KENDRION

WE MAGNETISE THE WORLD

INDUSTRIAL CONTROL SYSTEMS

# Instruction Manual

## Kuhnke FIO Controller 116

E 844 GB 25/08/2021

CODESYS

CANopen®

EtherCAT®

KUHNKE

# Table of Contents

# 1 Preface

## 1.1 Legal Notice

### Contact Details

Kendrion Kuhnke Automation GmbH
Industrial Control Systems
Lütjenburger Straße 101
D-23714 Malente, Germany

Tel. +49 4523 402-0
Fax +49 4523 402-201

Email sales-ics@kendrion.com
Web kuhnke.kendrion.com

### 1.1.1 Versions

| Modification history | |
|---|---|
| **Date** | **Comments / Modifications** |
| **17/09/2018** | Original version |
| **10/07/2021** | Warning added: Dangerous failures due to incorrect voltage supply |
| **26/04/2021** | Profinet Extension: Description extended<br>Retained data: As of firmware 1.25.0 retain and persistence declarations are supported. |
| **18.08.2021** | IT Security |

## 1.2 About this Manual

This technical information is primarily directed to system designers, project engineers and device developers. It does not contain any availability information. We reserve the rights for errors, omissions and modifications. Pictures are similar.

### 1.2.1 Limitation of Liability

Specifications are for description only and are not to be understood as guaranteed product properties in a legal sense. Exact properties and characteristics shall be agreed in the specific contract. Claims for damages against us - on whatever grounds - are excluded, except in instances of deliberate intent or gross negligence on our part.

### 1.2.2 Terms of Delivery

The general conditions of sales and service of Kendrion Kuhnke Automation GmbH & Co. KG shall apply.

### 1.2.3 Copyright

This instruction manual is proprietary and protected by copyright.

No part of this document may be reproduced or copied in any way or by any means except expressly permitted in writing by Kendrion Kuhnke Automation GmbH.

Microsoft®, Windows® and the Windows® logo are registered trademarks of Microsoft Corp. in the USA and other countries.

EtherCAT® is a registered trademark and patented technology, licenced by Beckhoff Automation GmbH, Germany.

Further information about the PLCopen organisation is available at www.plcopen.org. CiA® and CANopen® are registered joint brands of CAN in Automation e.V. Title to all companies and company names mentioned herein as well as to products and product names is held by the respective enterprises.

CODESYS© is a product of CODESYS GmbH.

### 1.2.4 Software licenses

Firmware

The firmware of the devices contains open source software. Parts of this software are licensed under the following and other open source licenses:

- GNU General Public License (GPL)
- MIT License
- BSD Zero Clause License
- GNU Lesser General Public License (LGPL)
- Mozilla Public License (MPL)
- FreeType License (FTL)

The source code of the free software can be requested from the product management Kendrion Kuhnke at cost price within three years after delivery of the device.

CODESYS

The installed CODESYS Runtime, like all CODESYS products, is subject to the terms of the End User License Agreement (EULA), of CODESYS GmbH, which can be viewed on the CODESYS website.

### 1.2.5 Warranty

Warranty is subject to the provisions of the conditions of sale of Kendrion Kuhnke Automation GmbH or any contractual agreements between the parties.

## 1.3 Reliability, Safety

### 1.3.1 Applicability

For reasons of personal safety and to avoid material damages when working with or handling this Kuhnke product, you are advised to take heed of the notes and information contained in this instruction manual.

### 1.3.2 Target Group of the Instruction Manual

This instruction manual contains all information necessary for the use of the described product (control unit, control terminal, software, etc.) according to instructions. It is written for design, project planning, servicing and commissioning experts. For proper understanding and error-free application of technical descriptions, instructions for use and particularly of notes of danger and warning, extensive knowledge of automation technology is compulsory.

### 1.3.3 Intended Use

Kuhnke's products are designed, developed and manufactured for standard industrial use. They must not be used for any other purposes than the ones specified in the catalogue or the associated technical documentation. Proper and safe operation depends on the products being transported, stored, lined up, mounted, installed, put into service, operated, and serviced correctly. Ambient conditions must be within the admissible limits. Notes and information in the associated documentation apply at all times.

### 1.3.4 Reliability

Reliability of Kuhnke products is brought to the highest possible standards by extensive and cost-effective means in their design and manufacture.

These include:

- selecting high-quality components,

- quality agreements with our suppliers,
- actions to avoid static charges when handling MOS circuits,
- worst case planning and design of all circuits,
- visual inspections at various stages of fabrication,
- computer-aided tests of all assemblies and their interaction in the circuit,
- statistical assessment of the quality of fabrication and of all returned goods for the immediate taking of appropriate corrective actions.

### 1.3.5 Hazard and Other Warnings

Despite the actions described in section 1.3.3, the occurrence of faults or errors in electronic control units - even if most highly improbable - must be taken into consideration.

Please pay particular attention to the additional notices which we have marked by symbols throughout this instruction manual. While some of these notices make you aware of possible dangers, others are intended as a means of orientation. They are described further down below in descending order of importance.

Every alert and hazard warning is made up as follows:

***Type and source of risk***

*Potential consequences of non-observance*

- ⇨ Preventive measures

| ⚠ | ***DANGER*** |
|---|---|
| | ***A DANGER warning makes you aware of an immediately hazardous situation which WILL cause a serious or fatal accident if not observed.*** |

| ⚠ | ***WARNING*** |
|---|---|
| | ***A WARNING makes you aware of a potentially hazardous situation which MAY cause a serious or fatal accident or damage to this or other devices if not observed.*** |

| ⚠ | ***CAUTION*** |
|---|---|
| | ***A CAUTION alert makes you aware of a potentially hazardous situation which MAY cause an accident or damage to this or other devices if not observed.*** |

| ❗ | ***NOTE*** |
|---|---|
| | ***A NOTE makes you aware of a potentially hazardous situation which MAY cause damage to this or other devices if not observed.*** |

### 1.3.6 Other Notices

| ℹ | ***Information*** |
|---|---|
| | ***This symbol draws your attention to additional information concerning the use of the described product. This may include cross references to information found elsewhere (e.g. in other manuals).*** |

## 1.3.7 Safety

Our products normally become part of larger systems or installations. The information below is intended to help you integrate the product into its environment without dangers to humans or material/equipment.

⚠ ***DANGER***

***Non-observance of the instruction manual***

*Measures for the prevention of dangerous faults or errors may be rendered ineffective or new hazard sources created.*

⇨ *Thoroughly read the instruction manual*

⇨ *Take particular heed of the hazard warnings*

ℹ ***Information***

*To achieve a high degree of conceptual safety in planning and installing an electronic controller, it is essential to exactly follow the instructions given in the manual because wrong handling could lead to rendering measures against dangers ineffective or to creating additional dangers.*

### Project Planning

- 24 VDC power supply: generate as electrically safely separated low voltage. Suitable devices include split-winding transformers built in compliance with European Standard EN 60742 (corresponds to VDE 0551).
- Power breakdowns or power fades: the program structure is to ensure that a defined state at restart excludes all dangerous states.
- Emergency-off installations must comply with EN 60204/IEC 204 (VDE 0113). They must be operative at any time.
- Safety and precautions regulations for qualified applications have to be complied with.
- Please pay particular attention to the notices of warning which, at relevant places, will make you aware of possible sources of dangerous mistakes or faults.
- Relevant standards and VDE regulations are to be complied with in every case.
- Control elements are to be installed in such a way as to exclude unintended operation.
- Lay control cables such that interference (inductive or capacitive) is excluded if this interference could influence controller operation or its functionality.

### Maintenance and Servicing

- Precautions regulation VBG 4.0 to be observed when measuring or checking a controller after power-up. This applies to section 8 (Admissible deviations when working on parts) in particular.
- Repairs must be carried out by specially trained Kuhnke staff only (usually in the main factory in Malente). Warranty expires in every other case.
- Only use parts approved of by Kuhnke. Only genuine Kuhnke modules must be used in modular controllers.
- Modular systems: always plug or unplug modules in a power-down state. You may otherwise damage the modules or (possibly not immediately recognisably!) inhibit their functionality.
- Always dispose of (rechargeable) batteries as hazardous waste.

## 1.3.8 IT Security

Kendrion Kuhnke products are designed for operation within closed industrial networks.

If the industrial networks are publicly accessible, e.g. through freely accessible network interfaces, or publicly accessible, e.g. through data connections via public data traffic (Internet), then suitable organisational and technical security measures must be taken by the integrator and operator to protect the internal network and ensure IT security.

***Information***

***Information for the secure operation of plants, systems, machines and networks can be found, among others, in the BSI information publications and IEC 62443.***

## 1.3.9 Electromagnetic Compatibility

### Definition

Electromagnetic compatibility is the ability of a device to function satisfactorily in its electromagnetic environment without itself causing any electromagnetic interference that would be intolerable to other devices in this environment .

Of all known phenomena of electromagnetic noise, only a certain range occurs at the location of a given device. These kinds of noise are specified in the applicable product standards.

The design and immunity to interference of programmable logic controllers are internationally governed by standard
IEC 61131-2 which, in Europe, has been the basis for European Standard
EN 61131-2.

***Information***

*Refer to IEC 61131-4, User's Guideline, for general installation instructions to be complied with to ensure that hardware interface factors and the ensuing noise voltages are limited to tolerable levels.*

### Interference emission

Interfering emission of electromagnetic fields, HF
compliant to EN 55011, limiting value class A, Group 1

***Information***

*If the controller is designed for use in residential areas, high-frequency emissions must comply with limiting value class B as described in EN 55011.*

*Fitting the controller into earthed metal cabinets and installing filters in the supply lines may produce a shielding compliant to the above standard.*

### General notes on installation

As component parts of machines, facilities and systems, electronic control systems must comply with valid rules and regulations, depending on their field of application.

General requirements concerning the electrical equipment of machines and aiming at the safety of these machines are contained in Part 1 of European Standard EN 60204 (same as VDE 0113).

## Electrical immission safeguard

To eliminate electromagnetic interference, connect the control system to the protective earth or functional earth conductor. Practice best cable routing.

## Cable routing and wiring

Keep power circuits separate from control circuits:

| | |
|---|---|
| ▪ DC voltages | 60 V ... 400 V |
| ▪ AC voltages | 25 V ... 400 V |

Joint laying of control circuits is allowed for:

- shielded data signals
- shielded analogue signals
- unshielded digital I/O lines
- unshielded DC voltages < 60 V
- unshielded AC voltages < 25 V

## Location of installation

Ensure that temperatures, contaminations, impact, vibration or electromagnetic interference are no impediment to the installation.

## Temperature

Consider heat sources such as general heating of rooms, sunlight, heat accumulation in assembly rooms or control cabinets.

## Contamination

Use suitable casings to avoid possible negative influences due to humidity, corrosive gas, liquid or conducting dust.

## Impact and vibration

Consider possible influences caused by motors, compressors, transfer lines, presses, ramming machines and vehicles.

## Electromagnetic interference

Consider electromagnetic interference from various local sources: motors, switching devices, switching thyristors, radio-controlled devices, welding equipment, arcing, switched-mode power supplies, converters / inverters.

## Particular sources of interference

## Inductive actuators

Switching off inductances (such as from relays, contactors, solenoids or switching magnets) produces surge voltages. It is necessary to reduce these extra voltages to a minimum.

Throttling elements could be diodes, Z-diodes, varistors or RC elements. Their rating should conform to the specifications provided by the manufacturer or supplier of the actuators.

# 2 System Description

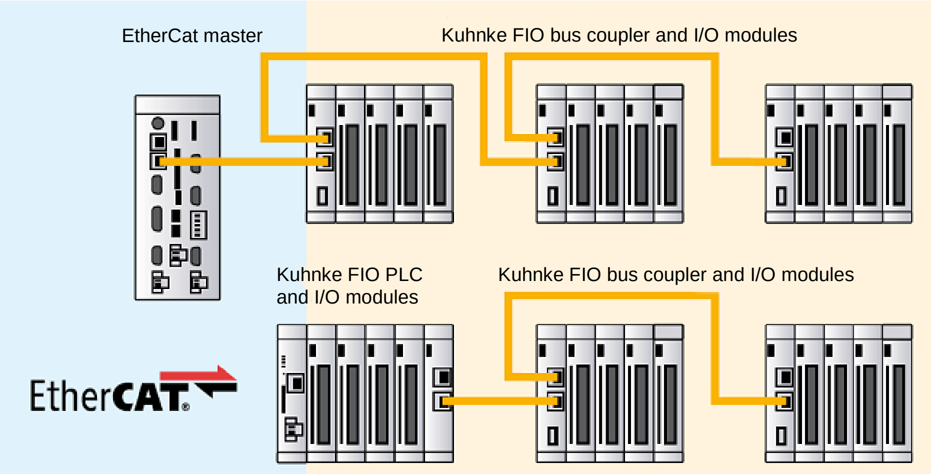
## 2.1 Kuhnke FIO

Kuhnke's FIO is a system of I/O modules for interconnecting the process signals in an EtherCAT network.

Kuhnke FIO consists of the Kuhnke FIO controller, Kuhnke FIO bus coupler and various Kuhnke FIO I/O modules.

Kunke FIO Controller 116 is a PLC equipped with a CODESYS V3 runtime system. It also supplies the system voltage to the FIO modules directly connected to it.

The Kuhnke FIO bus coupler converts the physical transfer technology (twisted pair) to LVDS (E-bus) and generates the system voltages required by the LVDS modules. The standard 100 Base Tx lines used for office network communications connect to the one side, the Kuhnke FIO I/O modules for the process signals connect to the other. This is how the Ethernet EtherCAT protocol is retained right through to the last I/O module.


EtherCat master
Kuhnke FIO bus coupler and I/O modules
Kuhnke FIO PLC and I/O modules
Kuhnke FIO bus coupler and I/O modules
EtherCAT

## 2.2 CODESYS V3

CODESYS is a software platform designed to handle many tasks of industrial automation technology. It is based on the IEC 61131-3 programming system. The tool benefits users with integrated solutions tailored to their practical work and aimed at providing hands-on support with whatever job needs to be done.

CODESYS supports all five languages specified in IEC 61131-3 (International Electrotechnical Commission):

- IL (Instruction List)
- ST (Structured Text), based on PASCAL for structured programming
- LD (Ladder Diagram)
- FBD (Function Block Diagram)
- SFC (Sequential Function Chart)

Apart from the standard IEC languages, CODESYS also supports:

- CFC (Continuous Function Chart) is a FD (function diagram) editor with a fully configurable graphical layout: whereas FD editors are network-based and automatically arrange the function blocks, CFC lets users place the blocks anywhere such that feedback effects can be created without any temporary variables. This feature makes the language the perfect choice for creating an overview of the application.

## Fieldbus technology

The CODESYS programming system allows the direct configuration of the CANopen and EtherCAT fieldbuses. Protocol stacks can be separately installed as CODESYS libraries for some systems.

## CODESYS Control

CODESYS Control is a soft PLC runtime system installed in Kuhnke Controller 116 and adapted to its hardware. It turns Kuhnke Controller 116 into an industrial controller in conformity with IEC 61131-3. This runtime system also features some extra functions that let the controller communicate with other components in its automation environment.

## CODESYS visualisation

An editor integrated in the CODESYS programming system allows users to create complex visualisation screens and to animate them by means of the application variables. The appropriate visualisation elements are provided by the software. Once created, the screens may help to test an application or to start it up in the programming system's online mode, for example. Running the optional visualisation clients **CODESYS HMI** and **CODESYS WebVisu** makes the screens a means of operating the machine or system.

## Software Releases

Various software options or combined software options are available for installation in the devices. Please ask product management for the combination that best fits your needs.

| Software Options | | |
|---|---|---|
| Option | **Identification** | Function |
| CODESYS Control | **V3** | This basic software package processes the programmed IEC 61131-3 code and debugs the code when working with CODESYS. This software is a main feature of all Scout-series devices |
| CODESYS TargetVisu | **TV** | CODESYS control unit extension for outputting visualisation screens to the control unit display. Generates the TargetVisualization immediately in the CODESYS Development System |
| CODESYS WebVisu | **WV** | A control unit running CODESYS WebVisu lets you display your own screens created in CODESYS in any web browser environment anywhere in the world. |
| CODESYS SoftMotion | **SM** | Use your standard IEC 61131-3 development interface to plan single or multi-axis movements up to and including the plotting of curves - together with the logic application.<br>A Motion Controller running CODESYS SoftMotion provides the PLC programming system with a kit of motion functions. Motion Controllers running CODESYS SoftMotion support many user project planning variants for the given motion tasks by providing PLCopen-certified motion modules, the entire functionality of the IEC 61131-3 programming interface and other tools. |
| CODESYS SoftMotion CNC+Robotics | **SM+CNC** | 3D-CNC motion control for Motion Controllers with a complete set of 3D-CNC and/or robotics functions including interpolator and kinematic transformations.<br>CODESYS SoftMotion CNC+Robotics provides you with an editor as a convenient way of configuring complex groups of robot axles. Run the editor, choose the appropriate kinematics, set its parameters and link them to the physical robot axles. Standardised function blocks process the robot functions in conformity with PLCopen MotionControl Part 4. |

For further details please visit the product pages of 3S-Smart Software Solutions GmbH at https://de.codesys.com

# 3 Product Description

## 3.1 General Description of FIO Controller 116

Kuhnke FIO Controller 116 is an ARM-based mini-IPC plus CODESYS PLC which is programmed by means of CODESYS V3.

Its SD card slot plus USB port provide a local storage volume for process and other data and the interchange of control programs. A digital interrupt input allows the controller to immediately respond to process events.

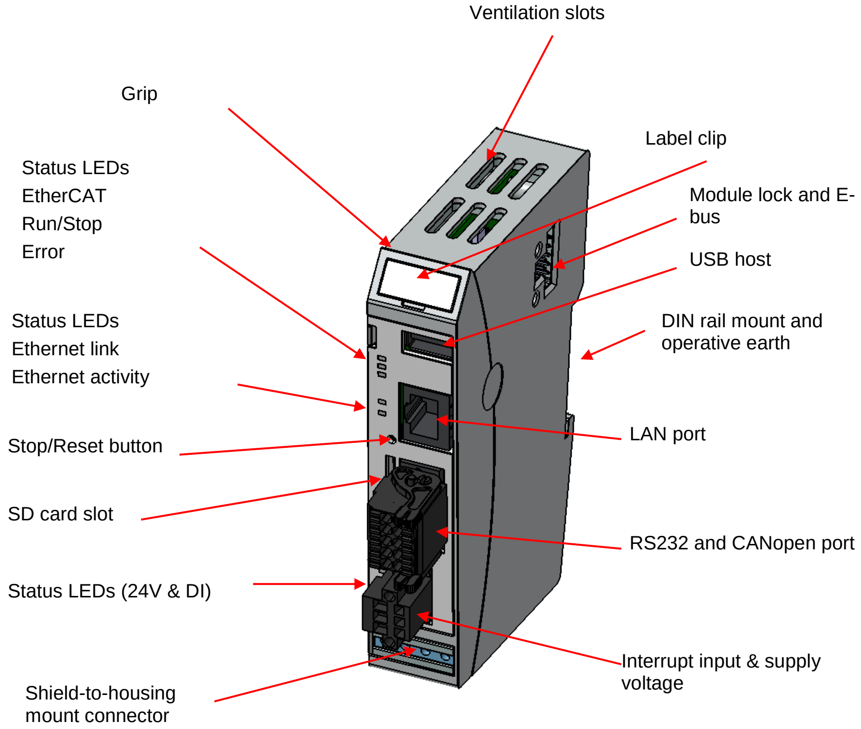
On one side of the modular control unit, there is an E-bus connector which provides for a flexible extension by Kuhnke FIO-series EtherCAT I/O modules. Or link in an extender module if you wish to actuate external EtherCAT slaves.

One way of exchanging data with other systems is to go through the on-board interfaces, i.e. Industrial Ethernet and RS232. Another way is to plug in interface and bus modules to provide further communication links and to facilitate system integration in existing control architectures.

### Properties

- Fanless ARM processor technology
- Expandable by Kuhnke FIO modules
- CODESYS V3 with EtherCAT®
- Optional CODESYS WebVisu

## 3.2 Front View



# 3.3 Application

## 3.3.1 Intended Use

Kuhnke FIO Controller 116 is a mini-IPC and a CODESYS PLC providing EtherCAT master functions for KUHNKE FIO system I/O modules which allow process signals to be directly attached to the control unit.

## 3.3.1 Foreseeable Misuse

### Place of installation

The unit is solely permitted for use in fully enclosed control cubicles or rooms.

Exhaust heat of the unit dissipates through the (top and bottom) ventilation slots and the aluminium U-profile. Verify that the place of installation is ventilated properly.

***NOTE***

***Damage to the unit***

*Choosing the wrong place of installation may cause damage to the unit.*

⇨ Check section 3.4 Technical Data Data for the admissible ambient conditions and the unit's mounting position.

# 3.4 Technical Data

## 3.4.1 Kuhnke FIO Controller 116 - General Specifications

| General Specifications | |
|---|---|
| **Product name** | FIO Controller 116 |
| **Article number** | 694 300 16 |
| **Processor** | i.MX6 SoloX Freescale 1 GHz |
| **RAM / remanent memory** | 256 MB / flash storage |
| **Drives** | 1x SD card slot (user) |
| **Software** | Operating system: RT Linux; Application: CODESYS Control V3, ... |
| **Interfaces** | With potential separation: 1x Ethernet 100 Mbit – RJ45, 1 x CAN,<br>Without potential separation: 1 x RS232, 1 x USB 2.0 Host, 1 x EtherCAT E-Bus |
| **Fieldbus** | CANopen® Master, EtherCAT® Master via E-bus system plug<br>Modbus RTU Master, Modbus TCP Master |
| **Integrated I/Os** | 1x DI, interrupt-enabled (Cable length < 30m) |
| **Clock** | Battery-buffered real-time clock |
| **Power supply** | 24 VDC (19.2 .. 28,8) |
| **E-bus power supply** | 1,5A @ 55°C; 2,0A @ 50 °C |
| **Output** | Approx. 3.5 W (@ 24 VDC) |
| **Noise immunity** | Zone B to EN 61131-2,<br>mounted on earthed DIN rail in earthed control cubicle |
| **Operating conditions** | |
| **Ingress protection** | IP20 |
| **Mounting position** | Vertical, stackable |
| **Storage temperature** | -25 °C...+70 °C |
| **Operating temperature** | 0 °C...+55 °C |
| **Rel. humidity** | 5% … 95%, non-condensing |
| **Mechanical properties** | |
| **Installation** | 35 mm DIN rail (top-hat rail) |
| **Dimensions** | 25mm x 120mm x 90mm (W x H x D) |
| **Housing mount** | Aluminium |
| **Shield** | Connected straight to module housing |

### 3.4.2 Specifications PROFINET IO Device Extension

| Additional Specifications | |
|---|---|
| **Product name** | FIO Controller 116 PROFINET IRT |
| **Article number** | 694 xx2 16 00x |
| **Interfaces** | 2x Ethernet (Switch) PROFINET® |
| **Fieldbus** | PROFINET IO Device |
| **PROFINET Chip** | Renesas TPS-1 |
| **Process data** | 256 Byte In/ 256 Byte Out |
| **Dimensions** | 42 mm x 120 mm x 90 mm (W x H x D) |

### 3.4.3 Specifications EtherCAT Slave Extension

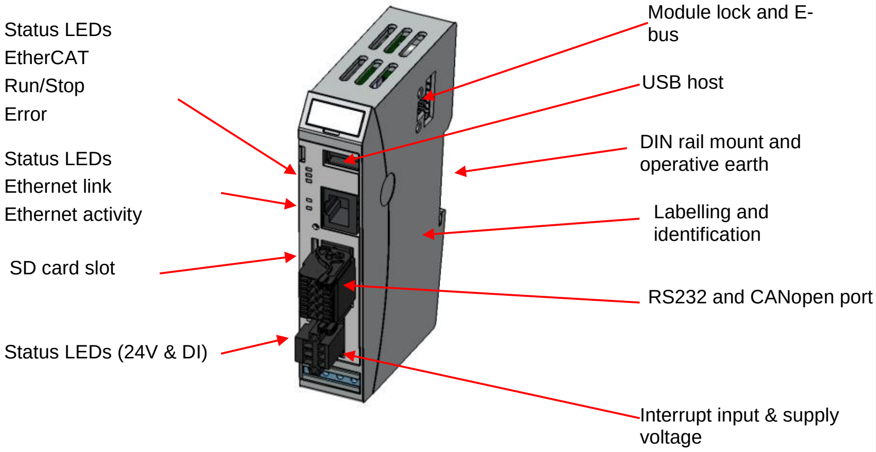
| Zusätliche Gerätedaten | |
|---|---|
| **Product name** | FIO Controller 116 EC |
| **Article number** | 694 xx3 16 00x |
| **Interfaces** | 2x EtherCAT (In / Out) |
| **Fieldbus** | EtherCAT Slave Device |
| **EtherCAT Slave Controller** | Microchip LAN9252 / Microchip LAN9253 |
| **Process data** | 256 Byte In/ 256 Byte Out |
| **Dimensions** | 42 mm x 120 mm x 90 mm (B x H x T) |

### 3.4.4 Specifications Ethernet Extension

| Zusätliche Gerätedaten | |
|---|---|
| **Product name** | FIO Controller 116 LAN |
| **Article number** | 694 xx1 16 00x |
| **Interfaces** | 2x Ethernet (Switch) 10Mbit, 100Mbit |
| **Dimensions** | 42 mm x 120 mm x 90 mm (B x H x T) |

# 4 Construction and Functionality

## 4.1 Brief Description



## 4.2 Labelling and Identification

Laser marking on the front and the right side wall
Label with serial number on aluminium frame
Label with software licence on aluminium frame

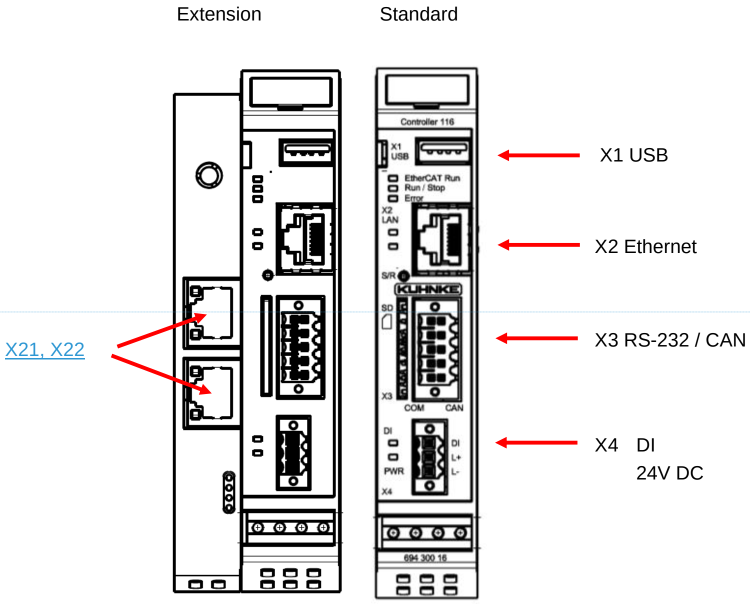
## 4.3 Contents of Package

The Kuhnke FIO Controller 116 package includes:

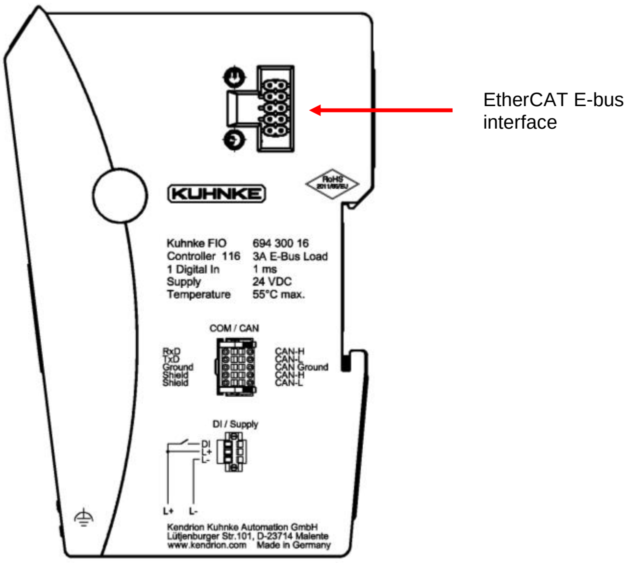
- Kuhnke FIO Controller 116
- Set of connectors (RS232/CAN and supply voltage / DI)

## 4.4 Connectors

While all external connectors plug in at the front of the unit, the modules of the FIO EtherCAT IO system connect to the EtherCAT E-bus interface on the side



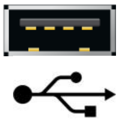
Gelöscht: <Objekt><Objekt>¶



## 4.4.1 X1 USB 2.0 (host)

The USB host port is made for attaching USB peripherals.

Pin wiring:



| **USB** | | |
|---|---|---|
| Connector | **Pin** | Function |
| <br>USB | B1 | VCC |
| | B2 | D- |
| | B3 | D+ |
| | B4 | GND |

***NOTE***

***USB port overload by excessive power consumption***

*Machine failure and/or data loss*

⇨ Only attach USB peripherals whose total power needs do not exceed 0.5 A or supply external power to the USB peripherals by adding an active hub to the line, for example.

***NOTE***

***Electromagnetic interference in industrial environments***

*Machine failure and/or data loss*

⇨ Mind that the USB interface is not designed for use in industrial environments but just as a service interface for loading software updates or data up into or down from the control unit.

### 4.4.1.1 USB Sticks

USB stick users should consider the following points:

- The USB port is mechanically designed to support up to 1,000 plug/unplug operations..
- Pulling off a USB stick during operation is allowed only when no more file operations are being performed. Failure to comply may render the USB stick useless!
  If a program still has a file open when the USB stick is pulled, you will find it impossible to close the directory that the file belongs to. In this situation, the system will not respond to any file or directory operation requests because it would be expected to read from a device that is no longer available. Therefore, before pulling off the USB stick, you should always ensure that no program still has an open file loaded from the stick.

  ➔ Before actually removing the USB stick, run function USB_UMountDisk from Kuhnke system library iMX6.
- You will find the USB sticks at:
  \media\usbx (x = 1..n number of USB sticks connected)
- Supported format(s): FAT32

### 4.4.2 X2 Ethernet "LAN"

The on-board 10/100 Mbit base-T Ethernet adapter attaches the unit to a network through its RJ-45 connector. The LAN configuration of the operating system lists this connector as eth0. The LEDs labelled "LNK" and "Activity" tell you whether the unit is properly connected to the network.

Pin wiring:

| LAN | | |
|---|---|---|
| Connector | **Pin** | Function |
| RJ45 | **1** | TX+ |
| | **2** | TX- |
| | **3** | RX+ |
| | **4** | 75 Ohm |
| | **5** | 75 Ohm |
| | **6** | RX- |
| | **7** | 75 Ohm |
| | **8** | 75 Ohm |

***Information***

*The E-bus connector on the side is provided for use as EtherCAT fieldbus interface. The LAN configuration of the operating system lists this connector as eth1.*

***NOTE***

***Unauthorised access to the computer***

*Controller failure and data loss*

⇨ Integration in networks granting public access requires the user to take appropriate measures aimed at preventing unauthorised access.

### 4.4.3 X3 Serial Communication RS-232 / 'CAN

The system uses COM1 for access to the **RS-232** port. The **RS-232** port is not electrically insulated.

Pin Wiring

***RS 232***

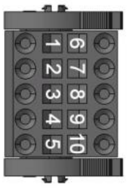
| RS 232 interface | |
|---|---|
| **Pin** | **Function** |
| **1** | RxD |
| **2** | TxD |
| **3** | GND |
| **4** | Shield |
| **5** | Shield |

The **CAN interface** complies with standard ISO 11898 and supports baud rates of up to 1 Mbps. This port is electrically insulated. A bus termination resistor of 120 Ω can be connected directly to the connector between CAN_H and CAN_L.

Pin Wiring

| CAN interface | |
|---|---|
| **Pin** | Function |
| **6** | CAN_H |
| **7** | CAN_L |
| **8** | CAN_GND |
| **9** | CAN_H |
| **10** | CAN_L |

Technical Data



| RS 232 | |
|---|---|
| Quantity | 1 |
| Signal | RS 232 |
| Baud rate | max. 115.200 Baud,<br>(57.600, 38400, 19.200, 9600 …) |
| Electrical insulation | no galvanic isolation |

| CAN 1 | |
|---|---|
| Signal | RS 485 |
| Baud rate | max. 1MBit/s,<br>(500, 250, 125 kBaud) |
| Electrical insulation | galvanic isolation |
| Termination | attached to the connector |

Connection Data

| Mating connector | |
|---|---|
| **Type** | female PUSH-IN spring connector<br>released by push-button |
| **Poles** | 10, two rows |
| **Order data** | Weidmüller: 1460190000 |
| **Wire cross section** | min. 0.14 mm²<br>max. 1.5 mm² |

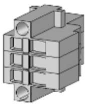
## 4.4.4 X4 Digital Input and Power Supply

### On-board power supply unit

The FIO Controller features an on-board power supply unit (PSU) designed for an input voltage of 24 VDC (18 V ... 32 V). The PSU is protected against reverse polarity. It supplies power to the FIO Controller's CPU core and to the FIO modules connected to the E-bus.

Verify that both the cord and the PSA are externally protected against short circuit and overload triggering at max. 10 A.

### Pin wiring:



| 24 VDC power supply | | |
|---|---|---|
| Connector | **Pin** | Function |
| Weidmüller | 1 | Digital input |
| | 2 | external power supply<br>24 VDC (18V ... 32V) |
| | 3 | external power supply, GND |

## 4.4.5 Extension moduls

| **Stecker** | **Profinet** | **EtherCAT** | **Ethernet** |
|---|---|---|---|
| **X21** | Port-001 | OUT | Port 1 |
| **X22** | Port-002 | IN | Port 2 |

**Gelöscht:** IN

**Gelöscht:** OUT

## 4.4.6 Operative Earth

Connect operative earth to the protective earth conductor of the switching cabinet or the system that Kuhnke FIO Controller 116 is installed in. Connection is made via the earthed DIN rail. In special cases you may attach the earth wire straight to the module.

Verify that the cross section is not less than 2.5 mm$^2$. Try to keep the lead to the cabinet terminal as short as possible.

***Information***

*A low-impedance earth conductor improves the dissipation of interference received via external power supply cables, signal cables or cables of peripheral units.*

### 4.4.7 SD Card

Kuhnke FIO Controller 116 features a SD card slot at its front. The slot is marked with a SD icon.

The SD card slot has a push-in/push-out plug & eject mechanism. You may use a flat object to help you plug in or eject the card.

You will find the SD card at: \media\sd

Features of the SD-card:

- Supported format(s): FAT32
- Maximum memory size: 32GB
- Maximum file size: 4GB – 1Byte
- Quality: Industrial Grade
- Technology: SLC

***NOTE***

***Electrostatic discharge (ESD)***

*Inappropriate handling will destroy the memory card*

⇨ SD cards are susceptible to electrostatic discharge (ESD). Please take account of the instructions on how to handle memory cards.

***Information***

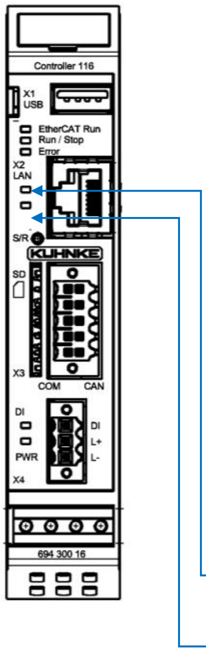
*The SD card slot has no hot-plug capability. They will therefore detect cards only if they were plugged in at the time of booting the unit. SD cards have no hot-swap capability.*

# 4.5 Indicators and Controls

## 4.5.1 Status LEDs

The status LEDs indicate the state of the FIO Controller's modules. RUN / STOP is indicative of the soft PLC status

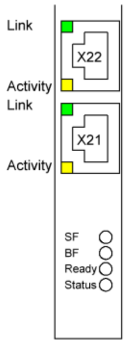
I



**LED Description**

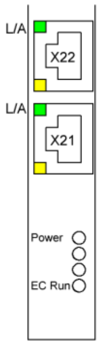
| Designation | LED | Explanation |
|---|---|---|
| EtherCAT Run | off | Initialising, no data exchange |
| | off/green, 1:1 | Pre-operational, no data exchange |
| | off/green, 5:1 | Safe operation, inputs readable |
| | green, on | Operational, unrestricted data exchange |
| RUN/STOP | off | Boot loader active or operating system is booting |
| | yellow | Low voltage |
| | yellow 1:1 0.5 Hz | Processing USB update |
| | yellow 1:1 2.5 Hz | Updating firmware |
| | yellow 1x flash | Service mode enabled |
| | yellow 2x flash | Reboot required, e.g. after an update |
| | green | CODESYS application is running |
| | red | CODESYS application stopped |
| Error | off | No error |
| | red | Error |
| LAN link / activity | off | Ethernet cable not connected |
| | green | Ethernet cable connected |
| | flashing | Ethernet activity |
| LAN speed | off | 10 Mbit |
| | yellow | 100 Mbit, 1000 Mbit |
| DI | off | No input signal |
| | green | Input signal received |
| PWR | off | Supply voltage not available |
| | blue | Supply voltage available |

## 4.5.2 Status LEDs PROFINET IO Device extension



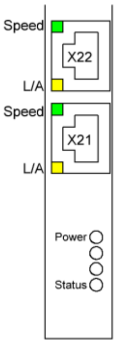
| Designation | LED | Explanation |
|---|---|---|
| Link | off | No physical connection |
| | green | Physical connection |
| Activity | off | No data exchange |
| | yellow | Data exchange |
| SF | off | No PROFINET Diagnosis existing |
| | red | PROFINET Diagnosis existing |
| BF | off | Active communication between PROFINET IO-Controller and this PROFINET IO-Device |
| | red | No connection status available |
| | ged flashing | Connection status OK: no communication with a PROFINET IO-Controller |
| Ready | off | TPS-1 is not started correctly |
| | green flashing | TPS-1 waiting for synchronization of Host CPU (Firmware-Start finished) |
| | green | TPS-1 started correctly |
| Status | off | n/a |
| | green | n/a |

## 4.5.3 Status LEDs EtherCAT extension



| Bezeichnung | LED | Bedeutung |
|---|---|---|
| Link/Activity | off | No physical connection |
| | green | Physical connection |
| | flashing | Data exchange |
| Power | off | Supply voltage missing |
| | green | Supply voltage OK |
| | | |
| | | |
| EC Run | off | Initialising, no data exchange |
| | off/green, 1:1 | Pre-operational, no data exchange |
| | off/green, 5:1 | Safe operation, inputs readable |
| | green | Operational, unrestricted data exchange |

## 4.5.4 Status LEDs Ethernet extension



| Bezeichnung | LED | Bedeutung |
|---|---|---|
| Speed | off | 10Mbit |
| | green | 100Mbit |
| Link/Activity | off | No physical connection |
| | yellow | Physical connection |
| | flashing | Data exchange |
| Power | off | Supply voltage missing |
| | green | Supply voltage OK |
| | | |
| | | |
| Status | off | Extension module not ready |
| | green | Extension module ready |

### 4.5.5 Stop/Reset Button

The Reset/Stop button is located underneath the status indicators at the front of the unit.

To avoid it being pushed unintentionally, the Stop/Reset button can be operated using a pointed object (pen, screwdriver) only.

Its function depends on the current state of FIO Controller 116.

#### CODESYS stop – start – reset

Briefly press the button to stop the running CODESYS application. Run/Stop changes from green to red.

Briefly press the button again to restart the CODESYS application. Run/Stop changes from red to green.

The type of reset function (warm, cold) when the button is pressed for more than 5 seconds can be parameterized. If the button is held down for longer than 5 seconds, the parameterized reset function is triggered (default: warm). After the time has elapsed, the Run / Stop LED flashes quickly and the button can be released. The controller is now in reset mode.

***NOTE***

***Loss of retain data***

*Important machine data can be deleted if the "Reset cold" function has been configured.*

⇨ Check the parameterization of the reset function for the stop/reset button.

#### Service mode

Service mode is enabled by turning off FIO Controller 116. Now press and hold the button to restart FIO Controller 116. Keep pressing the button until Run/Stop lights up yellow every 2 seconds.

# 5 Operation

## 5.1 Installation

### 5.1.1 Mechanical Installation

- ⇨ Kuhnke FIO I/O modules are intended for mounting rail installation (DIN EN 50022, 35 mm x 7.5 mm).

### To snap on a single module

- ⇨ Push up the module against the mounting rail from below, allowing the metal spring to snap in between mounting rail and mounting area as illustrated.
- ⇨ Push the top of the module against the mounting wall until it snaps in.

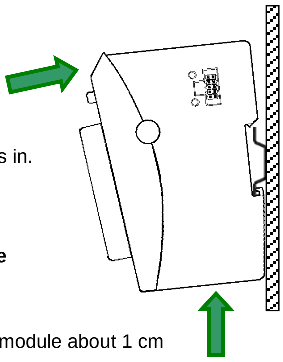


**Figure1: Rail mounting of module**

### To interconnect two modules

- ⇨ After snapping on the first module to the rail, snap on the second module about 1 cm away towards the right of the first module.
- ⇨ Push the second module along the rail towards the first module until you hear the locking device snap in.

### To disconnect two modules

- ⇨ Push down the unlock button (see Figure 2) of the module that you wish to disconnect from the module to the left of it.
- ⇨ With the button still pressed, push both modules away from one another until they are about 1 cm apart.

### To take down a single module

- ⇨ Push the module up and against the metal spring located on the underside of the rail guide.
- ⇨ Tip the module away from the rail as shown in the illustration.
- ⇨ Pull the module down and out of the mounting rail.

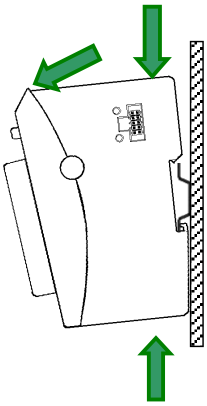
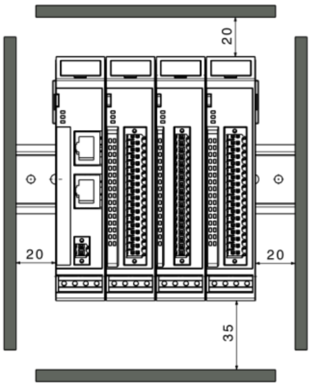


**Figure 2: Uninstalling a module**

## Position

Mount with rail horizontally with the modules' multiple socket connectors pointing away from the wall. To ensure that enough air gets in through the ventilation slots, leave at least 20 mm to the top and 35 mm to adjacent devices or cabinet surfaces. Leave at least 20 mm of lateral distance to third-party units and cabinet surfaces.



## Order of Modules in Multi-FIO Systems

| ❗ | ***NOTE*** |
|---|---|
| | *In order to ensure that the entire FIO system works properly, arrange the FIO modules by their specific E-bus load, placing the modules with the highest E-bus load immediately next to the head module (bus coupler or controller). Take account of the head module's maximum bus load.*<br>*If possible, place the Kuhnke FIO Safety I/O modules immediately next to the head module.* |

## 5.1.2 Electrical Installation

⚠ ***WARNING***

***Potentially hazardous failures due to wrong voltages supplied***

*Supplying the wrong voltages may damage or destroy the unit and may provoke potentially hazardous failures.*

*Preventive measures:*

- ⇨ We recommend to use PELV/SELV-ready power supply units to EN50178 or EN60950-1 to supply 24 VDC to bus couplers or compact PLCs.
- ⇨ Only use the GND terminal to connect the power supply unit to earth (PELV system). Do not use earthing variants that connect earth to +24V.
- ⇨ Remember that, even in case of a fault, a maximum voltage of U max. < 33 V maybe supplied to these assemblies. If you cannot rule out this risk, external protection of the power supply is mandatory.
- ⇨ To ensure that there is as little interference as possible, install a central power supply point and establish a star topology of as short wires as possible between the central point and the block of FIO modules.

### Earth

Connect the Kuhnke FIO modules to earth by attaching the metal housing to functional earth.

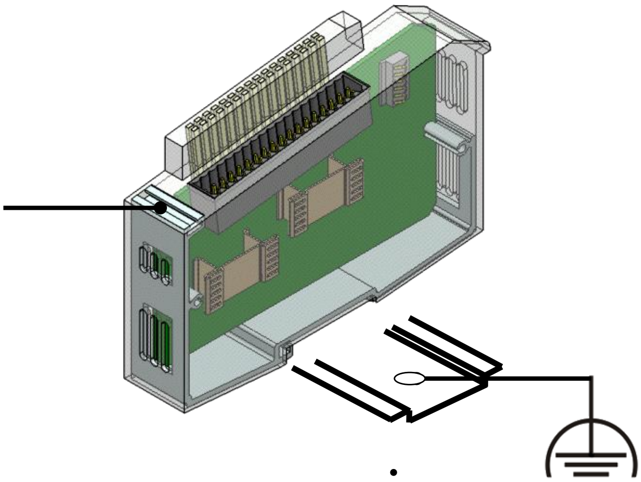
Since the functional earth connector dissipates HF currents, it is of utmost importance for the module's noise immunity.

HF interference is dissipated from the electronics board to the metal housing. The metal housing therefore needs to be suitably connected to a functional earth connector.

You will normally have to ensure that

- the connection between module housing and DIN rail conducts well,
- the connection between DIN rail and switching cabinet conducts well,
- the switching cabinet is safely connected to earth.

In special cases you may attach the earth wire straight to the module.



***Information***

*Earth wires should be short and have a large surface (copper mesh). Refer to http://de.wikipedia.org/wiki/ground_(electronics)* for further details

## Module interconnection

The FIO modules electrically connect by completely pushing the modules together. This automatically connects them to the EtherCAT bus system and supplies power to the EtherCAT communication modules. FIO Controller 116 is always the first module of a FIO I/O block.

Please note that the power supplied by FIO Controller 116 limits the number of FIO modules you may connect to a single block.

### Logic power supply (24 VDC)

Power to the logic circuitry is supplied through lines L+ and L- of the module plug.

***NOTE***

***Risk of electric voltage***

*Supply voltages outside of the admissible range may destroy the unit.*

⇨ Before turning on the supply voltage, verify that it is within the admissible voltage range.

### Digital Input

Ensure that the cable length at the digital input is shorter than 30m.

## 5.2 Configuration

A web interface is used to configure FIO Controller 116. After logging in, you can check and change various system settings and/or display system information.

### 5.2.1 IP Address

The FIO Controller is set to a permanent IP address and subnet mask in the factory.

IP address: 192.168.0.116

Subnet mask: 255.255.255.0

Before you connect to FIO Controller 116 for the first time, verify that your PC is set to the same IP address range as FIO Controller 116.

This many involve changing your PC's IP address.

Apart from the set IP address, FIO Controller 116  if in service mode - is assigned a unique IP address which allows you to access FIO Controller 116 in case you do not know the IP address settings.

IP address: 192.168.0.116

Subnet mask: 255.255.255.0

***NOTE***

***Duplicate IP addresses***

*Assigning the same IP address to two different devices may cause serious network problems.*

⇨ Check your network for duplicate IP addresses.

⇨ In order to put FIO Controller 116 into operation for the first time, we recommend using a direct network connection between and set IP addresses for FIO Controller 116 and the programming PC. You may have to use a cross-over network cable.

***Note***

***Unknown IP address***

⇨ If the connection with the FIO Controller 116 fails as the IP address is not known, please start the FIO Controller in service mode (chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**)

## 5.3 Web Interface

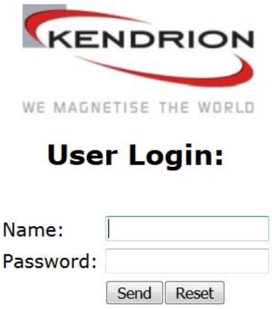
### 5.3.1 Login

First of all run a browser application on your PC to launch the web interface of FIO Controller 116. Internet Explorer, Chrome and Firefox are the current choice of browsers.

Type the following IP address of FIO Controller 116 into your browser's address bar:

http://<IP address>

The following web site displays:



The control unit hosts several user accounts for accessing the web interface. The user passwords default to the user names.

The following users may access the web interface:

**Overview**

| Name | Password | Rights |
|---|---|---|
| admin | admin | Unrestricted access |
| webuser | webuser | Restricted access |

**NOTE**

***Change of passwords***

*Change the passwords of the user accounts before setting the control unit to productive or network use. Otherwise, the control unit can easily be accessed by using the default passwords.*

⇨ Write down the password and keep it in a suitable place.

Provided that logging in is successful, another page showing the menu items below displays.

## 5.3.2 Menu – Configuration

### 5.3.2.1 "Network" menu item

Use this page to change various of the control unit's network settings. The new network settings will be enabled when you restart the unit.

**Host Name**

The host name identifies the control unit in a unique manner. In CODESYS V3, the host name is called **device name** and **computer name** in Windows. Host names are made up of one or several labels separated by a dot. A label consists of one or more characters.

A label may be comprised of up to 24 ASCII characters:

- a–z or A–Z (no distinction is made between upper and lower case)
- Numerals 0–9
- Hyphen/minus sign –

RFC952 disallows all other characters because they may cause problems. You may also use a Fully Qualified Domain Name (FQDN) such as *plc24.mycompany.de* as the host name.

Please note that the control unit does not run any DNS or WINS services. Accessing the control unit using the host name therefore requires an extra configuration (e.g. an entry in the network's DNS server).

**DNS Server**

Specify at least one valid DNS server if you with the control unit to gain access to the domain name system (DNS), that is to say, if you wish it to access other hosts by their name instead of their IP address. The second DNS server is a fallback name in case access to the first server fails. Retain the default for both entries (0.0.0.0) if you do not wish to use the DNS.

**Default Gateway**

Enter the IP address of the router or gateway you will need to connect the control unit to the Internet. You may retain the default (0.0.0.0) if the control unit will be used in the local area network only. The gateway settings only apply to the first network interface (eth0).

**Network Mode: Inactive**

This mode entirely disables the network interface.

**Network Mode: Static**

This mode lets you set a static IP address. Apart from the static IP address you will need the net mask (called NetMask or Subnetmask). This mode should also be enabled if you wish to use the interface for BACnet, Ethernet/IP or as a Modbus/TCP interface.

**Network Mode: DHCP**

This mode supports a DHCP server's IP address automatically assigned to the network interface when starting the control unit. Set fields IP Address and NetMask to 0.0.0.0 (default).

**Network Mode: ethercat**

This mode defines the network interface as an EtherCAT™ device. We recommend using eth1 for EtherCAT™. In CODESYS V3, you will then have to tell the EtherCAT™ master configuration that the interface selected for EtherCAT™ (e.g. "eth1") is a bus interface.

**Network Mode: profinet device**

This mode defines the network interface as a PROFINET™ device. In case you wish to use a PROFINET™ device together with an Ethernet, we recommend setting up eth0 as a PROFINET™ device and setting a second static IP address for interface eth0:1.

**ETH0**

ETH1 is the device name of the first network interface of the operating system. This interface supports various modes (inactive, static, dhcp, ethercat, profinet device).

The factory setting is a static IP address for the first network interface (mode: static).

**ETH0:1**

Virtual extension of network interfaces eth0 which allows you to set a second static IP address for the network interface concerned, e.g. for providing separate service access. Only supports mode: static and IP addresses from another IP address range than that of the basic interface. It will not activate if the basic interface mode is "ethercat".

**ETH1**

ETH1 is the device name of the second network interface of the operating system. This network interface is permanently set up as a EtherCAT™ device (mode: ethercat).

**ETH2**

These settings are only visible if the FIO Controller 116 is equipped with the Ethernet extension module. ETH2 is the device name of the third network interface in the operating system. This interface supports various modes (inactive, static, dhcp, ethercat, profinet device)

The factory setting is a static IP address for the first network interface (mode: static).

**ETH2:1**

Virtual extension of network interfaces eth2 which allows you to set a second static IP address for the network interface concerned, e.g. for providing separate service access. Only supports mode: static and IP addresses from another IP address range than that of the basic interface. It will not activate if the basic interface mode is "ethercat".

***Information***

***The ETH2 interface is not recommended for "hard" real-time applications.***

### 5.3.2.2 "CAN" Menu Item

The settings on this page allow you to omit CODESYS V3 and still operate the CAN interfaces at a specific baud rate. By default ("set by codesys"), the interface remains inactive until initialised by the CODESYS V3 application.

If you tick one of the available baud rates, the CODESYS V3 application will be ignored when booting the system and the CAN interface will be enabled at the set baud rate. The application can still access the interface.

### 5.3.2.3 "Time and Date" Menu Item

Use this page to set time of the control unit's real-time clock (RTC) and the time zone. The default time zone is set to the Coordinated Universal Time (UTC). What is making this a special setting is that UTC is equivalent to the control unit's RTC. Changing the UTC time also changes the RTC. Upon delivery, the control unit's RTC is set to the current German time.

You will need this information to change the time zone of the control unit in order to run functions such as changing between winter and daylight saving time. Remember to change the default German time to the actual UTC time before choosing a new time zone. The time and the changed time zone will not display correctly until you have properly set the RTC/UTC.

Please note that some of the CODESYS V3 system libraries read the time as RTC(UTC).

To show the time and the time zone together, the time reading must first be converted into the local time.

### 5.3.2.4 "VNC Server" Menu Item

Use this page to change the resolution and colour settings of the control unit's on-board VNC server.

### 5.3.2.5 "FTP server" Menu Item

Use this page to enable or disable the control unit's on-board FTP server which uses TCP port 21. The following users are privileged to log in to the FTP server:

| **Overview** | | | |
|---|---|---|---|
| **User Name** | **Root Directory** | **Change Root Directory** | **Rights** |
| **root** | /root | yes | read/write |
| **ftpuser** | /flash/ftpupload | no | read/write (ftpupload only) |
| **ftpadm** | /flash/ftpupload | yes | read/write (ftpupload only) |
| **ftpreader** | /flash/ftpupload | no | read |
| **ftp custom user** | adjustable | adjustable | adjustable (ftpreader/ftpuser/ftpadm) |

### 5.3.2.6 "Users" Menu Item

Use this page to change the passwords of the users registered with the control unit.

You may also create up to five custom "ftpusers", assign any user name and root directoy to them, enable or disable them and set their privileges based on the three default "ftpuser". Change the passwords of all or some users before running the control unit in a productive environment or make sure that nobody has physical access to the control unit or the network connected to it.

| **Overview** | | | |
|---|---|---|---|
| **User Name** | **FTP / Web** | **FTP Privileges** | **Web Privileges** |
| **root** | yes / yes | read/write | read/write |
| **admin** | no / yes | none | read/write |
| **ftpuser** | yes / no | read/write (ftpupload only) | none |
| **ftpadm** | yes / no | read/write (ftpupload only) | none |
| **ftpreader** | yes / no | read | none |
| **webuser** | no / yes | none | read |
| **ftpuser1-5** | yes / no | adjustable (ftpreader/ftpuser/ftpadm) | none |

### 5.3.2.7 "Reset Config" Menu Item

Use this page to restore the control unit's condition as delivered or its factory defaults. This includes the network, date/time, display and FTP server settings as well as all user passwords. Mind that all user data, CODESYS V3 applications and settings will be cleared in the process.

Only the licences installed in the control unit will be retained.

Restart the control unit after resetting the settings.

## 5.3.3 Menu – System

### 5.3.3.1 "Info" Menu Item

This page displays all major details of the control unit.

| Übersicht | | |
|---|---|---|
| **Option** | **Example** | **Explanation** |
| Part-Name | FIO Controller 116 | Name of control unit |
| Firmware-Version | X.XX.X.X (z.B. 1.21.1.0) | Release of the firmware currently installed in the control unit |
| Codesys RTS Version | 3.5.XX.XX (z.B. 3.5.13.30) | Release of the CODESYS V3 runtime system currently running in the control unit. The first two digits are the main CODESYS release, the third digit is the service pack, the fourth digit is the patch level and the fifth digit (if any) is the hotfix level. |
| Licenses | ETHERCAT-MASTER CANOPEN-MASTER MODBUS-TCP-MASTER MODBUS-RTU-MASTER | All licences installed in the control unit. Some libraries such as Modbus TCP requires extra licences which you may have to install separately. |
| System operation Time | 1612 hours 0 min | Total runtime since first putting the control unit into operation |
| System Uptime | 0 day 0 hour 19 min | Control unit runtime since last starting the operating system |

### 5.3.3.2 "Licenseinfo" Menu Item

This page displays all library licenses of the control unit.

| Übersicht | | | |
|---|---|---|---|
| **Library** | **Licensemodel** | **Link to licensetext** | **Remark** |
| Busybox-1.21.0 | Copyright | Julian R. Seward 2006 | archival/libarchive/bz |
| Bzip2-1.0.5 | Copyright | Julian R. Seward 1996 | |
| … | … | … | … |

### 5.3.3.3 "Update" Menu Item

Use this page to upload various files to the control unit in order to install firmware updates or further licences. Remember to stop all CODESYS V3 applications on the control unit before installing an update. First of all, pick the required file (e.g. firmware_mx6-plc_x.x.x.tgz) by clicking on "Durchsuchen…", then click on "Daten absenden" to upload the file. This may take up to several minutes, depending on the file size and the quality of the connection. After the upload, the web interface displays a description and the release of the uploaded file for you to review this information. Now click on "Start" to initiate the update process, which may take up to two minutes to complete, depending on the size of the .tgz file.

***NOTE***

***Damage to the unit***

*Once started, you must not interrupt the update process. Do not disconnect the unit from the power supply during that time. Prematurely interrupting the upload process will turn the unit into a case of repair.*

⇨ Make sure that the power supply cannot be interrupted by mistake.

Attention! You must restart the control unit after installing an update. The update process does not delete any applications or user data stored by the control unit. After the required restart, the control unit will try to change the status of the boot applications to "AS_RUN". The control unit will restart instantly after the update.

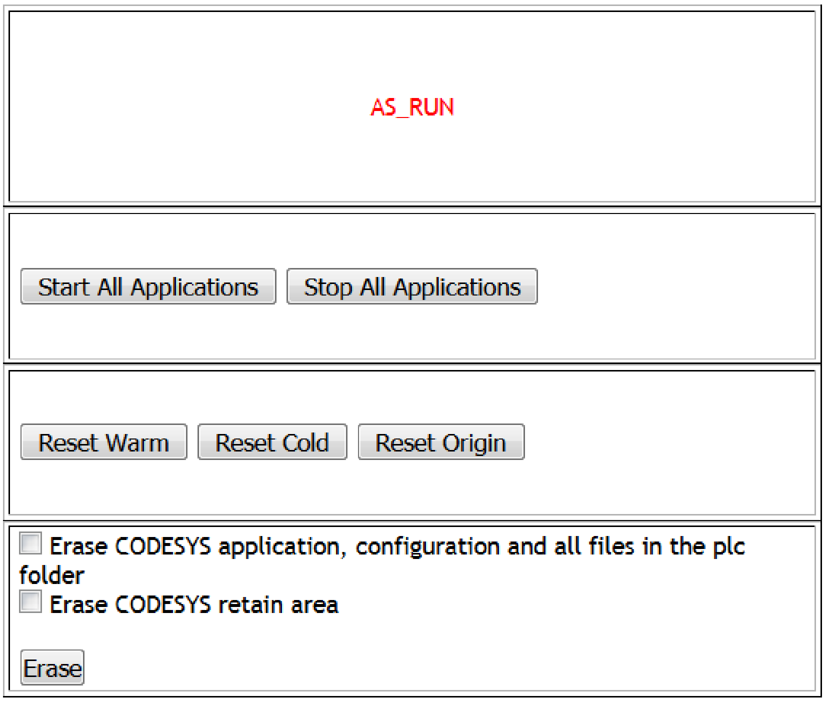
### 5.3.3.4 "Reboot" Menu Item

Use this page to restart the control unit. You have to restart the control unit when you change some of the control unit settings. This will interrupt all applications currently running in the control unit.

## 5.3.4 Menu – PLC Manager

### 5.3.4.1 Control

Use this page to control the CODESYS V3 applications hosted by the control unit.



AS_RUN
Start All Applications
Stop All Applications
Reset Warm
Reset Cold
Reset Origin
□ Erase CODESYS application, configuration and all files in the plc folder
□ Erase CODESYS retain area
Erase

Panel 1 displays the status of all applications hosted by the control unit.

| Status | Erklärung |
|---|---|
| **AS_PARTIALLY_STOPPED** | The status of at least one application is "AS_STOP" |
| **AS_RUN** | The status of all control unit applications is "AS_RUN" |
| **AS_STOP** | The status of all control unit applications is "AS_STOP" |
| **AS_NONE** | There are no control unit applications. |

Panel 2 allows you to start or stop all applications at once.

| Option | Explanation |
|---|---|
| **Start All Applications** | The software will try to change the status of all applications to "AS_RUN". |
| **Stop All Applications** | The software will try to change the status of all applications to "AS_STOP". |

Panel 3 lets you reset the control unit.

| Option | Explanation |
|---|---|
| **Reset Warm** | The status of all control unit applications changes to "AS_STOP" and all normal variables are reset. |
| **Reset Cold** | Same as "Reset Warm" except that all RETAIN variables will reset also. |
| **Reset Origin** | Erases all control unit applications and resets all control unit variables, including the RETAIN and RETAIN PERSISTENT variables of all applications. Afterwards, you will not be able to log in unless you download the applications to the control unit first. |

Panel 4 lets you reset the control unit.

| Option | Explanation |
|---|---|
| **Erase CODESYS application, configuration and all files in the plc folder** | Erases all control unit applications and resets all control unit variables, including the RETAIN and RETAIN PERSISTENT variables of all applications. The function also erases all files and folders in directory /home/plc/applications. This action has to be followed by restarting the control unit. |
| **Erase CODESYS retain area** | Erases the RETAIN and RETAIN PERSISTENT variables of all applications. This action has to be followed by restarting the control unit. |

### 5.3.4.2 Config

Use this page to change some special control unit settings.

- ⇨ PLC application on SD-Card.
  This option links in the SD memory card such that the control unit is able to run applications directly from the card.
  Warning! Before using this feature, remember that it requires a special type of SD card provided by Kuhnke. Off-the-shelf SD cards will be considered a mass storage device but not an extra system memory volume. If the option is ticked, the system will only boot and be able to load an application if a Kuhnke SD card has been inserted. However, you still have access to the web interface to disable the option again.

### 5.3.4.3 Application Info

This page displays details of the control unit applications.

- ⇨ Applicationname.
  Unique name of the application. Can be changed in the CODESYS V3 development environment by changing the name of the "Application" object.
- ⇨ Status.
  Current status of the application.
  AS_RUN: application is running.
  AS_STOP: application stopped either manually or because of an error during execution.
- ⇨ Projectname, Projectauthor, Projectversion, Projectprofile, Projectdescription
  Displays the project details specified on the CODESYS V3 development platform. To change any of these details, run CODESYS V3, open the "Project" menu from the menu bar and choose "Project Information".
- ⇨ Exception-ID
  Tells you whether an error occurred in the application. Exception ID 0x00000000 means that there is no error.
- ⇨ Exception
  Name of error state

### 5.3.4.4 Application Files

This page lists all files stored in the control unit. You can separately download every file or use one of the following additional options:

- ⇨ Download folder from PLC
  Creates a downloadable archive of all files in directory /home/plc and further re-inflating information. Neither modify or unzip the archive if you wish to upload it for use on another control unit.
- ⇨ Upload folder to PLC
  Uploads a previously downloaded archive to the control unit. Use this option to quickly produce and restore backup files or to quickly send application to many other control units which cannot be accessed via the network.
  Attention! The feature will overwrite existing files and applications without a security prompt. Restart the control unit after you have uploaded an image.
- ⇨ Clean folder
  Removes all applications from the control unit. CODESYS V3 configuration files are retained. To also remove the CODESYS V3 configuration files, run function "Erase CODESYS application, configuration and all files in the plc folder" on panel PLC Control as described in section 5.3.4.1.
  Restart the control unit after running this function.

### 5.3.4.5 Font Files

This page lists all fonts currently installed in the control unit. Fonts are distinguished as "System Fonts" and "PLC Fonts". Section 8.3 explains how to install new fonts in the control unit.

Please note that you can either use the system fonts or any custom fonts. In case you wish to use both, your own fonts and fonts previously installed in the control unit, you will first of all have to download the system fonts and upload them back to the control unit's "PLC Fonts" folder together with your own fonts.

- ⇨ System Fonts
  Lists the fonts previously installed in the control unit by default. You can neither modify nor delete these default fonts.
- ⇨ PLC Fonts
  Lists the fonts you uploaded to the control unit.

## 5.3.5 Menu – Diagnostics

### 5.3.5.1 PLC Log

This page shows the log of CODESYS V3 Runtime. Log details include:

- The CODESYS V3 release installed and active licences.
- The system libraries you are using plus their version.
- Network information.
- CODESYS V3 events such as user logins and logouts or the downloading of applications.
- Errors or exceptions that occurred in CODESYS V3 Runtime.

### 5.3.5.2 System Log

This page has two panels:

Panel "System Log" displays the system log located in the file system at /var/log/messages. It contains general details of the operating system and the services and programmes currently running. The lighttpd web server, for example, also logs accesses to the web interface.

Panel "System Diag" logs all interaction between the system and CODESYS V3 Runtime. Entries include information about changes to the retain memory, CODESYS V3 Runtime states, times of boot and power fail events.

### 5.3.5.3 Ethernet

Use this page to view details of the control unit's network interface. The difference to the "Network" item (see section x.x.x) is that you cannot make any changes on this page. Instead it lists detailed information such as the MAC address, IP address as well as the data packets and volumes sent and received.

### 5.3.5.4 CAN

This page displays details of the CAN interfaces. A counter of internal error states informs you of the bus state:

| | |
|---|---|
| can state : ERROR_ACTIVE | CAN active (<96 error frames) |
| can state : ERROR_WARNING | CAN active (<128 error frames) |
| can state : ERROR_PASSIVE | CAN inactive (<256 error frames) |
| can state : ERROR_BUS_OFF | CAN of (>=256 error frames) |
| can state : ERROR_SLEEPING | CAN running in stand-by mode |
| can state : STOPPED | CAN stopped |

Further details you may view include the current baud rate setting, data packets and volumes received and sent and the total number of error frames received.

### 5.3.5.5 Storage

Use this page to view details of the control unit's memory state. The most important information for you will be the flash memory state (green highlight) and the state of the external SD card (blue highlight), if any.

You will find the one or several USB memory sticks connected (via a hub if several) in column "/media/usbx" (orange highlight). The x marks the order in which the sticks have been mounted (i.e. 1 if one stick is connected).

```
Filesystem          Size       Used      Available      Use%         Mounted on
ubi0_0             47.5M      15.6M          31.9M       33%         /
devtmpfs          106.8M          0         106.8M        0%         /dev
/dev/ubi0_1        47.5M      31.5M          16.0M       66%         /usr
None              196.6M      32.0K         196.6M        0%         /tmp
None              122.9M          0         122.9M        0%         /media
None              122.9M     116.0K         122.8M        0%         /run
None              122.9M      60.0K         122.8M        0%         /var/log
None              122.9M     116.0K         122.8M        0%         /var/run
None              122.9M          0         122.9M        0%         /var/lock
None              122.9M          0         122.9M        0%         /var/tmp
/dev/ubi1_0       104.3M     596.0K         103.7M        1%         /flash
/dev/ubi1_0       104.3M     596.0K         103.7M        1%         /home/plc
/dev/ubi1_0       104.3M     596.0K         103.7M        1%         /usr/local
/dev/ubi1_0       104.3M     596.0K         103.7M        1%         /var/cache
/dev/ubi1_0       104.3M     596.0K         103.7M        1%         /var/spool
/dev/ubi0_1        47.5M      31.5M          16.0M       66%         /etc
None              196.6M      32.0K         196.6M        0%         /var/www/tmp
/dev/mmcblk0p1    945.7M       3.8M         941.9M        0%         /media/sd
/dev/sda1           7.5G      24.9M           7.5G        0%         /media/usb1
```

### 5.3.5.6 System Dump

This page allows you to create an image of the control unit's entire diagnostics realm. Use this function to analyse errors occurring in an application or the control unit. We recommend creating the image file immediately after the error occurred, i.e. without restarting the control unit first. It may take several minutes to create the image file. Once it is available, the browser will provide it for download. Save this file and send it to Kendrion Kuhnke Support for analysis.

## 5.3.6 Logout

Find the logout link in the bottom right corner of the screen.

# 5.4 System Functions

## 5.4.1 FTP Server

Run the FTP server of FIO Controller 116 to allow an FTP client to access the controller. Refer to section 5.3.2.5 "FTP server" Menu Item

FTP lets you exchange recipe data with the control unit, for example.

### File System and Folder Tree

Most of the control unit's file system folders and files can either not be accessed or just read by the user. To be able to use certain control unit functions, however, there are some directories that end users have write and use access to.

**Overview**

| Directory | Description |
|---|---|
| /flash | Control unit's internal flash memory |
| /flash/plc | CODESYS V3 root directory. Also mounted at /home/plc. |
| /flash/plc/applications | Location of the CODESYS V3 application |
| /flash/plc/applications/fonts | Location of some fonts which needs to be created once if you intend to use your own fonts. |
| /flash/ftpupload | FTP users root. Use this as your default directory for files downloaded or uploaded via FTP. |
| /media/sd | External flash memory (SD card); only available if a SD card is used. |
| /media/usb1 | External USB memory; only available if a storage medium has been plugged in. |

## 5.4.2 Service Mode

Service mode differs from normal operation in the following points:

- CODESYS V3 Runtime will not run, i.e. no control programme will be running. The factory defaults will be loaded when configuring the network to ensure that every control unit can be addressed by a unique IP address. Refer to section 5.2.1 to learn more about this kind of network configuration.
- Underneath the name of the control unit, the web interface will display "(maintenance)" instead of "(plcactive)".

Service mode is enabled by turning off FIO Controller 116. Now press and hold the button to restart FIO Controller 116. Keep pressing the button until Run/Stop lights up yellow every 2 seconds.

## 5.4.3 Installing More Fonts

By default, DejaVu series fonts are installed. If you wish to use other fonts such as Windows fonts for your WebVisu or TargetVisu visualisations, you may copy any set of Truetype fonts (*.ttf) to folder "/flash/plc/applications/fonts/" and use them afterwards.

If you have not used any own fonts yet, you will have to create this folder once. We recommend running the control unit's FTP server application and downloading the fonts into the control unit via FTP.

Attention! For system reasons, you will have to use lowercase letters for the font file extension because the system will otherwise fail to recognise the file and you may have to rename the file. For example, be sure Arial unicode font file is called "ARIALUNI.ttf". The control unit will not recognise "ARIALUNI.TTF" or "ARIALUNI.Ttf".

Attention! You can choose to use the on-board or additional fonts. To make parallel use of your own and the on-board DejaVu fonts, first of all download the DejaVu fonts from directory "/var/lib/fonts" or via the web interface and upload them to folder "/flash/plc/applications/fonts/" together with your own fonts.

| | ***Information*** |
|---|---|
| | ***Many fonts you may know from daily PC use are subject to a license! Arial, for example, is a Microsoft font covered by your Windows licence.***<br>***Before using alternative fonts, please verify that they non-licensed freeware or that you own the licence.*** |

### 5.4.4 Update- Function

The FIO Controller 116 can be updated via USB stick or SD card.

- USB stick: The update must be saved in the directory "usbupdate-mx6" on a USB stick.
- SD- card: The update must be saved in the directory "sdupdate-mx6" on a SD- card.

The file "usbupdate.ini" in the update directory ("usbupdate-mx6" or "sdupdate-mx6") is the central configuration file. In this file you can set which actions are performed when the device is booted.

If this file is not available, the file "usbupdate_TEMPLATE-ET.ini" or "usbupdate_TEMPLATE-PLC.ini" must be renamed to "usbupdate.ini" and then edited, depending on the controller type. The usbupdate.ini is divided into five sections and in each of these sections there is at least one pair of key and value.

Furthermore, the file "usbupdate.tgz" must be located in the directory "usbupdate-mx6".

A complete template can be found on our website at:

https://productfinder.kuhnke.kendrion.com/en/modular-plcs/

***Information***

*The SD card slot has no hot-plug capability. They will therefore detect cards only if they were plugged in at the time of booting the unit. SD cards have no hot-swap capability.*

All entries in usbupdate.ini have a short description of the function in English. Most of the values are of type "boolean" and can take the values "yes" or "no".

The following five sections exist:

- [firmware]: Settings for firmware upgrades or downgrades.
- [webtheme]: Settings for exchanging the logo visible in the web interface.
- [splashscreen]: Settings for exchanging the boot logo of a display controller.
- [sysconfig]: Settings to change the system configuration.
- [plcapp]: Settings to perform an application update or copy operation.

A folder of the same name exists on the USB memory for each of these sections. The files required for the respective execution are then stored in these folders.

Since usbupdate.ini is only a text file with a special formatting, it can be edited with any text editor. However, it is recommended to use an editor that supports the ini format and formats the text correctly, as well as a colored background, such as the free software Notepad++.

Each section and also each key/value pair can be used independently. So you can create a USB update which changes only one setting or copy files to the controller without doing an application update. The user can choose freely and create different USB memories for different occasions. Non-existent or commented entries are simply ignored.

In the following chapters the possible entries of all sections are briefly introduced.

### 5.4.4.1 Update: section [firmware]

The firmware section consists of the following keys:

| Overview | | |
|---|---|---|
| **key** | **value area** | **description** |
| do_update | yes / no | |
| firmware_name | File name | The file must be located in the "firmware" subfolder.<br>Example: firmware_mx6-plc_1.5.0.tgz |

### 5.4.4.2 Update: section [webtheme]

The Webtheme section consists of the following key:

| Overview | | |
|---|---|---|
| **key** | **value area** | **description** |
| do_update_webtheme | yes / no | |
| | | |

The logo must be a GIF image file named "logo.gif" and located in the "webtheme" subfolder. The image is scaled by the browser in the web interface, but it is still recommended to create the logo directly in the appropriate resolution for the intended use.

### 5.4.4.3 Update: Sektion [sysconfig]

The Sysconfig section consists of the following keys:

| Overview | | |
|---|---|---|
| **key** | **value area** | **description** |
| do_reset_syscfg_to_factory_defaults | yes / no | This key specifies whether the configuration of the controller is to be set to the delivery state. |
| do_sysconfig_from_file | yes / no | This key specifies whether the configuration of the controller should be updated with the settings from "configartion.ini” |
| replace_config_file_instead_of_merge | yes / no | This key specifies that the configuration of the controller is reset to the factory setting and then updated with the configuration from the "configuration.ini". |

Many control settings can be changed via an external file. The configuration settings to be updated are defined in the file "configuration.ini" in the "sysconfig" folder.

#### 5.4.4.4 Update: section [plcapp]

The Plcapp section consists of the following keys:

**Overview**

| **key** | **value area** | **description** |
|---|---|---|
| do_clean_plcfolder | yes / no | This key specifies whether the application folder on the controller (/flash/plc/applications/) is to be deleted completely. |
| do_update_plcapp | yes / no | This key specifies whether a program update of the control program is to be carried out. All files from the update archive are written to the controller, files with the same name are overwritten. |
| plcapp_name | file name | File name for the action "do_update_plcapp". This file must be located in the subfolder "application". An update file must first have been created and downloaded via the web interface on another controller.<br>See chapter 5.3.4.4 Application Files |
| do_copy_plcdata | yes / no | This key specifies whether all files contained in the subfolder "application\data" should be copied to the directory "/flash/plc/applications/". |

### 5.4.4.5 Update: Change control settings via the "configuration.ini" file

All control settings that can be set via the web interface can be changed automatically via a USB update. The file "configuration.ini" in the folder "sysconfig" serves as a basis.

As in "usbudate.ini", various sections with key/value pairs are stored in this file.

In this documentation only the standard sections and keys are introduced. If you need more information about available sections and keys, please contact technical support.

**Section [network]**

**Overview**

| key | Value area | description |
|---|---|---|
| eth0.mode | „static“ / „dhcp“ / „inactive“ | Interfcae mode |
| eth0.ip | xxx.xxx.xxx.xxx | IP address, e.g. 192.168.0.116 |
| eth0.netmask | xxx.xxx.xxx.xxx | Networkmask, e.g. 255.255.255.0 |
| default_gateway | xxx.xxx.xxx.xxx | IP address, e.g. 192.168.0.1 |

The second network card of the controller can of course also be configured using the corresponding eth1.xxx keys.

**Section [ftp]**

**Overview**

| key | Value area | description |
|---|---|---|
| enabled | 0 / 1 | 0 deaktiviert den FTP- Server<br>1 aktiviert den FTP- Server |
| | | |

# 5.5 Maintenance / Servicing

## 5.5.1 General

Only qualified persons are allowed to work on Kuhnke FIO Controller 116.

***NOTE***

***Do not plug, unplug, mount or touch the connectors during operation.***

*Risk of destroying the unit or provoking malfunctions*

⇨ Before working on the unit, turn off all power sources including those feeding power to peripherals such as externally fed sensors, programming devices etc.

***NOTE***

***Overheating***

*Risk of destroying the unit or provoking malfunctions*

⇨ Verify that the unit's ventilation slots are not covered and that air is able to circulate.

## 5.5.2 Servicing

Kuhnke FIO Controller 116 requires neither servicing for the specified service life nor any action if it is kept and operated at the admissible ambient conditions specified in section3.4 Technical Data.

## 5.5.3 Preventive Maintenance

### Cleaning

Prevent inadmissible contamination while operating and storing Kuhnke FIO Controller 116.

### To replace modules

Refer to section 5.1.1 Mechanical Installation

## 5.5.4 Repairs / Customer Service

***Information***

*Only the manufacturer or customer service providers authorised by the manufacturer are allowed to do repairs and perform corrective maintenance.*

## 5.5.5 Warranty

The statutory period and conditions of warranty apply. Warranty expires if unauthorised attempts are made to repair the unit / product or any other intervention is performed.

## 5.6 Error Handling

FIO Controller 116 features maintenance and error handling solutions you can operate without a monitor, keyboard or mouse.

## 5.7 Taking out of Service

### 5.7.1 Disposal

Before disposing of Kuhnke FIO Controller 116 you must disassemble it and completely take it apart. All metal components can be given to metal recycling.

#### Electronic scrap

Sort and dispose of electronic components by type. For details on proper disposal please check your national laws and regulations making sure that your method of disposal complies with them.

Treat the packaging as recyclable paper and cardboard.

# 6 CODESYS V3 Development Environment

## 6.1 Installing CODESYS on the Project Engineering PC

CODESYS is a device-independent system for programming control units. It conforms to standard IEC 61131-3 and supports all standardised IEC programming languages plus the integration of C code routines and object-orientated programming.

In conjunction with runtime system CODESYS Control Win V3 it also allows the use of "multi-device" and "multi-application" programs. Owing to its component-based architecture, it supports customer-specific configurations of and extensions to the user interface.

Before installing CODESYS, please read and take note of the system requirements:

Windows XP / 7 / 8 / 10 (32/64 bit), appropriate PC hardware matching the Windows version installed.

### 6.1.1 Installing the Device Description in CODESYS V3

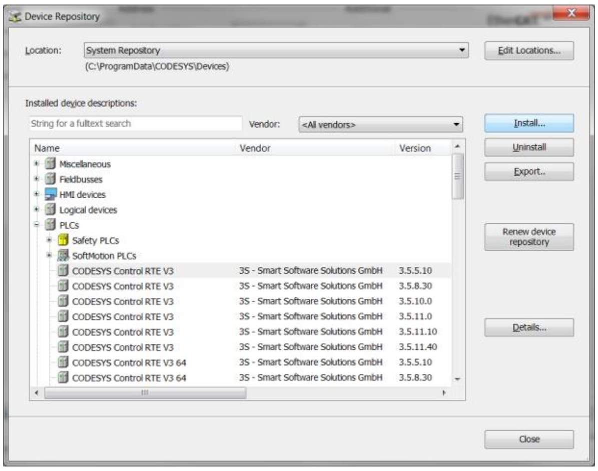
Before the IEC 61131-3 development tool CODESYS V3 can be used to operate a device, the device and its properties are to be made known to the runtime system.

A separate plug-in, the Device Repository, provides the local system and your projects with the device definition management functions. Among other features, it contains commands of category Devices which you will normally find in the Tools menu.

- ⇨ Open the Tools menu and pick Device Repository...
- ⇨ Screen Device Repository is displayed
- ⇨ Expand the tree at Control Units (PLC)

The Device Repository hosts the descriptions of all devices currently installed on the local system and makes the devices available for CODESYS programming. Devices are installed in and uninstalled directly from the Device Repository.

Screen Device Descriptions Installed displays the Name, Vendor and Version of every device that is currently installed. Click on the plus and minus signs to expand or collapse the branches.



Location: System Repository
(C:\ProgramData\CODESYS\Devices)

Installed device descriptions:

String for a fulltext search

Vendor: <All vendors>

| Name | Vendor | Version |
|---|---|---|
| Miscellaneous | | |
| Fieldbusses | | |
| HMI devices | | |
| Logical devices | | |
| PLCs | | |
| Safety PLCs | | |
| SoftMotion PLCs | | |
| CODESYS Control RTE V3 | 3S - Smart Software Solutions GmbH | 3.5.5.10 |
| CODESYS Control RTE V3 | 3S - Smart Software Solutions GmbH | 3.5.8.30 |
| CODESYS Control RTE V3 | 3S - Smart Software Solutions GmbH | 3.5.10.0 |
| CODESYS Control RTE V3 | 3S - Smart Software Solutions GmbH | 3.5.11.0 |
| CODESYS Control RTE V3 | 3S - Smart Software Solutions GmbH | 3.5.11.10 |
| CODESYS Control RTE V3 | 3S - Smart Software Solutions GmbH | 3.5.11.40 |
| CODESYS Control RTE V3 64 | 3S - Smart Software Solutions GmbH | 3.5.5.10 |
| CODESYS Control RTE V3 64 | 3S - Smart Software Solutions GmbH | 3.5.8.30 |

Click on Install... to install a new device on the local system and make it available for use in the programming system.

Dialog Device Descriptions Installed is displayed. Use it to search the system for a specific device description file. If you are looking for a standard device, set the file type filter to "*.**devdesc.xml**" (device description). Another option is to find vendor-specific description files such as *gsd  files of Profibus DP modules or *.eds and *.dcf files of CAN devices.

Clicking on OK confirms your settings, closes the screen and adds the device to the tree shown on screen Device Repository. Look at the bottom of screen Device Repository to find messages on any installation error (e.g. missing files referenced by the device description)t.

## 6.1.2 Installing Device-specific Libraries

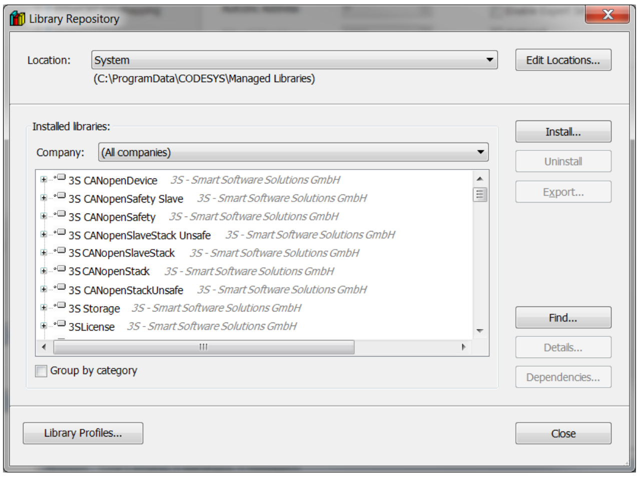
Similar to the device description files, CODESYS keeps libraries in a dedicated repository, i.e. the Library Repository in this case.

The following device-specific libraries are available for FIO Controller 116:

- Kuhnke System Library iMX6
  Library providing access to the system settings
- Kuhnke iMX6 Onboard IO Library
  Library of functions of the on-board IO module

⇨ Open the Tools menu and pick Library Repository...

⇨ The Library Repository dialog is displayed



Location: System
(C:\ProgramData\CODESYS\Managed Libraries)

Installed libraries:

Company: (All companies)

3S CANopenDevice 3S - Smart Software Solutions GmbH
3S CANopenSafety Slave 3S - Smart Software Solutions GmbH
3S CANopenSafety 3S - Smart Software Solutions GmbH
3S CANopenSlaveStack Unsafe 3S - Smart Software Solutions GmbH
3S CANopenSlaveStack 3S - Smart Software Solutions GmbH
3S CANopenStack 3S - Smart Software Solutions GmbH
3S CANopenStackUnsafe 3S - Smart Software Solutions GmbH
3S Storage 3S - Smart Software Solutions GmbH
3SLicense 3S - Smart Software Solutions GmbH

□ Group by category

Click on Install... to install a new library in the local system and make it available for use in the programming system.

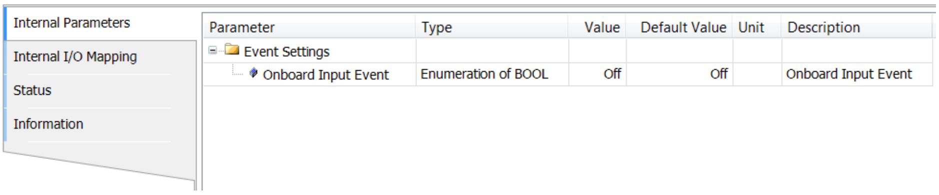
Browse to the location where your libraries are saved. The default filter is Compiled Libraries (*.compiled-library) which is the format in which libraries are normally made available. Choose the library you wish to install and click on Open. Once installed, the library will appear on tree Installed Libraries.

## 6.2 FIO Controller 116 and CODESYS

FIO Controller 116 supports various CODESYS options of capturing and processing signals. The signals can be provided locally or by various bus systems. This requires you to first of all add the devices concerned to the device list by right-clicking on "Device (FIO Controller 116)" and picking "Add device …" from the popup menu.

### 6.2.1 Digital Input

Now append the "Onboard_IO" to FIO Controller 116.

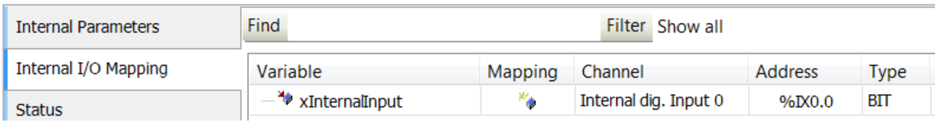


| Parameter | Type | Value | Default Value | Unit | Description |
|---|---|---|---|---|---|
| Event Settings | | | | | |
| Onboard Input Event | Enumeration of BOOL | Off | Off | | Onboard Input Event |

The digital input can be used as standard PLC input (default setting) as well as event input for fast signal processing.

#### 6.2.1.1 Standard SPS Eingang

To use the digital input, you can associate a variable with its address (mapping), e.g. in which you assign a name for the input under "Internal I / O Mapping".

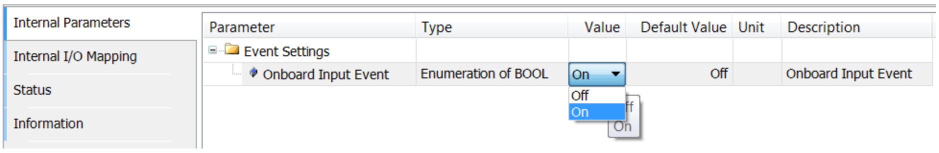


| Variable | Mapping | Channel | Address | Type |
|---|---|---|---|---|
| xInternalInput | | Internal dig. Input 0 | %IX0.0 | BIT |

### 6.2.1.2 Ereigniseingang

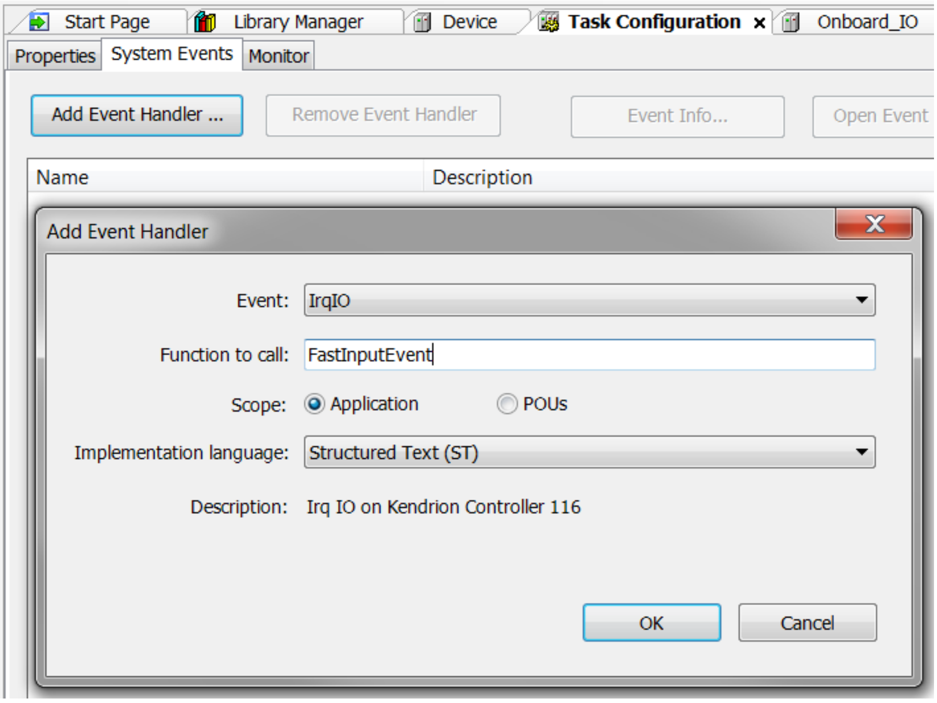
To process fast signals, the digital input can also be configured as an event input. Activate this under "Internal Parameter".

| | **Information** |
|---|---|
| | *In case the input should be used as event input the library „KICSOnboardIo_1.0.0.0“ has to be added to the library repository.* |



| Parameter | Type | Value | Default Value | Unit | Description |
|---|---|---|---|---|---|
| Event Settings | | | | | |
| Onboard Input Event | Enumeration of BOOL | On | Off | | Onboard Input Event |

Afterwards, a system event of the type "IrqIO" has to be defined in the task configuration.



Event: IrqIO

Function to call: FastInputEvent

Scope: ● Application ○ POUs

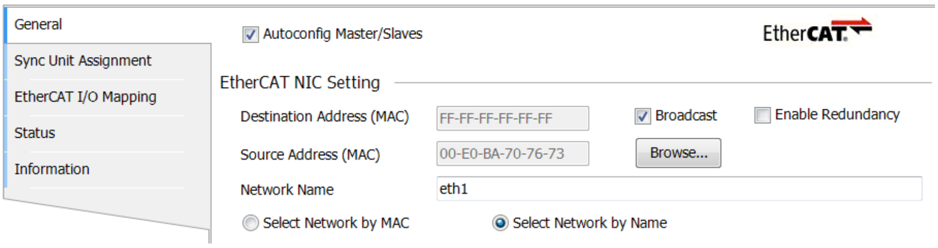
Implementation language: Structured Text (ST)

Description: Irq IO on Kendrion Controller 116

The function to be called is automatically created in the project tree. Here the response to the input signal can be programmed.

### 6.2.2 EtherCAT Master

Now append the "EtherCAT Master" to FIO Controller 116. Double-click to configure it. Go to the general settings and enter "eth1" as the interface:



☑ Autoconfig Master/Slaves

EtherCAT NIC Setting

Destination Address (MAC): FF-FF-FF-FF-FF-FF ☑ Broadcast ☐ Enable Redundancy

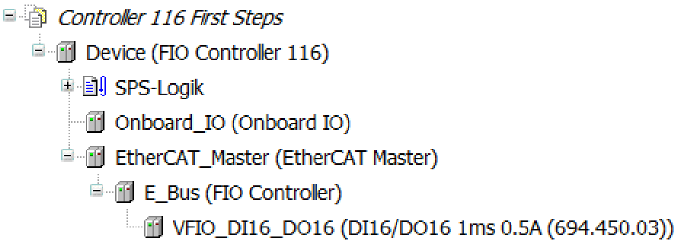
Source Address (MAC): 00-E0-BA-70-76-73 Browse...

Network Name: eth1

○ Select Network by MAC ● Select Network by Name

Now append the FIO Controller (Internal E-Bus) to the EtherCAT Master. The device is the starting point of appending all further Kuhnke FIO EtherCAT-series modules, e.g. FIO DI16/DO16

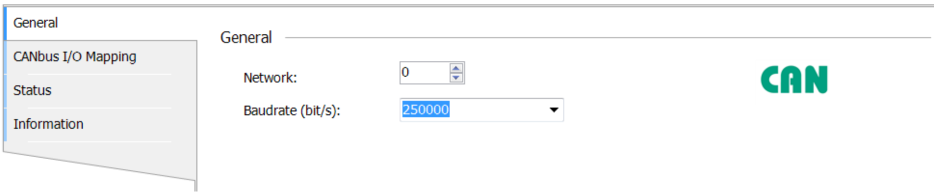
Devices explorer with EtherCAT



```
⊟ Controller 116 First Steps
  ⊟ Device (FIO Controller 116)
    ⊞ SPS-Logik
      Onboard_IO (Onboard IO)
    ⊟ EtherCAT_Master (EtherCAT Master)
      ⊟ E_Bus (FIO Controller)
          VFIO_DI16_DO16 (DI16/DO16 1ms 0.5A (694.450.03))
```

## 6.2.3 CANopen Master

Append the "CAN Bus" to FIO Controller 116. Double-click to configure it. Go to the general settings and pick the appropriate baud rate. Do not change the Network (0):

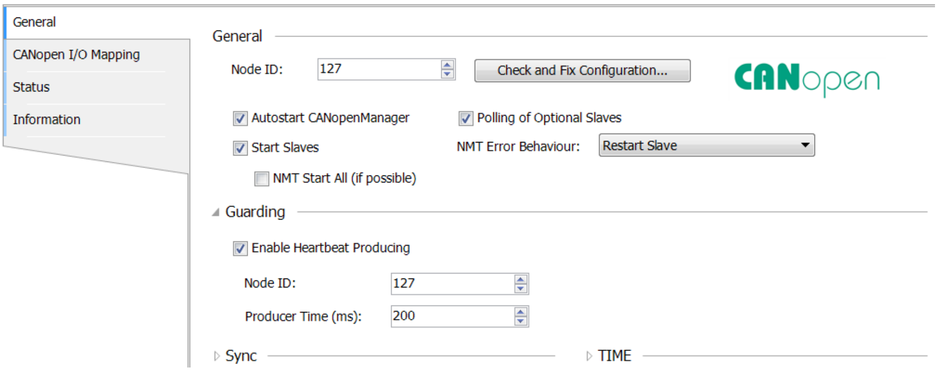


General

Network: 0

Baudrate (bit/s): 250000

Now append the CANopen_Manager to the CAN Bus. Its settings depend on and may have to be adapted to all later bus stations.



General

Node ID: 127

☑ Autostart CANopenManager

☑ Polling of Optional Slaves

☑ Start Slaves

NMT Error Behaviour: Restart Slave

□ NMT Start All (if possible)

Guarding

☑ Enable Heartbeat Producing

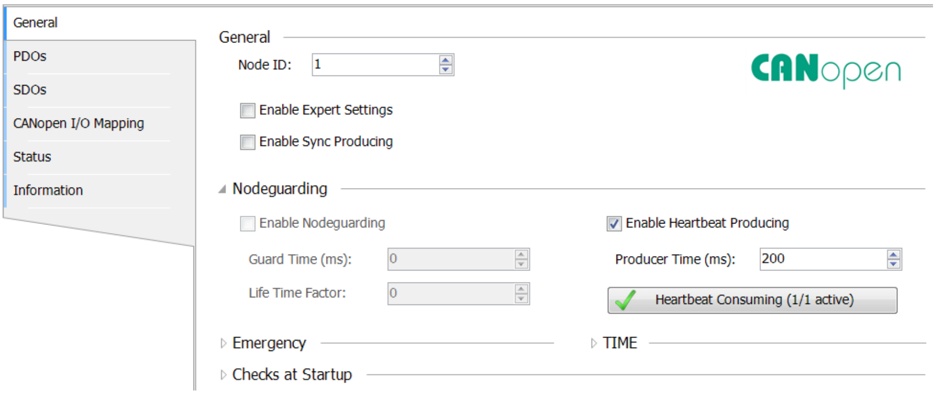
Node ID: 127

Producer Time (ms): 200

Sync

TIME

Now append the CANopen slaves to the CANopen_Manager, as appropriate. Refer to the slave manufacturer's instructions to know how to configure the slaves.



General

Node ID: 1

□ Enable Expert Settings

□ Enable Sync Producing

Nodeguarding

□ Enable Nodeguarding

☑ Enable Heartbeat Producing

Guard Time (ms): 0

Producer Time (ms): 200

Life Time Factor: 0

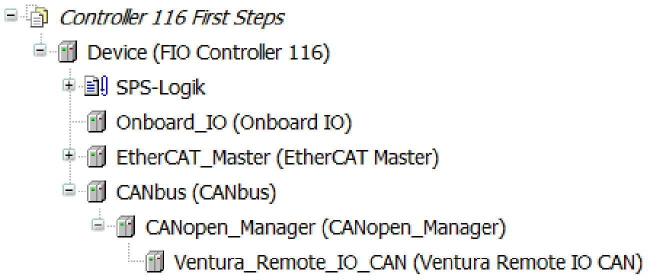
✓ Heartbeat Consuming (1/1 active)

Emergency

TIME

Checks at Startup

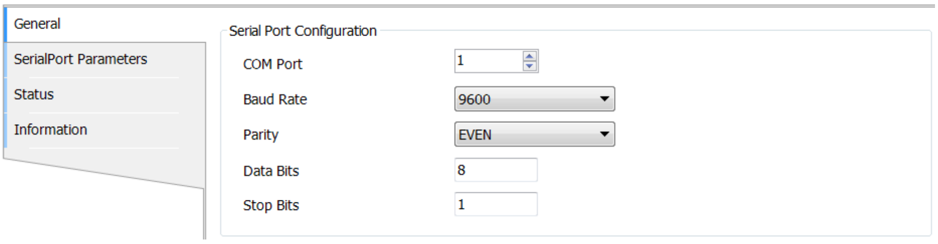
Devices explorer with CAN bus



- *Controller 116 First Steps*
  - Device (FIO Controller 116)
    - SPS-Logik
    - Onboard_IO (Onboard IO)
    - EtherCAT_Master (EtherCAT Master)
    - CANbus (CANbus)
      - CANopen_Manager (CANopen_Manager)
        - Ventura_Remote_IO_CAN (Ventura Remote IO CAN)

### 6.2.4 Modbus RTU Master

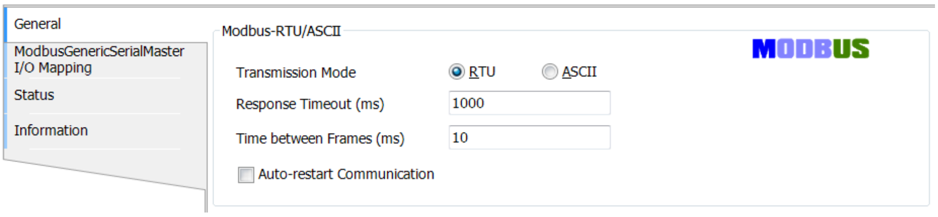
Append the "Modbus COM Port" to FIO Controller 116. Double-click to configure it. Go to the general settings and set up the COM Port (1). Its other settings such as the baud rate, parity, data bits and stop bits depend on and may have to be adapted to all later bus stations.



Serial Port Configuration

COM Port: 1
Baud Rate: 9600
Parity: EVEN
Data Bits: 8
Stop Bits: 1

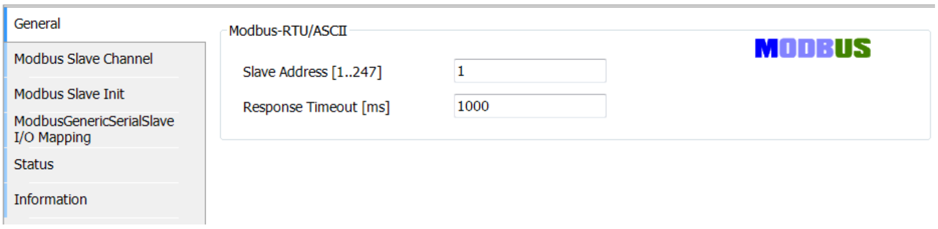
Now append a Modbus RTU Master. Again, its configuration depends on and may have to be adapted to lower-level bus stations.



Modbus-RTU/ASCII

Transmission Mode: ● RTU ○ ASCII
Response Timeout (ms): 1000
Time between Frames (ms): 10
□ Auto-restart Communication

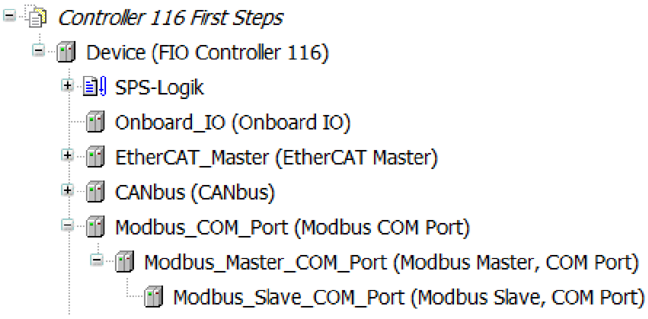
Add the required Modbus RTU slaves to the Modbus RTU Master. Refer to the slave manufacturer's instructions to know how to configure the slaves.



Modbus-RTU/ASCII

Slave Address [1..247]: 1
Response Timeout [ms]: 1000

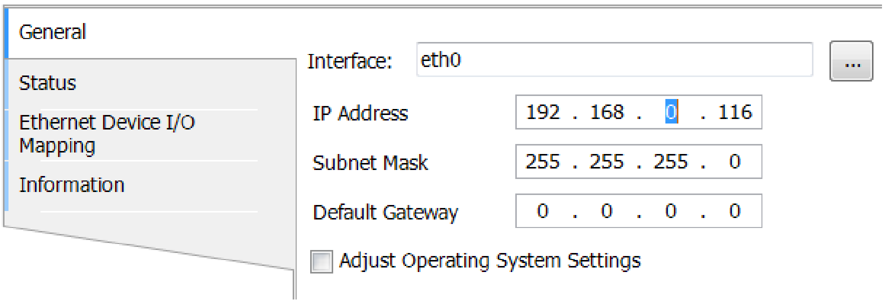
Devices explorer with Modbus RTU



```
Controller 116 First Steps
  Device (FIO Controller 116)
    SPS-Logik
    Onboard_IO (Onboard IO)
    EtherCAT_Master (EtherCAT Master)
    CANbus (CANbus)
    Modbus_COM_Port (Modbus COM Port)
      Modbus_Master_COM_Port (Modbus Master, COM Port)
        Modbus_Slave_COM_Port (Modbus Slave, COM Port)
```

## 6.2.5 Modbus TCP Master

Append the "Ethernet" device to FIO Controller 116. Double-click to configure it. Go to the general settings and enter "eth0" as the interface. If the device is connected to the control unit, click on ... and pick interface "eth0". The control unit will then accept the IP address, subnet mask and standard gateway.



Interface: eth0

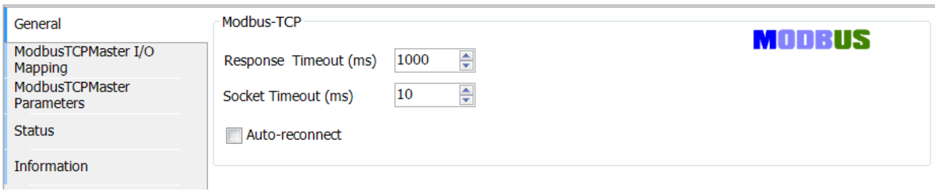
IP Address: 192 . 168 . 0 . 116

Subnet Mask: 255 . 255 . 255 . 0

Default Gateway: 0 . 0 . 0 . 0

□ Adjust Operating System Settings

Now append a Modbus TCP Master. Its settings depend on and may have to be adapted to all later bus stations.



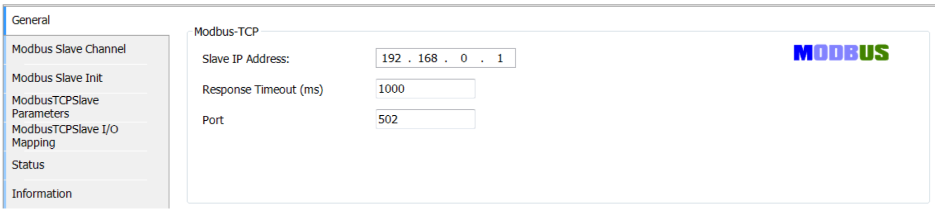
Modbus-TCP

Response Timeout (ms): 1000

Socket Timeout (ms): 10

□ Auto-reconnect

Add the required Modbus TCP slaves to the Modbus TCP Master. Refer to the slave manufacturer's instructions to know how to configure the slaves.



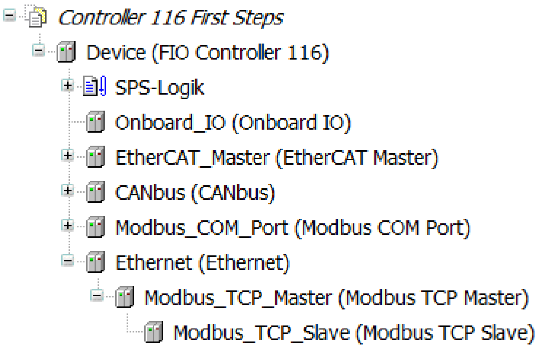
Modbus-TCP

Slave IP Address: 192 . 168 . 0 . 1

Response Timeout (ms): 1000

Port: 502

Devices explorer with Modbus TCP



- *Controller 116 First Steps*
  - Device (FIO Controller 116)
    - SPS-Logik
    - Onboard_IO (Onboard IO)
    - EtherCAT_Master (EtherCAT Master)
    - CANbus (CANbus)
    - Modbus_COM_Port (Modbus COM Port)
    - Ethernet (Ethernet)
      - Modbus_TCP_Master (Modbus TCP Master)
        - Modbus_TCP_Slave (Modbus TCP Slave)

## 6.2.6 PROFINET IO Device extension

The FIO Controller 116 communicates with the PROFINET IO Device extension via a shared memory. The system function SYF_GetAuxCoreMem is used to identify the start address of the shared memory.

The following dta are available for the Master:

### Cyclic data (process data):

- ARRAY [0..255] OF BYTE; (* cyclic data to the master*)
- ARRAY [0..255] OF BYTE; (* cyclic data from the master *)

### Acyclic Data (parameter data):

- ARRAY [0..199] OF INT; (* Index 100.. 299 *)
- ARRAY [0..199] OF UINT; (* Index 300.. 499 *)
- ARRAY [0..199] OF DINT; (* Index 500.. 699 *)
- ARRAY [0..199] OF UDINT; (* Index 700.. 899 *)
- ARRAY [0..199] OF REAL; (* Index 900..1099 *)
- ARRAY [0..19] OF STRING[63]; (* Index 1100..1119 *)

The data have to be defined in the above shown order.The symbolic names can be adapted individually.

### Programming example:

At first, a data type has to be defined as described above

```
TYPE tProfinetSharedMemory :
STRUCT
       bDataToMaster:   ARRAY[0..255] OF BYTE; (* cyclic output data *)
       bDataFromMaster:    ARRAY[0..255] OF BYTE; (* cyclic intput data *)
       (* azyklische Parameter Daten *)
       Int16:  ARRAY[0..199] OF INT;(* Index  100.. 299 *)
       UInt16: ARRAY[0..199] OF UINT;(* Index  300.. 499 *)
       Int32:  ARRAY[0..199] OF DINT;(* Index  500.. 699 *)
       UInt32: ARRAY[0..199] OF UDINT;(* Index  700.. 899 *)
       Real32: ARRAY[0..199] OF REAL;(* Index  900..1099 *)
       Strings: ARRAY[0..19]  OF STRING[63];(* Index 1100..1119 *)
END_STRUCT
END_TYPE
```

Afterwards a varaible of this data type as well as variable for the start address has to be defined, e.g. in the gobal variable list GVL.

```
ProfinetSharedMemory: POINTER TO tProfinetSharedMemory;
dwSharedMemoryAddress: DWORD;
```

Afterwards the SharedMomory must be initialized. For this purpose the function "KICSSYS.SYF_GetAuxCoreMem()" is called in an initialization sequence, e.g.

```
IF NOT xInitProfiNetIRT THEN
        // get shared memory A9/M4 address
        GVLProfinet.dwSharedMemoryAddress := KICSSYS.SYF_GetAuxCoreMem();
        // set structure pointer to address
        GVLProfinet.pProfinetSharedMemory := GVLProfinet.dwSharedMemoryAddress;
        xInitProfiNetIRT := TRUE;
ELSIF GVLProfinet.pProfinetSharedMemory <> 0 THEN
        // Mirroring inputs to outputs
        FOR iFor :=0 TO 255 BY 1 DO
                GVLProfinet.pProfinetSharedMemory^.bDataToMaster[iFor] :=
                        GVLProfinet.pProfinetSharedMemory^.bDataFromMaster[iFor];
        END_FOR
END_IF
```

### 6.2.7 EtherCAT Slave extension

The FIO controller 116 communicates with the EtherCAT slave extension via a shared memory. The start address of the shared memory is determined via the system function SYF_GetAuxCoreMem.

The following data are available for the master:

#### Cyclic data (process data):

- ARRAY [0..255] OF BYTE; (* cyclic data to the master*)
- ARRAY [0..255] OF BYTE; (* cyclic data from the master *)

#### Acyclic Data (service data):

- ARRAY [0..199] OF INT; (* Index  100.. 299 *)
- ARRAY [0..199] OF UINT; (* Index  300.. 499 *)
- ARRAY [0..199] OF DINT; (* Index  500.. 699 *)
- ARRAY [0..199] OF UDINT; (* Index  700.. 899 *)
- ARRAY [0..199] OF REAL; (* Index  900..1099 *)
- ARRAY [0..19] OF STRING[63]; (* Index 1100..1119 *)

The data have to be defined in the above shown order.The symbolic names can be adapted individually.

#### Programming example:

At first, a data type has to be defined as described above

```
TYPE tEtherCATSharedMemory :
STRUCT
       bDataToMaster:   ARRAY[0..255] OF BYTE; (* cyclic output data *)
       bDataFromMaster:    ARRAY[0..255] OF BYTE; (* cyclic intput data *)
       (* azyklische Service Daten *)
       Int16:  ARRAY[0..199] OF INT;(* Index  100.. 299 *)
       UInt16: ARRAY[0..199] OF UINT;(* Index  300.. 499 *)
       Int32:  ARRAY[0..199] OF DINT;(* Index  500.. 699 *)
       UInt32: ARRAY[0..199] OF UDINT;(* Index  700.. 899 *)
       Real32: ARRAY[0..199] OF REAL;(* Index  900..1099 *)
       Strings: ARRAY[0..19]  OF STRING[63];(* Index 1100..1119 *)
END_STRUCT
END_TYPE
```

Afterwards a varaible of this data type as well as variable for the start address has to be defined, e.g. in the gobal variable list GVL.

```
EtherCATSharedMemory: POINTER TO tEtherCATSharedMemory;
dwSharedMemoryAddress: DWORD;
```

Afterwards the SharedMomory must be initialized. For this purpose the function "KICSSYS.SYF_GetAuxCoreMem()" is called in an initialization sequence, e.g.

```
    IF NOT xInitEtherCAT THEN
        // get shared memory A9/M4 address
        GVLEtherCAT.dwSharedMemoryAddress := KICSSYS.SYF_GetAuxCoreMem();
        // set structure pointer to address
        GVLEtherCAT.pEtherCATSharedMemory := GVLEtherCAT.dwSharedMemoryAddress;
        xInitEtherCAT := TRUE;
    ELSIF GVLEtherCAT.pEtherCATSharedMemory <> 0 THEN
        // Mirroring inputs to outputs
        FOR iFor :=0 TO 255 BY 1 DO
            GVLEtherCAT.pEtherCATSharedMemory^.bDataToMaster[iFor] :=
                GVLEtherCAT.pEtherCATSharedMemory^.bDataFromMaster[iFor];
        END_FOR
    END_IF
```

## 6.3 Device-specific Libraries

### 6.3.1 Kuhnke System Library iMX6

Kuhnke System Library iMX6 provides you with various diagnostic and control unit configuration functions.

#### 6.3.1.1 Configuration

##### 6.3.1.1.1 FUNCTION CNFRTS_GetOperatorButtonDisable

Tells you whether the Stop / Reset button function is enabled or disabled.

Output parameters:

CNFRTS_GetOperatorButtonDisable: BOOL

FALSE: Stop / Reset button enabled

TRUE: Stop / Reset button disabled

##### 6.3.1.1.2 FUNCTION CNFRTS_GetOperatorButtonResetMode

Outputs the reset mode the Stop / Reset button is set to.

Input parameters:

peButtonResetMode: POINTER TO ButtonResetMode

0: COLD (cold reset)

1: WARM (warm reset)

Output parameters:

CNFRTS_GetOperatorButtonResetMode: UDINT

0: done

1: failed

##### 6.3.1.1.3 FUNCTION CNFRTS_SetOperatorButtonDisable

Enables or disables the Stop / Reset button function

Input parameters:

bDisable

TRUE: disable Stop/Reset button

FALSE: enable Stop/Reset button

Output parameters:

CNFRTS_SetOperatorButtonDisable: UDINT

0: done

1: failed

### 6.3.1.1.4 FUNCTION CNFRTS_SetOperatorButtonResetMode

Sets the Stop / Reset button's reset mode

Input parameters:

eButtonMode: ButtonResetMode

0: COLD (cold reset)

1: WARM (warm reset)

Output parameters:

CNFRTS_SetOperatorButtonResetMode: UDINT

0: done

1: failed

### 6.3.1.1.5 FUNCTION CNF_ApplySettings

Stores all configuration settings on the control unit's non-volatile memory. The settings will not take effect until you restart the control unit.

Output parameters:

CNF_ApplySettings: UDINT

0: done

1: failed

### 6.3.1.1.6 FUNCTION CNF_GetApplicationOnSd

Tells you whether the control application is stored on the SD card.

Output parameters:

CNF_GetApplicationOnSd: BOOL

FALSE: control application stored on internal flash memory

TRUE: control application stored on the SD card

### 6.3.1.1.7 FUNCTION CNF_GetDnsAddress

Reads the DNS address.

Input parameters:

sInterface: STRING

Values: dns0, dns1

Input / output parameters:

abyDnsAddress: ARRAY [0..3] OF BYTE

Output parameters:

CNF_GetDnsAddress: UDINT

0: done

1: failed

### 6.3.1.1.8 FUNCTION CNF_GetEthMode

Retrieves the current mode setting from the system configuration.

Input parameters:

sInterface: STRING

Values: eth0/eth1/...

Input / output parameters:

eMode: EthMode

Output parameters:

CNF_GetEthMode: UDINT

0: done

1: failed

### 6.3.1.1.9 FUNCTION CNF_GetGatewayAddress

Reads the gateway address.

Input parameters:

sInterface: STRING

Values: eth0, eth1

Input / output parameters:

abyGatewayAddress: ARRAY [0..3] OF BYTE

Output parameters:

CNF_GetGatewayAddress: UDINT

0: done

1: failed

### 6.3.1.1.10 FUNCTION CNF_GetHardwareRevisionString

Retrieves the hardware version ID from the control unit.

Input / output parameters:

sHwRevision: STRING

Output parameters:

CNF_GetHardwareRevisionString: UDINT

0: done

1: failed

### 6.3.1.1.11 FUNCTION CNF_GetHostName

Retrieves the HostName from the control unit.

Input / output parameters:

- sHostname: STRING

Output parameters:

- CNF_GetHostName: UDINT
  - 0: done
  - 1: failed

### 6.3.1.1.12 FUNCTION CNF_GetIpAddress

Input parameters:

- sInterface: STRING
  - Values: eth0, eth1

Input / output parameters:

- abyIpAddress: ARRAY [0..3] OF BYTE

Output parameters:

- CNF_GetIpAddress: UDINT
  - 0: done
  - 1: failed

### 6.3.1.1.13 FUNCTION CNF_GetMacAddress

Input parameters:

- sInterface: STRING
  - Values: eth0, eth1

Input / output parameters:

- abyMacAddress: ARRAY [0..5] OF BYTE

Output parameters:

- CNF_GetMacAddress: UDINT
  - 0: done
  - 1: failed

### 6.3.1.1.14 FUNCTION CNF_GetModuleName

Retrieves the name of the control unit module from the web interface.

Input / output parameters:

sModulName: STRING

Output parameters:

CNF_GetModuleName: UDINT

0: done

1: failed

### 6.3.1.1.15 FUNCTION CNF_GetModuleNumberString

Retrieves the article number of the control unit from the web interface.

Input / output parameters:

sModulNumber: STRING

Output parameters:

CNF_GetModuleNumberString: UDINT

0: done

1: failed

### 6.3.1.1.16 FUNCTION CNF_GetNetMask

Input parameters:

sInterface: STRING

Values: eth0, eth1

Input / output parameters:

abyNetMask: ARRAY [0..3] OF BYTE

Output parameters:

CNF_GetNetMask: UDINT

0: done

1: failed

### 6.3.1.1.17 FUNCTION CNF_GetSerialNumberString

Retrieves the article number of the control unit from the web interface.

Input / output parameters:

sSerial: STRING

Output parameters:

CNF_GetSerialNumberString: UDINT

0: done

1: failed

### 6.3.1.1.18 FUNCTION CNF_GetSkipUsbUpdateFlag

Checks whether or not the USB is to be updated automatically.

Output parameters:

CNF_GetSkipUsbUpdateFlag: UDINT

0: runs (does not skip) the USB update

1: skips the USB update

2: an error occurred

### 6.3.1.1.19 FUNCTION CNF_GetTimezone

Retrieves the time zone setting from the control unit.

Input / output parameters:

sTimezone: STRING

Output parameters:

CNF_GetTimezone: UDINT

0: done

1: failed

### 6.3.1.1.20 FUNCTION CNF_SetDnsAddress

Lets you set the DNS address. Use CNF_ApplySettings to permanently store the parameters.

Input parameters:

sInterface: STRING

dns0, dns1

abyDnsAddress: ARRAY [0..3] OF BYTE

DNS address

Output parameters:

CNF_SetDnsAddress: UDINT

0: done

1: failed

### 6.3.1.1.21 FUNCTION CNF_SetEthMode

Sets the EtherCAT mode. Use CNF_ApplySettings to permanently store the parameters.

Input parameters:

sInterface STRING

eth0, eth1

eMode: EthMode

INACTIVE := 0

STATIC :=1

DHCP:=2

ETHERCAT:=3

PROFINET:=4

Output parameters:

CNF_SetEthMode: UDINT

0: done

1: failed

### 6.3.1.1.22 FUNCTION CNF_SetGatewayAddress

Lets you set the gateway address. Use CNF_ApplySettings to permanently store the parameters.

Input parameters:

sInterface: STRING

eth0, eth1

abyGatewayAddress: ARRAY [0..3] OF BYTE

gateway address

Output parameters:

CNF_SetGatewayAddress: UDINT

0: done

1: failed

### 6.3.1.1.23 FUNCTION CNF_SetHostName

Lets you set the host name. Use CNF_ApplySettings to permanently store the parameters.

Input parameters:

sHostname: STRING

Host Name

Output parameters:

CNF_SetHostName: UDINT

0: done

1: failed

### 6.3.1.1.24 FUNCTION CNF_SetIpAddress

Lets you set the IP address. Use CNF_ApplySettings to permanently store the parameters.

Input parameters:

sInterface: STRING

eth0, eth1

abyIpAddress: ARRAY [0..3] OF BYTE

IP Address

Output parameters:

CNF_SetIpAddress: UDINT

0: done

1: failed

### 6.3.1.1.25 FUNCTION CNF_SetNetMask

Lets you set the net mask. Use CNF_ApplySettings to permanently store the parameters.

Input parameters:

- sInterface: STRING
  - eth0, eth1
- abyNetMask: ARRAY [0..3] OF BYTE
  - net mask

Output parameters:

- CNF_SetNetMask: UDINT
  - 0: done
  - 1: failed

### 6.3.1.1.26 FUNCTION CNF_SetSkipUsbUpdateFlag

Sets the xSkipUSBUpdateFlag. Use CNF_ApplySettings to permanently store the parameters.

Input parameters:

- xSkipUSBUpdate: BOOL
  - FALSE: runs the USB update
  - TRUE: skips the USB update

Output parameters:

- CNF_SetSkipUsbUpdateFlag: UDINT
  - 0: done
  - 1: failed

### 6.3.1.1.27 FUNCTION CNF_SetTimezone

Lets you set the time zone. Use CNF_ApplySettings to permanently store the parameters.

Input parameters:

- sTimezone: STRING
  - name of time zone

Output parameters:

- CNF_SetTimezone: UDINT
  - 0: done
  - 1: failed

### 6.3.1.2 Diagnosis

#### 6.3.1.2.1 FUNCTION DGN_GetDieTemperature

Takes a processor temperature reading and retrieves the minimum and maximum temperatures plus their time stamps.

Input parameters:

pTempValues: POINTER TO TempValues

Output parameters:

DGN_GetDieTemperature: UDINT

0: done

1: failed

#### 6.3.1.2.2 FUNCTION DGN_GetOperationHours

Output parameters:

DGN_GetOperationHours: DINT

-1: an error occurred

> 0: number of operating hours

### 6.3.1.3 Screen

#### 6.3.1.3.1 FUNCTION CNF_SaveScreenBrightness

#### 6.3.1.3.2 FUNCTION SCN_GetBrightness

#### 6.3.1.3.3 FUNCTION SCN_SetBrightness

### 6.3.1.4 Memory

#### 6.3.1.4.1 FUNCTION FS_DiskFree

Tells you how much free disk space a data volume has.

Input parameters:

sPath: STRING

memory path as string

Output parameters:

FS_DiskFree: UDINT

memory size in kbyte

#### 6.3.1.4.2 FUNCTION FS_DiskTotal

Tells you a data volume's total memory capacity.

Input parameters:

sPath: STRING

memory path as string

Output parameters:

FS_DiskFree: UDINT

memory size in kbyte

### 6.3.1.4.3 FUNCTION USB_GetMountStatus

Tells you whether a USB device has been properly mounted on the system. If not mounted properly, the USB device cannot be used.

Input parameters:

nUSBDevice: UINT

0..n (0=USB1, 1=USB2, …)

Output parameters:

USB_GetMountStatus:

0=device not connected

1=device connected

### 6.3.1.4.4 FUNCTION USB_GetPlugStatus

Tells you whether a USB device has been plugged in to a USB port.

Input parameters:

nUSBDevice: UINT

0..n (0=USB1, 1=USB2, …)

Output parameters:

USB_GetMountStatus:

0=device not plugged in

1=device plugged in

### 6.3.1.4.5 FUNCTION USB_MountDisk

Normally, USB devices are automatically mounted on the system. Use this function to manually mount a device. If not mounted properly, the USB device cannot be used.

Input parameters:

nUSBDevice: UINT

0..n (0=USB1, 1=USB2, …)

dwOptions: DWORD

0=default, see ENUM MountOptions for further options

Output parameters:

USB_MountDisk: INT

0: done

1: failed

### 6.3.1.4.6 FUNCTION USB_UMountDisk

Ends the system connection of a USB device. To avoid loss of data or damages to the file system, we recommend you to run this function before you unplug a USB device. Remember to close all open file and directory handles first.

Input parameters:

nUSBDevice: UINT

0..n (0=USB1, 1=USB2, …)

Output parameters:

USB_UMountDisk: INT

0: done

1: failed

### 6.3.1.5 Library Information

#### 6.3.1.5.1 FUNCTION GetLibVersion

#### 6.3.1.5.2 FUNCTION GetLibVersionNumber

#### 6.3.1.5.3 FUNCTION IsLibReleased

## 6.3.1.6 Data Types

### 6.3.1.6.1 ENUM ButtonResetMode

| | | | |
|---|---|---|---|
| COLD | INT | 0 | cold reset when button is triggered |
| WARM | INT | 1 | warm reset when button is triggered |

### 6.3.1.6.2 ENUM Errors

| | | | |
|---|---|---|---|
| ERR_OK | INT | 0 | No error |
| ERR_FAILED | INT | 1 | |
| ERR_PARAMETER | INT | 2 | Parameter invalid |
| ERR_NODEV | INT | 3 | Device invalid |
| ERR_UNDEFINED | INT | 255 | Unspecified error |

### 6.3.1.6.3 ENUM EthMode

| | | |
|---|---|---|
| INACTIVE | INT | 0 |
| STATIC | INT | 1 |
| DHCP | INT | 2 |
| ETHERCAT | INT | 3 |
| PROFINET | INT | 4 |

### 6.3.1.6.4 ENUM MountOptions

| | | | |
|---|---|---|---|
| MNT_NOSUID | DWORD | 1 | Ignore suid and sgid bits |
| MNT_NODEV | DWORD | 2 | Disallow access to device special files |
| MNT_NOEXEC | DWORD | 4 | Disallow program execution |
| MNT_NOATIME | DWORD | 8 | Do not update access times |
| MNT_NODIRATIME | DWORD | 16 | Do not update directory access times |
| MNT_RELATIME | DWORD | 32 | Update atime relative to mtime/ctime |
| MNT_RDONLY | DWORD | 64 | Mount read-only |
| MNT_SHRINKABLE | DWORD | 256 | |
| MNT_WRITE_HOLD | DWORD | 512 | |
| MNT_SHARED | DWORD | 4096 | change to shared |
| MNT_UNBINDABLE | DWORD | 8192 | change to unbindable |
| MNT_SHARED_MASK | DWORD | 8192 | |
| MNT_PROPAGATION_MASK | DWORD | 12288 | |
| MNT_INTERNAL | DWORD | 16384 | |
| MNT_LOCK_READONLY | DWORD | 4194304 | |
| MNT_LOCKED | DWORD | 8388608 | |

### 6.3.1.6.5 STRUCT TempValues

| | | |
|---|---|---|
| rCurrent | REAL | Currently measured temperature in degree celcius (°C) |
| rMax | REAL | Maximum measured temperature in degree celcius (°C) |
| sMaxTimestamp | STRING | String with timestamp at the time of maximum measured temperature |
| rMin | REAL | Minimum measured temperature in degree celcius (°C) |
| sMinTimestamp | STRING | String with timestamp at the time of minimum measured temperature |

## 6.3.2 Kuhnke iMX6 Onboard IO Library

### 6.3.2.1 Datentypen

### 6.3.2.1.1 STRUCT EVTPARAM_OnboardIo

| | | |
|---|---|---|
| xVal | BOOL | Internal input current state |
| uLiUsTimestamp | ULINT | Timestamp [µs] of the last edge |

### 6.3.2.1.2 VAR_GLOBAL CONSTANT EventIDs

Internal parameters

| | | |
|---|---|---|
| CMPID_CmpOnboardIo | DWORD | 16#1059210B |
| EVTPARAMID_CmpOnboardIo | WORD | 16#1 |
| EVTVERSION_CmpOnboardIo | WORD | 16#1 |
| EVT_IrqIo | DWORD | 16#10001 |

## 6.3.3 Systempriorities

For the design of real-time applications, it is important to have knowledge of system priorities.

Priorities sorted in descending order

- CODESYS Scheduler
- Event Task (e.g. interrupt input, if configured as event)
- EtherCAT
- CAN
- UART (RS232)
- PLC real-time task Prio 0
- PLC real-time task Prio 15
- Ethernet
- USB
- SD- card
- PLC normal Task Prio 16
- PLC normal Task Prio 31

## 6.4 Retain Variables

With version 1.25.0 (CODESYS 3.5 SP16 Patch40) retain variables can be declared on this controller with the keywords "RETAIN" or "PERSISTENT".

In older versions, the Persistence Manager of CODESYS must be used, see 6.4.2

### 6.4.1 Declaration of retain variables

Retain variables are declared by adding the keyword RETAIN in the declaration area. Example of declaration in the Global Variable List:

```
VAR_GLOBAL RETAIN
       udiCounter: UDINT;
END_VAR
```

To declare persistent variables, the object PersistentVars must be attached to the application. This global variable list contains the declaration of the persistent variables. For variables that are marked with the keyword PERSISTENT outside the persistence editor, instance paths are added there.

```
VAR_GLOBAL PERSISTENT RETAIN
       udiCounter: UDINT;
END_VAR
```

Retain and persistence variables differ in their lifetime:

| | Normale Variablen | RETAIN | PERSISTENT<br>RETAIN PERSISTENT<br>PERSISTENT RETAIN |
|---|---|---|---|
| Powerfail | 0 | X | X |
| Online-Change | X | X | X |
| Reset warm | 0 | X | X |
| Reset kalt | 0 | 0 | X |
| Laden | 0 | 0 | X (1) |
| Reset Ursprung | 0 | 0 | 0 |

X = Value of the variable is retained

0 = Variable is initialized

X (1) = Remains only if the structure within the persistent variable has not changed

## 6.4.2 Using the Persistence Manager

To store remanent variables the Pesistence Manager from the CODESYS application Composer can be used.

General information about the Persistence Manager can be found in the CODESYS Help:
https://help.codesys.com/webapp/f_application_composer_persistence_manager;product=core_Application_Composer;version=3.5.14.0

### 6.4.2.1 Activate "Modules" view

To use the Persistence Manager, the "Modules" view must be activated in CODESYS.

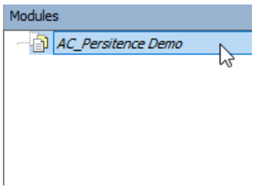
To do so, select "View" -> "Modules" from the menu

### 6.4.2.2 Add Module Library

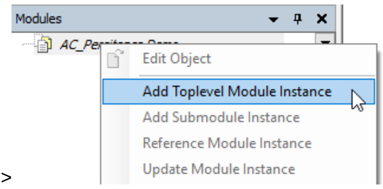
To do this, select "Composer" -> "Add module library to project" in the menu

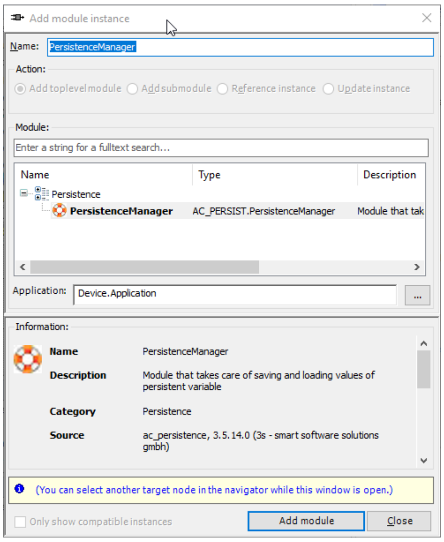
### 6.4.2.3 Add Persistence Manager

Under Modules add the "PersistenceManager" as top level instance.



-> right click ->





Add module instance

Name: PersistenceManager

Action:
● Add toplevel module ○ Add submodule ○ Reference instance ○ Update instance

Module:
Enter a string for a fulltext search...

| Name | Type | Description |
|---|---|---|
| Persistence | | |
| PersistenceManager | AC_PERSIST.PersistenceManager | Module that tak |

Application: Device.Application

Information:

| | |
|---|---|
| **Name** | PersistenceManager |
| **Description** | Module that takes care of saving and loading values of persistent variable |
| **Category** | Persistence |
| **Source** | ac_persistence, 3.5.14.0 (3s - smart software solutions gmbh) |

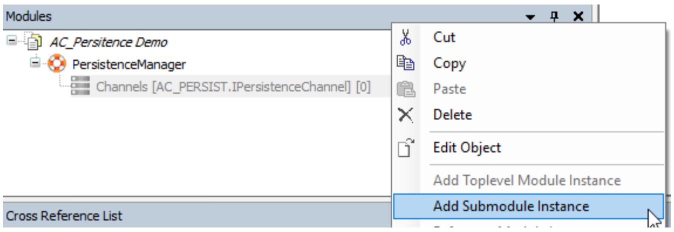
(You can select another target node in the navigator while this window is open.)

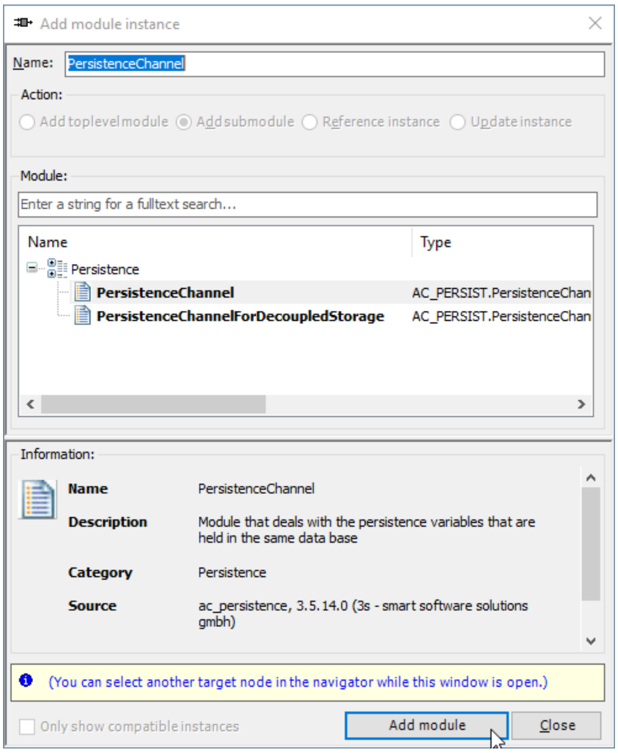
□ Only show compatible instances

The name can be adapted if necessary. Click on "Add module".

### 6.4.2.4 Define Persistence Channel

Now define a persistence channel under the PersistenceManager by right-clicking on "Channels" and selecting "Add submodule instance" from the context menu




Add module instance

Name: PersistenceChannel

Action:

○ Add toplevel module ● Add submodule ○ Reference instance ○ Update instance

Module:

Enter a string for a fulltext search...

| Name | Type |
|---|---|
| Persistence | |
| PersistenceChannel | AC_PERSIST.PersistenceChan |
| PersistenceChannelForDecoupledStorage | AC_PERSIST.PersistenceChan |

Information:

| | |
|---|---|
| **Name** | PersistenceChannel |
| **Description** | Module that deals with the persistence variables that are held in the same data base |
| **Category** | Persistence |
| **Source** | ac_persistence, 3.5.14.0 (3s - smart software solutions gmbh) |

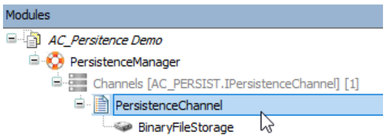
(You can select another target node in the navigator while this window is open.)

□ Only show compatible instances

The name can be adapted if necessary. If you use different persistence channels, we recommend to use a descriptive name. Click on "Add Module".

### 6.4.2.5 Parameterization of the Persistence Channel

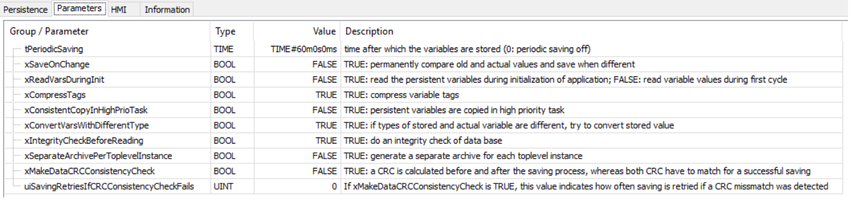
To parameterize the persistence channel, double-click on the desired entry in the module view:



Modules
- *AC_Persitence Demo*
  - PersistenceManager
    - Channels [AC_PERSIST.IPersistenceChannel] [1]
      - PersistenceChannel
        - BinaryFileStorage

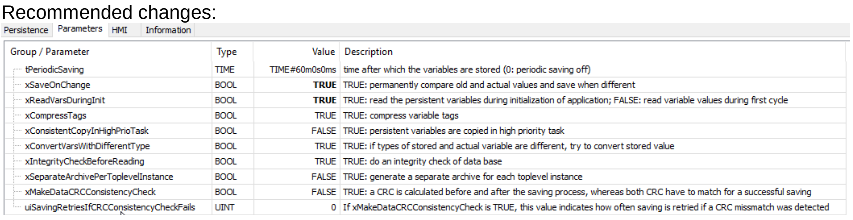
The module is opened in the Editing Area. In the "Parameters" tab you can adjust the settings to your needs.

Default parameter



| Group / Parameter | Type | Value | Description |
|---|---|---|---|
| tPeriodicSaving | TIME | TIME#60m0s0ms | time after which the variables are stored (0: periodic saving off) |
| xSaveOnChange | BOOL | FALSE | TRUE: permanently compare old and actual values and save when different |
| xReadVarsDuringInit | BOOL | FALSE | TRUE: read the persistent variables during initialization of application; FALSE: read variable values during first cycle |
| xCompressTags | BOOL | TRUE | TRUE: compress variable tags |
| xConsistentCopyInHighPrioTask | BOOL | FALSE | TRUE: persistent variables are copied in high priority task |
| xConvertVarsWithDifferentType | BOOL | TRUE | TRUE: if types of stored and actual variable are different, try to convert stored value |
| xIntegrityCheckBeforeReading | BOOL | TRUE | TRUE: do an integrity check of data base |
| xSeparateArchivePerToplevelInstance | BOOL | FALSE | TRUE: generate a separate archive for each toplevel instance |
| xMakeDataCRCConsistencyCheck | BOOL | FALSE | TRUE: a CRC is calculated before and after the saving process, whereas both CRC have to match for a successful saving |
| uiSavingRetriesIfCRCConsistencyCheckFails | UINT | 0 | If xMakeDataCRCConsistencyCheck is TRUE, this value indicates how often saving is retried if a CRC missmatch was detected |

Recommended changes:



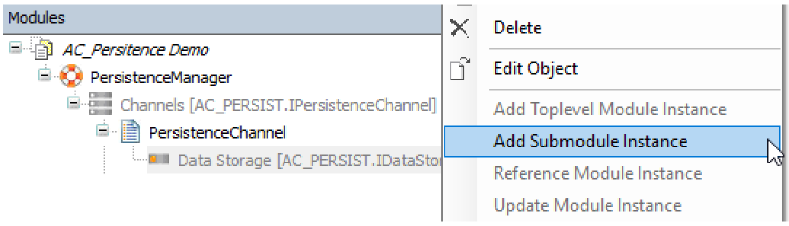
| Group / Parameter | Type | Value | Description |
|---|---|---|---|
| tPeriodicSaving | TIME | TIME#60m0s0ms | time after which the variables are stored (0: periodic saving off) |
| xSaveOnChange | BOOL | **TRUE** | TRUE: permanently compare old and actual values and save when different |
| xReadVarsDuringInit | BOOL | **TRUE** | TRUE: read the persistent variables during initialization of application; FALSE: read variable values during first cycle |
| xCompressTags | BOOL | TRUE | TRUE: compress variable tags |
| xConsistentCopyInHighPrioTask | BOOL | FALSE | TRUE: persistent variables are copied in high priority task |
| xConvertVarsWithDifferentType | BOOL | TRUE | TRUE: if types of stored and actual variable are different, try to convert stored value |
| xIntegrityCheckBeforeReading | BOOL | TRUE | TRUE: do an integrity check of data base |
| xSeparateArchivePerToplevelInstance | BOOL | FALSE | TRUE: generate a separate archive for each toplevel instance |
| xMakeDataCRCConsistencyCheck | BOOL | FALSE | TRUE: a CRC is calculated before and after the saving process, whereas both CRC have to match for a successful saving |
| uiSavingRetriesIfCRCConsistencyCheckFails | UINT | 0 | If xMakeDataCRCConsistencyCheck is TRUE, this value indicates how often saving is retried if a CRC missmatch was detected |

***Information***

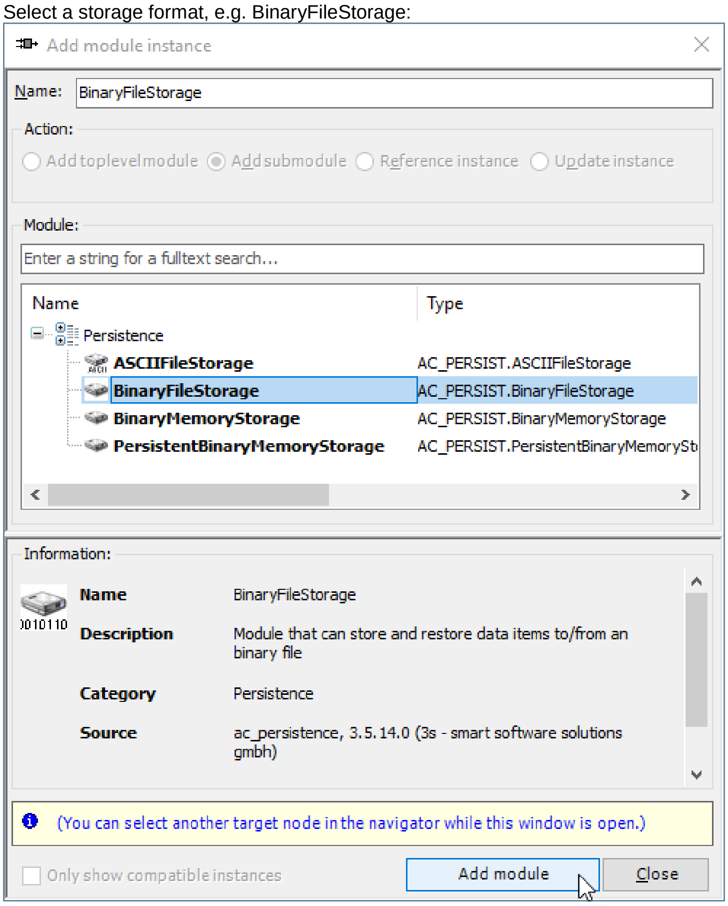
*xSaveOnChange: Use this setting with caution. Every change is followed by a write access to the flash memory.*

### 6.4.2.6 Configure data storage

Define the desired storage format of the Persistent Variables in the Data Storage submodule by right-clicking on "Data Storage" and selecting "Add Submodule Instance" from the context menu.


Modules
AC_Persitence Demo
PersistenceManager
Channels [AC_PERSIST.IPersistenceChannel]
PersistenceChannel
Data Storage [AC_PERSIST.IDataStor
Delete
Edit Object
Add Toplevel Module Instance
Add Submodule Instance
Reference Module Instance
Update Module Instance

Select a storage format, e.g. BinaryFileStorage:



Name: BinaryFileStorage

Action:
○ Add toplevel module ● Add submodule ○ Reference instance ○ Update instance

Module:
Enter a string for a fulltext search...

| Name | Type |
|---|---|
| Persistence | |
| ASCIIFileStorage | AC_PERSIST.ASCIIFileStorage |
| BinaryFileStorage | AC_PERSIST.BinaryFileStorage |
| BinaryMemoryStorage | AC_PERSIST.BinaryMemoryStorage |
| PersistentBinaryMemoryStorage | AC_PERSIST.PersistentBinaryMemorySt |

Information:

| | |
|---|---|
| Name | BinaryFileStorage |
| Description | Module that can store and restore data items to/from an binary file |
| Category | Persistence |
| Source | ac_persistence, 3.5.14.0 (3s - smart software solutions gmbh) |

(You can select another target node in the navigator while this window is open.)

□ Only show compatible instances

### 6.4.2.7 Declaration of variables

Set the following attribute before each variable that should be retentive:
{attribute 'ac_persist' := 'PersistenceChannel'}
The name, here 'PersistenceChannel' must match the persistence channel defined in 3.

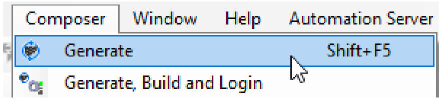
Example declaration of a variable:

```
VAR_GLOBAL
        {attribute 'ac_persist' := 'PersistenceChannel'}
        uiStartupCounter: UINT; // Counts the machine startups
END_VAR
```

To add the variable to the persistence channel, it must also be used in the project. The startups can be recorded in the PLC_PRG as follows, for example:

```
IF NOT xInit THEN
        uiStartupCounter := uiStartupCounter + 1;
        xInit := TRUE;
END_IF
```

Now the code must be generated, menu Composer -> Generate



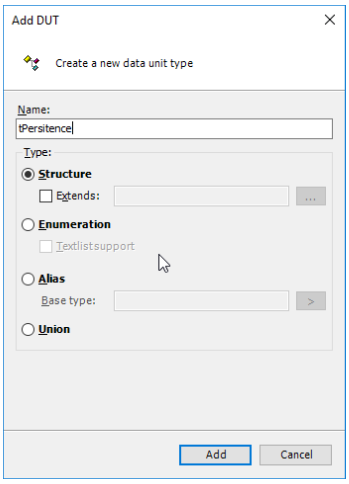
| | ***Information*** |
|---|---|
| | *In general, on systems with mass storage, it should be questioned how useful it is to use large amounts of retain data.* |

### 6.4.2.8 Persistent variables as data structure

We recommend to create a data structure for the persistent data when using the PersistenceManager. Then you have to set the attribute {attribute 'ac_persist' := 'PersistenceChannel'} only once in the variable declaration, so that all data from the data structure is remanent. We will be happy to provide you with an example project on request.

### Definition of a structure variable

Add a structure with a right click on the application -> Add Objects -> DUT.



Add DUT

Create a new data unit type

Name: tPersitence

Type:

● Structure

□ Extends:

○ Enumeration

□ Textlistsupport

○ Alias

Base type:

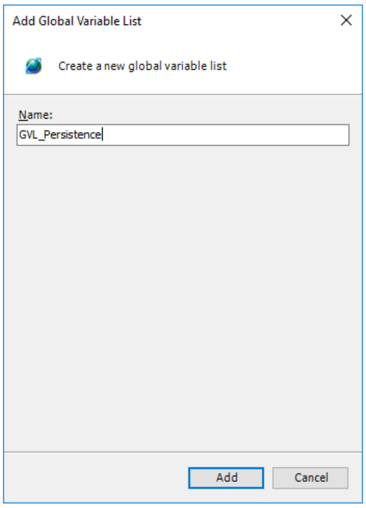
○ Union

Below you will find an example of the possible content of the data structure:

```
TYPE tPersistence :
STRUCT
        uiCounter: UINT; // Machine startup counter
        uiActState: UINT; // Actual machine state
        xModeAutomatic: BOOL; // Automatic mode
        xModeManual: BOOL; // Manual mode
END_STRUCT
END_TYPE
```

## Add a global variable list

Right-click on the application -> Add Objects -> Global Variable List to add a variable list for the remanent variables.



Name: GVL_Persistence

## Declaration of the retain data structure

Create a declaration for the data structure.

```
{attribute 'qualified_only'}
VAR_GLOBAL
	{attribute 'ac_persist' := 'PersistenceChannel'}
	Persistence: tPersistence;
END_VAR
```

## Using retain variables

The remanent variables can be used in program blocks as follows

```
IF NOT xInit THEN
	// Counts the machine startups
	GVL_Persistence.Persistence.uiCounter := GVL_Persistence.Persistence.uiCounter + 1;
	xInit := TRUE;
END_IF

IF NOT GVL_Persistence.Persistence.xModeAutomatic AND NOT GVL_Persistence.Persistence.xModeManual
THEN
	// If any mode is active, set the machine in manual mode
	GVL_Persistence.Persistence.xModeManual := TRUE;
END_IF
```

# 7 Appendix

## 7.1 Order Specifications

### 7.1.1 Basic Units

694: FIO Baureihe

x = 3: without Motion
x = 5: CODESYS Softmotion
x = 6: CODESYS Softmotion CNC & Robotics

y = 0: without Visualisization
y = 2: CODESYS Webvisualisization

z = 0: without expansion module
z = 1: Ethernet expansion module
z = 2: Profinet expansion module
z = 3: EtherCAT expansion module

16 = Prozessor platform

a = 0 (fix)

b = 0 (fix)

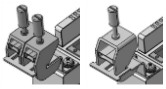
c = 0: without OPC UA
c = 1: with OPC UA

694 x y z 16 a b c

An up-to-date overview of the available combinations can be found on the Internet in our Product Finder at: https://productfinder.kuhnke.kendrion.com/de/modulare-sps/

Other combinations on request.

### 7.1.2 Accessories



- 694 412 01 Kuhnke FIO Shield terminal 2x8mm
- 694 412 02 Kuhnke FIO Shield terminal 1x14mm

## 7.2 Sales & Service

Please visit our Internet site to find a comprehensive overview of our sales and service network including all the relevant addresses. Feel free to also contact us at our headquarters in Malente/Germany.

### 7.2.1 Malente Headquarters

Kendrion Kuhnke Automation GmbH
Industrial Control Systems
Lütjenburger Straße 101
D-23714 Malente, Germany

| | |
|---|---|
| Phone: | +49 4523 402-0 |
| Fax: | +49 4523 402-201 |
| Email: | sales-ics@kendrion.com |
| Internet: | https://www.kendrion.com/en/products/control-technology |