Application Manual

Language	**English**
	Translation
Document No.	5.16008.02
Part Number	462104
Status	July 10, 2017

be in motion be in motion

BAUMÜLLER

Control system
PCC-04

E	5.16008.02

Read the Instruction handbook before starting any work!

Preliminary information

Warning Insofar as this document is identified as being preliminary information, the following applies:
this version is regarded as providing advance technical information to users of the described devices and their functions at an early enough time in order to adapt to any possible changes or expanded functionality.
This information must be regarded as being preliminary, as it has not yet passed through Baumüller's internal review process. In particular, this information is still subject to changes, thus no legal liability can be derived from this preliminary information. Baumüller assumes no liability for damages that might arise from this possibly faulty or incomplete version.
If you detect or suspect any content errors and/or major form errors in this preliminary information, we request that you notify the Baumüller support specialist responsible for you. Please provide us, via this employee, with your insights and comments so that we can take them into account and include them when transitioning from the preliminary information to the final information (as reviewed by Baumüller).
The conditions stipulated in the following section under „Obligatory“ are invalid in case of preliminary information.

Obligatory

This Application Manual are a part of the equipment/machine. This Application Manual must be available to the operator at all times and must be in legible condition. If the equipment/machine is sold or moved another location, this Application Manual must be passed on by the owner together with the equipment/machine.
After any sale of the equipment/machine, this original and all copies must be handed over to the buyer. After disposal or any other end use, this original and all copies must be destroyed.

When the present Application Manual is handed over, corresponding sets of Instruction handbooks of a previous version are automatically invalidated.
Please note that the specifications/data/information **are current values according to the printing date**. These statements are **not legally binding** with regard to measurements, computation or calculations.
Baumüller Nürnberg GmbH reserves the right, in developing its products further, to change the technical specifications and handling of it products concerned without prior notice.

No liability can be accepted concerning the correctness of this Application Manual unless otherwise specified in the General Conditions of Sale and Delivery.

© **Baumüller Nürnberg GmbH**

Ostendstr. 80 - 90
90482 Nuremberg
Germany

Tel. +49 9 11 54 32 - 0
Fax: +49 9 11 54 32 - 1 30

Email: mail@baumueller.de
Internet: www.baumueller.de

Table of Contents

Table of Contents

1

GENERAL

1.1 Information about Application Manual

The application manual **Control system PCC-04** provides important information in regard to handling the device.

A prerequisite for safe working is compliance with all specified safety information and handling instructions.

Furthermore, the local accident prevention regulations and general safety requirements applicable to the area of application of the device must be observed.

Before starting any work on the device, completely read through the application manual, in particular the chapter on safety information. The application manual is an integral part of the product and must be kept in the immediate vicinity of the device in order to be accessible to personnel at all times.

1.2 Explanation of symbols

Warnings

Warnings are identified by symbols in this application manual. The notices are introduced by signal words which express the degree of the danger.

Observe the notices without exception and exercise caution to prevent accidents, personal injury and damage to property.

DANGER!

....warns of an imminently dangerous situation which will result in death or serious injury if not avoided.

WARNING!

....warns of a potentially dangerous situation which may result in death or serious injury if not avoided.

CAUTION!

....warns of a potentially dangerous situation which may result in minor or slight injury if not avoided.

NOTICE!

....warns of a potentially dangerous situation which may result in material damage if not avoided.

Recommendations

NOTE!

....points out useful tips and recommendations, as well as information for efficient, trouble-free operation.

1.3 Limitation of liability

All specifications and information have been compiled taking account of the applicable standards and regulations, the state of the art and also our many years of expertise and experience.

The manufacturer accepts no liability for damage resulting from:

- Non-compliance with the Operating Manual
- Non-intended use
- Use of untrained personnel

The product actually supplied may deviate from the versions and illustrations described here in the case of special versions, the use of additional ordering options or as a result of the latest technical changes.

The user is responsible for carrying out servicing and maintenance in accordance with the safety regulations in the applicable standards and all other relevant national or local regulations concerning conductor dimensioning and protection, grounding, isolation switches, overcurrent protection, etc.

The person who carried out the assembly or installation is liable for damage arising during assembly or upon connection.

1.4 Copyright

Treat the application manual confidentially. It is intended exclusively for persons involved with the device. It must not be made available to third parties without the written permission of the manufacturer.

NOTE!

The details, text, drawings, pictures and other illustrations contained within are copyright protected and are subject to industrial property rights. Any improper exploitation is liable to prosecution.

EnDat®	is a registered trademark of Dr. Johannes Heidenhain GmbH, 83301 Traunreut, Germany
EtherCAT®	is a registered trademark of Beckhoff Automation GmbH, 33415 Verl, Germany
Hiperface®	is a registered trademark of SICK STEGMANN GmbH, 78166 Donaueschingen, Germany
Intel®	is a registered trademark of Intel Corporation, Santa Clara, CA 95054-1549. USA
Intel Atom™ **Intel Core™**	is a registered trademark of Intel Corporation

NOTE!

Please note, that BAUMÜLLER is not responsible to examine whether any (industrial property) rights of third parties are infringed by the application-specific use of the BAUMÜLLER products/components or the execution.

1.5 Other Applicable Documents

Components from other manufacturers were built into the device. Hazard assessments for these purchased parts have been performed by the respective manufacturers. The compliance of the designs with the applicable European and national regulations has been declared by the respective manufacturers of the components.

1.6 Guarantee Conditions

The guarantee conditions are located as a separate document in the sales documents.

Operation of the devices described here in accordance with the stated methods/ procedures / requirements is permissible. Anything else, e.g. even the operation of devices in installed positions that are not shown here, is not permissible and must be checked with the factory in each individual case. If the devices are operated differently than described here, any guarantee will be invalidated.

1.7 Customer service

Our customer service department is available for technical information.

Information concerning the responsible contact person can be obtained at any time by telephone, fax, e-mail or over the internet.

1.8 Terms used

Besides the term „**Control system PCC-04**“, the term „Device“ is used for our product.

1.9 List of associated documentations

	Doc. No.	Part Number German	Part Number English
Instruction Handbook Control System PCC-04	5.12045	462899	462901
Application Manual EtherCAT	5.10030	440084	
Online Help ProMaster			
Online Help ProProg			
Online Help ProViz			

2

SYSTEM OVERVIEW

2.1 Real time operating system and Windows 7

The **Control system PCC-04** combines a Microsoft Windows PC system and a b maXX-PLC with EtherCAT-Master in one device.

The system is divided into two parts - real time capable part for the PLC + EtherCAT and a part without real time capability for Windows applications. Both parts are considered separated parts and can be accessed with separated IP addresses, for example.

- Base of the non real time part is Microsoft Windows® Embedded Standard 7 (WES7) operating system.
- Base of the real time part is the real time operating system (RTOS) INtime RTOS by TenAsys. The b maXX PLC and the EtherCAT master are operated under RTOS.

This division is supported by the hardware as well. The processor core of the dual core processor is exclusively used by the RTOS. A core is exclusively used by the RTOS as well at quad core processors. The others are used by Windows.

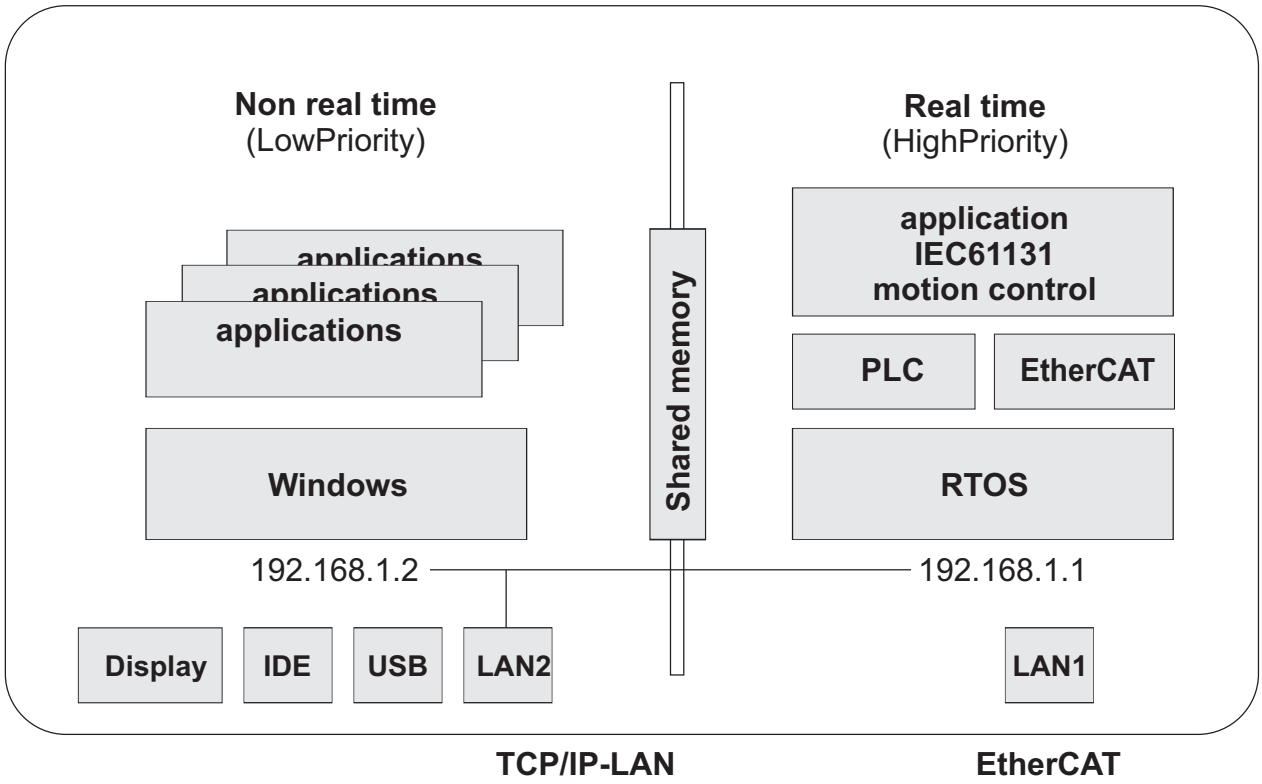


Figure 1: Separation real time/ non real time

The standard hardware interfaces of the **Control system PCC-04** is available in Windows.
Exception: Network card j LAN1 (labeled „LAN1“) must be used for EtherCAT and may be accessed with RTOS, only.

The network card LAN2 (labeled „LAN2“) is used for TCP/IP communication with Windows as well as with RTOS.

2.2 ProMaster, ProEtherCAT, ProProg and MotionControl

The real time part of the **PCC-04** can be regarded as a complete b maXX PLC with EtherCAT master. Therefore, engineering insignificantly differs from the engineering of the b maXX controller PLC (BMC-M-PLC-0x) and of the b maXX drive PLC (BM4-O-PLC-01).

Tool is **Engineering Framework „ProMaster“**.

ProEtherCAT

„ProEtherCAT“ configures EtherCAT fieldbus accordant to the application requirements.
This configuration is saved in the ProMaster project and is linked with other components.
The configuration is also transmitted to the **PCC-04**. The **PCC-04**'s software component EtherCAT master controls the fieldbus accordingly.

ProPLC

ProEtherCAT provides the fieldbus-based information of another ProMaster component **„ProPLC“**.
This configures the Motion Control core within the b maXX PLC and exports the fieldbus-based information (e.g. program variables for process data) into the ProProg project.

ProProg

ProProg is an **IEC 61131-3 programming system** (.. an IDE) to set PLC application (operation) with as well as without use of PLCopen Motion Control functions.

ProProg can be operated from ProMaster or as an independent tool.
The ProProg project (.... the PLC application) can be managed as part of the ProMaster project (the total project) or self-contained and outside of its scope (as a MWT file and index or packed as ZWT file).

The export of process data and motion control objects from ProMaster into the ProProg project is indispensable. Then the ProProg project can be operated separately, which is an advantage in larger project teams.

If changes are made on the fieldbus configuration this export must be updated. ProMaster project and ProProg project must be consistent.

ProMaster supports the setting of ProProg application by the generation of templates. Optional ProMaster can generate a new ProProg project with an application body and including motion control initialization, which complies with the configured fieldbus and can be used as basis for the specific PLC application.

MotionControl

The term **„MotionControl“** describes the functional programming of drives with IEC 61131 function blocks according to standards of user organization PLCopen. Baumüller supports this with libraries of these function blocks, a MC core within the b maXX PLC and with a configuration tool in ProMaster.

The PLCopen function blocks serve as commander in the user program to the motion control core of the b maXX PLC. It coordinates the command processing on the drives, notifies of the drive state etc.

The function blocks allow the programming of the application on a high abstraction level: e.g. „Positioning of position A to position B“, „Switch to cam No. 4“, „Engage the gear synchronization“ etc.

The PLC user program does not parameterize drives. Prior to this, the PLC user program is made once only with the ProMaster component „ProDrive“. The MC core of b maXX PLC regards or finishes other details in the background

The communication connection to the drive (normally a fieldbus, e.g. EtherCAT) must not be programmed in the IEC 61131-3 code, anymore. It has already been parameterized with the ProMaster component „“ProEtherCAT“. The process data, which are required to and from the drive required by motion control were already regarded together with the application-specific and user-defined process data and were transferred to the EtherCAT master as well as to the PLC program and the MC core. Even the initiation of the fieldbus is executed automatically in the background by the MC core of the b maXX PLC.

This interaction accelerates the application setup and eliminates error sources.

2.3 SCADA Visualization with web technologies and classical

Integrated Web Server

Besides the b maXX controller PLC (BMC-M-PLC-0x) and the b maXX drive PLC (BM4-O-PLC-01) there is the new **Integrated Web Server**. They allow HMI applications (Human Machine Interface = user interfaces of a machine or installation) and by profiting of the internet protocol HTTP are able to be displayed in each internet browser.

The user access to the machine or installation can be executed via a TCP/IP network (LAN) with PC or smartphone. As a matter of course with proper automation devices, e.g. the b maXX HMI panel (BM-HMI-035W).

The ProMaster component **„ProViz“** is the editor for these HMI applications.

SCADA Server

Web-based HMI applications are hosted with **SCADA Server** just as with the eWebServer. However, compared to the eWebServer, the **SCADA Server** is not a component of the PLC. The **SCADA Server** runs with Microsoft Windows. It can directly connect to other SCADA- and eWebServers and to b maXX PLC(s) and b maXX drive(s) and obtain data therefrom. High-level languages can be programmed optionally. Due to this, it can be extended by business logic components, which concentrate and process the data of these devices.

SCADA Server is configured with the ProMaster component **„ProViz“**.

The b maXX **SCADA Server** can be operated on Windows PCs. However, it can also be operated on the Windows page of the PCC-03 (in the not real-time capable part).

ProViz Editor

Analog to PLC application and ProProg, ProViz can be used as an independent tool. The ProViz project can be managed within ProMaster project and independently (separation in large project teams)

OPC Server

The conventional visualization via an **OPC Server** is possible. The selected visualization (SCADA application) and the OPC Server are processed with Microsoft Windows, e.g. with Windows of **PCC-04**.

b maXX OPC-Server **„ProOPC II“** is used for this.

3

MOUNTING AND INSTALLATION

NOTE!

Mounting and installation must be made by personnel of the manufacturer or by qualified personnel.

Detailed description see **Instruction Handbook for Control System PCC-04**. See ▹List of associated documentations◃ on page 8.

4

COMMISSIONING

4.1 First steps

NOTE!

A USB keyboard and a USB mouse as well as a display port screen should be connected to the **PCC-04**.

After switching on 24 V_{DC} supply voltage **PCC-04** boots.

At first the BIOS is displayed. Then Windows® 7 Embedded from system data medium is started.

NOTE!

Only at the very first start Windows is preconfigured and the system is booted a second time automatically.

NOTE!

If problems occur, when starting the system, then see instructions in the instruction handbook, chapter „Operation, first boot up“!

After Windows has been started, RTOS starts and two text windows „RT I/O Console ..“ with black background appear. These are the reporting outputs of the b maXX PLC and of the EtherCAT master.

In the background of the Windows screen Windows 7 (which was modified by Baumüller) is displayed. The version of the **Baumüller base system** is to be found in „System Control - System“ (press buttons „Win“ and „Break“ at the same time)**.**

4.2 Windows and pre-installed components

The basic operating system is Microsoft Windows® Embedded Standard 7 (WES7) in English.

The change between German and American keyboard is possible (not in „Control panel - Keyboard“, but in **„Control panel - Region and Language“** and there from tab „Keyboards and Languages“).

User account

Default setting of user account „bm“ with administrator rights and without password was made. However, after completing application, a user account with limited rights according to requirements and to security concept of the end user or of the machine manufacturer should be established.

In regularly intervals Baumüller will create a current basic system image. The updates, however, may not be applied by the user - they will be applied by Baumüller in Nürnberg.

RTOS

The real-time component is a real-time operating system (RTOS) INtime for Windows of TenAsys. INtime is displayed in application list in Windows.

Controlled by RTOS. b maXX-PLC, EtherCAT-Master, eWeb Server and FTP Server are operated. Windows recognizes them as files, only and not as executing applications.

WES7

First of all WES7 is differentiated by its adaptability and its scalability from the normal Windows 7. The **PCC-04** possesses a lean basic system.

- Instead of „Windows Prompt“, the **PCC-04** provides Microsoft „Windows Power Shell“ for an efficient operation on a command line.
- Microsoft Explorer is an existing integral system part. However, an additional file manager is missing.
- The standard **Web Browser** is the Microsoft Internet explorer.
  The existing b maXX **MicroBrowser** is specialized on the eWebServer of the b maXX - PLC. This means, specialized on HMI-/SCADA applications, which were created with ProViz.
  The b maXX **SCADA Server** was pre-installed as service.
- A **PDF Reader** is not installed (self-installation if required).
- A Microsoft Windows **Firewall** is not activated.
  A **Virus scanner** is missing (according to the security concept install the appropriate product).
  Please consider that the virus scanner deeply takes action in the Windows system and it can affect the real-time capability of the total system.
- The FTP server „FileZilla“, the multi-purpose „Notepad++“, some of the Windows Accessories“ and Microsoft „On-Screen keyboard were installed, also.
  The „Visual C++ 2008 Redistributable“ and the „Dot Net Framework 4.0“ were installed, also.
- Links to the starting as well as to the stopping scripts, to FileZilla, to the SystemProtection Tool and so on are provided.

NOTE!

In general the Baumüller components, which were installed should not be removed. Other components should be installed carefully and the extended total system should be tested precisely.

NOTE!

Do not install development kits such as ProMaster, ProProg or ProDrive to the control system PCC-04. These development kits need to be installed to a workstation or a notebook PC of the application engineer and not on the automation device.

For PCC-04 there is one special version of ProDrive.

4.3 Overview of partitions and file system

The **PCC-04** is delivered with a SSD/mSATA (the only persistent mass storage). Mechanical drives are done without.

This data medium is divided into system partition and data partition.

A CFast® Card can be used optionally (see Instruction Handbook **PCC-04**).

System partition

The system partition (**Windows drive C:**) includes system software WES7 and INtime.

The following registries are to be found in its Windows 7 common use:

- C:\Windows System files
- C:\Program Files installed programs
- C:\Users User account specific settings, private data, ...
- Additional registries - partly hidden

Besides that, there is a file C:\BM for most of the components of Baumüller. This file makes the industrial PC a **PCC-04**.

- C:\BM\bin\
- C:\BM\bin\INtime b maXX-PLC, EtherCAT-Master
- C:\BM\bin\Windows Auxiliary programs, e.g.:
- C:\BM\bin\Windows\SystemWriteProtection see chapter 4.5
- C:\BM\scripts\ Start- and stop scripts

Data partition

Data partition (**Windows drive D:**) contains its application and its data.

- D:\BM\boot Boot project of the b maXX PLC, EtherCAT master configuration
- D:\BM\eweb\hmi eWebServer project including alarm server and trend server.
- D:\BM\eweb\logs Log files of alarm server and trend server
- D:\BM\log Log files, in general
- D:\BM\misc Cams, optional MotionControl configuration, etc.
- D:\BM\retain Retain data file for PLC application
- D:\BM\zip Program sources of the PLC application, optional
- D:\SCADA optional components, SCADA server
- D:\SCADA\hmi SCADA server project
- D:\SCADA\Logs Its log files
- D:\userdata Reserved for application

As the file system under Windows is freely accessible with administrator access rights (or can be made accessible), a firmware update is possible by exchanging a file.

Update

An update or change of the application can be done by exchange of the folder D:\BM or single sub-directories or files.

For further information about „Firmware update“ refer to ▹Firmware update◃ on page 74.

The disadvantage of the free accessibility is a vulnerability of the system by non intentional access. Therefore a protection mode is necessary, set this mode with the system protection tool.

4.4 Installation mode, application mode and protected mode

Installation mode

The recently provided **PCC-04** is in installation mode. In this state it is possible to upgrade system partitions (window drive C:\) with required components (installation of Windows and RTOS applications, driver for USB hardware) or system settings can be changed (e.g. network settings and activation of optional components).

Application mode

Change to application mode as soon as possible.
This mode enables changing the boot project of b maXX-PLC, of EtherCAT and motion control configuration, of WebServer and SCADA project etc. - short the OEM application of the machine builder. The system partition is protected by the Enhanced Write Filters of Microsoft. Made system changes are lost when Windows is restarted. The PCC-04 can be switched-off, similar to other b maXX devices, without waiting for the normal shut down of the windows PC. Please note: A switch-off while active download of e.g. a boot project causes its loss. Retain data of the PLC are lost after switch-off.

Protected mode

The third mode, protected mode is used, if the application is completely developed and the machine is ready for the end consumer. In this mode only certain data partition areas are writable (e.g. retain data file, log files of WebServer and SCADA server and the directory D:\userdata). The system partition and the application on the data partition is write-protected.

The set protection mode is displayed, if the mouse moved over the icon (divided into four parts) in windows system tray (taskbar, at the bottom right)..

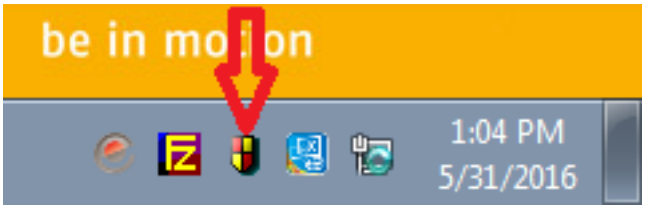


Figure 2: Symbol „Protection mode“ in windows system tray

The protection mode can be changed in the context menu of this symbol (click with right mouse key).

4.5 RTOS

A link to the start script for the b maXX PLC is in the Windows Autostart.

This script starts RTOS first and then the b maXX PLC. The b maXX PLC starts the EtherCAT master.

Displayed are two text windows „RT I/O Console .." with black background (reports of the b maXX PLC and of the EtherCAT master) and prior to this a color change of an icon in the Windows system tray is visible.

A red symbol displays an active INtime-RTOS, a yellow symbol displays a stopped RTOS.

The tool tip message „The INtime kernel is running“ is displayed when moving the mouse cursor over the icon of an active INtime-RTOS.

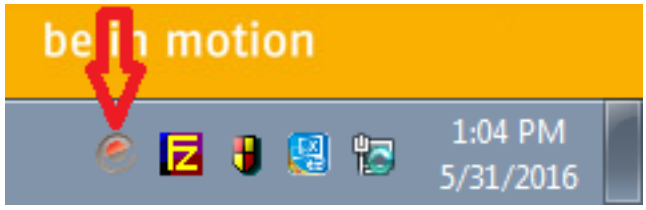


Figure 3: Symbol „RTOS“ in Windows System Tray

The context menu of this symbol (right mouse key click) enables access to version information „About INtime“ and access to tools, e.g. „INtime Explorer“ or „INtime Configuration“. No changes by the application engineer are necessary in this area, normally. Unintended changes here can cause failure of the **PCC-04**.

Links to start and stop scripts can be found on the Windows desktop (in C:\bm\script).

Start RTOS

You start the RTOS and the based components RTOS-TCP stack, b maXX PLC, EtherCAT master and eWeb server with „PCC_Start“ manually. The RTOS and the based components are closed, if they are already active.

Stop RTOS

You stop the RTOS and its dependent components with „PCC_Stop“ manually.

The scripts „PCC_Shutdown“ and „PCC_Reboot“ shut down the complete **PCC-04**, Windows included.

Please note the difference to stop, cold and warm start of the PLC (see ▹Remanent data, cold start, warm start and hot start◃ on page 48).

As soon as the RTOS and the b maXX PLC is started, the access to the b maXX PLC via TCP communication network is possible.

4.5.1 Local access of Windows to the b maXX PLC

The local access is done via the TCP communication network.

For testing a prompt is recommended:

Windows Power Shell

The **PCC-04** provides in addition to the „Windows prompt“ the Microsoft „Windows Power Shell“.

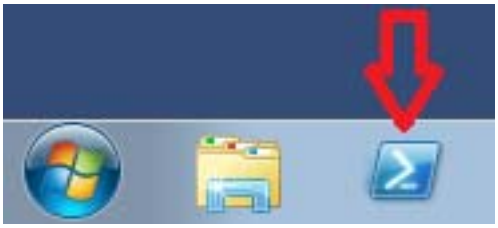


Figure 4: Symbol „Windows Power Shell“ in Windows start bar

Windows prompt

Open the conventional „Windows prompt“ in start menu - all programs - accessory or with „Win key + R“ -> system dialog „execute“ (insert „cmd“ in the input box and press button „OK“).

➲ Please open any prompt and enter „ipconfig“.

The TCP/IP address of the **PCC-04 Windows page** is displayed.
The default setting is 192.168.1.2.

The IPv4 address of **PCC-04 RTOS page** is 192.168.1.1 at delivery status.
A ping command on this address should be successful as long as the RTOS is active.

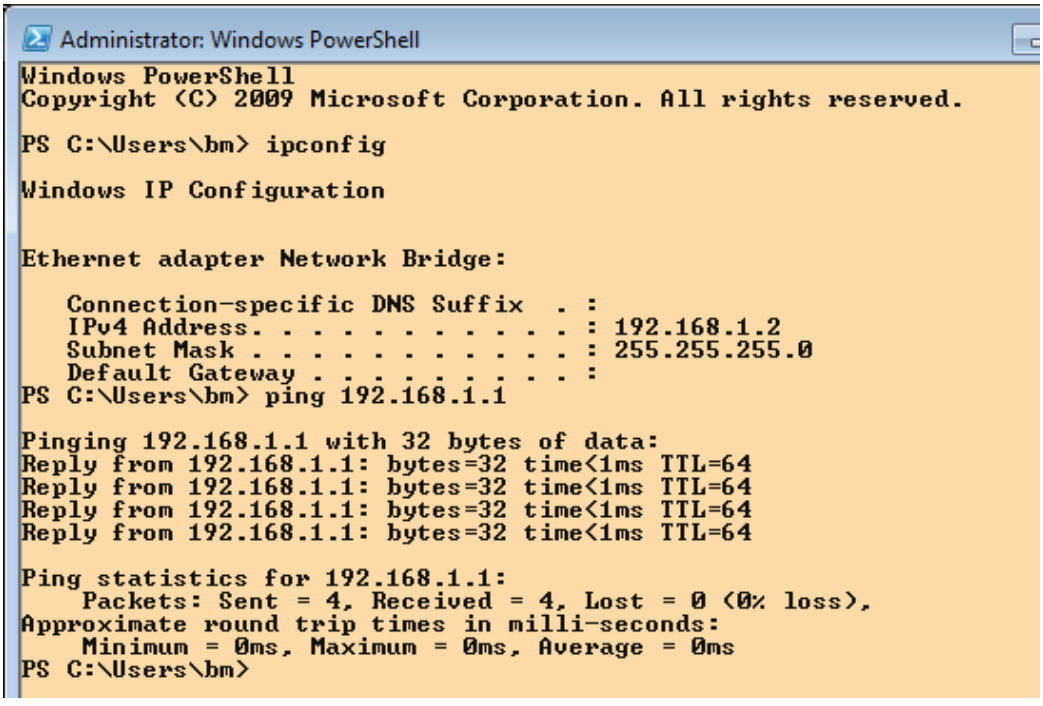


Figure 5: IP connection to RTOS sided TCP stack

Further information to „TCP/IP settings“, see ▹TCP/IP network configuration◃ from page 26.

The standard web browser on the **PCC-04** has access to the b maXX PLC. Enter „http://192.168.1.1/“ in the address prompt of the browser and a standard test website appears. For generation of an own Web-Visu application, see ▹Integration of visualization◃ from page 60.

It is preset, that starting the MicroBrowser enables an access to this website directly.

If a ProProg or ProMaster would be installed on the Windows side of the **PCC-04**, this tools could establish a connection to b maXX PLC and program the b maXX PLC.

This tools are not installed on the notebook PC of the application engineer, normally. They should not be installed on the automation device.

4.5.2 External or remote access to b maXX PLC

The external access (e.g. from a notebook PC of an application engineer) is done via TCP communication network, too.

- Connect the Ethernet network interface „LAN2“ of PCC-04 with a CAT5 cable to the network interface of the notebook PC.
- The PC network interface must be assigned to an IP address from the same subnetwork (as determined beforehand), e.g. 192.168.1.50.

Now both the IP address of the **PCC-04 Windows side** (192.168.1.2) and the IP address of the **PCC-04 RTOS side** (192.168.1.1) can be pinged to.

As soon this is successful, the notebook web browser has access to the web server of the **PCC-04** b maXX PLC.

- Enter „http://192.168.1.1/“ to the prompt of the browser and
  check the proxy settings of your web browser (perhaps an exception for the address range 192.168.*.* is necessary) and your firewall.

As stated above, a simple website for testing is displayed.
The generation of own web-visu applications is shown in chapter ▹Integration of visualization◃ from page 66.

The ProProg, installed on the notebook PC, should have access to the **PCC-04** b maXX PLC as well, see ▹Figure 6◃ on page 25.

A corresponding project is necessary for this
(settings see ▹Applications and Operation◃ from page 37).

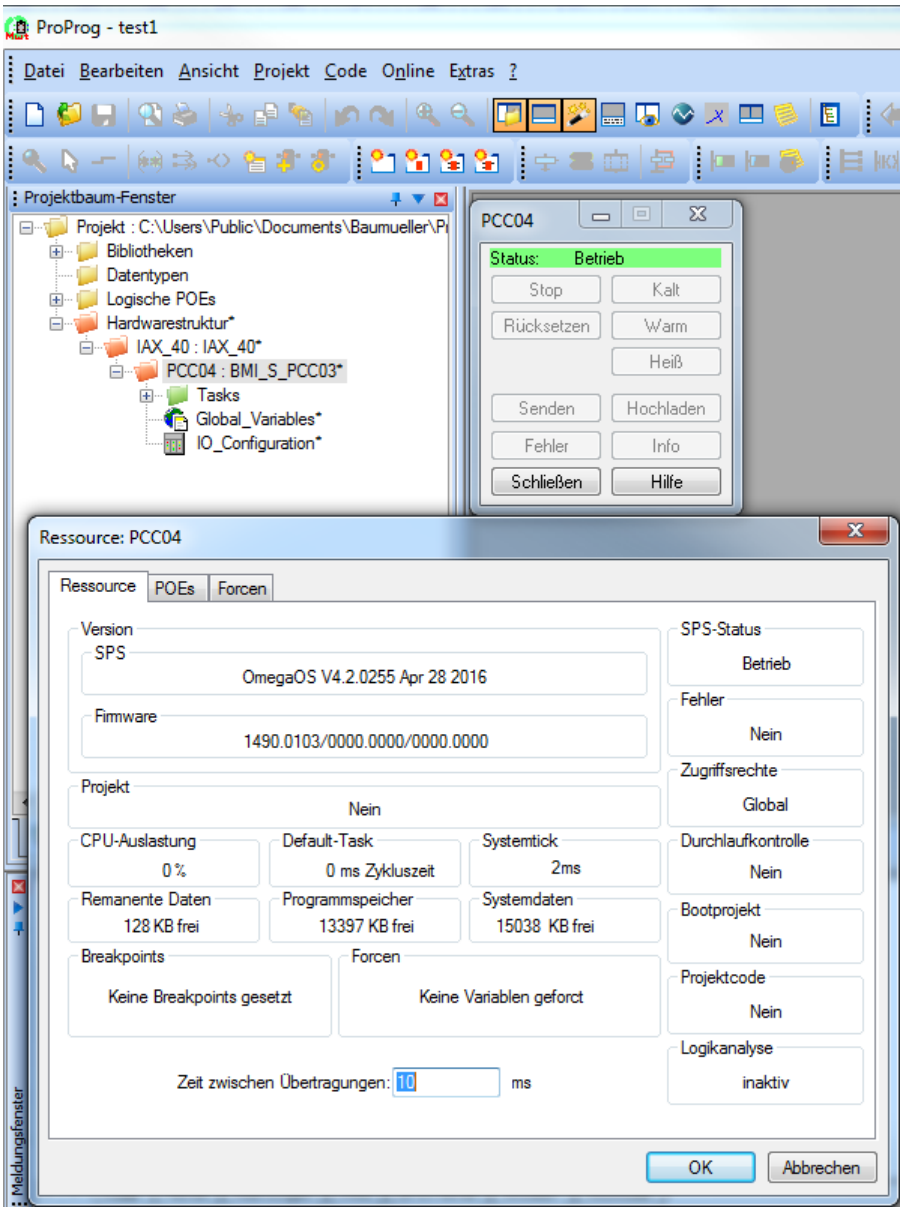


Figure 6: First access with ProProg to the b maXX PLC of PCC-04

The „PLC state“ is „On“. Without boot project no project (=IEC-61131 application) is load while starting. The PLC is ready now and waiting for the ProProg. The „PLC status“ changes to „Stop“ after successful download of a (RAM) project to the RAM memory. The PLC changes to the „PLC status“ „Run“ when initiating an cold start. In this status the application is executed.

The RAM project is lost after switching off the PLC (not only stopping the PLC).
A so-called boot project is necessary. This boot project is loaded automatically to the RAM after switching on the PLC and is started with a warm start.

This behavior is shown by all other b maXX PLC.

4.6 TCP/IP network configuration

The basics for this topic are shown in chapter 3 of „application handbook EtherCAT“.

Please note, that the **PCC-04** TCP/IP features are not linked to the „EtherCAT master“ module (in contrast to b maXX controller PLC or b maXX drive PLC)
The chapter 3.2.3 „Setting of IP address and subnetwork mask on communication modules“ of „application handbook EtherCAT“ is **not** valid for **PCC-04**.

The **PCC-04** provides **two integrated network interface cards** (NICs, adapter) both type „Intel(R) PRO/1000 PL Network Connection“ (that means two 1GBit/s adapter)

- **LAN1** ➨ EtherCAT
- **LAN2** ➨ Windows network

The first NIC is necessary for the EtherCAT master and is assigned to the RTOS „INtime“ (LAN1). This NIC is not visible for Windows.

The second NIC is assigned to the operation system Windows (LAN2) and is displayed as adapter **„Win-LAN“** (see ▹Figure 7◃ on page 26.

The **virtual adapter „RTOS“** of type **„TenAsys Virtual Ethernet Adapter“** is used for the link to the RTOS network stack.

The **„Network Bridge“** of type **„MAC Bridge Miniport“** is generated by configuration.

Both original adapters retreat into the background and refer to the bridge. Windows wants to show that linked adapters share settings and exchange data transparently.

In summary:

- Window provides a real NIC (LAN2) and a virtual NIC,
- both are linked
- the NIC hardware (LAN1) used for EtherCAT is not visible in Windows

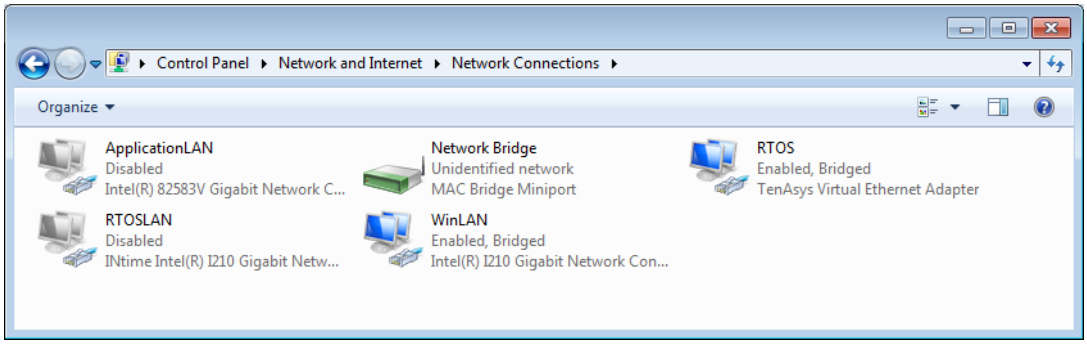


Figure 7: Windows view

Further network adapters can be displayed, when additional components are installed (e.g. USB Ethernet adapter, virtual machines). You decide, whether this adapters are added to the „Network Bridge“ and can communicate with the RTOS side or whether this adapters are available for the Windows TCP/IP stack, only.

This Windows specific point of view is integrated to the TCP/IP communication diagram of the **PCC-04**.

Communication overview of PCC-04

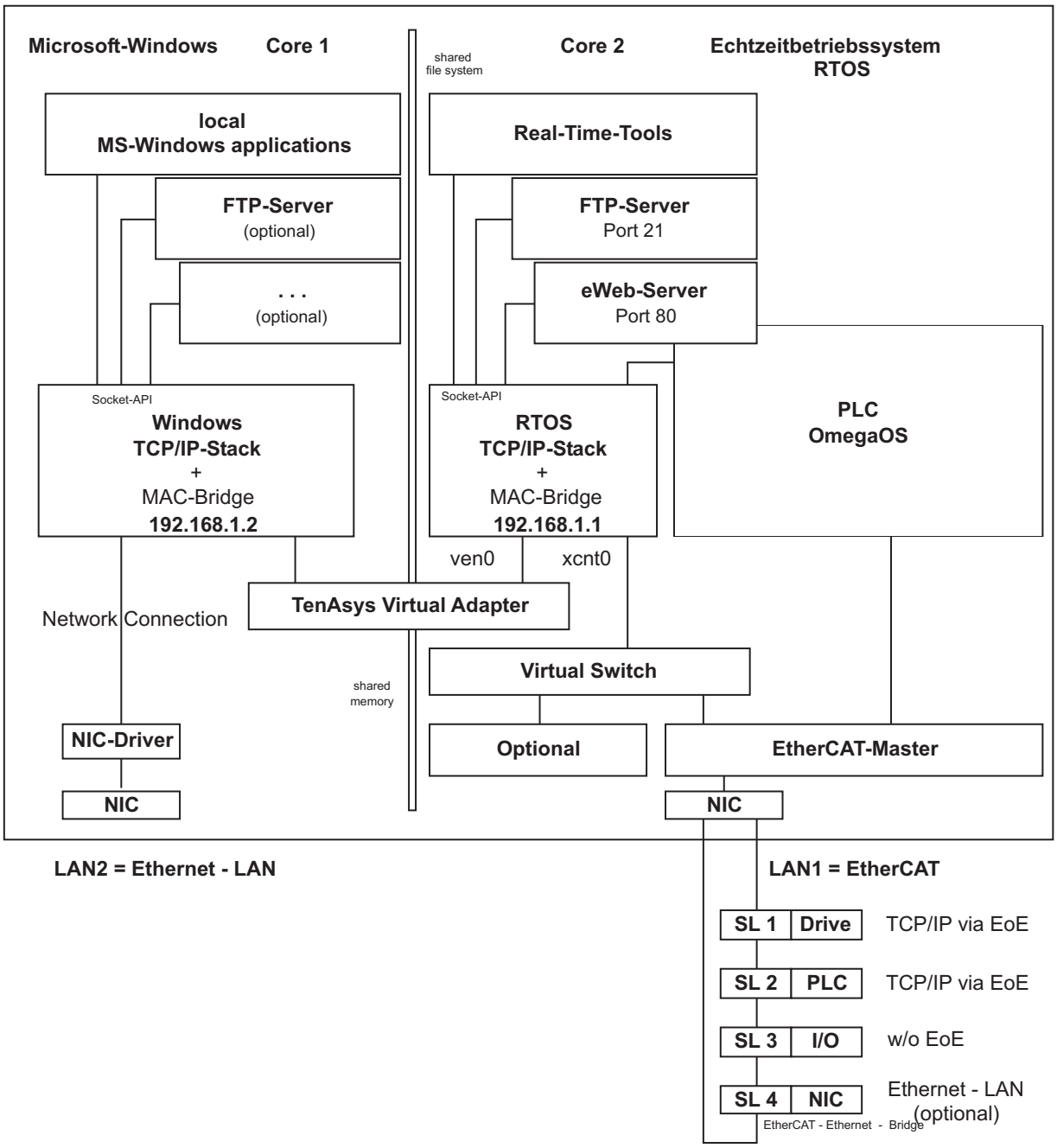


Figure 8: TCP/IP communication scheme of PCC-04

IP address 192.168.1.1 is linked to the real time side of **PCC-04**.

IP address 192.168.1.2 is linked to the Windows side of **PCC-04**.

Of course the IP address and the corresponding subnetwork mask can be changed.

4.6.1 Change IPv4 network settings

Please note prior to change of the IP address and corresponding subnetwork mask that the address of real time side of **PCC-04** and Windows side are **in the same subnetwork**. Further note the basics in chapter 3.2.2 of „application handbook EtherCAT“.

Precondition for change of the network settings is that the **PCC-04** is in **installation mode** (see ▹Installation mode, application mode and protected mode◃ from page 21). Only in this mode the settings remain in effect after switching off the device.

A

A further condition is that the **RTOS has to be restarted** to apply the new IP settings. The Windows sided IP settings are active without restart.
It is recommended to stop the RTOS (and the b maXX PLC) before changing network settings.

- Open the command prompt (see ▹Local access of Windows to the b maXX PLC◃ from page 23), then enter command „**ipconfig**“

The TCP/IP address of the **PCC-04** Windows side appears as answer.

- New settings are entered with command „**pcc_config**“

A short overview of the command options appears when entering the command without parameters.

Example 1:

The IP address should be set as follows:

- new IP address RTOS side 192.168.0.5
- new IP address WIN side 192.168.2.5
- new subnetwork mask 255.255.0.0

The command is:

```
pcc_config /ip1=192.168.0.5 /ip2=192.168.2.5 /ipm=255.255.0.0
```

Example 2: The default settings should be restored, the command is:

```
pcc_config /ip=default
```

equal to:

```
pcc_config /ip1=192.168.1.1 /ip2=192.168.1.2 /ipm=255.255.255.0
```

NOTE!

Please add the parameter `/ip2name="My Bridge"` when calling the tool „pcc_config“ in case the adapter „Network Bridge“ has been renamed with Windows tools. The new name (here e.g. „My Bridge“) has to be set within quotation marks.

➲ Restart the RTOS and the b maXX PLC using the script „**PLC_Start**“, next.

The tool „pcc_config“ can be used in own scripts, too. The return „Errorlevel = 1" indicates an error, „Errorlevel = 0" indicates a successful execution.

4.6.2 Change/add IPv4 network settings for Windows only

The command prompt tool described above covers all standard applications.

Use Windows board tools when using e.g. an additional IP address on Windows side or further features.

The network IP address of the **PCC-04** Windows side must be set in the Bridge.

- Right click, context menu Network Bridge Properties:

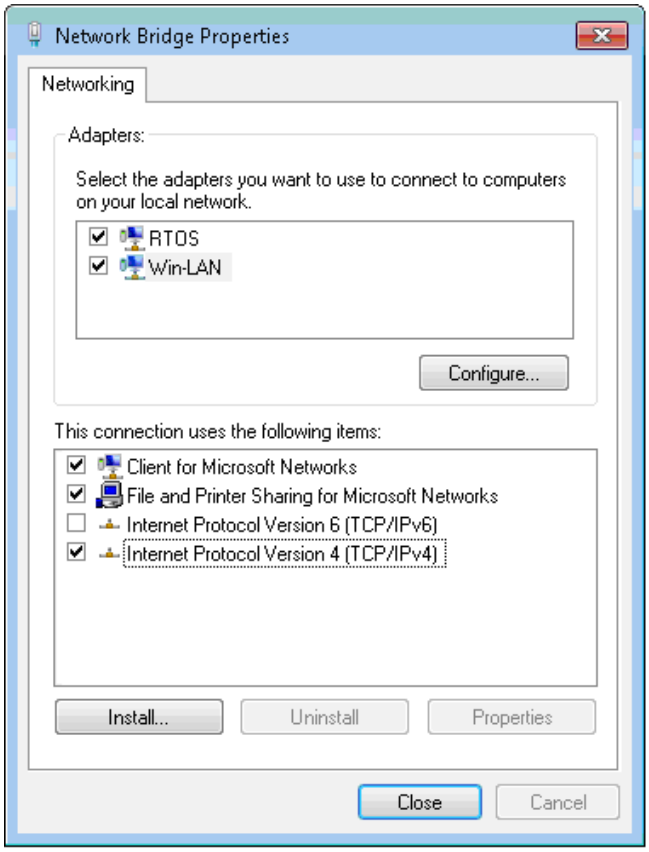


Figure 9: Network settings of bridged adapter

- Set the IP address of Windows side (192.168.1.2) after double click on „Internet Protocol V4“:

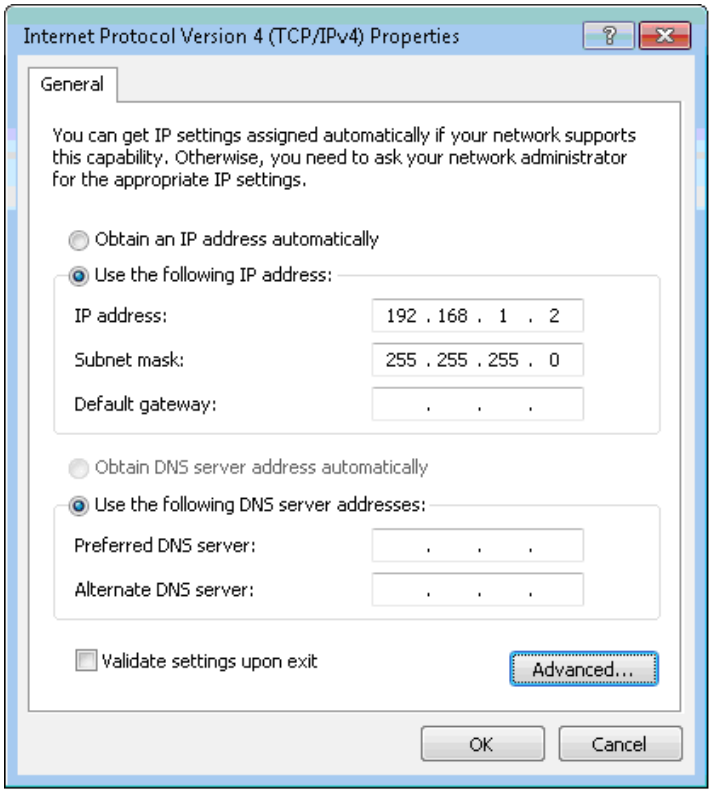


Figure 10: IPv4 settings of bridged adapter

Click on „Advanced“ to set e.g. a further IP address.

See chapter 3.2.2 of „Application handbook EtherCAT“ for further information.

4.6.3 Operating in the local network instead of in the Cloud

The PCC-04 is intended to be used within a separated machine LAN, only. This LAN has to be protected spatial as well as organizational against unauthorized accesses.

NOTE!

The PCC-04 is not an IoT device!

It does not provide safety schemes which allow a direct access to the Internet (or the "Cloud").

A remote maintenance VPN router should be used for advanced requirements such as remote maintenance or integration in the company LAN.

Contact the technical sales and distribution department of Baumüller if support is required to implement the conception of such a remote maintenance and Cloud access.

We dissuade from harden the Windows system software of the PCC-04. Irrespective of the probable failure of such an undertaking, the real time features of the PCC-04 can be affected.

4.7 FTP server on RTOS side

The FTP server on RTOS side (ftpd.rta) is started by the start script „PLC_Start“ together with b maXX PLC and EtherCAT master.

This FTP server provides the standard FTP access, e.g. saving the EtherCAT master configuration, deployment of ProViz project on eWeb server of PLC, firmware update ...

The FTP server is addressed by the RTOS side IP address (default 192.168.1.1), port 21 with set user approval and pass words.

A configuration by the application engineer is not necessary and not provided.

Use the FTP server on Windows side (see next chapter) if a FTP server is necessary for the application.

4.8 FTP server on Windows side

Pre-installed is the software „FileZilla“.

There is an icon to the control panel of „FileZilla Server“ in the Windows 7 system tray (bottom right on the edge of the desktop). There the necessary settings are done (menu Edit/Settings“, „Edit/Users“) and the server can be started or stopped (menu „Server/Active“ bzw. „Server/Lock“).

The server can be reached at Windows IP address (default: 192.168.1.2) and port 21 via LAN.

We suggest a Personal Computer with **Microsoft Windows 7** for testing. The PC must be connected to the PCC-04 according chapter ▹External or remote access to b maXX PLC◃ on page 24 (Windows 10 is supported limited, only).

- Open an instance of Windows explorer (shortcut Win+E) and enter the destination address „ftp://192.168.1.2“ (see ▹Figure 11◃ on page 33) to the address line.
  - Windows explorer opens a dialog to enter the access data.
- Enter name „BM_USER“ and pass word „12345678“.

The access to the directory „D:\userdata\“ of PCC-04 is linked to this account.

This directory consists of one file „reame.txt“ in delivery status. In the following example the directory includes two files „test.bin“ and „test.csv“. This both files should be displayed by the Windows explorer of the application PC, connected via FTP.

The control panel of the „FileZilla Server“ on PCC-04 records the connection establishment automatically.

The application engineer must enter the necessary settings, e.g. user approval, accounts. The FTP server should be inhibited or removed if it isn't required.

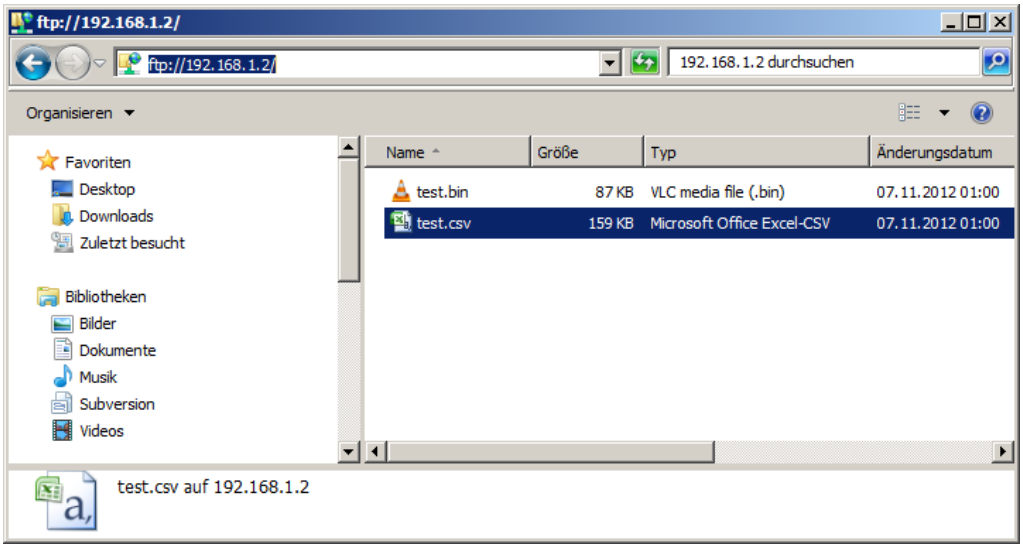


Figure 11: FTP-Server on Windows side

4.9 eWebServer

The **eWebServer** is an integrated part of the b maXX PLC and is started with the PLC.

The **eWebServer** can be reached by IP address on RTOS side (default 192.168.1.1), port 80.

We suggest a Personal Computer with **Microsoft Windows 7** for testing. The PC must be connected to the PCC according chapter ▹External or remote access to b maXX PLC◃ on page 24.

Enter „http://192.168.1.1“ into address line of the web browser of the test PC (note proxy settings and firewall!). It should display the eWebServer page or your ProViz web visualization project.

The settings of this WebVisu application and its use within the eWebServer is shown in chapter ▹ProViz - visualization by means of web technologies◃ on page 67.

4.10 SCADA server

The SCADA server is a pre-installed service, but is not started automatically.

If required the SCADA server can be started with the prompt command

- **net start scadaserver,**

it can be stopped with

- **net stop scadaserver.**

Start SCADA server automatically

The start mode of the SCADA server must be changed from „manual“ to „automatic“, if the service should start with Windows, automatically.

- Open the management terminal „services.msc“, see ▹Figure 13◃, by
  - entering services.msc on a command prompt or
  - with the script „%windir%\system32\services.msc“ or
  - via the „ControlPanel-AdministrativeTools“ see ▹Figure 12◃.

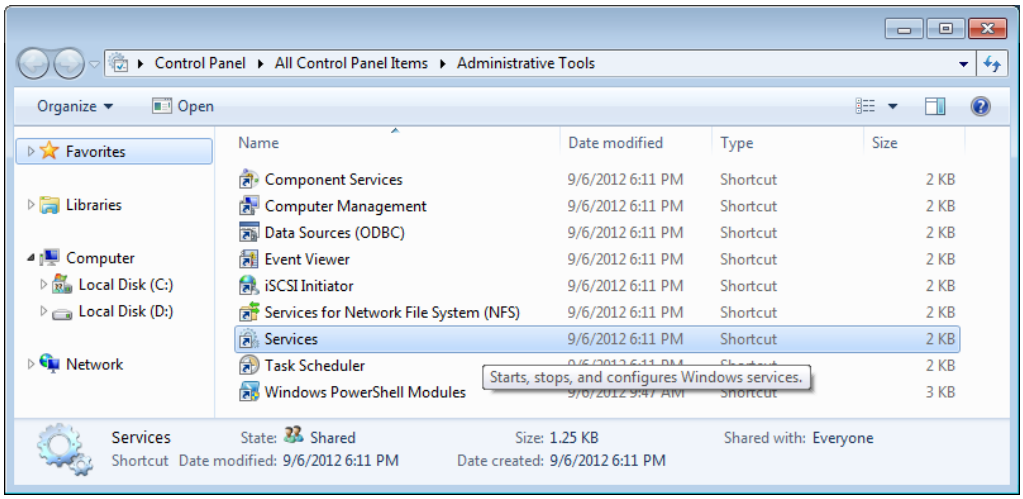


Figure 12: Services

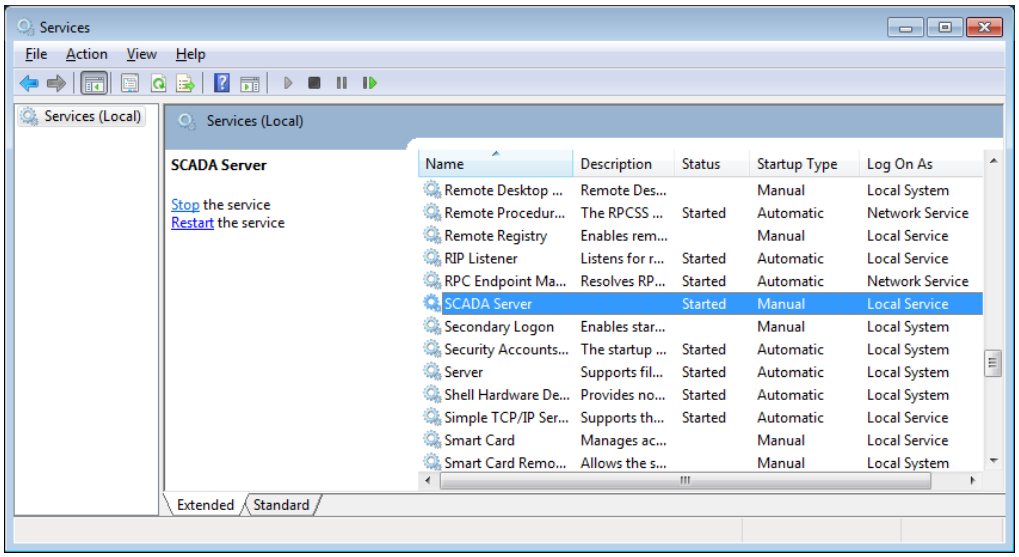


Figure 13: SCADA server

- A double click on the entry „SCADA **Server**“ opens the property dialog. There can be changed the start type.

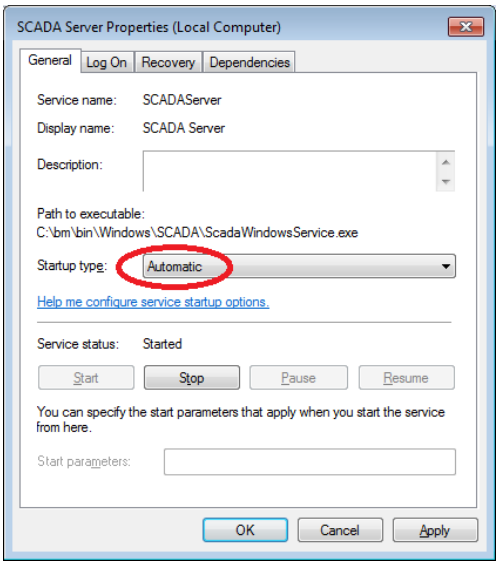


Figure 14: SCADA server

Special settings can be done in SCADAControlPanel.exe.

Access via start menu:

- Start/All Programs / SpiderControl SCADA Web Server
- SCADAControlPanel.exe
- SCADAWebServerSetup.pdf

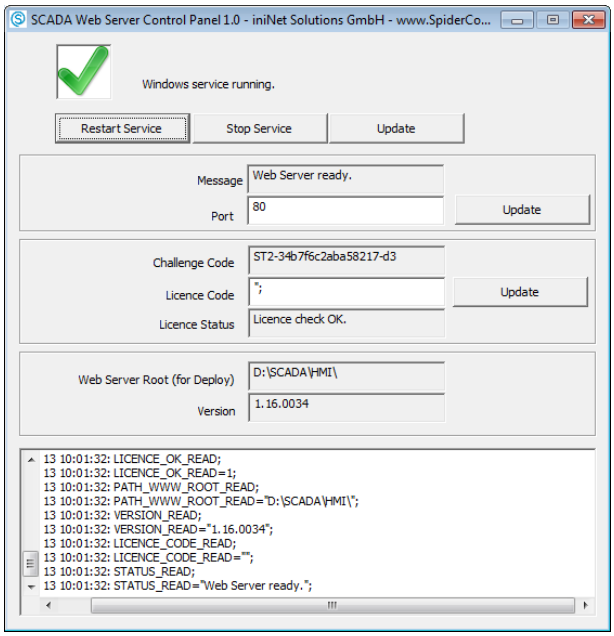


Figure 15: SCADA ControlPanel

Further details on applying the SCADA server see separate application handbook or on-line help /pdf file referring to SCADA server.

4.11 MicroBrowser

The MicroBrowser can be started by a link on the desktop or by a script via direct start of „MicroBrowser_XP.exe“ in directory „**C:\bm\bin\Windows\MicroBrowser**“.

Important for configuration is the path to the MicroBrowser start project. Normally, the path is the IP address on RTOS side of the local b maXX PLC and the start site (e.g. start.html) is the eWebServer application created with ProViz.
For instance:

- http://192.168.1.1/start.html

The ProViz application starts immediately, if the IP address is set correctly and the eWebServer of the corresponding b maXX PLC is ready for use.

Please refer to the manual „MircoBrowserSetup.pdf“ in above mentioned directory for further configuration options.
The access code mentioned there is not necessary.

Example 1

A display connected to „Disply Port“ of PCC-04 shall display the ProViz application, the MicroBrowser shall be executed in full screen mode and always in foreground.

In „mBrowser.xml“ the value <kiosk> must be set to 2.

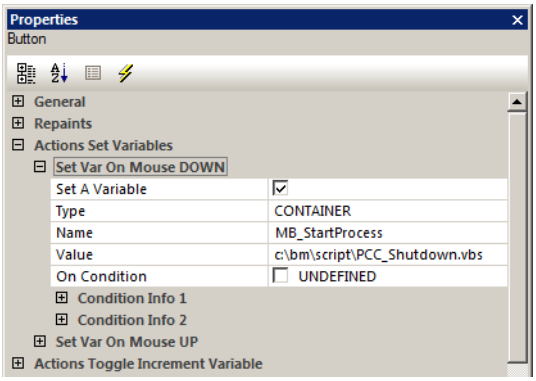
No shortcuts must be interpreted because the panel is only equipped with an USB keyboard in service mode.

For service mode the MicroBrowser can be closed with e.g. shortcut Shift X. Then the service engineer has access to the Windows operation system.

Example 2

A button for switching-off or shutting down the PCC-04 should be added to the ProViz application of example 1

- Place a painter of type „Button“ on a Teq-View in ProViz-Editor
- Choose under „Painters Property“ the menu point „Actions Set Variables“ and than „Set Var On Mouse DOWN“
  - set the check mark „Set A Variable“
  - type: CONTAINER
  - name: MB_StartProccess
  - value: C:\bm\script\PCC_shutdown.vbs



5

APPLICATIONS AND OPERATION

A complete b maXX PLC with EtherCAT master is included in the real-time part of the automation device **PCC-04**. Therefore the engineering is not considerably different to the engineering of b maXX controller PLC (BMC-M-PLC-0x) or b maXX drive PLC (BM4-O-PLC-01).

5.1 ProMaster - system configuration

The **PCC-04** configuration is done via **ProMaster component „ProPLC“.** A detailed description of this component is to be found in the online help of ProMaster.

ProMaster from version 1.17 is necessary.

- Start a new, empty ProMaster project, shortcut **„Strg-N“** on ProMaster start site.

1 Open the ProMaster catalogue.
The ProMaster catalogue includes an entry for **PCC-04**.

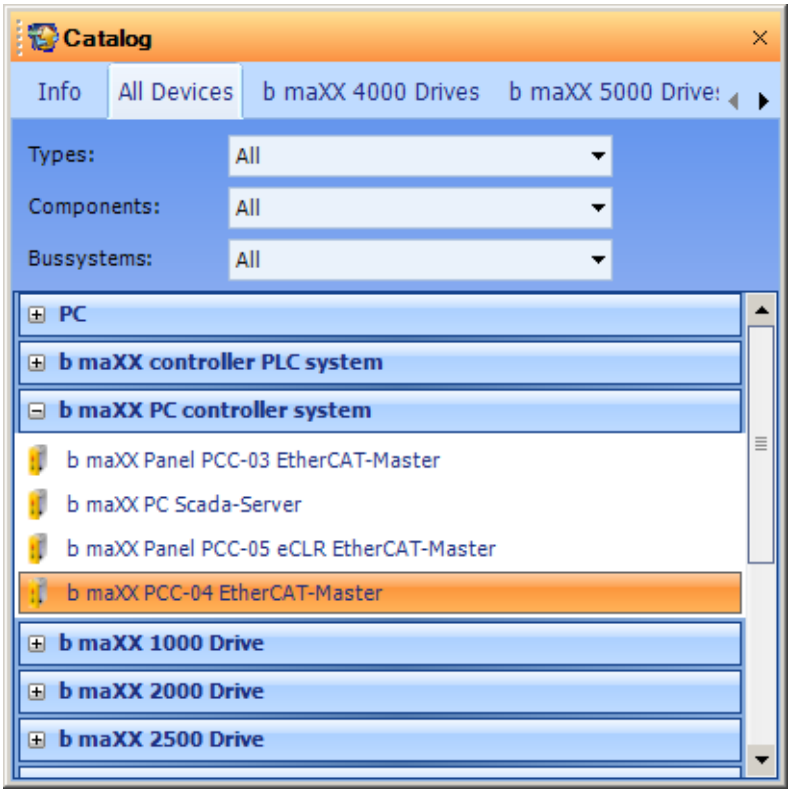


Figure 16: ProMaster catalog with settings for PCC-04

➲ Generate a device instance of this type on the empty workspace with drag and drop.

- Set the IPv4 address of the RTOS side in communication settings. The default value 192.168.1.1 is suitable in our case.

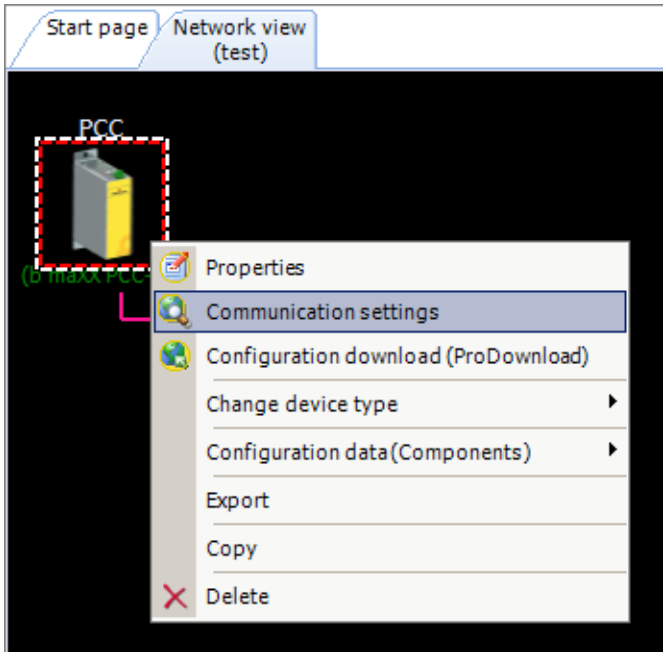


Figure 17: ProMaster context menu of a PCC-04 device instance

All other configuration components can be found with this context menu. An alternative access is the „ProMaster workspace“ (normally located right beside the workspace)

- Open the component „ProPLC“ (see below).

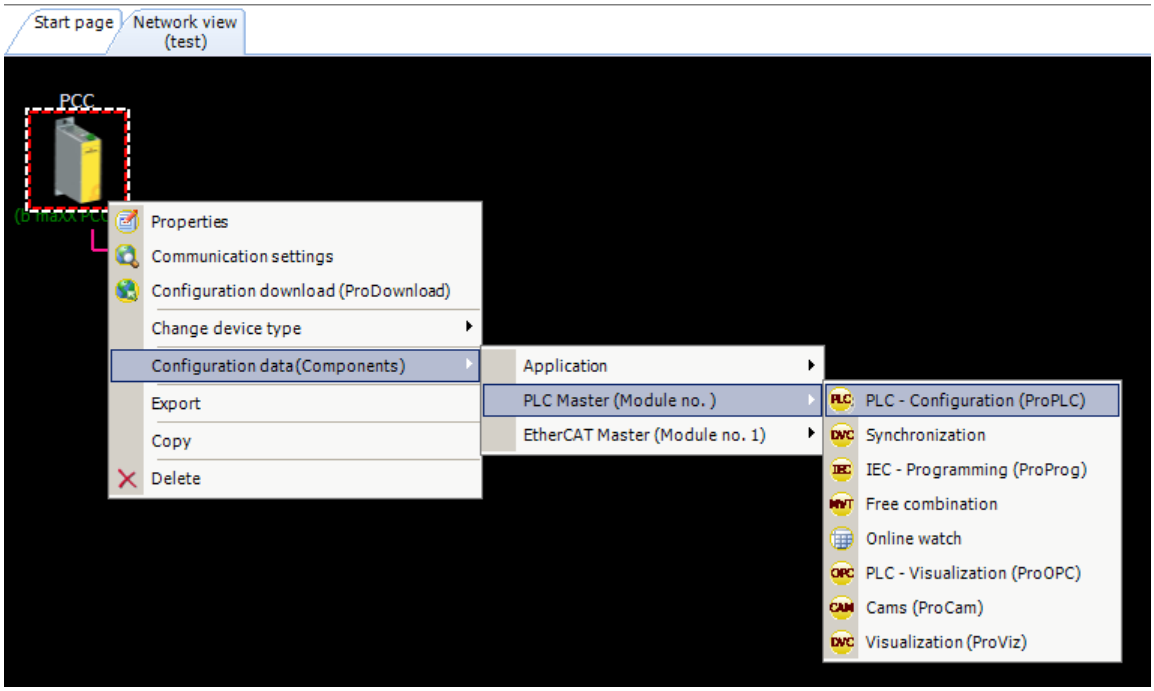


Figure 18: PLC configuration

ProPLC is started with page 1 / tab „IEC“, which is empty at the moment.

- Change to page 2 / tab „PLC“ and click on button „Connect“ beside the grey bar „OFFLINE“.

The bar changes to color green and displays the PLC state (POWER ON = without boot project) after a successful connection is established.

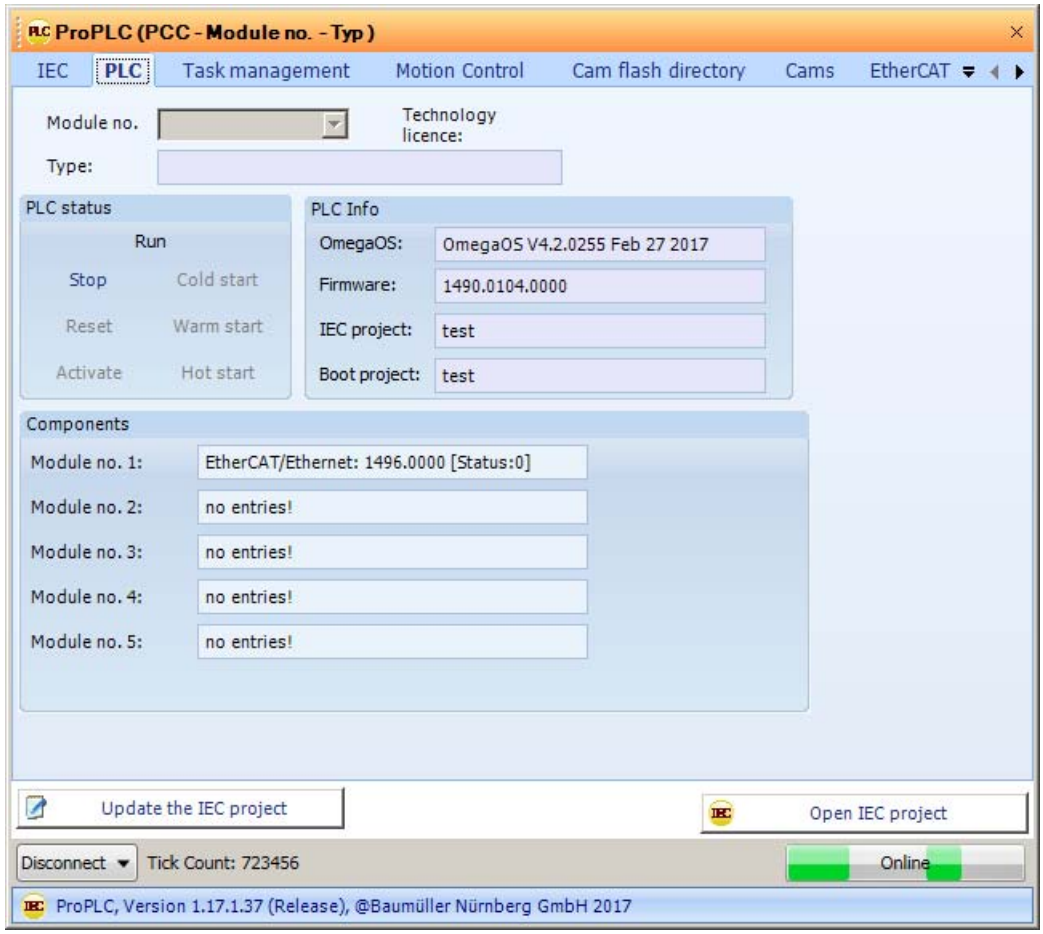


Figure 19: ProMaster connected with b maXX-PLC of PCC-04

The version information of PLC „OmegaOS“ and of EtherCAT master are displayed.

The further pages/tabs of ProPLC are empty to a large extent, because the **PCC-04** without fieldbus EtherCAT have no connection to the automation process and to the drives.

Next the fieldbus is configured. Please return after that to the ProPLC - tab „IEC“ to create or link a ProProg application (PLC program) and export the fieldbus dependent configuration files to the PLC program and the Motion Control configuration.

For further information see chapter 5.4 „Export to ProProg - user program and download to device“ of „Application handbook EtherCAT“.

5.2 ProEtherCAT - fieldbus configuration

The fieldbus EtherCAT is of particular importance for **PCC-04**. This fieldbus is the only connection of the **PCC-04** to the automation process (other drives and other I/O devices).

For general information fieldbus EtherCAT see chapter 4 „Application handbook EtherCAT“.

The **PCC-04** firmware includes an EtherCAT master corresponding to the information in chapter 5 of „Application handbook EtherCAT“.

The EtherCAT master of **PCC-04** is no real module, but a software module.
The EtherCAT master works like a module in „slot G / module number 1“.
The remarks in chapter 2.2.3 „Recognition of modules / version information in user program“ and 2.2.4 „DPRAM interface between EtherCAT modules and PLC“ of the „Application handbook EtherCAT“ are also valid.

In difference to real modules e.g. **no real BACI sync. signals** are generated. This signals are emulated and the bypass event task for MotionControl (see following chapter 5.4 „ProProg - application generation with IEC 61131") works analog to bypass event task „Motion1“ of b maXX controller PLC or b maXX drive PLC.

The **event synchronous buffer change of process data** (compare chapter 5.1.4 in „Application handbook EtherCAT“ is not available at the **PCC-04**.

The configuration of fieldbus EtherCAT is done via ProMaster component „ProEtherCAT“. For further information see „Online help“ of ProMaster. The basics are described in chapter 5.2 „Configuration with ProMaster“ in „Application handbook EtherCAT“.

5.3 ProProg - generation of IEC 61131 application

ProProg functions as programming system according IEC 61131-3 for generating an PLC application.
The interaction of ProMaster and ProProg is explained in ▹ProMaster, ProEtherCAT, ProProg and MotionControl◃ from page 11.

5.3.1 Generate ProProg project for PCC-04 manually

Proceed as follows:

- Start ProProg (via start menu, desktop shortcut and so on. - but do not start from ProMaster exceptionally)
- Close the automatically loaded last project
- Generate with shortcut „Strg+N“ a new project.

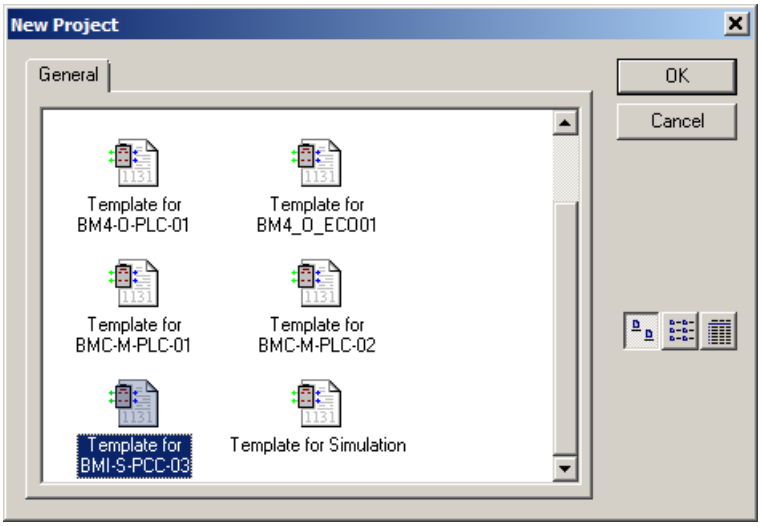


Figure 20: Project template for new ProProg project

The wizzard provide different project templates

- Choose „Template for BMI_S_PCC03“

The following figure shows the generated project from the project template.

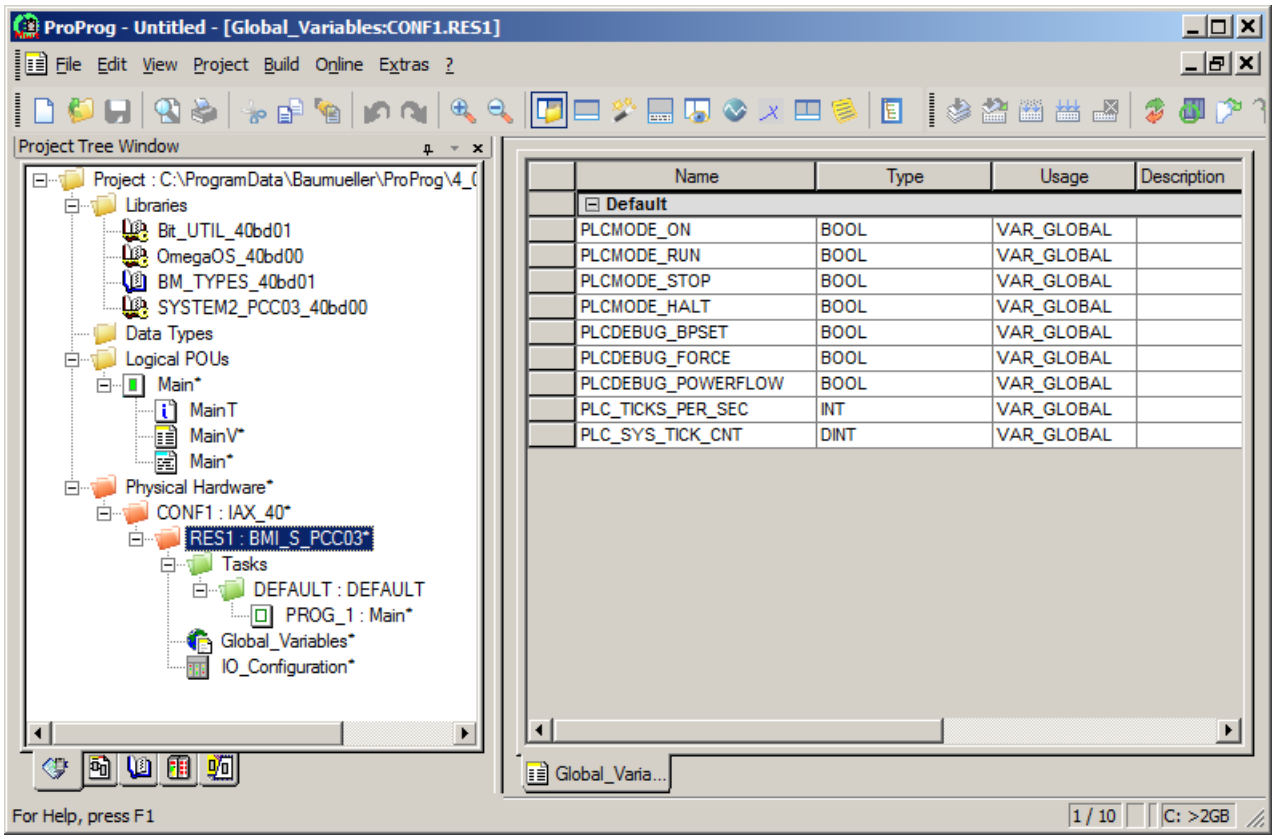


Figure 21: Generated project from project template

This project has a simple and clear appearance.
You can find a resource „RES1“ of type „BMI_S_PCC03“ under the project tree „Hardware structure“. The PCC-03 has been the previous model of **PCC-04**.

The

The higher-ranking configuration type is „IAX_40“. ProProg generates code for INtime based OmegaOS runtime systems on basis of Intel x86 processors.

- Select the entry „RES1 : BMI_S_PCC03“ of the project tree and open its settings with the Context Menu.

Adjust the IP address if necessary and enable both options of „Marked variables“ in window „PDD“ and „OPC“

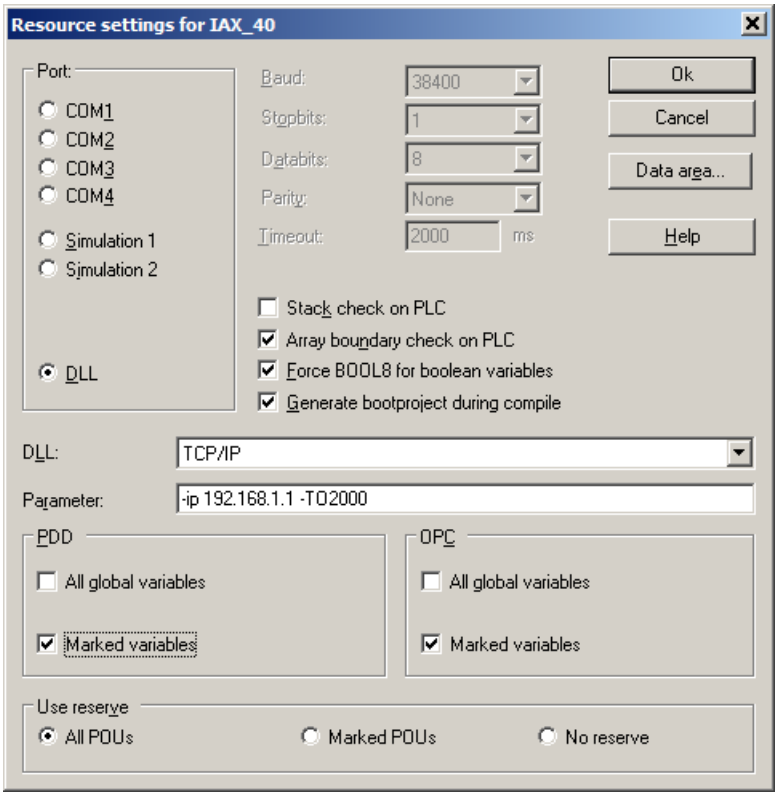


Figure 22: Resource settings

NOTE!

for expert user of other b maXX PLCs:

The following explanation up to ▹Integration of visualization◃ from page 66 are valid for all b maXX PLCs and should be known. The following information is for newcomers.

- The application can be compiled (key F9) and sent to the control unit (menu „Online“, „Check project..“) after checking the resource settings.

Figure 23: Control dialog of resource „RES1“

- Use the button „Send“ to send a application (a project).

The window „Send“ opens. Here you can select what shall be sent.

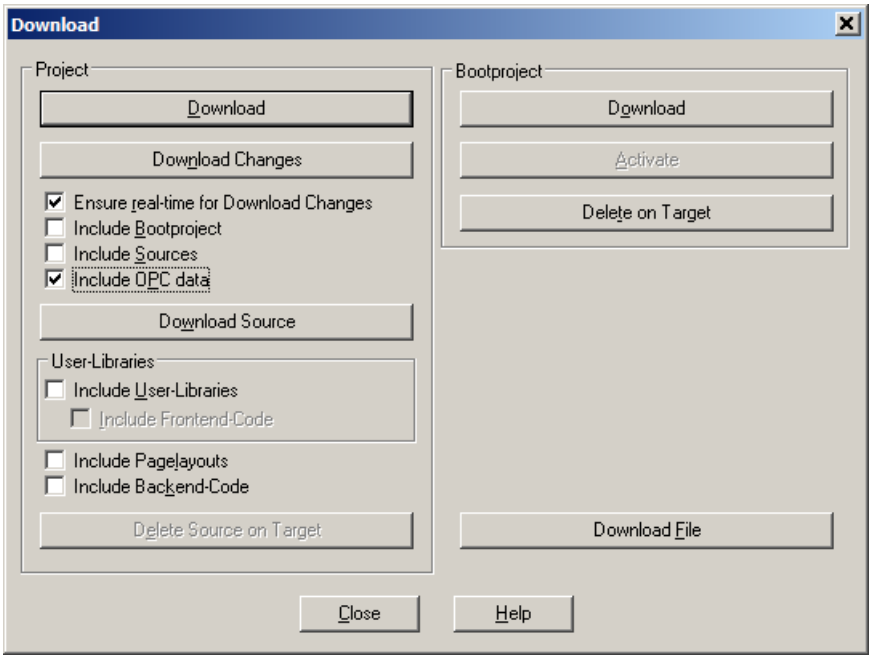


Figure 24: Window „Send“

Difference between RAM project and boot project

An application is run in RAM. ProProg compiles the project, sends the compiled code to the b maXX PLC and its runtime system OmegaOS executes the code (within the RAM).

The b maXX PLC must run without continual connection to ProProg for an efficient use. The application must be saved at the b maXX PLC - this is the boot project. The boot project is load from the persistent memory (e.g. flash EEPROM) to the RAM and executed there, when switching-on the PLC.

The application is available twice: in boot memory and RAM.

Window „boot project“

In window „Send“ you can find the frame „Boot project2 with 3 buttons.

- „Send“ transmits a new boot project from ProProg to the persistent memory.
- „Delete at destination“ deletes this boot memory. No boot project is available on the PLC now.
- „Activate“ loads the boot project from the persistent memory to the RAM.
  This generates a new RAM project. The former RAM project must be explicitly deleted before - control dialog „RessourceName“, button „Reset“.
  This newly load RAM project is not executed yet.

Frame „Project“

A direct access to the RAM project is desired while application development, as an alternative to the activation of a boot project (that has been transmitted to the boot memory before). The option for activating this can be found in the left frame „project“ of „Send“ window.

- „Send“ transmits a new code (entire new RAM project) from ProProg to RAM. This newly load RAM project has not been executed yet.
- „Send changes“ transmits and activates changes without stopping the PLC. Please note the remarks below.
- „Send program source“ and „Delete program source at destination“ are not available at b maXX controller PLC (BMC-M-PLC-0x) or b maXX drive PLC (BM4-O-PLC-01). The PC-based PLC system offers enough memory to save the entire ProProg project on the **PCC-04** (as ZipFile.zwt in directory D:\bm\zip\ ).

The options „Include boot project“, „Include program sources“ and „Include OPC data“ are related to the RAM helping to provide consistent data on the destination system.

Remark

The feature „Send changes“ has a „little brother“, the command „Online changes“ (shortcut ALT+F9 or in menu „Code“).

This command transmits incremental changes of a single POE and switches to the debug mode automatically. This command should be used preferred to the extensive feature „Send changes“ while application development.

For further information to this feature see „OmegaOS specific online help“ of ProProg.

For a first test it is no boot project necessary. The application is sent to the RAM.

- Use the button „Send“, on the top left in frame „project“.

Please not the ProProg message window, tab „Info“.

After successful download ProProg returns to control dialog „RessourceName“.

- Now a cold start (button „Cold“, on top right) and then the state is displayed (button „Info“ on bottom right).

The following window (/ the dialog „Resource“) is opened. The „PLC specific help“ of ProProg describes this window as „Dialog Resource“ ➙ button „Help“

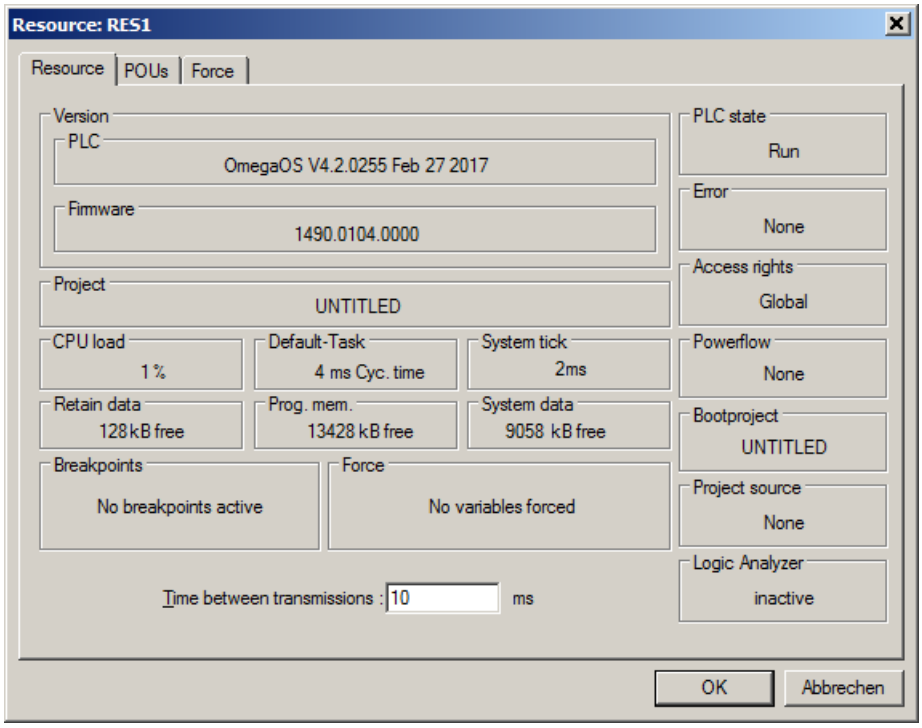


Figure 25: Dialog ‚Resource'

The minimal program generates no CPU load (.. the display is rounded up to 1%) and the memory ranges are empty.

The system tick of the runtime system is 2 ms.

- Close the info window

System tick, cycle task, default task

A task is a execution context, a frame within the assigned programs (more precisely: instances of program POEs) are executed from the runtime system.

There are cyclic executed tasks (cyclic task) , tasks as reaction to external events (event task) and tasks as reaction to internal events (system task).

Cyclic tasks do the main work of a PLC application, normally.
Examples for event tasks and system tasks are stop of the PLC, cold start or exceptions e.g. division by zero.

Event tasks can be cyclical, too. But the occurrence of the event and therefore the cycle are not controlled by the runtime system. There can be compared with interrupts in the easiest comparison. Examples are the synchronization with the fieldbus EtherCAT in the bypass event task „Motion1“, see ▹Fieldbus communication in eventtask „Motion1“◃ from page 55.

The system tick of the runtime system is the time basis of the cyclic task generated by the runtime system.

Only event task can be run in a smaller interval.

The default task is a special form of a cyclic task. The default task has no set interval, but is run as often as possible. The default task must share the unoccupied cycle time with a hidden, system task (amongst others for ProProg communication) because its priority is low. In an unloaded system the interval of the default task is (2 x system tick) = 4 ms, as shown before. The default task is blocked, if another cyclic task (with set interval) is ready for execution. Event task have a higher priority in general and lead to a delay of default tasks.

For further details refer to the online help.

Remanent data, cold start, warm start and hot start

Remanent data (program variables) of the PLC remain after the PLC was switched off and are available again after switch-on. A file on the data medium (SSD hard disk drive) is used as a retentive data memory.

NOTE!

Remanent data remains if the PCC-04 was correctly shut down (uninterrupted power supply is used if required)!

Remanent program variables must be declared as remanent.

- In the worksheet of variables (of a program POE or „Global Variables“) the variables are declared remanent by setting a checkmark behind the desired variable in column „Remanent“.

Three buttons „Cold“, „Warm“ and „Hot“ can be seen in the control dialog of a resource. This are the three possibilities to restart the stopped PLC.

- **Cold start**: Both remanent and normal program variables are initialized to the initial values (or default value 0) set in the ProProg project (boot project). For a new downloaded application only a cold start is possible.
- **Warm start**: The remanent variables are set to the current (last saved) value from the remanent data storage. The normal program values are initialized to the initial values set in the ProProg project (boot project). A warm start is the default start when switching-on the PLC. A cold start will be executed and a warning will be displayed in ProProg (in case the ProProg is connected to the PLC), if a inconsistence of the remanent data storage is detected.
- **Hot start:** Both types of program variables are set to the current value. This can be applied only while application development with ProProg online. After switch-off no hot start is possible.

For further details refer to the online help.

At this point the **debug mode** (key F10 or menu Online / Debug).

- A double click on „Global Variables“ in project tree opens the variables table in the workspace (if it is not already open). Now the variables table is in online mode: the current values of the variables are displayed.

This mode is valid for code worksheets, too. A double click on „Main“ in project tree opens the code worksheet of this POE (programming organization unit, a definition of IEC-61131-3, generic term for functions, function blocks or programs).
The current values of the used variables are displayed here, too. Editing is not possible in online mode.

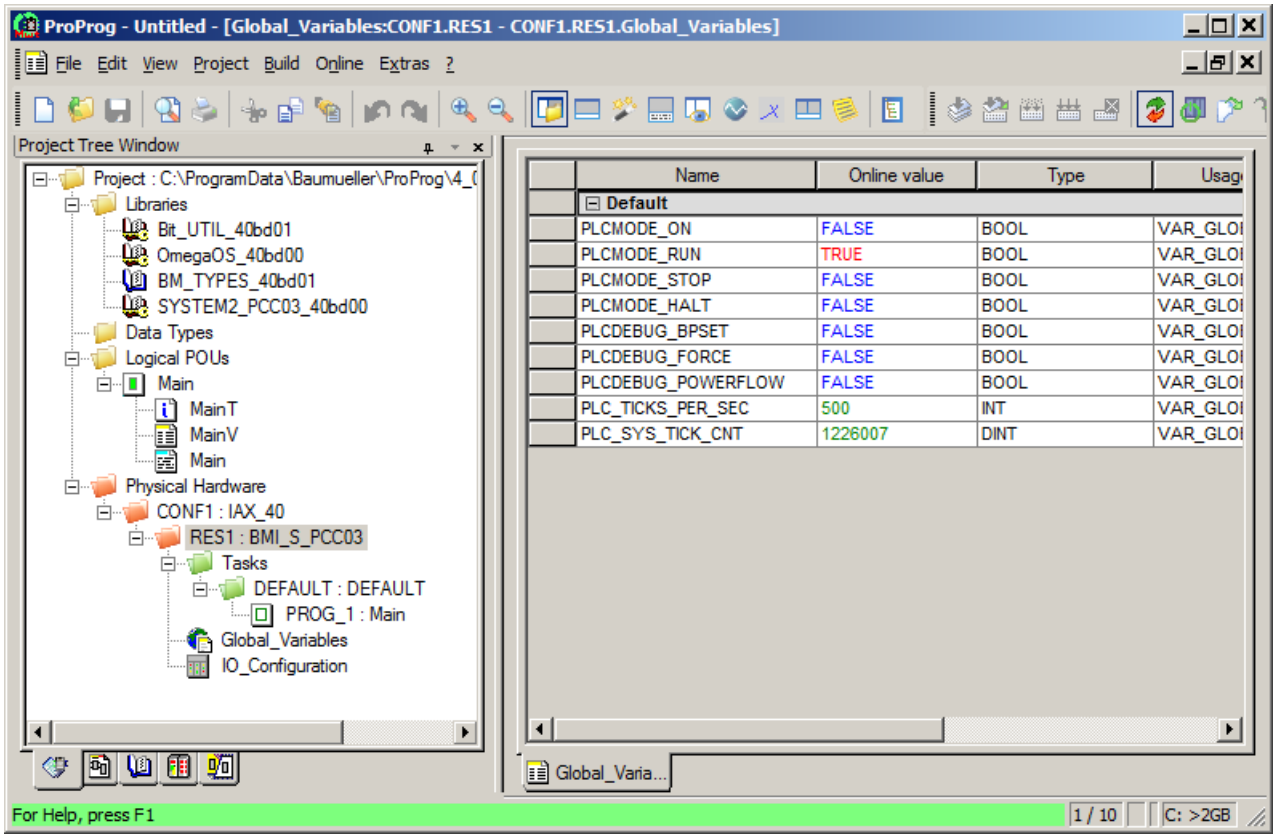


Figure 26: Worksheet „Global_Variables“ in debug mode

Storing of the ProProg project

The new generated project is not saved on the workstation PC, yet. No name is assigned, yet. The ProProg has assigned a name in background. As shown in the figure above, it is save with name „UNTITLED.mwt“ in hidden directory „C:\ProgramData\Baumueller\ProProg\4_0_10225\“.

There are two options, because it is deleted there when ProProg is closed.

- The ProProg project can be saved within the ProMaster project (see ▹Create a ProProg project (template) with ProMaster◃ from page 61). Then it is a component of the ProMaster project and e.g. is zipped to *.bmzip while saving.
- It is also possible to save and manage it separately.
  The simplest method is the continuous storing of „*.zwt“ (zipped ProProg project).
  The use of a version management is recommended for larger projects, e.g. Microsoft „Team Foundation Server“. Information to this subject can be found in online help of ProProg, index SCC (Source Code Control system).
- "Debug" must be switched off before saving.
- Then the ProProg command in the menu "File"
- "Save project under / Zip project under .. ". For example, save the project in "test.mwt" in "C: Users\ Public\ Documents Baumueller\ ProProg_4.0 Projects\".

Further application development

The generated PLC program demonstrates the procedure, but has no real function because of the missing fieldbus connection (= exclusive process connection of **PCC-04**).

Theoretically the extension of application fieldbus connection can be made manually. This procedure is not recommended.

All information for the desired and necessary fieldbus connection is set in the ProMaster component „ProEtherCAT“, before. At least by now it is mandatory, because the EtherCAT master must know what do to.

Furthermore the ProMaster can export this fieldbus dependent settings or MotionControl settings to ProProg.

Although ProProg can be used as stand-alone program, the export of process data and MotionControl objects from ProMaster to the ProProg project is essential.

Procedure

- After setting and testing the fieldbus configuration in ProMaster change back to **ProPLC / tab „IEC“**.

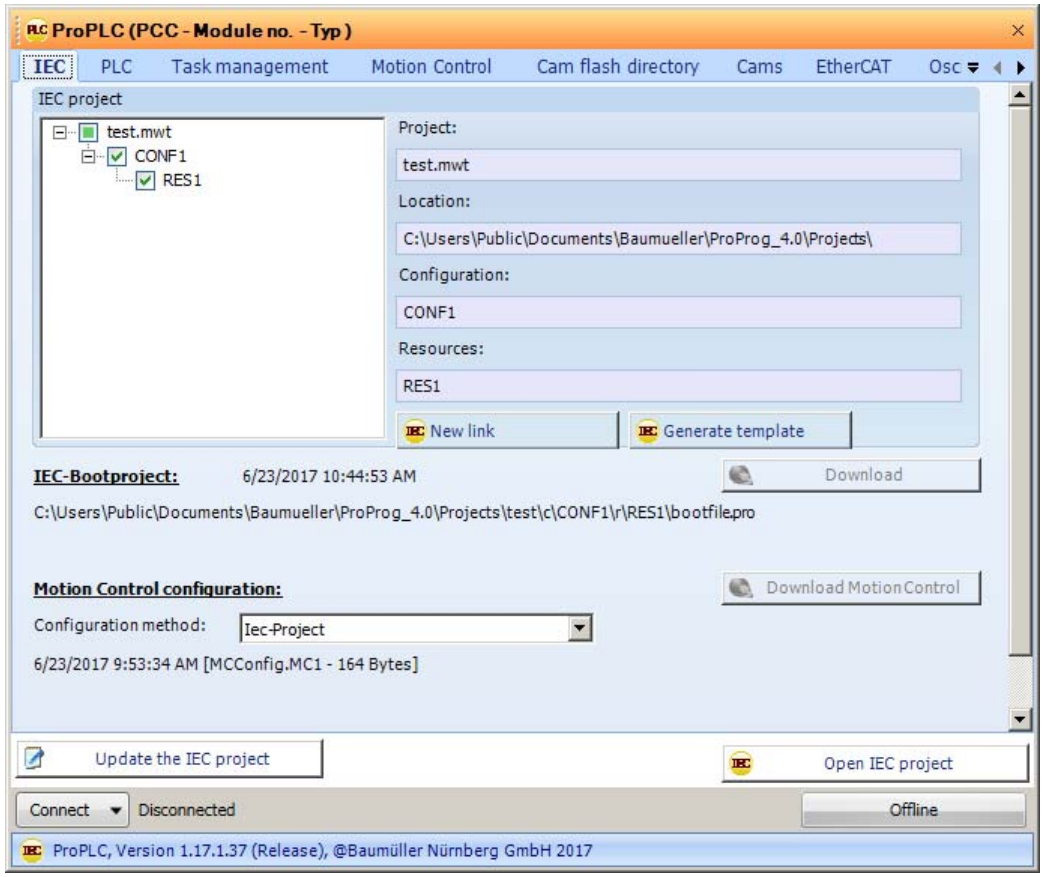


Figure 27: ProPLC / tab „IEC“

- Use the button „New link“ (almost exactly in the window center) to inform ProMaster about the name and directory of your ProProg project:
  e.g. „test.mwt“ in directory „C:\Users\Public\Documents\Baumueller\ProProg_4.0\Projects\“ for instance.
- Export fieldbus variables, motion control variables, settings, etc. into the new linked ProProg project using the button „Update the IEC project“ (on bottom left).

ProMaster starts ProProg, opens the set project and writes via remote control. Beside the fieldbus dependent configuration data the Motion Control configuration is written to the ProProg project, too.

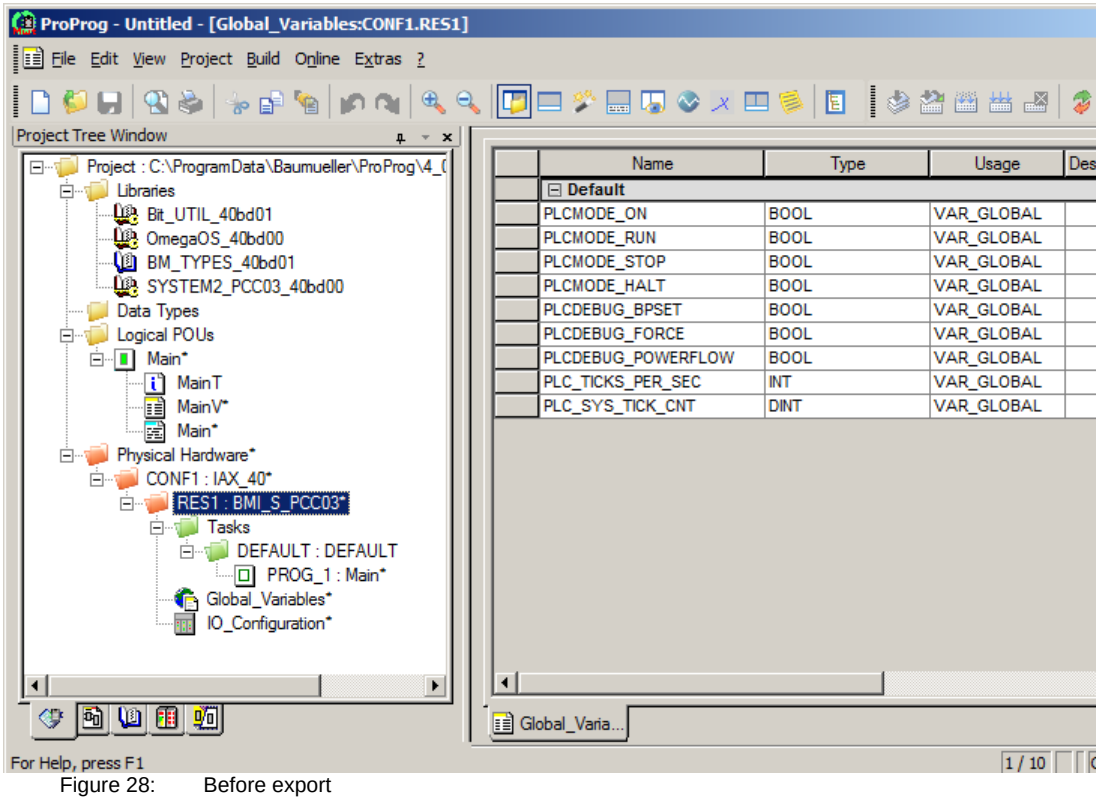


Figure 28: Before export

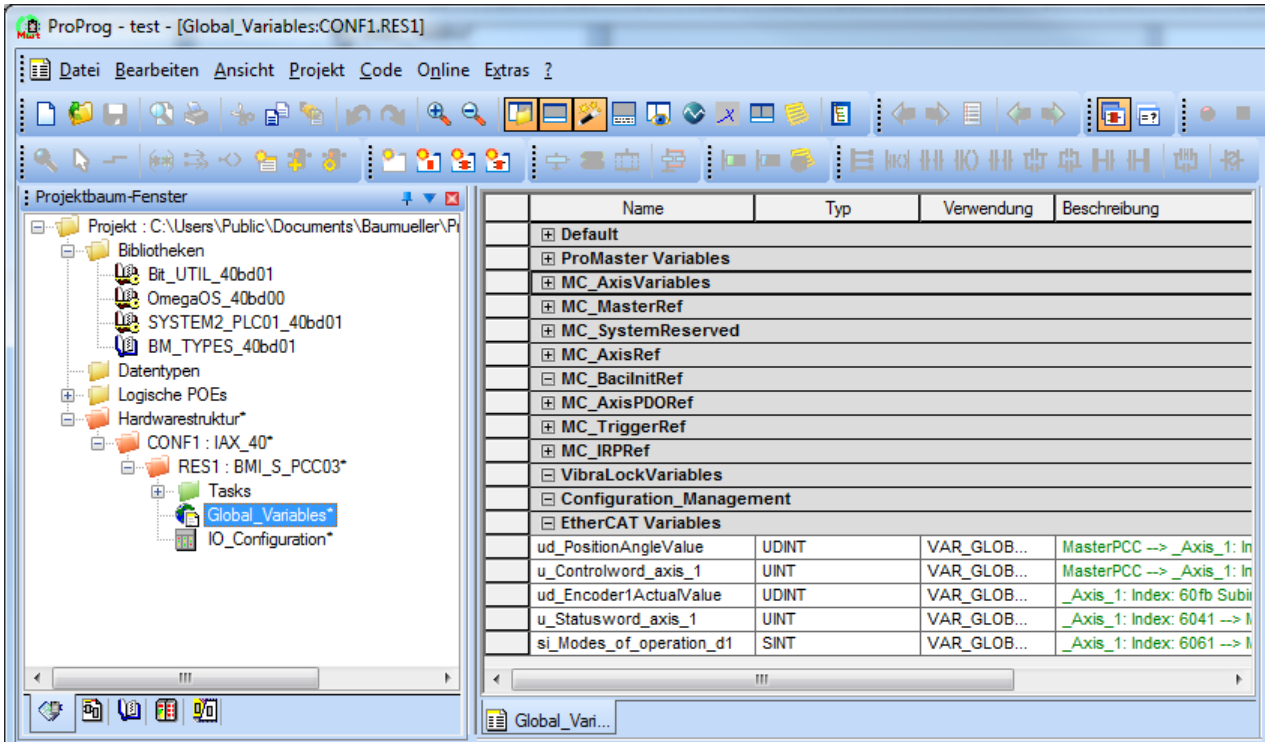


Figure 29: After export

Before export only the group „Default“ is existent in „Global_Variables“.

The export upgrades this: Total 8 variable groups MC_*** and the variable group „EtherCAT variables“ are written.

NOTE!

If in the variable group "MC_AxisVariables" at the variable "a_Axis" an error is displayed, click with the right mouse button on the variable and select "inhibit variable / release" in the context menu. Then the variable is inhibited.

NOTE!

The library „MOTION_TYPES_40bdXX“ must be embedded, in order that the extended project can be compiled.

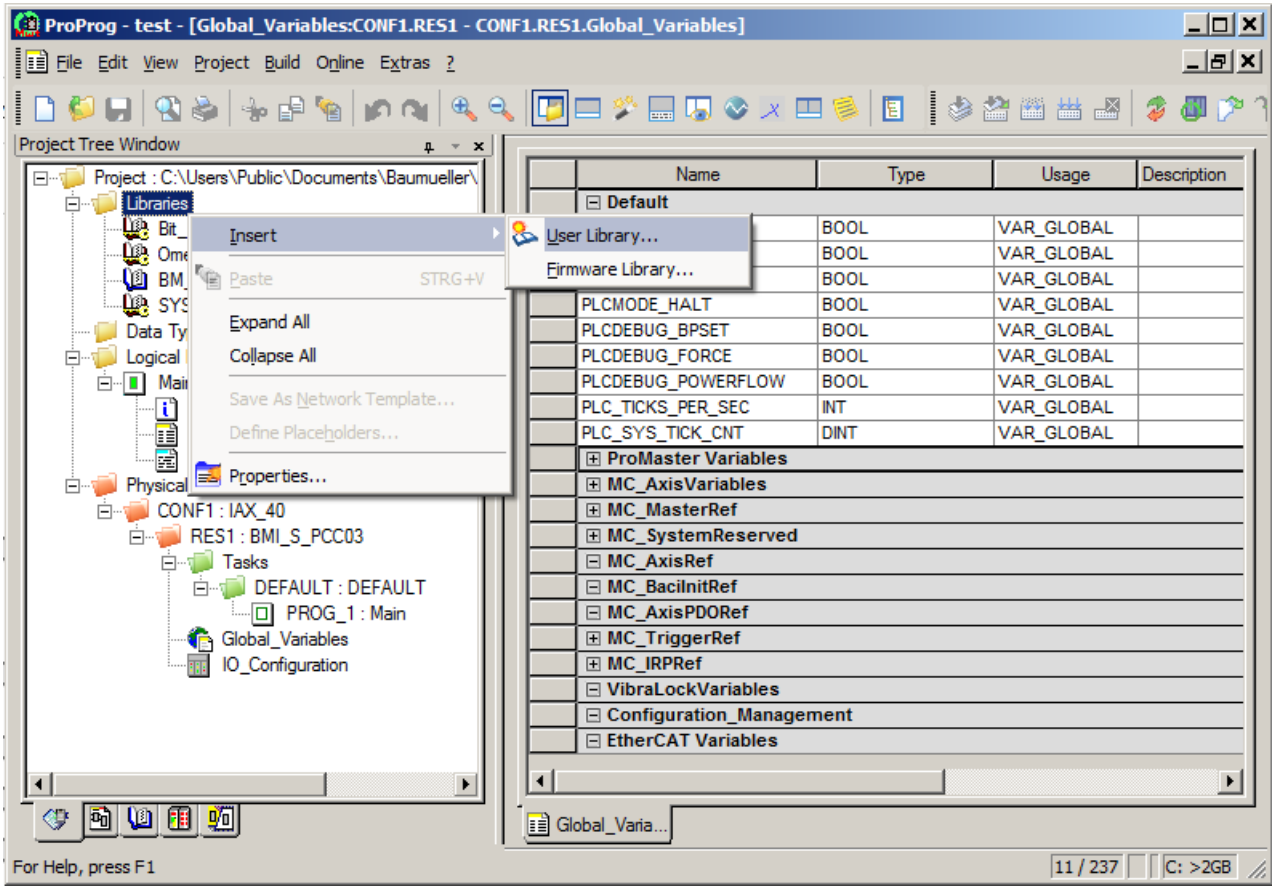


Figure 30: Embedding a library

For detailed description see chapter 5.4 „Export to the ProProg user program and download to the device“ of the „Application handbook EtherCAT“.

NOTE!

This export must be updated when fieldbus configuration is changed later. ProMaster project and ProProg project must be hold consistently.

Automatic start of fieldbus via MC core

Caused by export the MotionControl core was configured to start the fieldbus EtherCAT automatically, too.

For testing the completely fieldbus corresponding to the set configuration is necessary. The created configuration must be transferred to the EtherCAT master of the **PCC-04**, before.
The installation should be tested with ProEtherCAT, already. That means, it should be possible to change operation mode to OPERATIONAL in online mode, manually (see application handbook EtherCAT 5.10030).

- Reset the fieldbus to INIT or RESET with ProEtherCAT and leave the ProEtherCAT online connection active.
- Compile the extended ProProg project and load this project into PLC (RAM project).
- The ProEtherCAT bus state should be „OPERATIONAL“ after a cold start of the PLC (control dialog)

Fieldbus communication in eventtask „Motion1“

The fieldbus has been automatically started by the MC core of the b maXX PLC. No program code is necessary for that in the ProProg project.

The EtherCAT process data (both the process data necessary for MotionControl and the user defined process data) are available in „Global_Variables“.

Up to now no program code uses this process data and there is no task, allowed to run such POEs (program code).

This task must be run synchronously to the process data interval of the fieldbus.

For simple digital inputs and outputs (digital IOs) this is abdicable. For higher control and regulation functions e.g. a virtual leading axis for drives (synchronous target position setting) this synchronism is absolutely required.

Such a task is an event task, more precisely a bypass event task on the event 9, reserved for MotionControl.

Procedure to create a bypass event task:

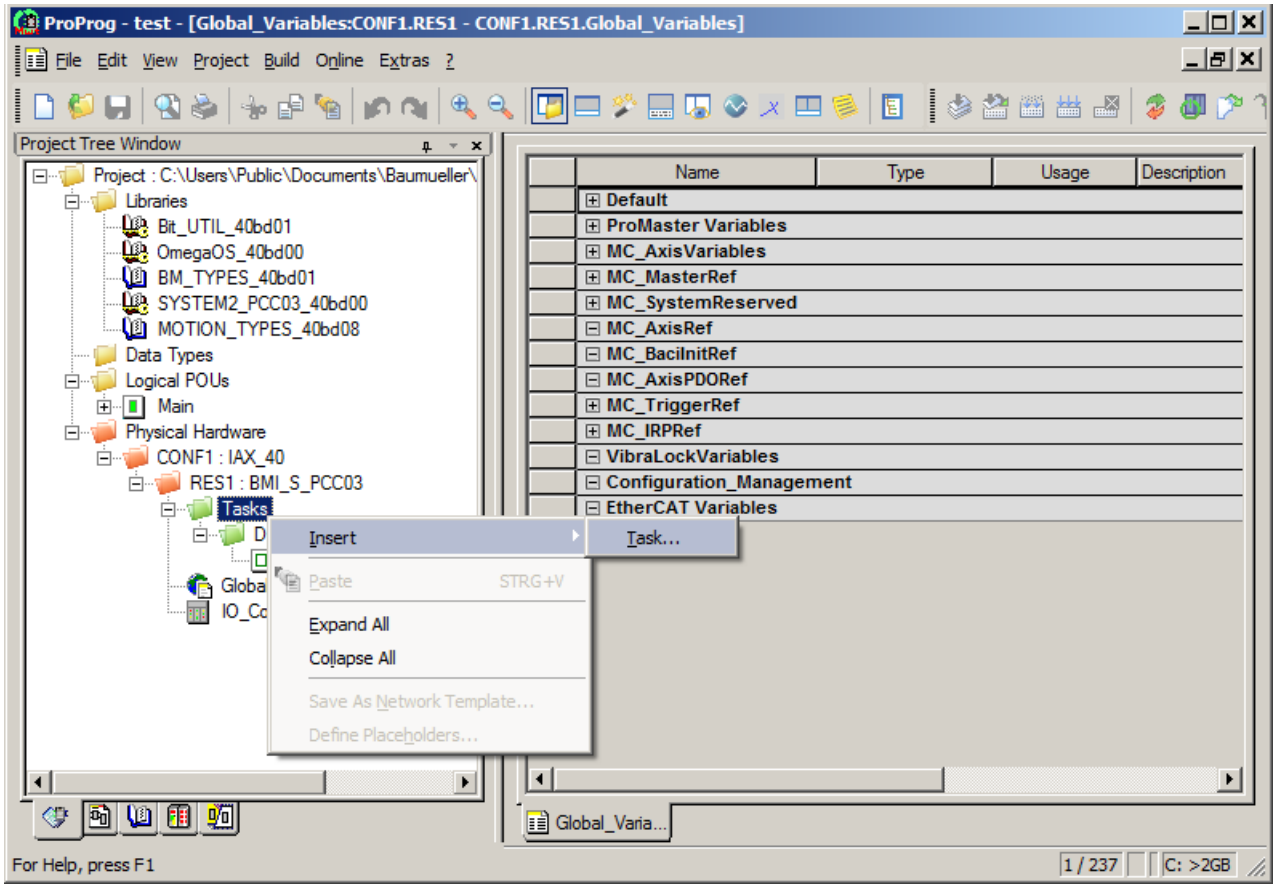


Figure 31: Create new task

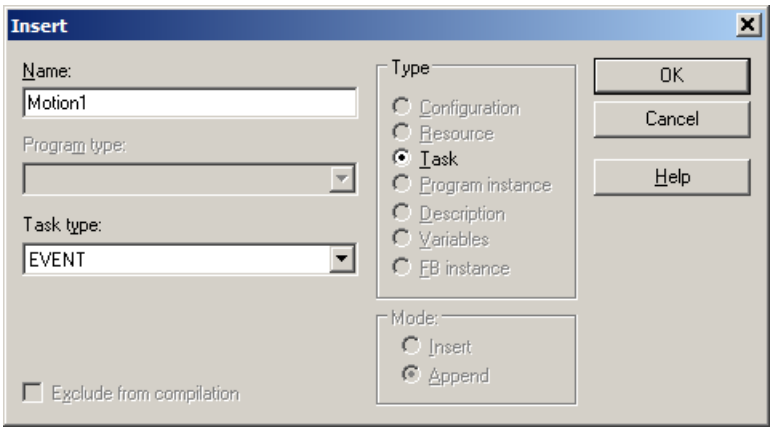


Figure 32: Attributes of the new task 1

The task name is irrelevant - because of consistency and overview the name „Motion1“ is selected.

- Important is the task type „EVENT“.
  This event task is specified in the next dialog.

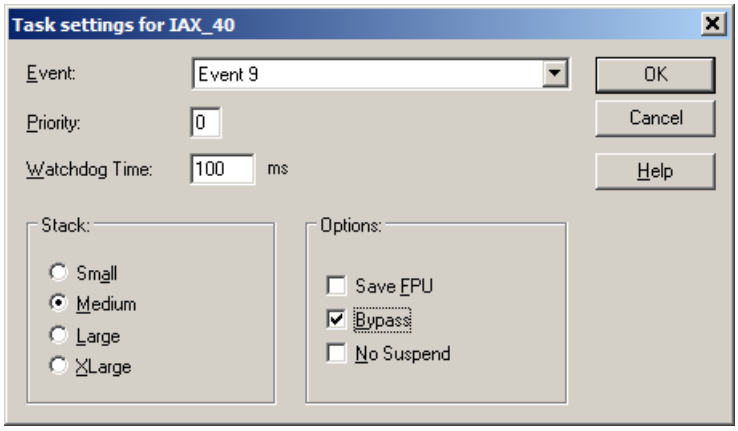


Figure 33: Attributes of the new task 2

- Important are the settings „Event9“ and „BYPASS“.

Each task must be assigned at least to one program POE, otherwise the project cannot be compiled.

To simplify matters we use the only available program POE „main“.

- Embedding of a new instance of the program „main“ via the context menu of the task entry in the project tree.

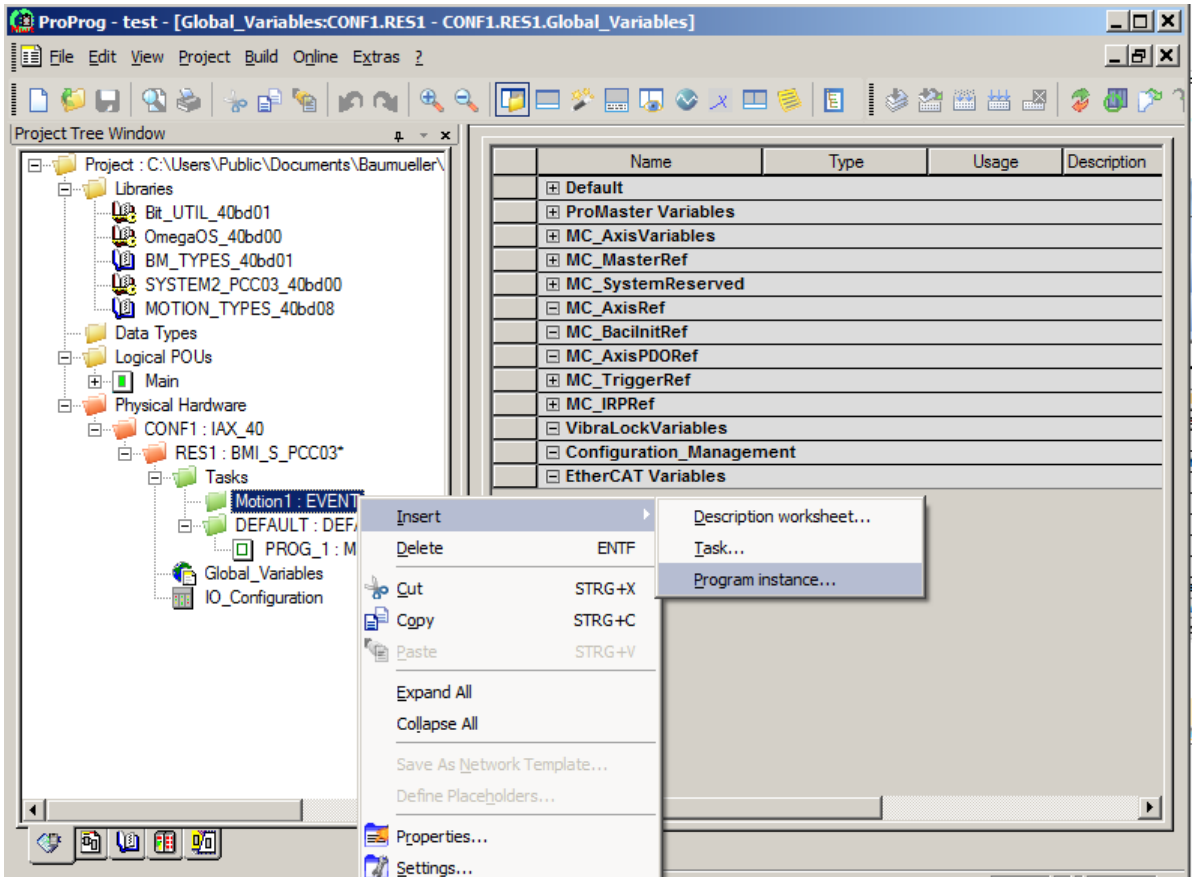


Figure 34: Embedding of a program instance

- The instance name is irrelevant - use e.g. „ev_00“.

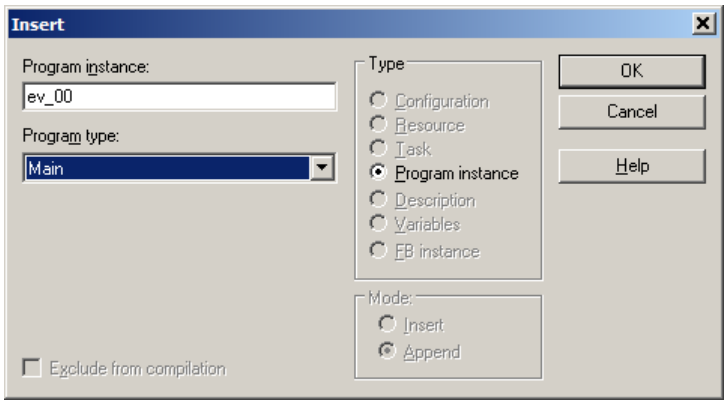


Figure 35: Embedding of a program instance

A second instance of the program POE „main“ is generated. The first instance is in the cyclic DEFAULT task.

- Compile the project, send it to the PLC RAM and initiate a cold start via control system then.

NOTE!

The project may not have any errors, otherwise it can't be compiled.

- Switch ProProg with key F10 in debug or online mode.
- Open the entry POE „main“ in project tree and open it in debug mode by double-clicking on the (bottom) code worksheet.

Set the instance before in Proper.

- Choose the instance „ev_00“ created before and confirm with „OK“.

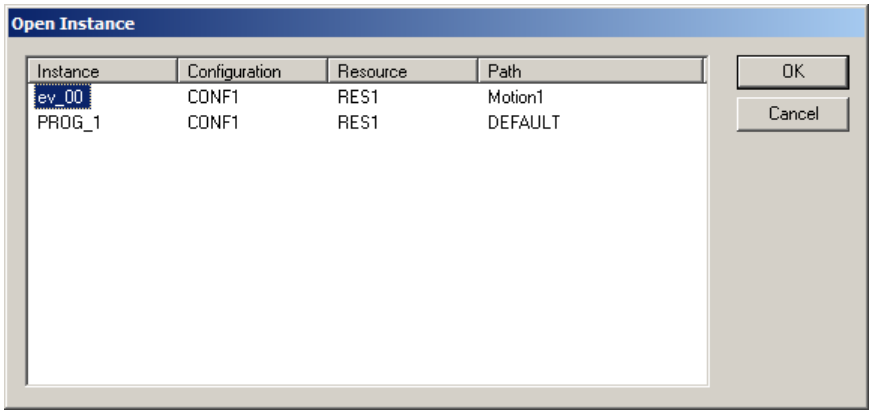


Figure 36: Choose instance

You can see only one task (incrementing the counter) in the debug view of the code worksheet. This code is operated once with each task call and because the task is synchronous to the process data interval of the EtherCAT master (e.g. each 1 ms), the counter should be incremented in the process data interval (e.g. each 1 ms). This counter is not linked to other counters of other instances of the same POE „main“, because it is a local variable.

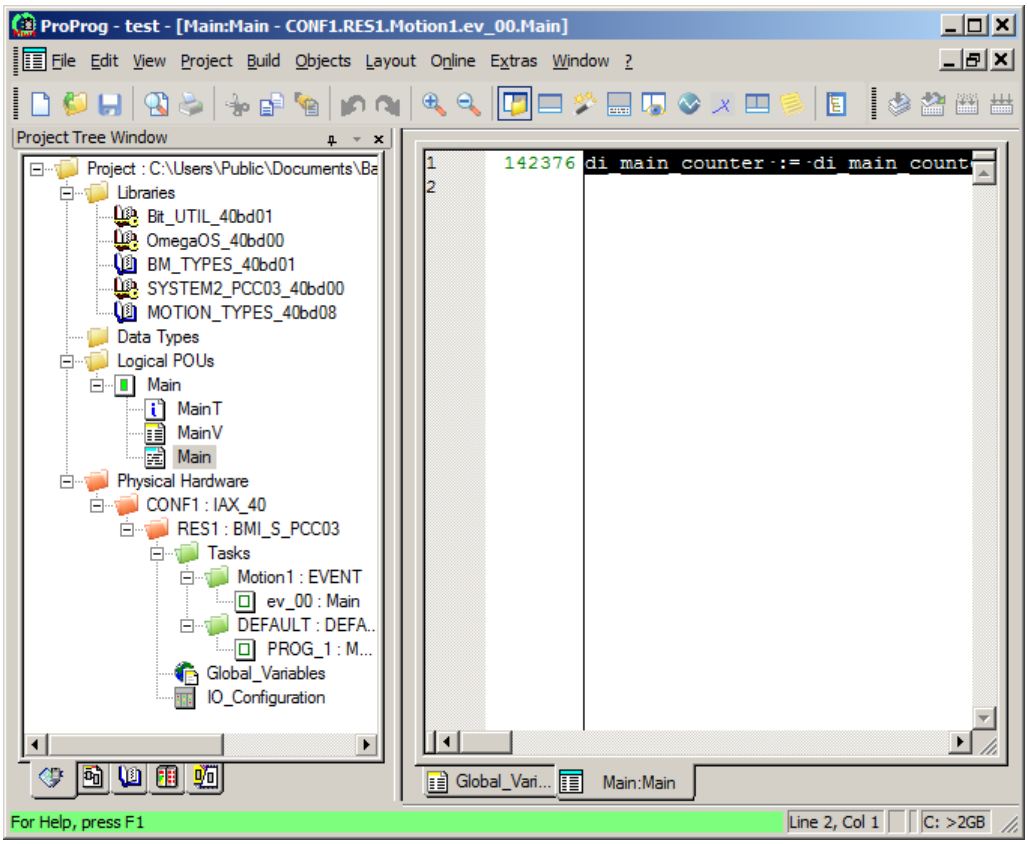


Figure 37: Debug view of the code worksheet

Use of the bypass event task

The program instances of POEs belong to this task. This task have to be run synchronously to the process data interval of the fieldbus, because they have to create e.g. new set values for the fieldbus via control and regulation algorithm (output signals) or have to react to input signals of the fieldbus (actual values) as fast as possible. Furthermore the consistency of related fieldbus input signals are assure only within this task.

POEs, that do not meet this demands should not included in the bypass event task.

The just used instance of „main“ is not suitable therefore (in case, the instance would not contain a counter, as in our example).

The generated template of ProMaster, shown in the following chapter demonstrates how to split code between event task and cyclic task.

Please note, that the bypass event task is optimized to optimal performance.

The optional attribute „Bypass“ effects in the invalidation of monitoring and protection (for performance reasons) mechanism of the runtime system of this task. The user POEs run in this task have to note that.

The whole PLC is blocked (incl. communication abort to the programming system), if a bypass event task overloads its computing time.

Please not in addition, that it is not written on MotionControl used EtherCAT process data.

The EtherCAT process data is marked with a grey background in ProEtherCAT, but is not specially marked in ProProg. For instance it is useless to write to the drive control word of a MotionControl axis, because the drive control word is overwritten by the MotionControl core, after execution of all assigned user POEs of the bypass event task „motion1“.

The further development of the ProProg project depends on the special demands of the application.

An example for a simple MotionControl application is shown in the following chapter.

5.3.2 Create a ProProg project (template) with ProMaster

All steps described before and in addition the start of the code generation can be run by ProMaster.

ProMaster generates a new ProProg project by request, including an application body (incl. MotionControl initialization, which comply with the configured fieldbus development). This can be used as basis for your PLC application.

- Open the (own) ProMaster project and mark the device **PCC-04**.
- Open ProPLC and change to tab „IEC“.

The button „Generate template“ is near center. This button opens the „IEC Project Wizzard“. There you can choose between different basic applications, specific for your configuration.

The new generated project is stored within the ProMaster project in sub-directory „IecTemplate“ by default. It is also possible (as shown before) to select a directory outside the ProMaster project and refer to this directory in the ProMaster project.

Regardless of template type: In all types a state machine is implemented in cyclic part (task „CY_20ms“) of POE „CY_MyApplication“. This state machine switches-on, positions (homing) and sets the selected operation mode (disc cam, synchronous mode, ...).
For the reasons already given, the bypass event task includes the POEs for leading axis generation (MC_MasterEngine), disc cam execution ...), only.

In the online help of the MotionControl libraries is for each MotionControl FB described, wether its instances have to be run in the motion event task or in the cyclical task.

In this context, it is especially important to note POE „EV_GetTime“. It provides the monitoring of bypass event task used by MotionControl. At least while application development this POE should be embedded in the motion event task and the logged maximum times should be considered!

5.3.3 Embedding specific libraries

ProProg libraries provide tested IEC-61131 code for re-use and help to structure more comprehensive applications.

User libraries can be created yourself, firmware libraries are provided by Baumüller.

For embedding a user library see ▹Embedding a library◃ from page 54.

The embedding of firmware libraries is different only differ to a very small extent.

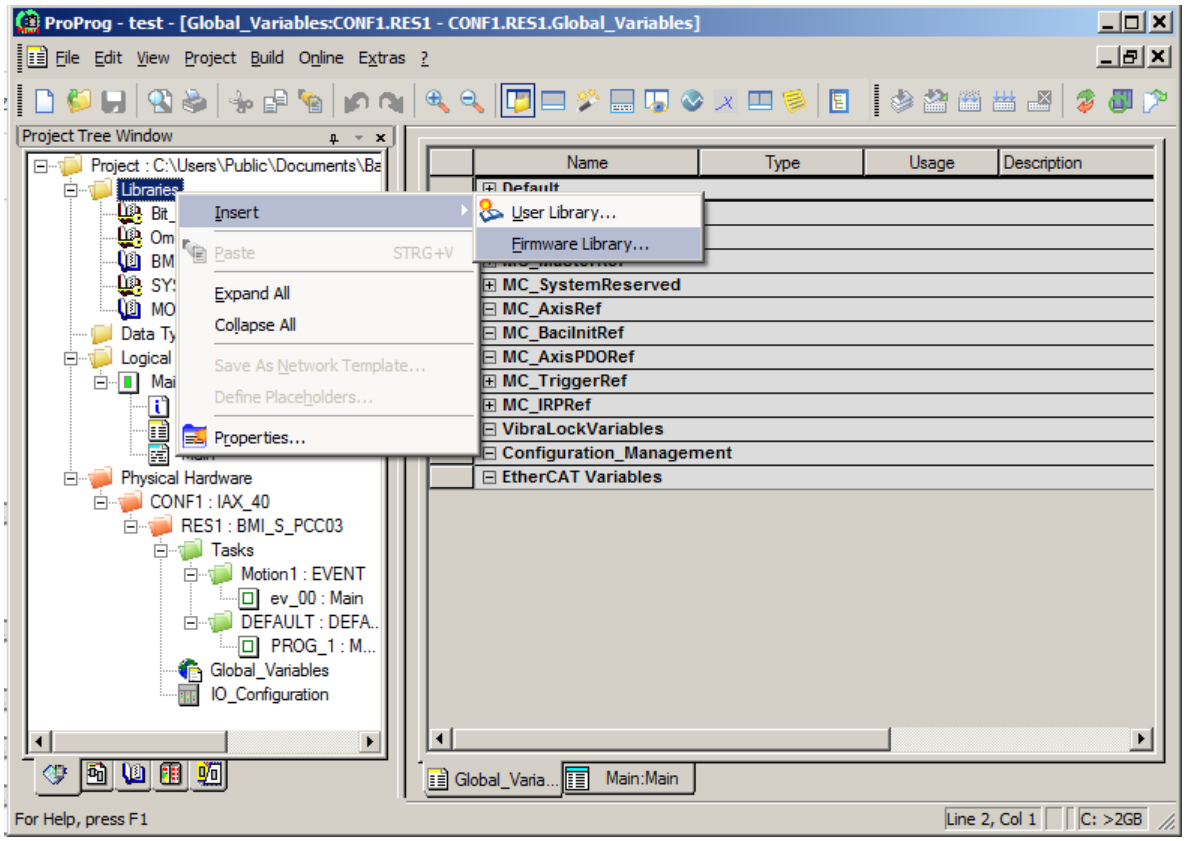


Figure 38: Embedding of firmware libraries

Libraries and POEs can be limited for use only with one configuration type and/or resource type. Such a library is displayed only in the „Editor Assistant“ (magic wand icon), if the opened, active POE on the worksheet corresponds with this limitations (has a suitable resource type).

A resource type independent POE (e.g. as from the template „Main“) are not able to embed function blocks from the in the following described firmware library, so it is not listed by the „Editor Assistant“.

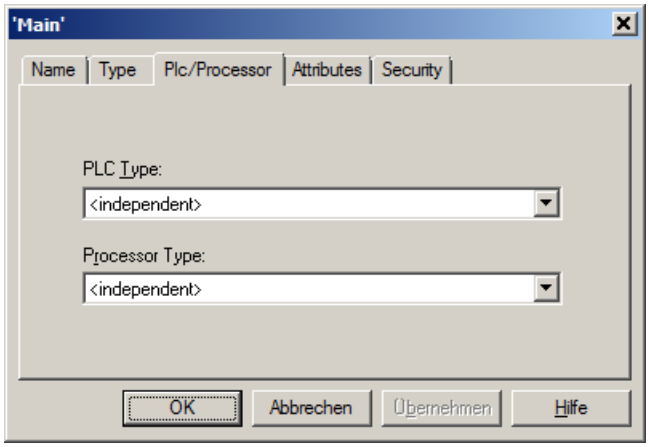


Figure 39: Attributes of a POE without limitation of the resource type

- In a suitable specialized POE the function blocks of the following library can be embedded.

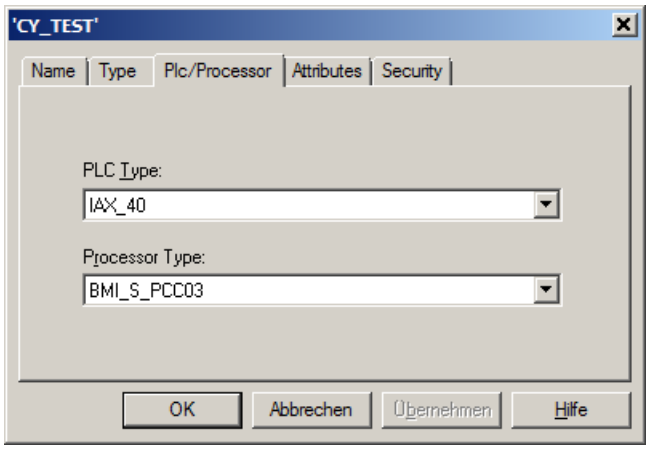


Figure 40: Attributes of a specialized POE

- The type assignment is set when creating of a new POE (see following figure, checkboxes in line at the bottom).

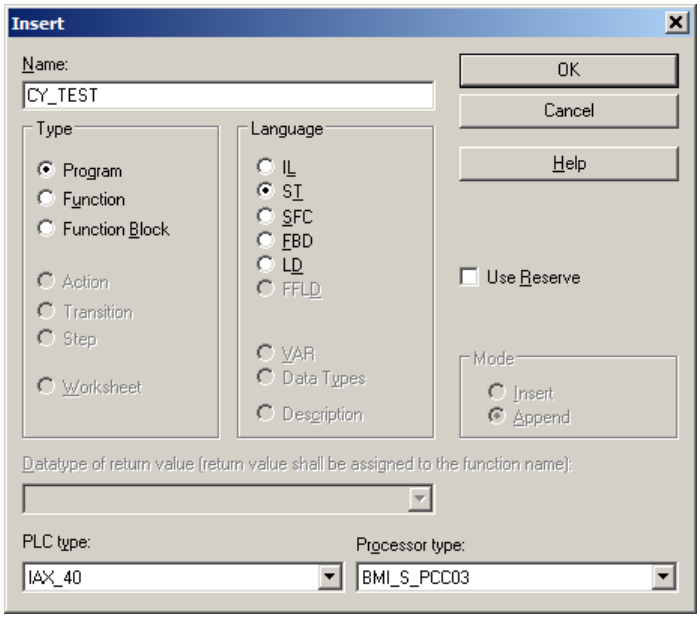


Figure 41: Create a new POE specialized for processor type

An example for a **firmware** library exclusively available on **PCC-04** is the FILE_PCC03_40bd01. It enables the access of the IEC application to the file system of the **PCC-04** in the enabled directories (e.g. D:\userdata).

All function blocks and notes for using them can be found in the online help of the library.

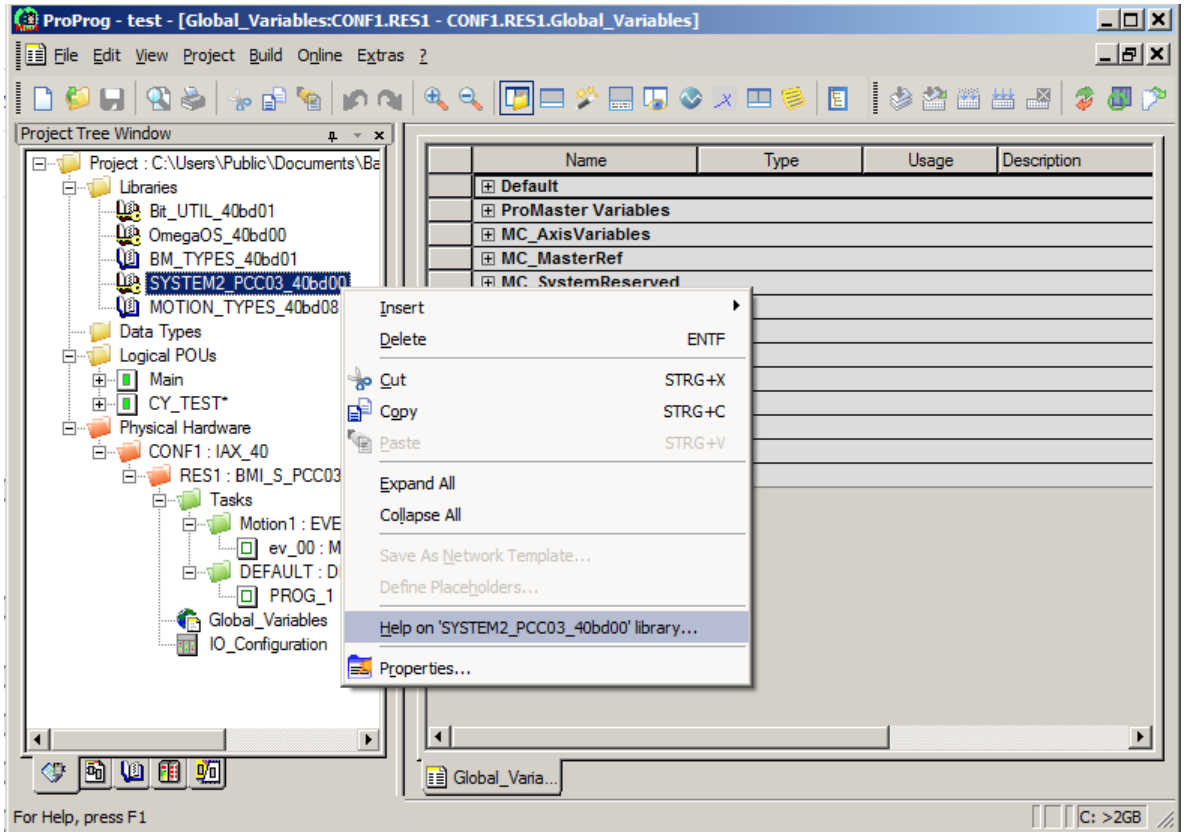


Figure 42: Online- help of a FWL

For additional information to this subject see online help of ProProg, index „Libraries“.

5.4 Integration of visualization

An HMI visualization or a SCADA application displays the state of the machine or system for the end user in a clear way. It provides control possibilities in the automation process, e.g. the control system of the machine or system.

An introductory overview was given in chapter ▹SCADA Visualization with web technologies and classical◃ from page 12.

This will be technically realized by providing the state of selected variables of the PLC program for the visualization. The PLC programmer determines in ProProg, which variables may be requested and read by the visualization.

5.4.1 ProOPC - Visualization via OPC server

A typical OPC-based visualization or HMI application acts as OPC client and accesses selected variables in the user program of the b maXX-PLC (the PLC application) via the b maXX OPC server „ProOPC II“. These variables are called OPC variables and have to be declared as OPC variable.

- Set the checkmark behind the requested variable in the column „OPC“ in the variable worksheet (a program POE or „Global Variables).

Furthermore there is a global release / blocking for OPC variables in the dialogue „resource settings“ (▹Figure 22◃ on page 44).

- Set or remove the checkmark here at „marked variables“ in order to release or block all OPC variables.

The checkmark „All global variables“ is generally not to be recommended, as otherwise the OPC object contains too many variables.

Note for programmers:

The standardized OPC interface between OPC client (e.g. HMI visualization) and b maXX OPC server works with the names of the OPC variables. However, the communication link between b maXX OPC server and b maXX PLC is based on logic addresses.

The corresponding allocation between names and logic address of the variables is written in the OPC project of the OPC server, in addition to the communication parameters for the corresponding PLC.

If this allocation between variable names and logic address of the variables will change, which may occur on each recompiling of the ProProg project, even the OPC project has to be updated.

These new allocations can be avoided, if fixed logic addresses are allocated manually to all of the OPC variables (e.g. %MB 1000 up to %MB 2000).

The complete procedure of creating and testing OPC server projects is described in the manual „b maXX OPC server ProOPC II“ and in the online help of the OPC configurator being installed with the b maXX OPC server.

5.4.2 ProViz - visualization by means of web technologies

A client server principle is used, similar to the OPC-based visualization.

In contrast to the OPC-based visualization, the whole HMI surface is stored on the eWeb server of the b maXX PLC. The Web client is a standard web browser or e.g. a specialized micro browser on small HMI panels (= Thin-Client) - it has no knowledge of the HMI visualization application, except of the URL (e.g. „http://192.168.1.1/Demo0815.htm“). It is placed completely and central on the eWeb server.

Contrary to the OPC server with its OPC variables, the eWeb server accesses the so-called PDD variables of the PLC application.

These PDD program variables have to be declared and labeled.

- For this purpose, set a checkmark behind the requested variable in column „PDD“ in the variables worksheet (a program POE or „global variables“)

There is a global release / blocking for the PDD variables in the dialogue „resource settings“, similar to the OPC variables (▹Figure 22◃ on page 44).

- Set or remove the checkmark here at „Marked variables“, in order to release or block all PDD variables.

The checkmark „All global variables“ is generally not to be recommended, as otherwise the eWebserver project contains too many variables.

The ProMaster component „ProViz“ is the editor for the HMI applications of the eWeb server of the b maXX PLC.

- Open ProViz either via the workspace or via the context menu (as to be seen in the following illustration)

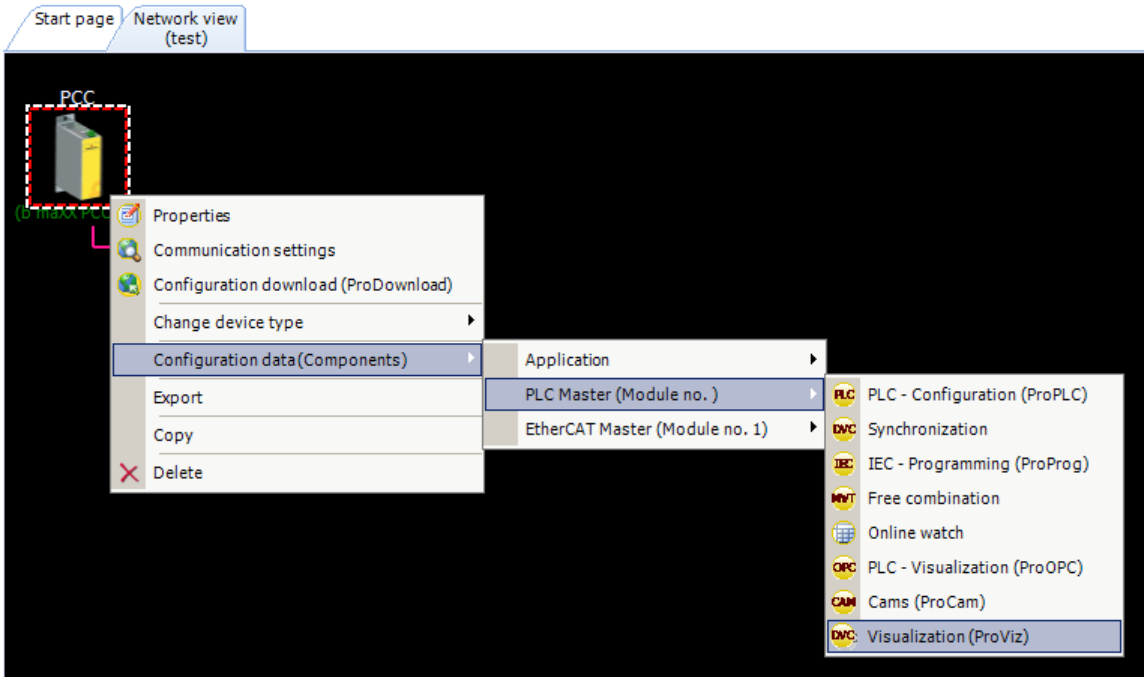


Figure 43: Start ProViz from ProMaster

A dialogue window for linking ProMaster project and ProViz project opens.

- Create a new project with the button „New visualization project..“

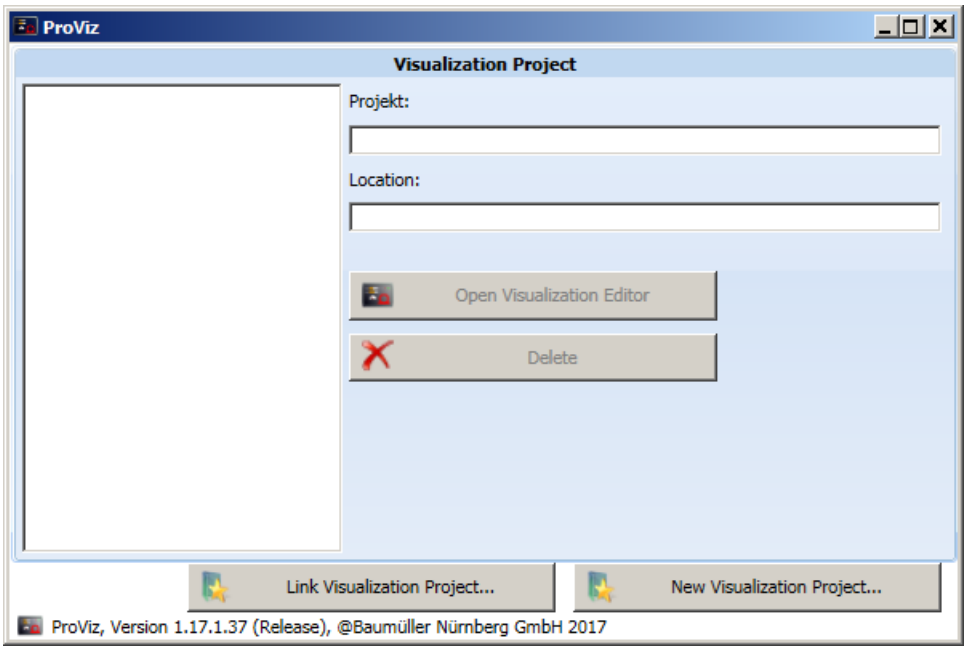


Figure 44: Link between ProMaster project and ProViz project

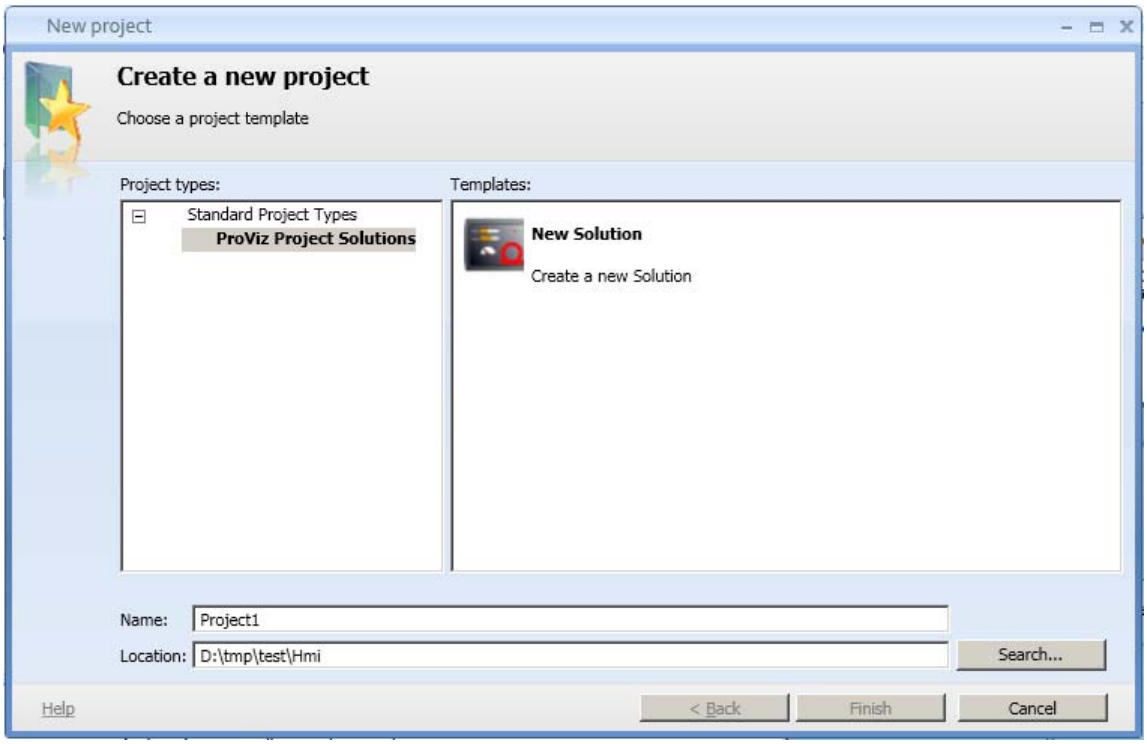


Figure 45: Project templates and wizard for ProViz-HMI projects

- The button „Open visualization editor“ opens the editor in the ProMaster dialogue.

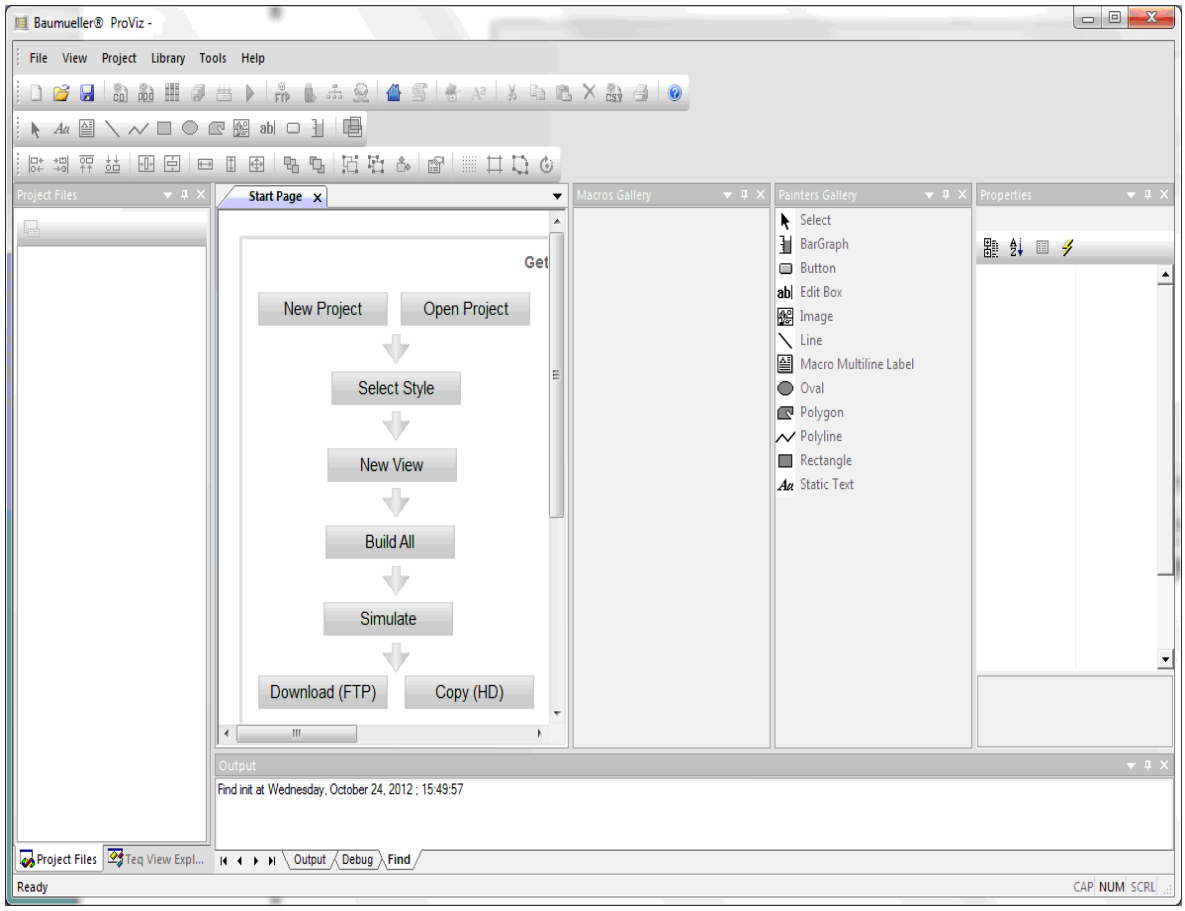


Figure 46: ProViz with the newly created project

The further procedure does not involve any **PCC-04** specific subject and would exceed the scope of this application instruction. It is therefore described in a separate manual and in the online help of the visualization editor.

5.5 Integration of a local Win7 application

5.5.1 Overview

The PCC-04 is a MotionControl based on the industrial PC engineering of Baumüller. It provides a real time operation system to execute control tasks. At the same time it provides a flexible application specific integration to higher automation levels by Microsoft Windows 7.

Besides the communication links of the general controls such as OPC, WebServer (CGI-Bin) or customer specific TCP/IP-communication general control, a direct input over "shared memory" (SHM) is possible at the local Win7 of the PCC-04.

After release in the central configuration file of the control, this application program range can be approached by a permanent base address. At first, the Windows application must embed a Win32 DLL of Baumüller. The position and size of the SHM now must be requested for. Then it can be read and written to.

As long as the PLC and RTOS are processed the SHM remains in force. It must be assured that the SHM is not used as soon as the PLC and RTOS are shut down switch-off sequence when switching off/shutting down the PCC-04).

The SHM can be accessed by two programs which are executed asynchronously on different processor cores and on different operating systems. Writing and reading accesses (IEC-61131-data types up to 32 bit) are consistent.

Further synchronizations (such as the block consistency) must be made by applications. Messages are not planned (events, callbacks).

5.5.2 Procedure

- The PLC must be notified if the feature is to be used

In order to do this edit the file to the PCC-04.

D:\BM\BOOT\PCC_PLC_001.INI

Remove the semicolon in front of the line in the section 'BACI modules'
'SlotH/M2=SHM_2':

```
[BACI-Modules]
SlotG/M1=ECM_1          ; EtherCAT master
SlotH/M2=SHM_2          ; BACI DPRAM emulation for Win32 DLL access
;SlotJ/M3=SHM_3
```

Note:

The legacy b maXX PLCs (drive-based BM4-O-PLC-01 and standard BMC-M-PLC-02) are constructed modularly.

The basic PLC can be extended by a pluggable BACI module such as a fieldbus interface connection. These modules have an own micro controller and communicate by a dual-ported RAM (BACI-DPRAM) with the basic PLC.

In correspondence to the drive-based BM4-O-PLC-01 the PCC-04 provides an EtherCAT master in slot G as a virtual module.

Due to this configuration a SHM type module is now added in slot H (virtual). This type is a placeholder for Win32 programs which use the NTX-DLL „am734.dll“.

- Application user program of the PLC

After restarting the PLC (not stop/start of the PLC user program but restarting the PCC-PLC-001.RTA) the access on a virtual BACI module in slot H is possible in the user program of the PLC.

That means that the BACI-DPRAM addresses %MB3.3000000 ... %MB3.3262144 remain in force and can be read and written by the user program.

The range is 262144 = 0x40000 = 256 kByte.

- Windows application

The Windows program requires an access to a memory range which is made available from the RTOS.

For this the Windows program must integrate the Win32-NTX-DLL "am734.dll" (as well as "am734.lib"). Then request for the position and the size of the SHM on the PLC by "LoadRtosSharedMemory" function to read and write.

The following example describes the procedure for C++ under VisualStudio.

Due to the integration of the associated LIB file in the link settings of your C++ project the DLL and the used functions are loaded during operation time.

Refer to the following:

```
// need to include .\release\am734.lib at linker settings
#define AM734_DLL_EXPORT  __declspec(dllimport) PASCAL // for 32-bit
extern "C"
{
 LPVOID AM734_DLL_EXPORT LoadRtosSharedMemory(LPCTSTR pszNodeName, LPCTSTR pszRtaName,
                         LPCTSTR pszMemName, DWORD* pByteLength);
 VOID   AM734_DLL_EXPORT UnloadRtosSharedMemory(LPVOID pMem);
 LPCSTR AM734_DLL_EXPORT getLastRtosErrorMessage(void);
} // end  extern "C"
```

Your functional code could start as follows:

```
CString csNodeName(_T("NodeA"));
CString csRtaName(_T("PCC_PLC_001"));
CString csMemName(_T("SHM_2_DPRAM"));
DWORD dByteLength= 0L;

LPVOID pMem= LoadRtosSharedMemory(csNodeName, csRtaName, csMemName, &dByteLength);
if (pMem)
{
        CString csReport;
        csReport.Format(_T("got pMem=%08X and len=%i."), pMem, dByteLength);
        REPORT(csReport);
}
REPORT(getLastRtosErrorMessage());

if (NULL == pMem) { REPORT(_T("sorry !")); return; }
```

```
My_access_to_SharedMemory(pMem, dByteLength);

UnloadRtosSharedMemory(pMem);
REPORT(getLastRtosErrorMessage());
```

In the following are the designations of the required SHM range:

- "NodeA" =The name of the RTOS instance
  (besides Windows there can be several separated RTOS on one multi-core processor.)
- "PCC_PLC_001" = The name of the real-time application, here the first OmegaOS-PLC.
- "SHM_2_DPRAM" = The name of the required SHM range
  (or "SHM_3_DPRAM" for line 'SlotJ/M3=SHM_3' in the INI file to access the address range from '%MB3.4000000')

The provided address '%MB3.3000000' in the user program of the PLC accords to the address in the pointer 'pMem'.

Note:

For testing purposes (e.g. if there is no PLC user program that writes on '%MB3.3000000' yet) you can read the BACI-DPRAM of the EtherCAT master, as well.

This SHM range has another name and points out other addresses in the user program of the PLC.

```
CString csMemName(_T("ECM_01_DPRAM"));
```

From Byte-Offset 8 is the BoardClassID of the EtherCAT-Master:

```
DWORD dBoardClassID= *(((DWORD*)pMem) + 2); // Byte 8
```

Here, the value 0x00018207 should be readable.

The firmware identification of the EtherCAT master is from Byte-Offset 1102 onward:

```
WORD wFwNumber= *(((WORD*)pMem) + 551); // Byte 1102
```

Here the value 1496 should be readable.

Other accesses (in particular writing accesses are not authorized!

The function call

```
My_access_to_SharedMemory(pMem, dByteLength)
```

is a place holder for your application on the Windows side.

5.5.2.1 Shared memory (SMH)

Please consider that this SHM can be accessed by two programs which are asynchronous to the different processor cores and can be executed by different operating systems.

Writing and reading accesses to basic data are consistent (IEC-61131-data types up to 32 bit).

Other synchronizations (such as with regard to the block consistence) must be solved by applications. Notifications (events, callbacks) are not provided.

The SHM remains applicable as long as the PLC and RTOS are operated.

Therefore, assure that the Windows application does no longer use the SHM as soon as the PLC and the RTOS are shut down (note switch off sequence during switch-off/shut-down of the PCC-04).

5.6 Firmware update

The software components of the PCC-04 not being assigned to the customer's application and delivered by Baumüller directly together with the hardware are determined as firmware, by analogy with usual controls („embedded systems").
These are the operating systems (Windows embedded and RTOS), the PLC runtime system, the EtherCAT master, the SCADA server and other.

The whole firmware of the PCC-04 is stored at the SSD/mSATA hard disk.

(Other firmware of an industrial PC, e.g. the BIOS of the mainboard, is considered to belong to the hardware.).

The firmware update is possible only at Baumüller's. Therefore, the PCC-04 must be sent back (see Instruction Manual PCC-04).

5.6.1 Manual replacement of single files

Many components of the firmware can be accessed separately via the file system of the Windows operating system.
The component EtherCAT Master, for example, is stored in file „PCC_ECM_001.rta" in the directory „C:\BM\bin\INtime" and can therefore be replaced in the installation mode of the PCC-04 (see ▷Installation mode, application mode and protected mode◁ from page 21).

This procedure requires the explicit attention to internal dependencies (e.g. „the new version 1.23 of the ECM.rta also requires a new version of the PLC.rta"). Thus, such a procedure demands accuracy and is recommended for skilled application engineers only.

5.6.2 Firmware update by ProMaster

Of course it is more convenient and safer to use a tool-supported firmware update by ProMaster or ProService. Follow the instructions of the relating software tools.

APPENDIX A - ABBREVIATIONS

API	Application Program Interface
ARP	Address Resolution Protocol
CAN	Controller Area Network
CF	Compact Flash
CiA	CAN in Automation e. V.
COB	Communication Object
COB-ID	Communication Object Identifier
CPU	Central Processing Unit
CSMA/CD	Carrier Sense Multiple Access / Collision Detection
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
DIN	German Institute for Standardization
DLL	Dynamic Link Library
DPRAM	Dual Ported RAM
DRAM	Dynamic RAM
EMV	Electromagnetic compatibility EMC
EN	European standard
EPROM	Erasable Programmable Read Only Memory
ESD	Electrostatic sensitive device
EXT, ext	External
FTP	File Transfer Protocol
HAL	Hardware Abstraction Layer
HD	Hamming distance
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
IEC	International Engineering Consortium
I/O	input/output
I/O-bus	bus for input modules and output modules
IP	Internet Protocol
ISO	International Standard Organization
LAN	Local Area Network
MAC	Media Access Control
NIC	Network Interface Card
NOVRAM	**No**n-**v**olatile RAM
OPC	**O**LE for **P**rocess **C**ontrol (OLE: Object Linking and Embedding)
PLC	Process loop control
POE	Power
ProProg	Tool for programming the b maXX PLC and the control system PCC
RAM	Random access memory
RTOS	Real time operating system
SCADA	Supervisory Control and Data Acquisition
SDRAM	Synchronized Dynamic RAM
SPS	programmable logic control
SW	Software
VDE	Association for Electrical, Electronic and Information Technologies e.V
Windows XPe	Windows XP embedded
WES7	Microsoft Windows® Embedded Standard 7
www	World Wide Web
16#	Prefix for hexadecimal number

Index

Overview of Revisions

Version	Status	Changes
5.16008.01	22.07.2016	Creation
5.16008.02	10.07.2017	Ch. 4.6.3 added Ch. 4.10 SCADA server and 4.11 MicroBrowser added Ch. 5.5 Integration of a local Win7 application added Ch. 5.6.2 added

Notes:

be in motion

Baumüller Nürnberg GmbH Ostendstraße 80-90 90482 Nürnberg T: +49(0)911-5432-0 F: +49(0)911-5432-130 **www.baumueller.de**