[SL
  EMPTY-SLOT[SL
  EMPTY-SLOT[SL
  EMPTY-SLOT[SL
  CanMaster[VAR]
    WINbloc CAN 3AI UI 1AO UI (EDS) [VAR]
      %QB2 Ca
        AT %
      %IB4 Can
        AT %
        AT %
        AT %
```

Base parameters | CAN parameters | PDO-Mapping Receive | PDO-Mapping Send | Service Data O

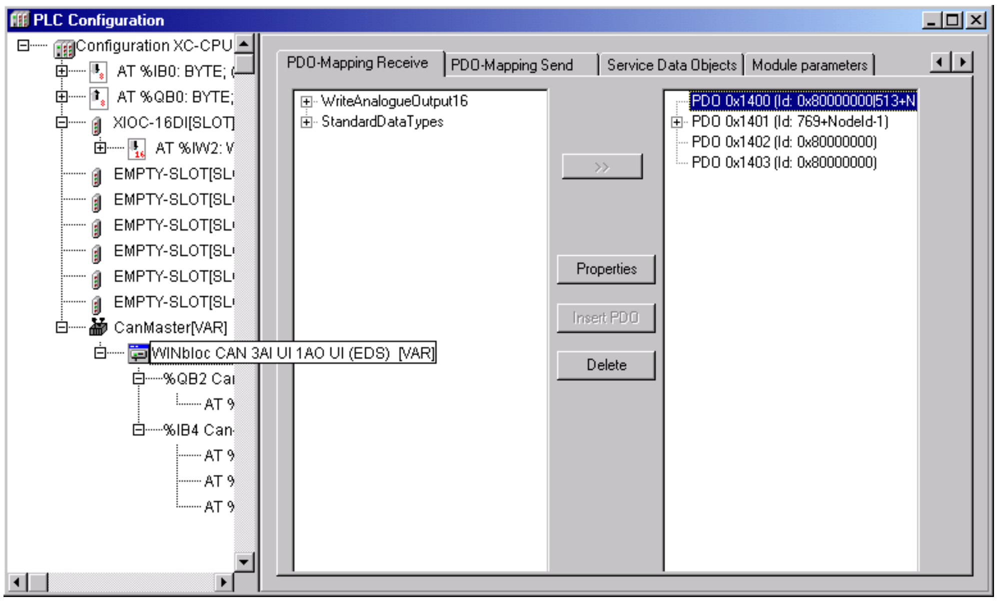
General
Node ID: 2
Write DCF: ☐ Create alle SDO's: ☐ Optional device: ☐
Reset Node: ☐ No initialization: ☐

Nodeguard
☑ Nodeguarding
Guard COB-ID: 1793+NodeID-1
Guard Time (ms): 0
Life time factor: 0

Info...

Heartbeat settings
☐ Activate heartbeat generation
Heartbeat producer time: 0 ms
☐ Activate heartbeat consumer

Figure 66: CAN-WINbloc configuration – CAN parameters



PLC Configuration

```
Configuration XC-CPU
  AT %IB0: BYTE;
  AT %QB0: BYTE;
  XIOC-16DI[SLOT]
    AT %IW2: V
  EMPTY-SLOT[SL
  EMPTY-SLOT[SL
  EMPTY-SLOT[SL
  EMPTY-SLOT[SL
  EMPTY-SLOT[SL
  EMPTY-SLOT[SL
  CanMaster[VAR]
    WINbloc CAN 3AI UI 1AO UI (EDS) [VAR]
      %QB2 Ca
        AT %
      %IB4 Can
        AT %
        AT %
        AT %
```

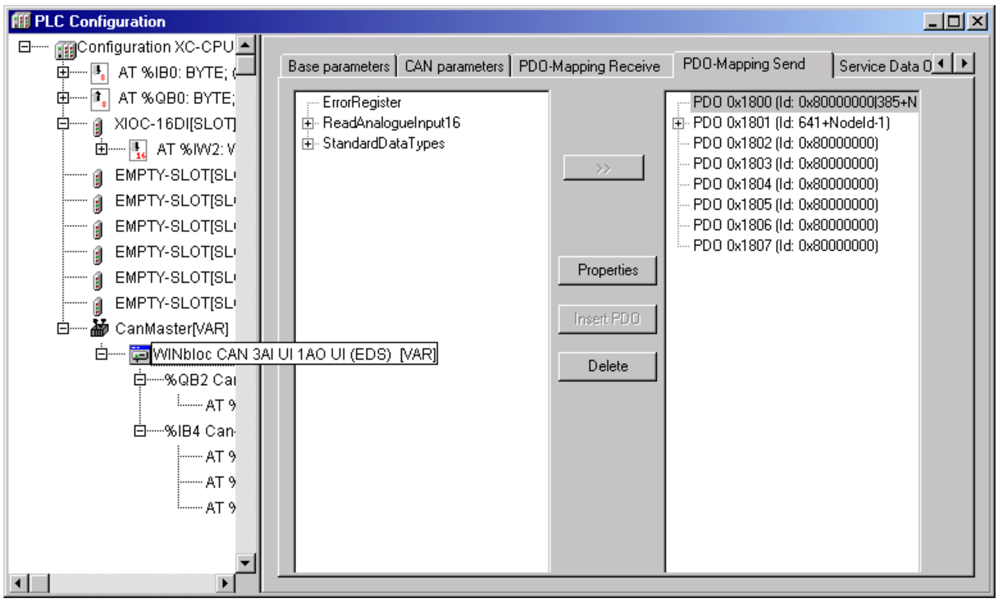
PDO-Mapping Receive | PDO-Mapping Send | Service Data Objects | Module parameters

```
WriteAnalogueOutput16
StandardDataTypes
```

```
PDO 0x1400 (Id: 0x80000000|513+N
PDO 0x1401 (Id: 769+NodeId-1)
PDO 0x1402 (Id: 0x80000000)
PDO 0x1403 (Id: 0x80000000)
```

Figure 67: CAN-WINbloc configuration – receive PDO mapping

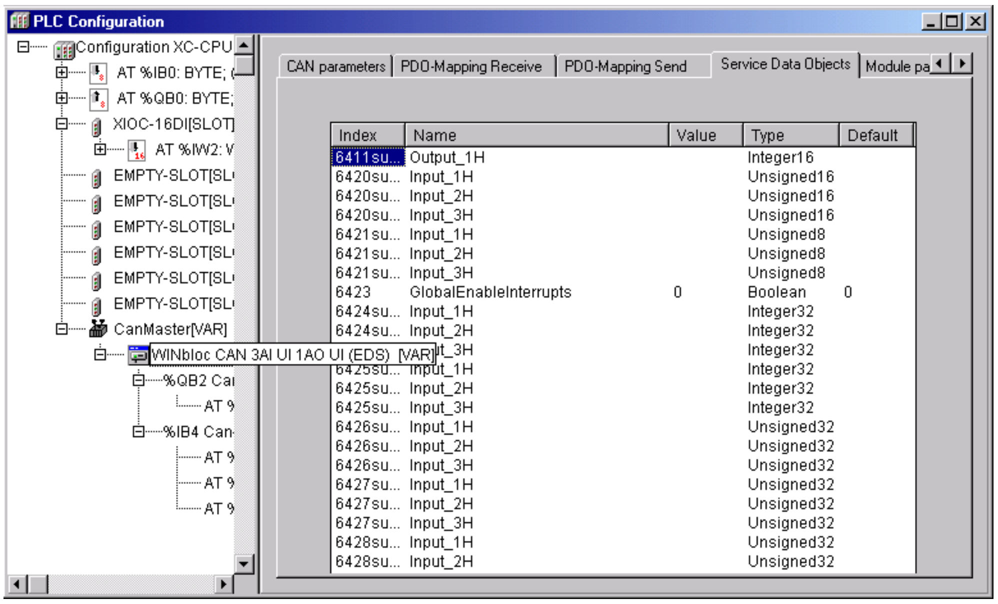
For Immediate Delivery call KMParts.com at (866) 595-9616



ErrorRegister
ReadAnalogueInput16
StandardDataTypes

PDO 0x1800 (Id: 0x80000000|385+N
PDO 0x1801 (Id: 641+NodeId-1)
PDO 0x1802 (Id: 0x80000000)
PDO 0x1803 (Id: 0x80000000)
PDO 0x1804 (Id: 0x80000000)
PDO 0x1805 (Id: 0x80000000)
PDO 0x1806 (Id: 0x80000000)
PDO 0x1807 (Id: 0x80000000)

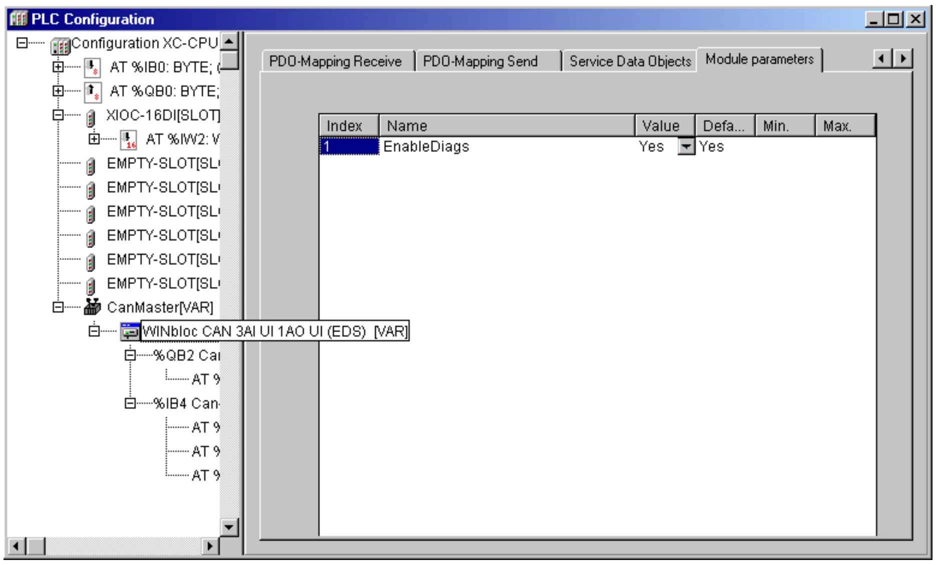
Figure 68: CAN-WINbloc configuration – transmit PDO mapping



Index	Name	Value	Type	Default
6411su...	Output_1H		Integer16	
6420su...	Input_1H		Unsigned16	
6420su...	Input_2H		Unsigned16	
6420su...	Input_3H		Unsigned16	
6421su...	Input_1H		Unsigned8	
6421su...	Input_2H		Unsigned8	
6421su...	Input_3H		Unsigned8	
6423	GlobalEnableInterrupts	0	Boolean	0
6424su...	Input_1H		Integer32	
6424su...	Input_2H		Integer32	
[illegible]	t_3H		Integer32	
6425su...	Input_1H		Integer32	
6425su...	Input_2H		Integer32	
6425su...	Input_3H		Integer32	
6426su...	Input_1H		Unsigned32	
6426su...	Input_2H		Unsigned32	
6426su...	Input_3H		Unsigned32	
6427su...	Input_1H		Unsigned32	
6427su...	Input_2H		Unsigned32	
6427su...	Input_3H		Unsigned32	
6428su...	Input_1H		Unsigned32	
6428su...	Input_2H		Unsigned32	

Figure 69: CAN-WINbloc configuration – Service Data Objects

For Immediate Delivery call KMParts.com at (866) 595-9616



Index	Name	Value	Defa...	Min.	Max.
1	EnableDiags	Yes	Yes		

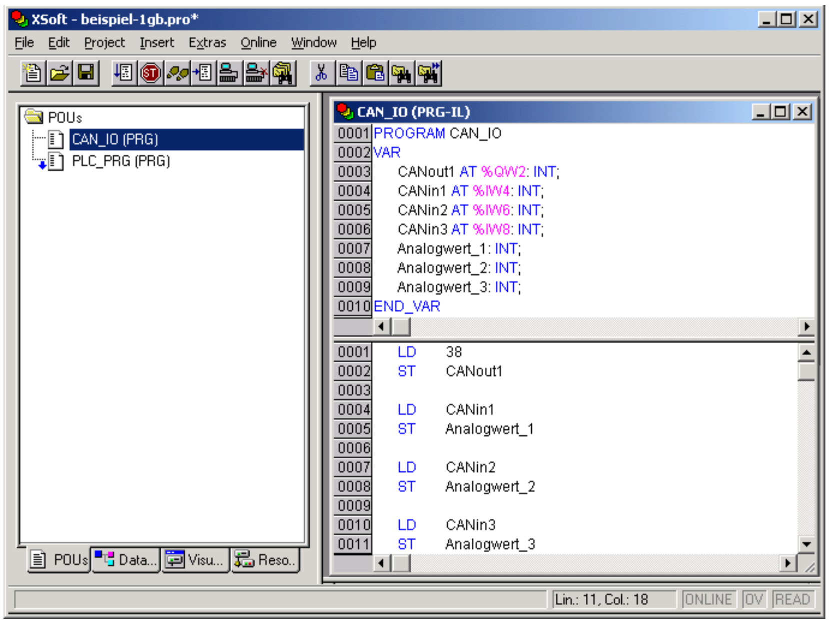
Figure 70: CAN-WINbloc configuration – module parameters

Create program for decentralised inputs/outputs on the CANopen bus

The module that is used as an example can read in 3 analog values, and output 1 analog value.

The example is saved as a POU “CAN_IO”, and called from the POU “PLC_PRG”. The following steps are required to program the new POU:

- ▶ Mark the directory “POUs” in the “POUs” register.
- ▶ Select the action “Insert Object”.
- ▶ Select the type “Program” and the language “IL” for the POU.
- ▶ Assign the name “CAN_IO” and confirm this with “OK”.
- ▶ Declare the variables and enter the program:



```
PROGRAM CAN_IO
VAR
    CANout1 AT %QW2: INT;
    CANin1 AT %IW4: INT;
    CANin2 AT %IW6: INT;
    CANin3 AT %IW8: INT;
    Analogwert_1: INT;
    Analogwert_2: INT;
    Analogwert_3: INT;
END_VAR
```

```
    LD      38
    ST      CANout1

    LD      CANin1
    ST      Analogwert_1

    LD      CANin2
    ST      Analogwert_2

    LD      CANin3
    ST      Analogwert_3
```

Figure 71: Program for CANopen expansion

- ▶ In the POU “PLC_PRG” insert the instruction “CAL CAN_IO”, so that the program “CAN_IO” can be called up from the controller program.
- ▶ Save the project.
- ▶ Compile the project and transfer it to the controller for testing.

For Immediate Delivery call KMParts.com at (866) 595-9616

Routing

From programming PC to CANopen fieldbus slave

It is possible to establish a connection to a lower level XC100 PLC (CANopen fieldbus slave) from the programming PC via a master control. The CANopen fieldbus station can be programmed and operated via this routing.

The following online functionality's for the CAN master control and for the CAN fieldbus slave are available:

- Program download
- Online modification
- Test and commissioning
- Create bootable project
- Filing source code.

The communication path from the programming PC via the XC100 master to the CANopen slave is indicated in Figure 72:

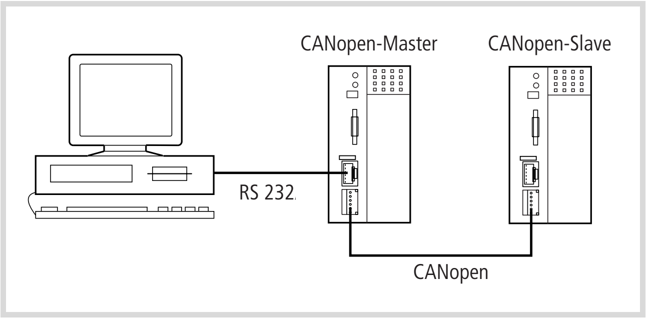
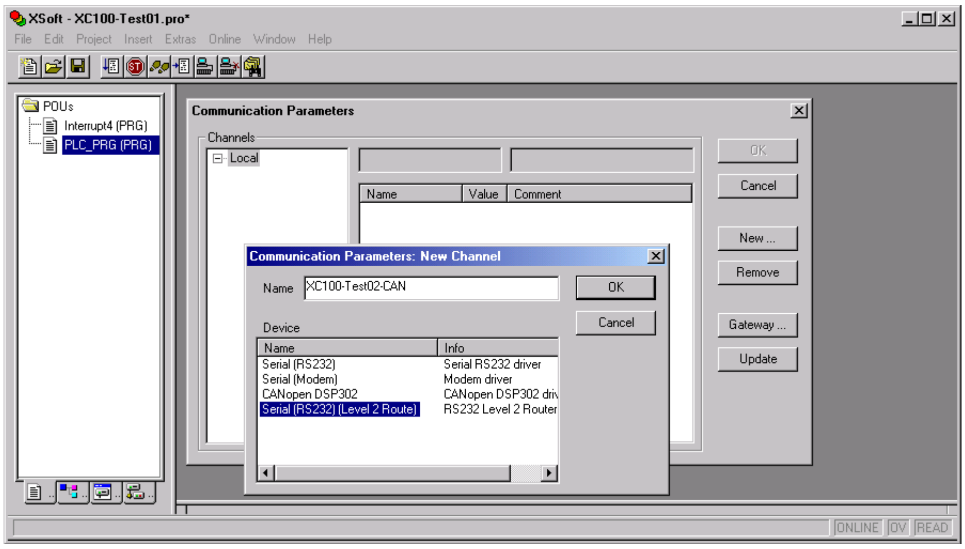


Figure 72: Routing RS 232 – CAN fieldbus slave

The communication relationship between the programming PC and the CANopen fieldbus slave can be established in the XSoft by the selection of the respective communication parameters. The following figures indicate the steps required leading to the parameter setting.

▶ Call up the ‹Online → Communication Parameters› in the XSoft and establish a new communication channel.

Select the "Serial [RS232] [Level 2 Route"] default setting in the "Device" field in the window which opens and confirm with OK.



Communication Parameters: New Channel

Name: XC100-Test02-CAN

Device

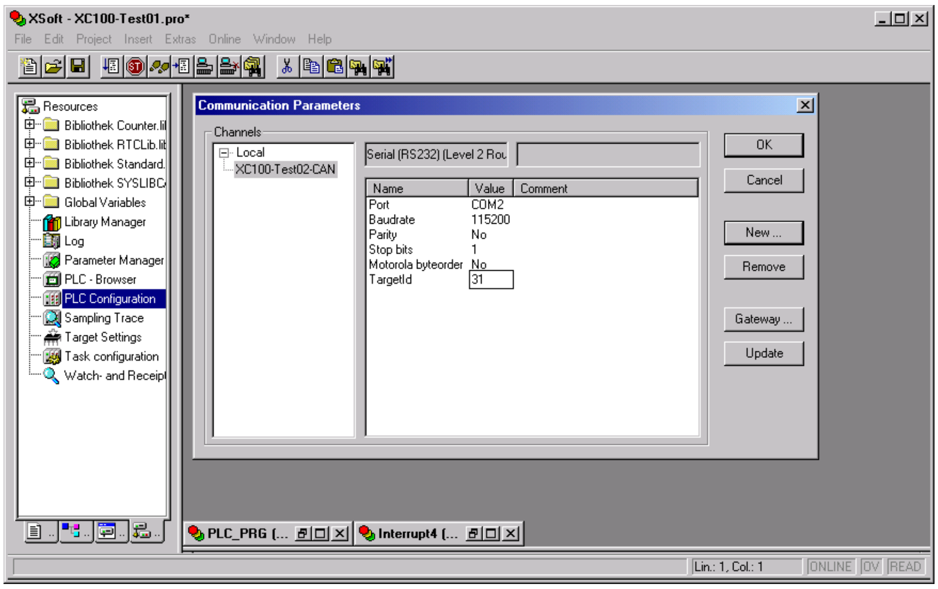
Name	Info
Serial (RS232)	Serial RS232 driver
Serial (Modem)	Modem driver
CANopen DSP302	CANopen DSP302 driv
Serial (RS232) (Level 2 Route)	RS232 Level 2 Router

Figure 73: Setting the RS 232 – CAN slave communication parameters

You will receive the following figure.

▶ Enter the selected "TargetID", e. g. "31", and confirm with OK in this window.
▶ Ensure that this "Target-ID" is also entered for the respective CAN fieldbus slave.

For Immediate Delivery call KMParts.com at (866) 595-9616



Communication Parameters

Channels

- Local
  - XC100-Test02-CAN

Serial (RS232) (Level 2 Rou

Name	Value	Comment
Port	COM2	
Baudrate	115200	
Parity	No	
Stop bits	1	
Motorola byteorder	No	
TargetId	31	

Figure 74: Setting the RS 232 – CAN communication parameters, e. g. Target-ID = 31

With these communication parameter settings, you contact the first CAN fieldbus station with the Target-ID “31” from the PRG-PC via the connected XC100.

▶ Establish the Online connection via “Login” and verify that the application functions.

Routing in a CANopen network

In a CANopen network, it is possible to access a CANopen slave fieldbus station with a programming PC. Figure 75 shows an example of a simple CANopen network with an integrated programming PC. This arrangement can be expanded to suit requirements. A limit to the expansion is only posed by the physical and electrical CANopen bus parameters.

If the XC100 is used as a master or CANopen slave station, the following additional conditions must also be observed:

- Device compatibility
- Assignment of the TargetID in the CANopen network.

Device compatibility

The dependance between the XC100 hardware version and the operating system version with the master PLC or CANopen slave station is indicated functionally in the following table.

Table 4: Device compatibility with routing

Hardware version	**Operating system version**	**Routing master station**	**Fieldbus station, programmable via CAN**
V02	2.0	✓	✓
V01	1.3	–	✓
CAN Target-ID	–	–	1 to 127

For Immediate Delivery call KMParts.com at (866) 595-9616

Assignment of the TargetID in the CANopen network

With the assignment of the TargetID with a CANopen slave fieldbus node (stations 6.1 and 6.2 in Figure xxx), the operating system version must be considered in dependance on the XC100 – CAN slave.

The following topology serves as an example:

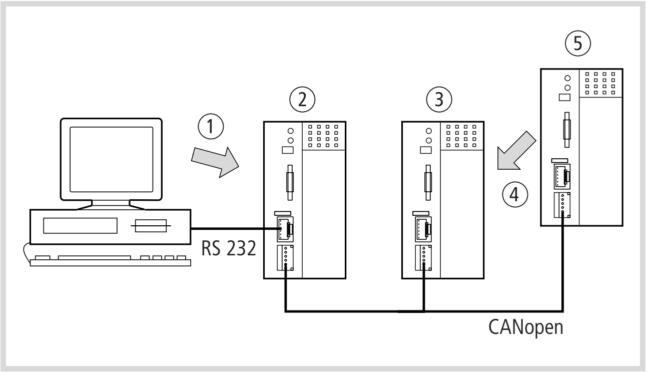


Figure 75: Routing in a CANopen network

① Master control slave XC100, V02

② Fieldbus connection 1:
Routing: Programming PC → CANopen slave station ⑥

③ CANopen slave station: XC100 with operating system 1.3/2.0

④ Fieldbus connection 2:
Routing: CANopen master to CANopen slave station

⑤ CANopen master

Procedure

- TargetID with XC100 with operating system version 1.3 as a CANopen slave station

If fieldbus connections 1 and 2 are operated simultaneously, a separate unique TargetId is required for each communication path ② + ④ in the CANopen slave!

- TargetID with XC100 with operating system version 2.0 as a CANopen slave station

If fieldbus connections 1 and 2 are operated simultaneously, only one common TargetId is required for communication path ② + ④ in the CANopen slave!

RS 232 interface in transparent mode

The transparent mode is used for data transfer with the data end user devices (e. g. terminals, printers, PC, measurement devices, etc.) via the RS 232 serial interface connection. The data is a transfer is a transparent transfer; i. e. the data is transferred without any further interpretation.

This functionality is provided on the XC100 by the "XC100_COM.LIB" library. The library must be integrated into the "Library Manager". The library contains functions for opening and closing the interface, for sending and receiving data and for setting the interface parameters.

If control lines on the RS 232 interface of the XC100 are contacted in the function module, they will not function as the control lines do not physically exist. However, these "XC100_COM.LIB functions" are implemented for reasons of compatibility.

If transparent mode is active, no communication is possible with the "XSoft" programming system. Transparent mode must be deactivated beforehand. When transparent mode is closed, the original communication parameters are re-initialized. The transparent mode is forcibly deactivated when the PLC changes to the STOP mode.

A prerequisite for operating the RS 232 interface in transparent mode is hardware version 02 and operating system version V02.00.

Demands placed on the functionality of the transparent mode

- "SysComOpen" function

The function opens the RS 232 interface for transparent mode.

After the interface has been successfully opened, the function returns a value greater than 0.

▶ Enter this value for the following functions as the "dwHandle" parameter.

If an error occurs, the return value is equal to zero. Transparent mode of the interface will not be enabled.

For Immediate Delivery call KMParts.com at (866) 595-9616

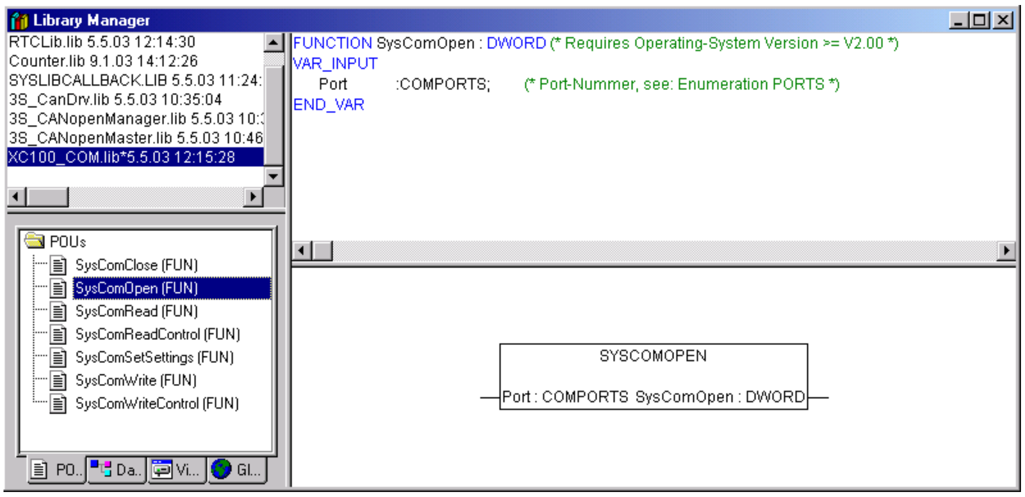

```
FUNCTION SysComOpen : DWORD (* Requires Operating-System Version >= V2.00 *)
VAR_INPUT
    Port        :COMPORTS;      (* Port-Nummer, see: Enumeration PORTS *)
END_VAR
```

Figure 76: SysComOpen function

Parameters

Port	Interface selection	
	Possible parameters:	1: with RS 232 interface of the XC100.
SysComOpen	Return value 0: Return value > 0:	Opening of the RS 232 interface was unsuccessful. Opening of the RS 232 interface was successful.

- **“SysComClose” function**

The function closes any RS 232 interface opened in transparent mode. During closing, the original communication parameters which were set are restored.

The function returns the TRUE return value when the action has been completed successfully.

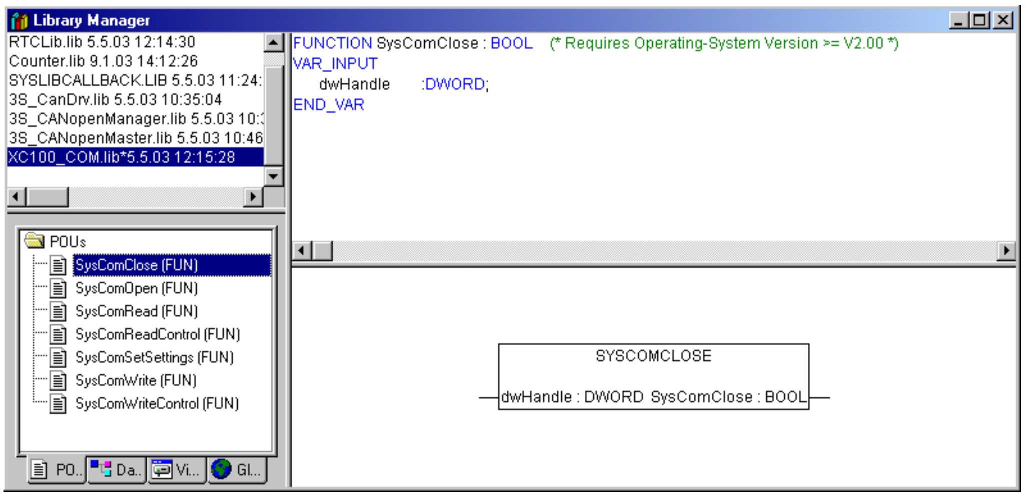

```
FUNCTION SysComClose : BOOL   (* Requires Operating-System Version >= V2.00 *)
VAR_INPUT
    dwHandle      :DWORD;
END_VAR
```

Figure 77: SysComClose function

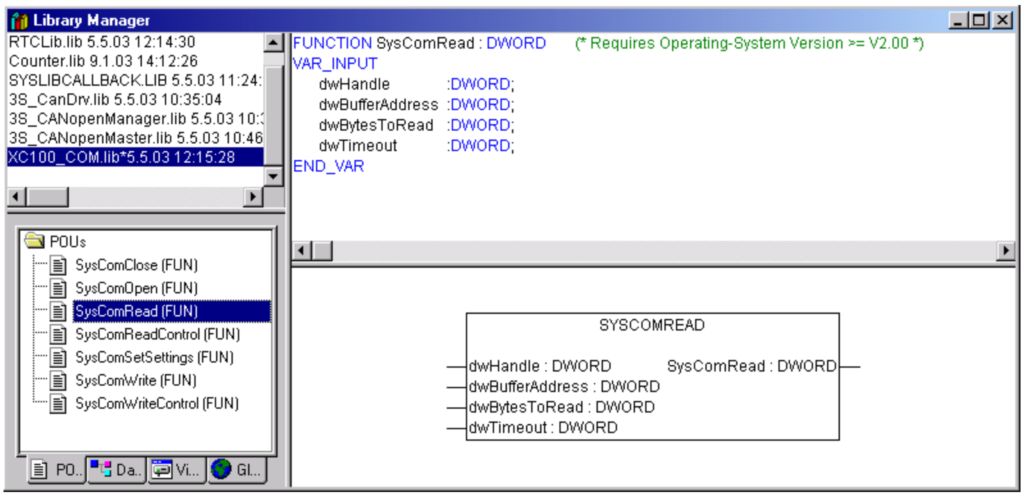
Parameters

dwHandle	Return value of the “SysComOpen” function
SysComClose	Return value “TRUE”: transparent mode; closing of the RS 232 interface was successful

For Immediate Delivery call KMParts.com at (866) 595-9616

- **“SysComRead” function**

The data received in transparent mode via the RS 232 interface can be read with this function.



```
FUNCTION SysComRead : DWORD     (* Requires Operating-System Version >= V2.00 *)
VAR_INPUT
    dwHandle         :DWORD;
    dwBufferAddress  :DWORD;
    dwBytesToRead    :DWORD;
    dwTimeout        :DWORD;
END_VAR
```

Figure 78: SysComRead function

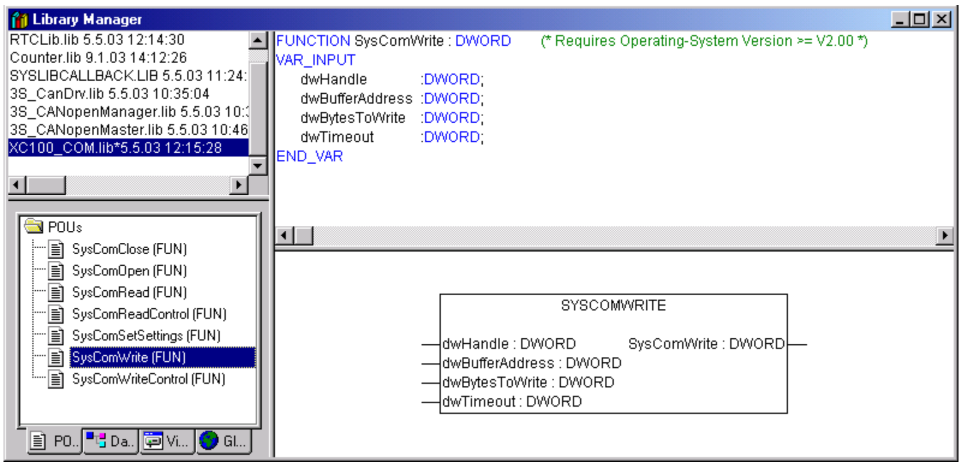
Parameters

dwHandle	Return value of the “SysComOpen” function
SysComClose	Return value “TRUE”: transparent mode; closing of the RS 232 interface was successful
dwBufferAddress	Address under which the read data is stored
dwBytesToRead	Limitation of the max. number of data bytes
dwTimeout	Parameter without meaning
SysComRead	Return value: Informs you about the number of read data bytes.

Note!
Test of the buffer address or the buffer size does not occur!

- **“SysComWrite” function**

This function allows output of data via the RS 232 interface.



```
FUNCTION SysComWrite : DWORD     (* Requires Operating-System Version >= V2.00 *)
VAR_INPUT
    dwHandle         :DWORD;
    dwBufferAddress  :DWORD;
    dwBytesToWrite   :DWORD;
    dwTimeout        :DWORD;
END_VAR
```

Figure 79: SysComWrite function

For Immediate Delivery call KMParts.com at (866) 595-9616

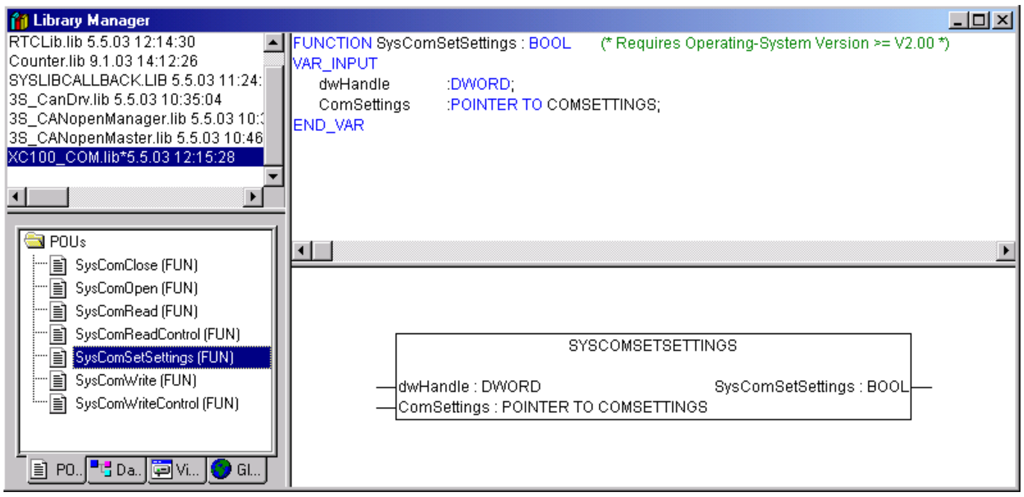
Parameters

dwHandle	Return value of the “SysComOpen” function
dwBufferAddress	Address under which the output data is stored
dwBytesToWrite	Number of data bytes to be sent
dwTimeout	Parameter without meaning
SysComWrite	Return value: Informs you about the amount of sent data.

Note!
Test of the buffer address or the buffer size does not occur!

• “SysComSetSettings” functions

With this function, the interface parameters of the RS 232 interface can be set for transparent mode.


Library Manager
RTCLib.lib 5.5.03 12:14:30
Counter.lib 9.1.03 14:12:26
SYSLIBCALLBACK.LIB 5.5.03 11:24:
3S_CanDrv.lib 5.5.03 10:35:04
3S_CANopenManager.lib 5.5.03 10:
3S_CANopenMaster.lib 5.5.03 10:46
XC100_COM.lib*5.5.03 12:15:28
POUs
SysComClose (FUN)
SysComOpen (FUN)
SysComRead (FUN)
SysComReadControl (FUN)
SysComSetSettings (FUN)
SysComWrite (FUN)
SysComWriteControl (FUN)
PO..
Da..
Vi...
Gl...
SYSCOMSETSETTINGS
dwHandle : DWORD
ComSettings : POINTER TO COMSETTINGS
SysComSetSettings : BOOL

```
FUNCTION SysComSetSettings : BOOL    (* Requires Operating-System Version >= V2.00 *)
VAR_INPUT
    dwHandle        :DWORD;
    ComSettings     :POINTER TO COMSETTINGS;
END_VAR
```

Figure 80: SysComSetSettings function

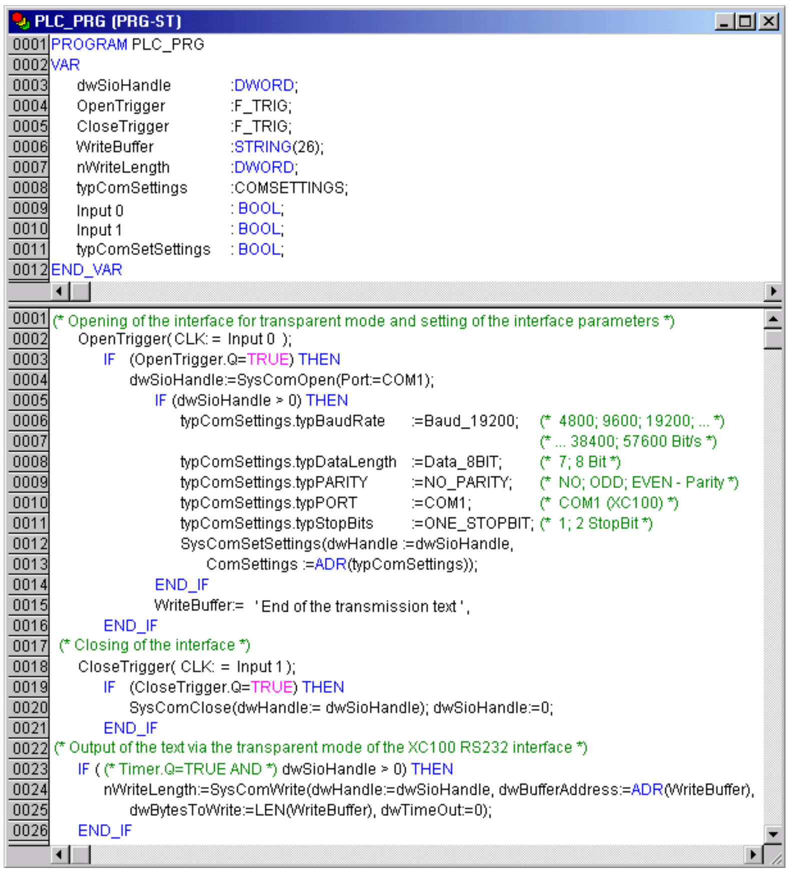
Parameters

dwHandle	Return value of the “SysComOpen” function
ComSettings	Pointer which points to the memory range in which the interface parameter is stored
SysComSetSettings	Return value “TRUE” if the interface has been parameterized successfully

For Immediate Delivery call KMParts.com at (866) 595-9616

Example

The example opening, a text output and closing of the RS 232 interface with the XC100.



PLC_PRG (PRG-ST)

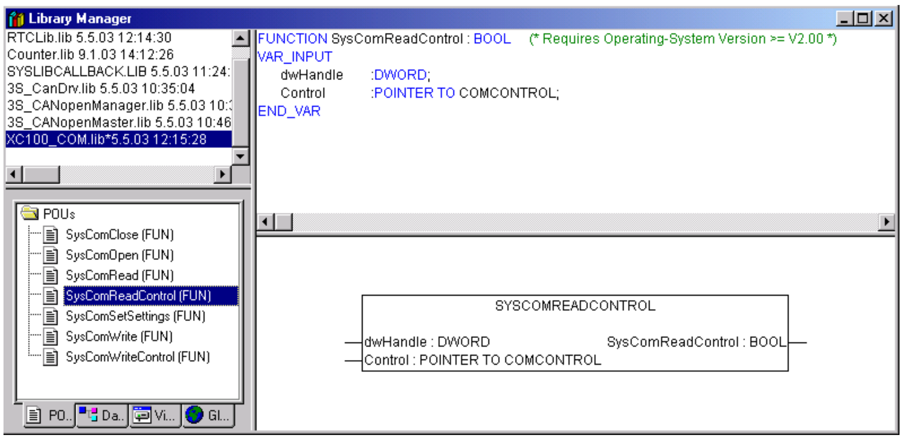
```
PROGRAM PLC_PRG
VAR
    dwSioHandle          :DWORD;
    OpenTrigger          :F_TRIG;
    CloseTrigger         :F_TRIG;
    WriteBuffer          :STRING(26);
    nWriteLength         :DWORD;
    typComSettings       :COMSETTINGS;
    Input 0              : BOOL;
    Input 1              : BOOL;
    typComSetSettings    : BOOL;
END_VAR
```

```
(* Opening of the interface for transparent mode and setting of the interface parameters *)
    OpenTrigger(CLK: = Input 0 );
        IF  (OpenTrigger.Q=TRUE) THEN
            dwSioHandle:=SysComOpen(Port:=COM1);
                IF (dwSioHandle > 0) THEN
                    typComSettings.typBaudRate     :=Baud_19200;   (* 4800; 9600; 19200; ... *)
                                                                   (* ... 38400; 57600 Bit/s *)
                    typComSettings.typDataLength   :=Data_8BIT;    (* 7; 8 Bit *)
                    typComSettings.typPARITY       :=NO_PARITY;    (* NO; ODD; EVEN - Parity *)
                    typComSettings.typPORT         :=COM1;         (* COM1 (XC100) *)
                    typComSettings.typStopBits     :=ONE_STOPBIT;  (* 1; 2 StopBit *)
                    SysComSetSettings(dwHandle :=dwSioHandle,
                        ComSettings :=ADR(typComSettings));
                END_IF
                WriteBuffer:=  ' End of the transmission text ' ,
        END_IF
 (* Closing of the interface *)
    CloseTrigger( CLK: = Input 1 );
        IF  (CloseTrigger.Q=TRUE) THEN
            SysComClose(dwHandle:= dwSioHandle); dwSioHandle:=0;
        END_IF
(* Output of the text via the transparent mode of the XC100 RS232 interface *)
    IF ( (* Timer.Q=TRUE AND *) dwSioHandle > 0) THEN
        nWriteLength:=SysComWrite(dwHandle:=dwSioHandle, dwBufferAddress:=ADR(WriteBuffer),
            dwBytesToWrite:=LEN(WriteBuffer), dwTimeOut:=0);
    END_IF
```

Figure 81: Transparent mode program example

• “SysComReadControl” function

This function assumes control lines with the RS 232 interface. However, control lines have been omitted with the RS 232 interface of the XC100. For this reason, the function is not available and only provided for reasons of compatibility.



Library Manager

```
RTCLib.lib 5.5.03 12:14:30
Counter.lib 9.1.03 14:12:26
SYSLIBCALLBACK.LIB 5.5.03 11:24:
3S_CanDrv.lib 5.5.03 10:35:04
3S_CANopenManager.lib 5.5.03 10:
3S_CANopenMaster.lib 5.5.03 10:46
XC100_COM.lib*5.5.03 12:15:28
```

```
POUs
  SysComClose (FUN)
  SysComOpen (FUN)
  SysComRead (FUN)
  SysComReadControl (FUN)
  SysComSetSettings (FUN)
  SysComWrite (FUN)
  SysComWriteControl (FUN)
```

```
FUNCTION SysComReadControl : BOOL    (* Requires Operating-System Version >= V2.00 *)
VAR_INPUT
    dwHandle        :DWORD;
    Control         :POINTER TO COMCONTROL;
END_VAR
```

```
                 SYSCOMREADCONTROL
dwHandle : DWORD                     SysComReadControl : BOOL
Control : POINTER TO COMCONTROL
```

Figure 82: SysComReadControl function

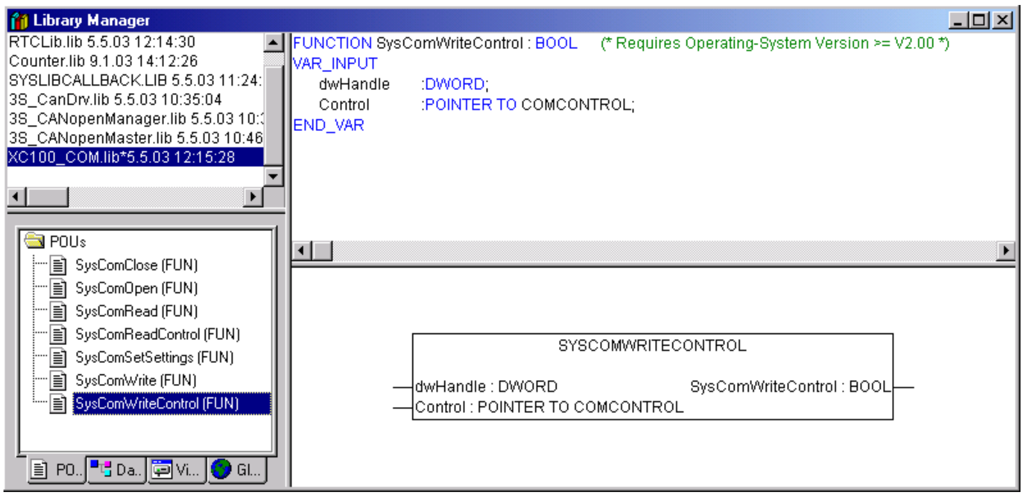
For Immediate Delivery call KMParts.com at (866) 595-9616

Parameters

dwHandle	Return value of the “SysComOpen” function
Control	Function omitted as no control lines are available on the control lines of the RS 232 interface of the XC100.
SysComReadControl	

• “SysComWriteControl” function

This function assumes control lines with the RS 232 interface. However, control lines have been omitted with the RS 232 interface of the XC100. For this reason, the function is not available and only provided for reasons of compatibility.


Library Manager
RTCLib.lib 5.5.03 12:14:30
Counter.lib 9.1.03 14:12:26
SYSLIBCALLBACK.LIB 5.5.03 11:24:
3S_CanDrv.lib 5.5.03 10:35:04
3S_CANopenManager.lib 5.5.03 10:
3S_CANopenMaster.lib 5.5.03 10:46
XC100_COM.lib*5.5.03 12:15:28
POUs
SysComClose (FUN)
SysComOpen (FUN)
SysComRead (FUN)
SysComReadControl (FUN)
SysComSetSettings (FUN)
SysComWrite (FUN)
SysComWriteControl (FUN)
PO..
Da..
Vi...
Gl...

```
FUNCTION SysComWriteControl : BOOL    (* Requires Operating-System Version >= V2.00 *)
VAR_INPUT
    dwHandle        :DWORD;
    Control         :POINTER TO COMCONTROL;
END_VAR
```

SYSCOMWRITECONTROL
dwHandle : DWORD
Control : POINTER TO COMCONTROL
SysComWriteControl : BOOL

Figure 83: SysComWriteControl function

Parameters

dwHandle	Return value of the “SysComOpen” function
Control	Function omitted as no control lines are available on the control lines of the RS 232 interface of the XC100.
SysComWriteControl	

Automatic closing of the interface

The transparent mode is automatically ended by the operating system with a PLC state change of the XC100 to STOP. The interface is reinitialized with the original interface parameters.

For Immediate Delivery call KMParts.com at (866) 595-9616

For Immediate Delivery call KMParts.com at (866) 595-9616

Appendix

Technical data

		XC-CPU101-C64K-8DI-6DO(-XV)	XC-CPU101-C128K-8DI-6DO(-XV)
General			
Standards and regulations		IEC/EN 61131-2 EN 50178	
Ambient temperature	°C	0 to +55	
Storage	°C	–25 to +70	
Mounting position		horizontal	
Relative humidity, no condensation (IEC/EN 60068-2-30)	%	10 to 95	
Air pressure (in operation)	hPa	795 to 1080	
Vibration resistance		10 to 57 Hz ±0.075 mm 57 to 150 Hz ±1.0 g	
Mechanical shock resistance		15 g/11 ms	
Overvoltage category		II	
Pollution degree		2	
Enclosure protection		IP20	
Rated insulation voltage	V	500	
Interference emission		EN 50081-2, Class A	
Interference immunity		EN 50082-2	
Battery (lifetime)		Worst case 3 years, typ. 5 years	
Weight	kg	0.23	
Dimensions (W × H × D)	mm	90 × 100 × 100	
Connecting terminals		Plug-in terminal block	Plug-in terminal block
Conductor cross-section			
Screw terminals			
stranded, with bootlace ferrule	mm2	0.5 to 1.5	0.5 to 1.5
solid core	mm2	0.5 to 2.5	0.5 to 2.5
Spring-loaded terminal			
stranded	mm2	0.14 to 1.0	0.14 to 1.0
solid core	mm2	0.34 to 1.0	0.34 to 1.0
Electromagnetic Compatibility (EMC)			
Electrostatic discharge (IEC/EN 61000-4-2, Level 3, ESD)			
Contact discharge	kV	4	4
Radiated (IEC/EN 61 000-4-3, RFI)	V/m	10	10
Burst Impulse (IEC/EN 61000-4-4, Level 3)			
Supply cables	kV	2	2
Signal cables	kV	1	1
Surge (IEC/EN 61 000-4-5)	kV	0.5	0.5
Conducted (IEC/EN 61 000-4-6)	V	10	10

For Immediate Delivery call KMParts.com at (866) 595-9616

			XC-CPU101-C64K-8DI-6DO(-XV)	XC-CPU101-C128K-8DI-6DO(-XV)
Supply voltage for the CPU (24 V/0 V)				
Hold-up time on supply drop-out				
Drop-out duration		ms	10	10
Repeat rate		s	1	1
Input voltage		V DC	24	24
Permissible range		V DC	20.4 to 28.8	20.4 to 28.8
Power consumption		W	max. 26	max. 26
Residual hum and ripple		%	≦ 5	≦ 5
Maximum power dissipated (without local I/O)	P_v	W	6	6
Overvoltage protection			Yes	Yes
Polarity protection			Yes	Yes
External supply filter			Yes, e. g. from Schaffner	
Internal supply filter			Yes	Yes
Switch-on current surge		$\times I_n$	Not limited, (limiting only by a supply-side 24 V DC PSU)	
Output voltage for the signal modules				
Nominal value		V DC	5	5
Output current		a	3.2	3.2
Off-load stability			Yes	Yes
Short-circuit proof			Yes	Yes
Electrically isolated from supply voltage			No	No
CPU				
Microprocessor			Infineon C164	
Memory				
Program code and program data		kByte	64	128
Markers and/or retained data		kByte	4	8
Cycle time for 1 k instructions (Bit, Byte)		ms	0.5	0.5
Interfaces				
Multimedia card			Yes, optional, 16 MB or 32 MB, to be ordered separately	
Serial interface (RS 232) without handshake line				
Data transmission rate		kBit/s	57.6	57.6
Connection by			RJ 45	RJ 45
Electrical isolation			No	No
CANopen				
Maximum data transmission rate		Bits/s	125000	125000
Electrical isolation			Yes	Yes
Device profile			to DS301V4	to DS301V4
PDO type			asyn., cyc., acyc.	asyn., cyc., acyc.
Connection			Plug-in spring-loaded terminal block, 6-pole	
Bus termination resistors			External	External
Participant		No.	max. 126	max. 126
Watchdog			Yes	Yes
RTC (Real-Time Clock)			Yes	Yes

For Immediate Delivery call KMParts.com at (866) 595-9616

			XC-CPU101-C64K-8DI-6DO(-XV)	XC-CPU101-C128K-8DI-6DO(-XV)
Supply voltage for the local inputs/outputs (24 V_Q/0 V_Q)				
Input voltage		V DC	24	24
Voltage range		V DC	19.2 to 30, observe polarity	19.2 to 30, observe polarity
Electrical isolation				
Between supply and CPU voltage			Yes	Yes
Overvoltage protection			Yes	Yes
Polarity protection			Yes	Yes
Digital inputs				
Input current per channel at rated voltage		mA	typ. 3.5	typ. 3.5
Power dissipation per channel			typ. 85 mW	typ. 85 mW
Switching levels as per EN 61131-2				
Limit values type “1”			low < 5 V DC, high > 15 V DC	
Input delay				
Off → On		ms	typ. 0.1	typ. 0.1
On → Off		ms	typ. 0.1	typ. 0.1
Inputs		No.	8	8
Channels with common reference potential		No.	8	8
Status indicator			LED	LED
Digital outputs				
Power dissipation per channel				
QX0.0 to QX0.3		W	0,08	0,08
QX0.4 and QX0.5		W	0,16	0,16
Load circuits				
QX0.0 to QX0.3		a	0,5	0,5
QX0.4 and QX0.5		a	1	1
Output delay				
Off → On			typ. 0.1 ms	typ. 0.1 ms
On → Off			typ. 0.1 ms	typ. 0.1 ms
Channels		No.	6	6
Channels with common reference potential		No.	6	6
Status indicator			LED	LED
Switching capacity			EN 947-5-1, usage category DC-13	
ON-time (duty cycle)		% ED	100	100
Simultaneity factor		g	1	1

For Immediate Delivery call KMParts.com at (866) 595-9616

08/03 AWB2724-1453GB

For Immediate Delivery call KMParts.com at (866) 595-9616

Index

A

Abbreviations . . . 3
Accumulator buffering . . . 29

B

Battery . . . 11
Battery change . . . 11
Baud rates . . . 45
Block diagram
  CPU . . . 7
  Supply module . . . 6
Breakpoint . . . 16
Browser commands . . . 29

C

Cabinet layout . . . 13
Cable routing . . . 13
Cold start . . . 15
Communication parameters . . . 31
Connect PC . . . 31
Connect programming device . . . 31
Connect text displays . . . 10
Connections, PSU and local I/O . . . 6
CPU module . . . 5
CRC checks . . . 15
Create bootable project . . . 30
Create program . . . 49
Creating a program (sample project) . . . 41
Cycle time, max. . . . 16
Cycle-time monitoring . . . 16

D

Data remanence . . . 29
Data transmission rate . . . 45
Data-saving . . . 7, 11
Declaration . . . 41, 49
Direct peripheral access
  Error code . . . 22

E

Electromagnetic contamination . . . 13
EMPTY-SLOT . . . 40
Error code with “direct peripheral access” . . . 22
Ethernet patch cable . . . 8

F

Forcing . . . 16
Function blocks . . . 17
Functions, XIOC
  DisableInterrupt . . . 25
  EnableInterrupt . . . 25
  ReadBitDirect . . . 19
  ReadWordDirect . . . 20
  WriteBitDirect . . . 21
  WriteWordDirect . . . 22

H

Hardware timer . . . 16

I

Inductors . . . 13
Insert CANopen module (in XSoft) . . . 43
Insert the input module (in XSoft) . . . 40
Installation, CPU . . . 11
Interface
  CANopen . . . 10
  Communication (programming device) . . . 8
Interference factors . . . 13
Interrupts . . . 24

J

Jitter . . . 16

L

Layout of units . . . 13
LED indicators . . . 6
Library . . . 17
Lighting protection . . . 14
Loading the operating system . . . 23

M

Memory Card . . . 8
Memory, application program . . . 7
Monitoring, system voltage . . . 7
Mounting position . . . 13
Multimedia Card . . . 8

O

Operating mode selector switch . . . 8
Operating states . . . 30

P

Parameterization . . . 16
PDO mapping . . . 47
POU . . . 49
Program processing . . . 16
Program run interrupted . . . 15
Program transfer . . . 30
Programming cable
  for RS 232 interface . . . 31
Project engineering . . . 13
PSU . . . 5

R

Real-time clock . . . 10, 17
Reset . . . 16
Residual cycle . . . 29
RJ 45 connector . . . 8
RS 232 interface . . . 8

For Immediate Delivery call KMParts.com at (866) 595-9616

S
Sample project 33
Select POU type 37
Self-test 15
Service Data Objects 48
Shielding 13
Single-cycle mode 16
Single-step mode 16
Start behaviour 15
Start behaviour at Power-On 16
Stop behaviour 15
Supply interruption 29
Suppressor circuitry for interference sources 13
Switch-off behaviour 15
Switch-on behaviour 15
Symbols 3
System time 16

T
Terminal assignments 6
Test and commissioning 15

U
Uninterruptible power supply 29

V
Ventilation 13
Versions, CPU 7

W
Warm start 15
Wiring 13
Wiring example
  PSU 14
  Terminal block 14

X
XC100 configuration 39
XIOC modules 5

For Immediate Delivery call KMParts.com at (866) 595-9616