Rexroth Bosch Group

Rexroth IndraLogic L40 04VRS System Description

R911323262
Edition 01

Operating and Programming Instructions

Title Rexroth IndraLogic L40 04VRS
System Description

Type of Documentation Operating and Programming Instructions

Document Typecode DOK-CONTRL-IC*L40**V04-AW01-EN-P

Internal File Reference RS-0795aa16c34a7b470a6846a000f1acf9-1-en-US-5

Purpose of Documentation This documentation provides an overview on the system components belonging to the control IndraLogic L40 and describes their project planning and programming.

Record of Revision

Edition	Release Date	Notes
120-0401-B346-01/EN	05.2008	First edition

Validity The specified data is for product description purposes only and may not be deemed to be guaranteed unless expressly confirmed in the contract. All rights are reserved with respect to the content of this documentation and the availability of the product.

Published by Bosch Rexroth AG
Bgm.-Dr.-Nebel-Str. 2 ■ 97816 Lohr am Main, Germany
Phone +49 (0)93 52/ 40-0 ■ Fax +49 (0)93 52/ 40-48 85
http://www.boschrexroth.com/
Dep. BRC/EAY2 (MK/PiGe)
Dep. BRC/EAH2 (BA)

Note This document has been printed on chlorine-free bleached paper.

Table of Contents

Page

Table of Contents

Page

Table of Contents

Page

Page

Page

Page

1 System Overview

1.1 Components

Control The control IndraLogic L40 DP consists of:

- Hardware: IndraControl L40 including connector set
- Firmware: IndraLogic L40 DP (on CompactFlash card)

The modular and scalable IndraControl L40 hardware platform can be used in combination with the IndraLogic L40 DP firmware for PLC applications.

Software The "IndraWorks Logic" software serves to commission and project the IndraLogic L40 DP. It consists of the following components:

- **IndraWorks**: Project planning, configuration
- **IndraLogic**: PLC programming
- **IndraWorks HMI**: Visualization and user interface as well as diagnostic functions (ProVi)
- **IndraWorks WinStudio**: Engineering tool to create user screens for IndraWorks HMI
- **IndraLogic L40 DP TSP**: Target system files (Target Support Package) to edit the IndraLogic L40 DP using IndraWorks and IndraLogic
- **Target Manager**: Tool for TSP data administration and installation, Inline module description files and libraries

All components are automatically installed.

1.2 Related Documentation

No.	Title	Identification
/1/	Rexroth IndraControl L40; Project Planning Manual	DOK-CONTRL-IC*L40*****-PR..-EN-P
/2/	PLC Programming with Rexroth IndraLogic; Operating and Programming Instructions	DOK-CONTRL-IL**PRO*V02-AW..-EN-P
/3/	Rexroth Inline Profibus DP; Application Manual	DOK-CONTRL-R-IL*PBSSYS-AW..-EN-P
/4/	Rexroth Inline Profibus DP Terminal and Module Supply; Functional Description	DOK-CONTRL-R-IL*PB*-BK-FK..-EN-P
/5/	Rexroth IndraWorks Engineering; Operating and Programming Instructions	DOK-IWORKS-ENGINEE*V..-AW..-EN-P
/6/	Rexroth WinStudio; Overall View	DOK-CONTRL-WIS*PC**V06-KB..-EN-P
/7/	Rexroth PLCopen Function Blocks For Field Bus Drives; Application Manual	DOK-CONTRL-PLCOPENFB*D-AW..-EN-P

Fig.1-1: Related documentation

2 Important Instructions on Use

2.1 Appropriate Use

2.1.1 Introduction

Rexroth products represent state-of-the-art developments and manufacturing. They are tested prior to delivery to ensure operational safety and reliability.

⚠ WARNING

Physical injury and material damage might result from inappropriate use of the products!

The products are designed for use in an industrial environment and may therefore only be used for the intended purpose. If they are not used as intended, situations causing personal injury as well as material damage can occur.

☞ Rexroth disclaims as manufacturer any warranty, liability or damages occurring due to inappropriate use of the products. Furthermore, Rexroth is not paying any compensation. The user is responsible for any risks resulting from inappropriate use of the products.

Before using Rexroth products, the following requirements must be met to ensure appropriate use of the products:

- Anyone handling one of the Rexroth products in any way has to read and understand the respective safety-related guidelines as well as the instructions on appropriate use.
- Hardware products have to remain in their original state, in other words, modifications regarding their design are not allowed. Software products must not be decompiled and their source codes must not be modified.
- Damaged or faulty products must not be implemented or put into operation.
- It must be ensured that the products are installed as specified in the documentation.

2.1.2 Areas of Use and Application

The IndraLogic L40 DP from Rexroth is intended for logic applications.

It can be necessary to connect additional sensors and actuators to control and monitor the IndraLogic L40 DP.

☞ The IndraLogic L40 DP may only be used with the accessories and add-on components specified in this documentation. Components that are not mentioned explicitly must neither be mounted nor connected. The same is applicable for cables and wires.

Operation may only be carried out in the component configurations and combinations specified and with the software and firmware specified in the respective functional description.

Each drive control device has to be programmed before commissioning so that the motor executes the application-specific functions.

Typical areas of application of the IndraLogic L40 DP:

- [Handling systems and assembly systems]
- [Packaging and processing machines]

- [Printing machines and paper processing machines]
- [Machine tools]

The IndraLogic L40 DP may only be operated under the assembly and installation conditions, in the position of application and under the ambient conditions (temperature, degree of protection, humidity, EMC, etc.) specified in this documentation.

2.2 Inappropriate Use

Applications of IndraLogic L40 DP that are not within the specified areas of application or under operating conditions deviating from the operating conditions and technical data specified in the documentation are considered as "inappropriate".

The IndraLogic L40 DP must not be used if ...

- it is exposed to operating conditions that do not fulfill the ambient conditions specified (for example, operation under water, under extreme temperature fluctuations or extreme maximum temperatures is not allowed);
- Rexroth has not explicitly released the intended applications – please also note the general statements in the general safety-related guidelines;
- it is used in household devices or devices belonging to categories 1 to 7 and 10 specified in Appendix IA of the Directive 2002/96/EC ("WEEE").

3 Safety Instructions for Electric Drives and Controls

3.1 Safety Instructions - General Information

3.1.1 Using the Safety Instructions and Passing them on to Others

Do not attempt to install or commission this device without first reading all documentation provided with the product. Read and understand these safety instructions and all user documentation prior to working with the device. If you do not have the user documentation for the device, contact your responsible Bosch Rexroth sales representative. Ask for these documents to be sent immediately to the person or persons responsible for the safe operation of the device.

If the device is resold, rented and/or passed on to others in any other form, these safety instructions must be delivered with the device in the official language of the user's country.

⚠ WARNING

Improper use of these devices, failure to follow the safety instructions in this document or tampering with the product, including disabling of safety devices, may result in material damage, bodily harm, electric shock or even death!

Observe the safety instructions!

3.1.2 How to Employ the Safety Instructions

Read these instructions before initial commissioning of the equipment in order to eliminate the risk of bodily harm and/or material damage. Follow these safety instructions at all times.

- Bosch Rexroth AG is not liable for damages resulting from failure to observe the warnings provided in this documentation.
- Read the operating, maintenance and safety instructions in your language before commissioning the machine. If you find that you cannot completely understand the documentation for your product, please ask your supplier to clarify.
- Proper and correct transport, storage, assembly and installation, as well as care in operation and maintenance, are prerequisites for optimal and safe operation of this device.
- Only assign trained and qualified persons to work with electrical installations:
    - Only persons who are trained and qualified for the use and operation of the device may work on this device or within its proximity. The persons are qualified if they have sufficient knowledge of the assembly, installation and operation of the product, as well as an understanding of all warnings and precautionary measures noted in these instructions.
    - Furthermore, they must be trained, instructed and qualified to switch electrical circuits and devices on and off in accordance with technical safety regulations, to ground them and to mark them according to the requirements of safe work practices. They must have adequate safety equipment and be trained in first aid.
- Only use spare parts and accessories approved by the manufacturer.

- Follow all safety regulations and requirements for the specific application as practiced in the country of use.
- The devices have been designed for installation in industrial machinery.
- The ambient conditions given in the product documentation must be observed.
- Only use safety-relevant applications that are clearly and explicitly approved in the Project Planning Manual. If this is not the case, they are excluded. Safety-relevant are all such applications which can cause danger to persons and material damage.
- The information given in the documentation of the product with regard to the use of the delivered components contains only examples of applications and suggestions.

  The machine and installation manufacturer must

  - make sure that the delivered components are suited for his individual application and check the information given in this documentation with regard to the use of the components,
  - make sure that his application complies with the applicable safety regulations and standards and carry out the required measures, modifications and complements.
- Commissioning of the delivered components is only permitted once it is sure that the machine or installation in which they are installed complies with the national regulations, safety specifications and standards of the application.
- Operation is only permitted if the national EMC regulations for the application are met.
- The instructions for installation in accordance with EMC requirements can be found in the section on EMC in the respective documentation (Project Planning Manuals of components and system).

  The machine or installation manufacturer is responsible for compliance with the limiting values as prescribed in the national regulations.
- Technical data, connection and installation conditions are specified in the product documentation and must be followed at all times.

National regulations which the user must take into account

- European countries: according to European EN standards
- United States of America (USA):
  - National Electrical Code (NEC)
  - National Electrical Manufacturers Association (NEMA), as well as local engineering regulations
  - regulations of the National Fire Protection Association (NFPA)
- Canada: Canadian Standards Association (CSA)
- Other countries:
  - International Organization for Standardization (ISO)
  - International Electrotechnical Commission (IEC)

3.1.3 Explanation of Warning Symbols and Degrees of Hazard Seriousness

The safety instructions describe the following degrees of hazard seriousness. The degree of hazard seriousness informs about the consequences resulting from non-compliance with the safety instructions:

Warning symbol	Signal word	Degree of hazard seriousness acc. to ANSI Z 535.4-2002
⚠	Danger	Death or severe bodily harm will occur.
⚠	Warning	Death or severe bodily harm may occur.
⚠	Caution	Minor or moderate bodily harm or material damage may occur.

Fig.3-1: Hazard classification (according to ANSI Z 535)

3.1.4 Hazards by Improper Use

⚠ DANGER

High electric voltage and high working current! Risk of death or severe bodily injury by electric shock!

Observe the safety instructions!

⚠ DANGER

Dangerous movements! Danger to life, severe bodily harm or material damage by unintentional motor movements!

Observe the safety instructions!

⚠ WARNING

High electric voltage because of incorrect connection! Risk of death or bodily injury by electric shock!

Observe the safety instructions!

⚠ WARNING

Health hazard for persons with heart pacemakers, metal implants and hearing aids in proximity to electrical equipment!

Observe the safety instructions!

⚠ CAUTION

Hot surfaces on device housing! Danger of injury! Danger of burns!

Observe the safety instructions!

⚠ CAUTION

Risk of injury by improper handling! Risk of bodily injury by bruising, shearing, cutting, hitting or improper handling of pressurized lines!

Observe the safety instructions!

CAUTION

Risk of injury by improper handling of batteries!

Observe the safety instructions!

3.2 Instructions with Regard to Specific Dangers

3.2.1 Protection Against Contact with Electrical Parts and Housings

☞ This section concerns devices and drive components with voltages of **more than 50 Volt**.

Contact with parts conducting voltages above 50 Volts can cause personal danger and electric shock. When operating electrical equipment, it is unavoidable that some parts of the devices conduct dangerous voltage.

DANGER

High electrical voltage! Danger to life, electric shock and severe bodily injury!

- Only those trained and qualified to work with or on electrical equipment are permitted to operate, maintain and repair this equipment.
- Follow general construction and safety regulations when working on power installations.
- Before switching on the device, the equipment grounding conductor must have been non-detachably connected to all electrical equipment in accordance with the connection diagram.
- Do not operate electrical equipment at any time, even for brief measurements or tests, if the equipment grounding conductor is not permanently connected to the mounting points of the components provided for this purpose.
- Before working with electrical parts with voltage potentials higher than 50 V, the device must be disconnected from the mains voltage or power supply unit. Provide a safeguard to prevent reconnection.
- With electrical drive and filter components, observe the following:

  Wait **30 minutes** after switching off power to allow capacitors to discharge before beginning to work. Measure the electric voltage on the capacitors before beginning to work to make sure that the equipment is safe to touch.
- Never touch the electrical connection points of a component while power is turned on. Do not remove or plug in connectors when the component has been powered.
- Install the covers and guards provided with the equipment properly before switching the device on. Before switching the equipment on, cover and safeguard live parts safely to prevent contact with those parts.
- A residual-current-operated circuit-breaker or r.c.d. cannot be used for electric drives! Indirect contact must be prevented by other means, for example, by an overcurrent protective device according to the relevant standards.
- Secure built-in devices from direct touching of electrical parts by providing an external housing, for example a control cabinet.

☞ For electrical drive and filter components with voltages of **more than 50 volts**, observe the following additional safety instructions.

DANGER

High housing voltage and high leakage current! Risk of death or bodily injury by electric shock!

- Before switching on, the housings of all electrical equipment and motors must be connected or grounded with the equipment grounding conductor to the grounding points. This is also applicable before short tests.
- The equipment grounding conductor of the electrical equipment and the devices must be non-detachably and permanently connected to the power supply unit at all times. The leakage current is greater than 3.5 mA.
- Over the total length, use copper wire of a cross section of a minimum of 10 mm^2 for this equipment grounding connection!
- Before commissioning, also in trial runs, always attach the equipment grounding conductor or connect to the ground wire. Otherwise, high voltages may occur at the housing causing electric shock.

3.2.2 Protection Against Electric Shock by Protective Extra-Low Voltage

Protective extra-low voltage is used to allow connecting devices with basic insulation to extra-low voltage circuits.

All connections and terminals with voltages between 5 and 50 volts at Rexroth products are PELV systems.[1] It is therefore allowed to connect devices equipped with basic insulation (such as programming devices, PCs, notebooks, display units) to these connections and terminals.

WARNING

High electric voltage by incorrect connection! Risk of death or bodily injury by electric shock!

If extra-low voltage circuits of devices containing voltages and circuits of more than 50 volts (e.g. the mains connection) are connected to Rexroth products, the connected extra-low voltage circuits must comply with the requirements for PELV.[2]

3.2.3 Protection Against Dangerous Movements

Dangerous movements can be caused by faulty control of connected motors. Some common examples are:

- improper or wrong wiring of cable connections
- incorrect operation of the equipment components
- wrong input of parameters before operation
- malfunction of sensors, encoders and monitoring devices
- defective components
- software or firmware errors

Dangerous movements can occur immediately after equipment is switched on or even after an unspecified time of trouble-free operation.

[1] *"Protective Extra-Low Voltage"*

[2] *"Protective Extra-Low Voltage"*

The monitoring in the drive components will normally be sufficient to avoid faulty operation in the connected drives. Regarding personal safety, especially the danger of bodily harm and material damage, this alone cannot be relied upon to ensure complete safety. Until the integrated monitoring functions become effective, it must be assumed in any case that faulty drive movements will occur. The extent of faulty drive movements depends upon the type of control and the state of operation.

DANGER

Dangerous movements! Danger to life, risk of injury, severe bodily harm or material damage!

- Ensure personal safety by means of qualified and tested higher-level monitoring devices or measures integrated in the installation.

  These measures have to be provided for by the user according to the specific conditions within the installation and a hazard and fault analysis. The safety regulations applicable for the installation have to be taken into consideration. Unintended machine motion or other malfunction is possible if safety devices are disabled, bypassed or not activated.

To avoid accidents, bodily harm and/or material damage:

- Keep free and clear of the machine's range of motion and moving parts. Possible measures to prevent people from accidentally entering the machine's range of motion:
  - use safety fences
  - use safety guards
  - use protective coverings
  - install light curtains or light barriers
- Fences and coverings must be strong enough to resist maximum possible momentum.
- Mount the emergency stop switch in the immediate reach of the operator. Verify that the emergency stop works before startup. Don't operate the device if the emergency stop is not working.
- Isolate the drive power connection by means of an emergency stop circuit or use a safety related starting lockout to prevent unintentional start.
- Make sure that the drives are brought to a safe standstill before accessing or entering the danger zone.
- Additionally secure vertical axes against falling or dropping after switching off the motor power by, for example:
  - mechanically securing the vertical axes,
  - adding an external braking/ arrester/ clamping mechanism or
  - ensuring sufficient equilibration of the vertical axes.
- The standard equipment motor brake or an external brake controlled directly by the drive controller are **not sufficient to guarantee personal safety**!
- Disconnect electrical power to the equipment using a master switch and secure the switch against reconnection for:
  - maintenance and repair work
  - cleaning of equipment
  - long periods of discontinued equipment use
- Prevent the operation of high-frequency, remote control and radio equipment near electronics circuits and supply leads. If the use of such devices cannot be avoided, verify the system and the installation for possible malfunctions in all possible positions of normal use before initial startup. If necessary, perform a special electromagnetic compatibility (EMC) test on the installation.

3.2.4 Protection Against Magnetic and Electromagnetic Fields During Operation and Mounting

Magnetic and electromagnetic fields generated by current-carrying conductors and permanent magnets in motors represent a serious personal danger to those with heart pacemakers, metal implants and hearing aids.

⚠ WARNING

Health hazard for persons with heart pacemakers, metal implants and hearing aids in proximity to electrical equipment!

- Persons with heart pacemakers and metal implants are not permitted to enter following areas:
  - Areas in which electrical equipment and parts are mounted, being operated or commissioned.
  - Areas in which parts of motors with permanent magnets are being stored, repaired or mounted.
- If it is necessary for somebody with a pacemaker to enter such an area, a doctor must be consulted prior to doing so. The noise immunity of present or future implanted heart pacemakers differs greatly so that no general rules can be given.
- Those with metal implants or metal pieces, as well as with hearing aids, must consult a doctor before they enter the areas described above. Otherwise health hazards may occur.

3.2.5 Protection Against Contact with Hot Parts

⚠ CAUTION

Hot surfaces at motor housings, on drive controllers or chokes! Danger of injury! Danger of burns!

- Do not touch surfaces of device housings and chokes in the proximity of heat sources! Danger of burns!
- Do not touch housing surfaces of motors! Danger of burns!
- According to the operating conditions, temperatures can be **higher than 60 °C**, 140°F during or after operation.
- Before accessing motors after having switched them off, let them cool down for a sufficiently long time. Cooling down can require **up to 140 minutes**! Roughly estimated, the time required for cooling down is five times the thermal time constant specified in the Technical Data.
- After switching drive controllers or chokes off, wait 15 minutes to allow them to cool down before touching them.
- Wear safety gloves or do not work at hot surfaces.
- For certain applications, the manufacturer of the end product, machine or installation, according to the respective safety regulations, has to take measures to avoid injuries caused by burns in the end application. These measures can be, for example: warnings, guards (shielding or barrier), technical documentation.

3.2.6 Protection During Handling and Mounting

In unfavorable conditions, handling and mounting certain parts and components in an improper way can cause injuries.

CAUTION

Risk of injury by improper handling! Bodily injury by bruising, shearing, cutting, hitting!

- Observe the general construction and safety regulations on handling and mounting.
- Use suitable devices for mounting and transport.
- Avoid jamming and bruising by appropriate measures.
- Always use suitable tools. Use special tools if specified.
- Use lifting equipment and tools in the correct manner.
- If necessary, use suitable protective equipment (for example safety goggles, safety shoes, safety gloves).
- Do not stand under hanging loads.
- Immediately clean up any spilled liquids because of the danger of skidding.

3.2.7 Battery Safety

Batteries consist of active chemicals enclosed in a solid housing. Therefore, improper handling can cause injury or material damage.

CAUTION

Risk of injury by improper handling!

- Do not attempt to reactivate low batteries by heating or other methods (risk of explosion and cauterization).
- Do not recharge the batteries as this may cause leakage or explosion.
- Do not throw batteries into open flames.
- Do not dismantle batteries.
- When replacing the battery/batteries do not damage electrical parts installed in the devices.
- Only use the battery types specified by the manufacturer.

☞ Environmental protection and disposal! The batteries contained in the product are considered dangerous goods during land, air, and sea transport (risk of explosion) in the sense of the legal regulations. Dispose of used batteries separate from other waste. Observe the local regulations in the country of assembly.

3.2.8 Protection Against Pressurized Systems

According to the information given in the Project Planning Manuals, motors cooled with liquid and compressed air, as well as drive controllers, can be partially supplied with externally fed, pressurized media, such as compressed air, hydraulics oil, cooling liquids and cooling lubricating agents. Improper handling of the connected supply systems, supply lines or connections can cause injuries or material damage.

⚠ CAUTION

Risk of injury by improper handling of pressurized lines!

- Do not attempt to disconnect, open or cut pressurized lines (risk of explosion).
- Observe the respective manufacturer's operating instructions.
- Before dismounting lines, relieve pressure and empty medium.
- Use suitable protective equipment (for example safety goggles, safety shoes, safety gloves).
- Immediately clean up any spilled liquids from the floor.

☞ Environmental protection and disposal! The agents used to operate the product might not be economically friendly. Dispose of ecologically harmful agents separately from other waste. Observe the local regulations in the country of assembly.

4 Project Planning and Programming

4.1 Requirements and Overview of the Proceeding

Requirements

- The IndraWorks Logic package is installed.
- The target system information (Target Support Package) of the IIndra-Logic L40 DP are available in IndraWorks. All relevant data are automatically generated during the installation of IndraWorks.
- IndraWorks Engineering is started.
- An IndraWorks project exists and is indicated in the project explorer.

☞ For detailed information on the user interface and the handling of IndraWorks refer to the IndraWorks documentation /5/ or online help.

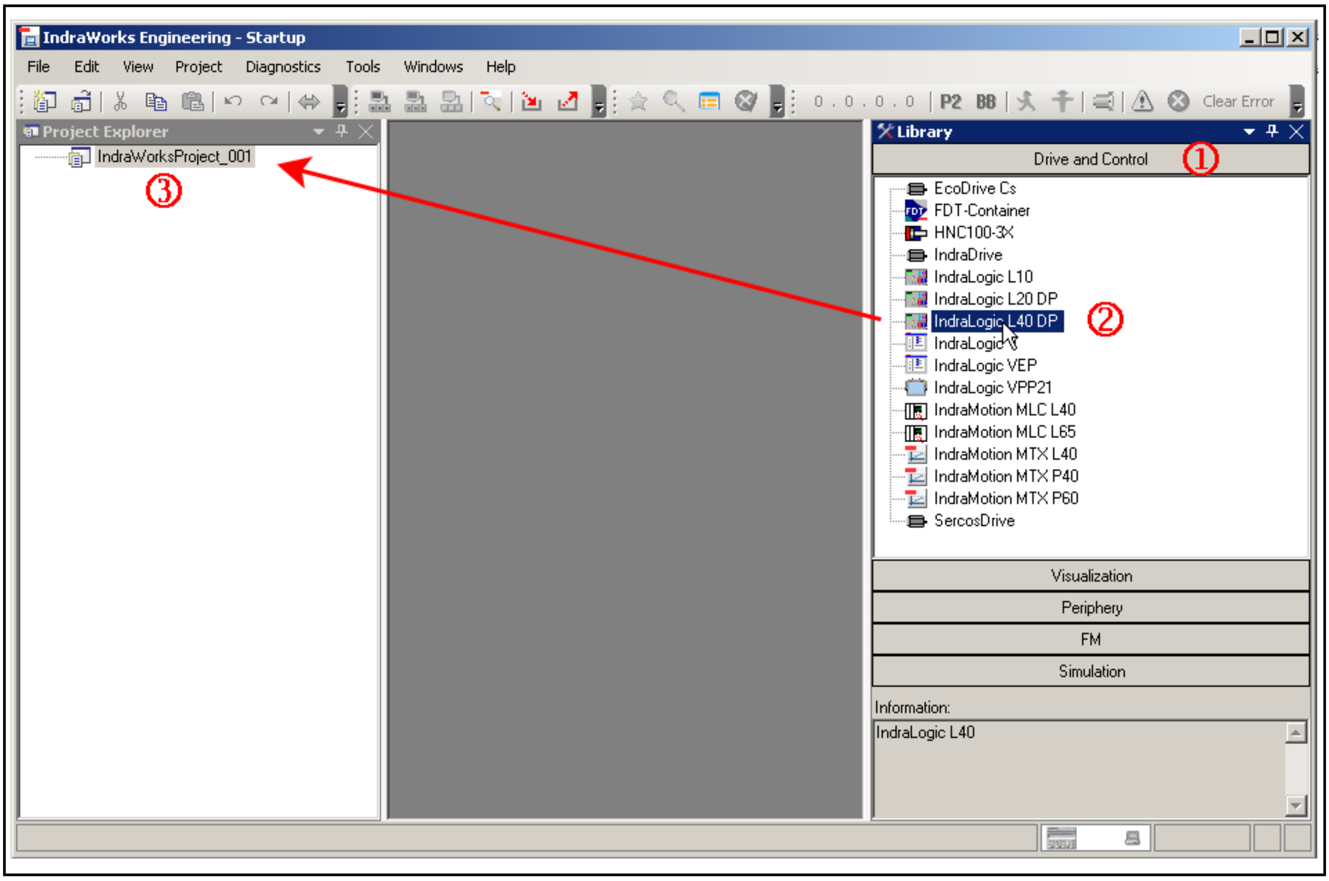
Required Steps

1. Creating a new IndraLogic L40 DP device with IndraWorks.
2. Defining basic settings with IndraWorks.
3. Defining field bus and I/O configuration with IndraWorks.
4. PLC programming with IndraLogic.
5. Loading configuration and PLC program data into the hardware of the IndraLogic L40 DP and activating them with IndraWorks.

4.2 Creating a New IndraLogic L40 Device

Drag the "IndraLogic L40 DP" device from the "Drive and Control" library to the desired project in the project explorer. Alternatively, you can also use the **Copy** and **Paste** functions of the particular context menu (right mouse button).

☞ For more information, please refer to the IndraWorks /5/ documentation or online help.



① "Drive and Control" library

② "IndraLogic L40" device

③ Project explorer

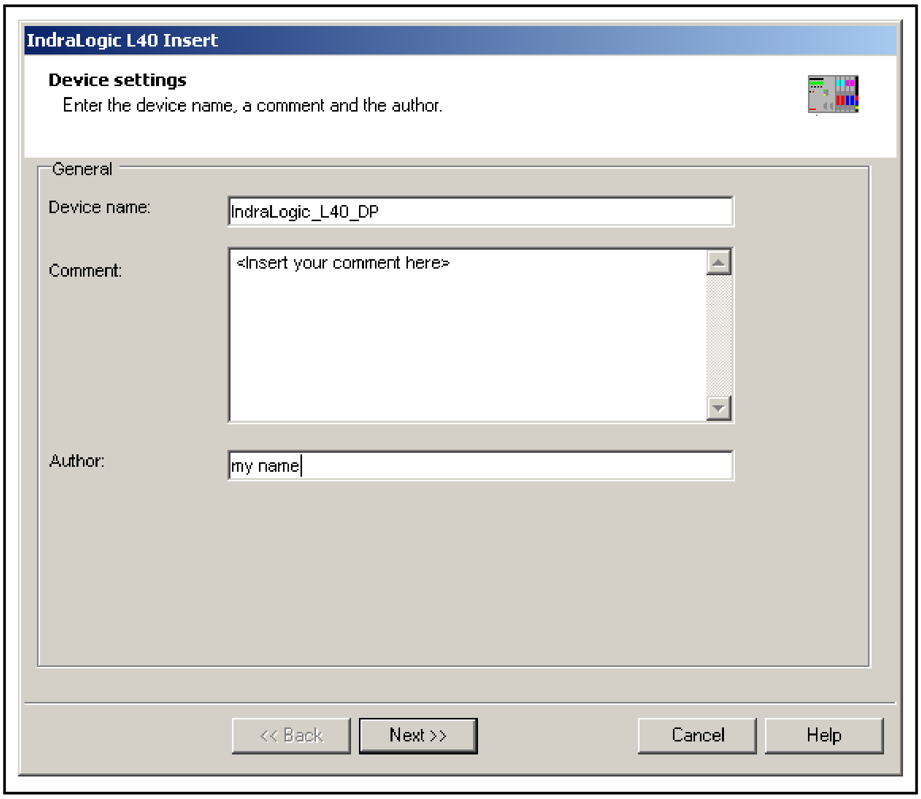
Fig.4-1: Insert the IndraLogic L40 in the project explorer

Then a wizard starts automatically to guide you through the definition of the necessary basic settings.

4.3 Specifying Basic Settings (Wizard-Guided)

4.3.1 Device Settings

Enter general information on the inserted device.



IndraLogic L40 Insert

Device settings

Enter the device name, a comment and the author.

General

Device name: IndraLogic_L40_DP

Comment: <Insert your comment here>

Author: my name

Fig.4-2: Device Settings

Device Name Enter any device name desired. The default setting of this field is the name specified in the library. The name entered here will be device name displayed in the project explorer.

Comment Enter any comment to describe the device in detail.

Author Enter the project author's name.

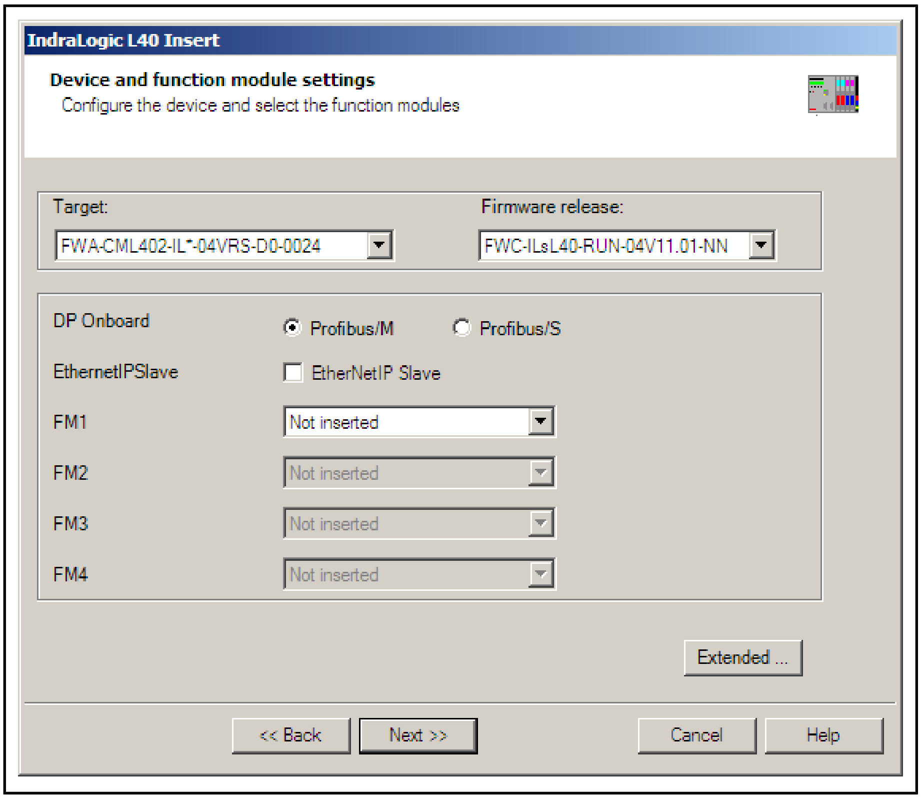
☞ These details can still be changed later, please refer to chapter 4.3.5 "Modifying Basic Settings" on page 24.

Confirm the settings with "Next >>".

4.3.2 Device and Function Module Settings

General Information

Select the firmware used for the device. Depending on the selected firmware, additional settings concerning the existing field bus interfaces (Profibus, Ethernet/IP) can be made.



IndraLogic L40 Insert

Device and function module settings

Configure the device and select the function modules

Target: FWA-CML402-IL*-04VRS-D0-0024

Firmware release: FWC-ILsL40-RUN-04V11.01-NN

DP Onboard ● Profibus/M ○ Profibus/S

EthernetIPSlave □ EtherNetIP Slave

FM1 Not inserted

FM2 Not inserted

FM3 Not inserted

FM4 Not inserted

Fig.4-3: Entering the firmware version and the settings for the field bus interfaces

☞ After integrating the new IndraLogic L40 DP device to the project explorer, the "version" and the "DP onboard" setting cannot be changed!

Target and Firmware Release

Select the **target system version** and the **firmware** for this version. This also affects further settings provided in this dialog.

DP Onboard

"**Profibus/M**": If you select this option, subordinate Profibus DP slaves will be coupled to the IndraLogic L40 DP via the onboard Profibus DP interface (X7P). In this case, the IndraLogic L40 DP will be the master of the slaves connected there.

"**Profibus/S**": If you select this option, the IndraLogic L40 DP will be a subordinate Profibus DP slave coupled to a superordinate Profibus DP master via the onboard Profibus DP interface (X7P).

Ethernet/IP Slave

Activate this option to use the IndraLogic L40 DP via the onboard Ethernet interface (X7E) as follows:

- As **CIP data server**(CIP: Communication Industrial Protocol), Allows simultaneous access for a maximum of 6 CIP data clients to enabled variables of the IndraLogic L40 DP via acyclic DataTable services ("Unconnected Explicit Messaging" and "Class 3 Connected Messaging").
- As **CIP data client**: Allows access to enabled variables via the "RIL_EtherNetIP" library of a CIP data server via acyclic DataTable services.
- As **Ethernet/IP slave** (target): Allows cyclic exchange of I/O data blocks with a superordinate control (master/originator) via an "Exclusive Owner connection" (transport class 1) and a further "Listen Only connection" (transport class 1).

FMx Select the desired **Function module** coupled to IndraLogic L40 DP. Take the numbering FM1 to FM4 into consideration which has to correspond to the switch positions for the module slot numbers and the function module position.

The following function modules are permissible for IndraLogic L40 DP 04VRS:

- **DeviceNet/M**: DeviceNet master
- **Profibus/M**: Profibus DP master
- **Fast I/O**: Fast inputs and outputs

Previous firmware versions of IndraLogic L40 DP do not support function modules.

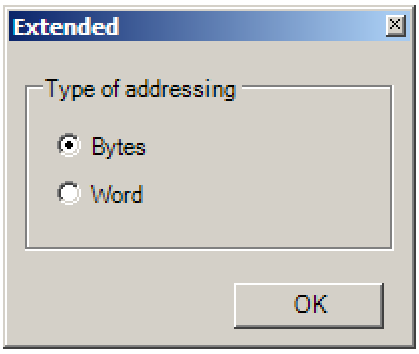
Several DP masters can be used but max one DeviceNet master. Otherwise, arbitrary function module combinations are possible.

Type of Addressing

Click on the "Extended..." button to set the desired I/O addressing type (bytes or word).

☞ Basic settings of the I/O addressing type:

- Lower than firmware version 03 VRS: Word
- Higher than firmware version 03 VRS: Bytes



Extended

Type of addressing

● Bytes

○ Word

Fig.4-4: Setting the I/O addressing type

Select the setting which corresponds to the type of addressing in the PLC programs to be used. For example, the following applies to word addressing: output byte 10 (QB10) is addressed by "QW5". In contrast, the following applies to byte addressing: output byte 10 (QB10) is addressed by "QW10".

☞ Once the new IndraLogic L40 DP device is included in the project explorer, the type of addressing can no longer be modified!

Confirm the settings with "Next >>".

This opens the dialog for defining the communication settings.

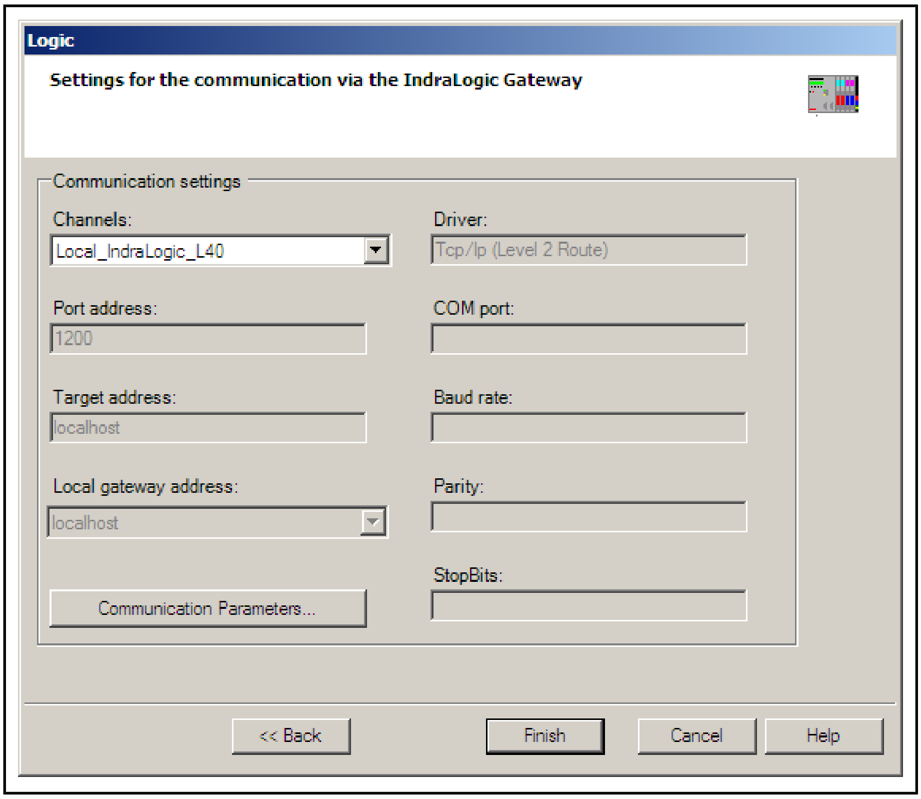
4.3.3 Communication Settings

General Information

Define the settings for communication between IndraWorks and the IndraLogic L40 DP. Opening of this dialog automatically starts the IndraLogic gateway server used to enable communication. If already provided in the gateway server, communication settings can be selected from the "Channels" dropdown list.

☞ If there is no entry yet, a dialog to create a communication channel appears automatically. If there is no suitable entry, you can either create a new or edit an existing communication channel. Click on the "Communication parameters..."button.

Proceed as described in chapter "Communication Parameters" on page 20.



Logic

Settings for the communication via the IndraLogic Gateway

Communication settings

Channels: Local_IndraLogic_L40

Driver: Tcp/Ip (Level 2 Route)

Port address: 1200

COM port:

Target address: localhost

Baud rate:

Local gateway address: localhost

Parity:

StopBits:

Fig.4-5: Communication settings of the IndraLogic L40

The gray fields show the parameters of the entry selected from "Channels".

The communications settings have to correspond to the actual setting of the IndraLogic L40 DP, also refer to chapter 7.4.3 "Default Menu " on page 188 or chapter 7.5.5 "Network Configuration " on page 198.

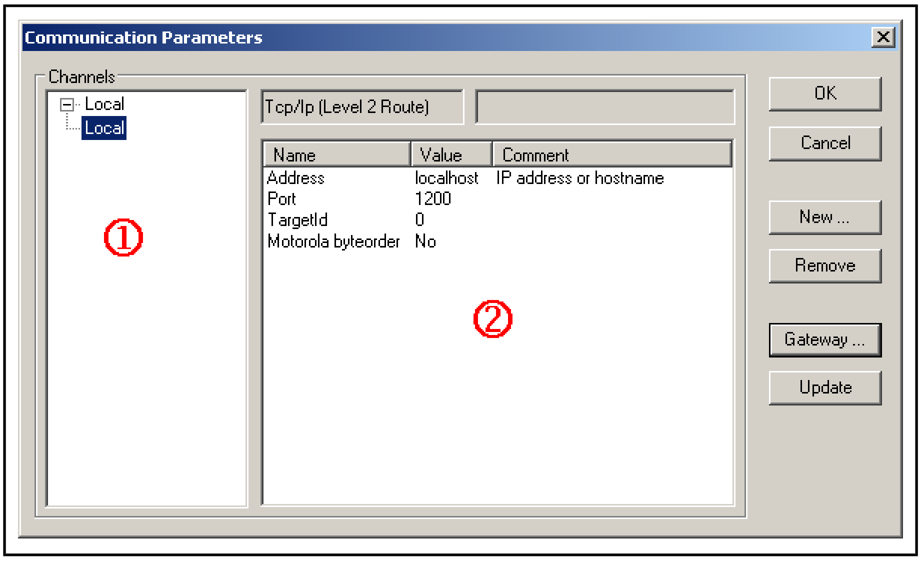
☞ For more information about the subject of "Communication", please refer to the IndraLogic help or to the IndraLogic manual /2/.

To complete your settings, click on the "Finish" button. This completes the wizard-guided basic settings. The new IndraLogic L40 DP device appears in the project explorer. The basic settings can be changed later, refer to chapter 4.3.5 "Modifying Basic Settings" on page 24.

For more information on the device, please refer to chapter 4.3.4 "Components of the IndraLogic L40 Device" on page 22.

Communication Parameters

New communication channels (communication instances) can be created, modified or deleted in the "Communication Parameters" dialog.



Name	Value	Comment
Address	localhost	IP address or hostname
Port	1200	
TargetId	0	
Motorola byteorder	No	

① Communication channels

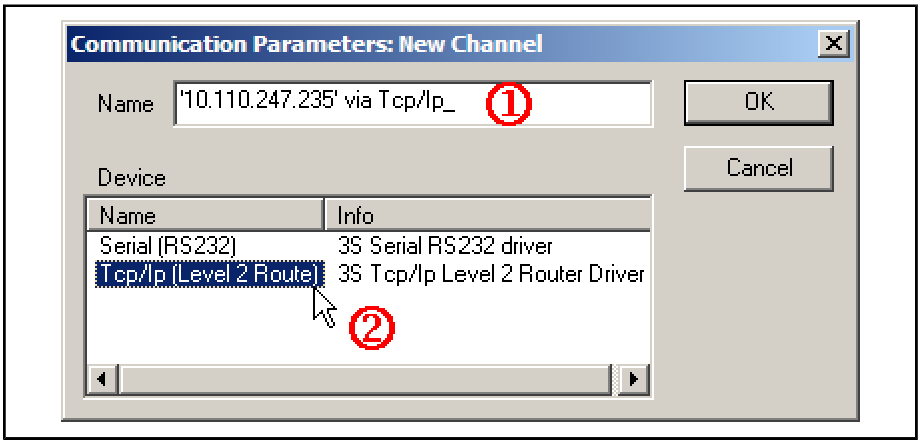
② Parameters of the selected communication channel

Fig.4-6: Communication Parameters

Create a new communication channel:

Click on the "New... "button to create a new communication channel.

The following dialog appears:



Name '10.110.247.235' via Tcp/Ip_

Device

Name	Info
Serial (RS232)	3S Serial RS232 driver
Tcp/Ip (Level 2 Route)	3S Tcp/Ip Level 2 Router Driver

① Name

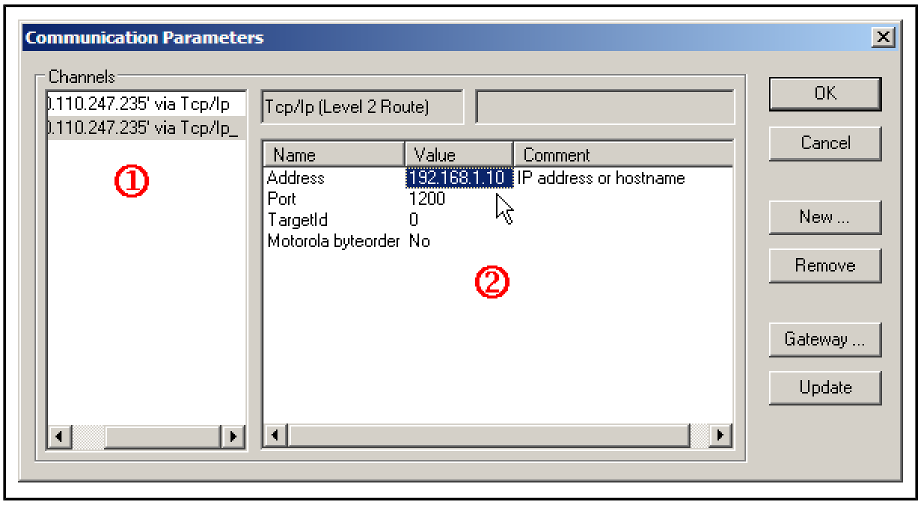
② Device

Fig.4-7: Creating a new communication channel

Name Enter a name for the new communication channel in this box.

Device Select the desired communication driver to enable communication between IndraWorks and the IndraLogic L40 DP. Click on the appropriate line in the "Name" column and confirm your settings with "OK".

The new communication channel appears in area ① in fig. 4-8 "Adjusting the parameters of a communication channel" on page 22.



Name	Value	Comment
Address	192.168.1.10	IP address or hostname
Port	1200	
TargetId	0	
Motorola byteorder	No	

Fig.4-8: Adjusting the parameters of a communication channel

Changing a Communication Channel:

Left-click to select the new communication channel and adjust its parameters. Double-click on the appropriate line in area ② (refer to fig. 4-8 "Adjusting the parameters of a communication channel" on page 22) and change the entry in the "Value" column according to your requirements. Confirm the modifications with the <Enter> key.

Removing a Communication Channel:

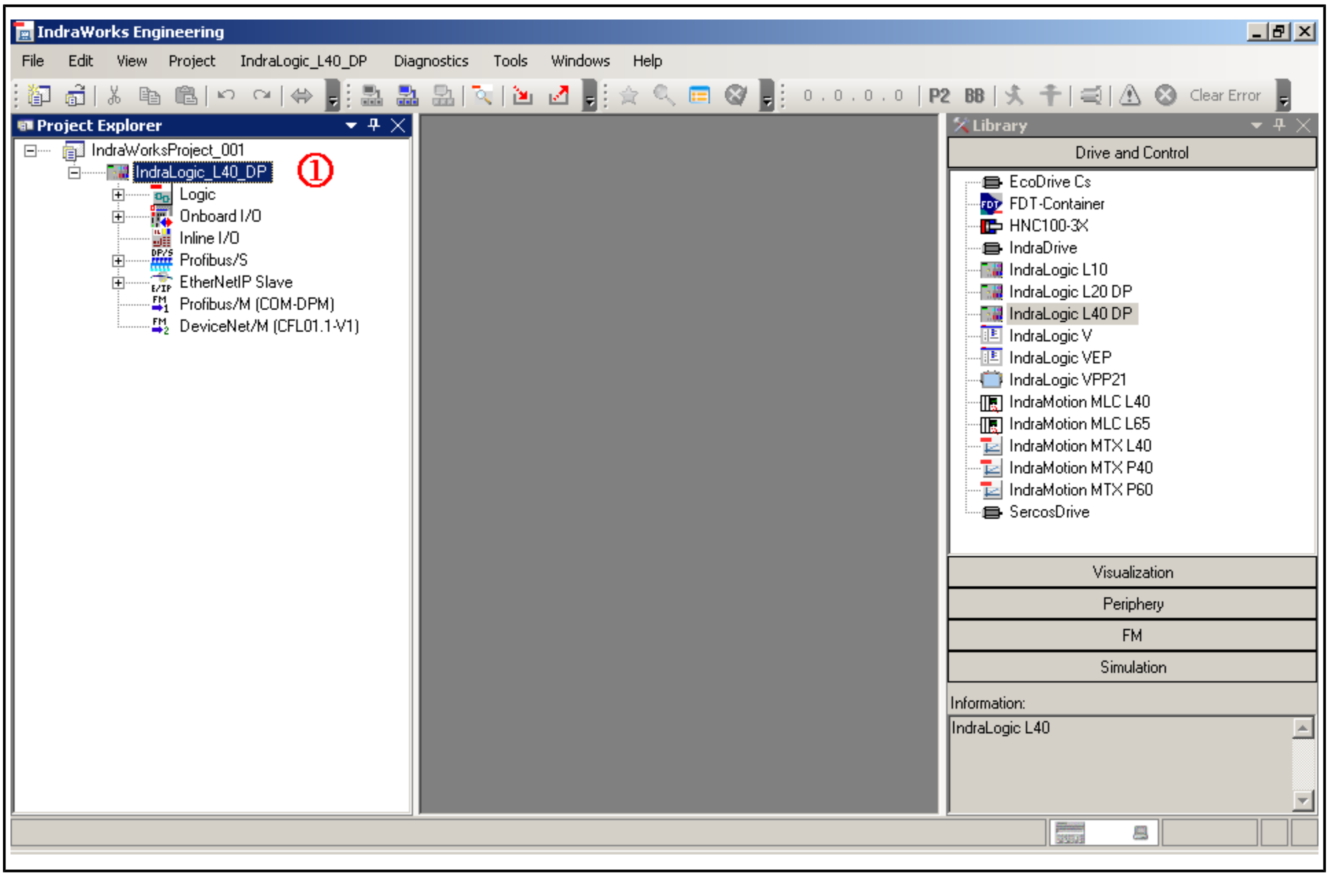
Select the desired communication channel and click on the "Remove" button.

☞ You will find further information on the individual parameters in the IndraLogic help, or in the IndraLogic manual /2/.

4.3.4 Components of the IndraLogic L40 Device

General Information

To display all subordinate objects of the new IndraLogic L40 DP device, click on the plus symbol of the object tree in the project explorer.



① IndraLogic L40 device

Fig.4-9: IndraLogic L40 device with subordinate objects

The object tree of the IndraLogic L40 DP can contain the following objects:

- "**Logic**": Part of the IndraLogic L40 DP device responsible for processing all I/O signals through the software. This includes e.g. PLC modules (POUs), the PLC task management and global variables. Double-click on the "Logic" object to start IndraLogic with the "IndraLogic L40 DP" target system.
- "**Onboard I/O**": Local I/O area of the IndraLogic L40 DP (Onboard I/O). This area is equipped with 8 high-speed inputs and 8 high-speed outputs each. The configuration is described in chapter 4.4 "Configuring the Onboard I/O " on page 27.
- "**Inline I/O**": Local Inline I/O area of the IndraLogic L40 DP. This area contains the I/Os of Rexroth Inline modules that can be coupled to the IndraLogic L40 DP. The configuration is described in chapter 4.5 "Projecting Rexroth Inline Modules " on page 31.
- "**Profibus/M**": Only available, if the onboard Profibus DP interface (X7P) has been configured as "Profibus/M" (Profibus master) or, alternatively, a Profibus/M function module has been configured, refer to fig. 4-3 "Entering the firmware version and the settings for the field bus interfaces" on page 18. This object allows parameterization of important Profibus operating data and integration of Profibus slaves, refer to chapter 4.6 "Configuring IndraLogic L40 as Profibus DP Master " on page 34. The object icon indicates the interface selected: DPM = Onboard-Profibus-DP interface (X7P) FM1 = Profibus-DP interface on function module.
- "**Profibus/S**": Only available, if the onboard Profibus DP interface (X7P) has been configured as "Profibus/S" (Profibus slave), refer to fig. 4-3

"Entering the firmware version and the settings for the field bus interfaces" on page 18. This object allows configuration of the Profibus slave settings of the IndraLogic L40 DP, refer to chapter 4.7 "Configuring IndraLogic L40 as Profibus DP Slave " on page 46.

- "**Ethernet/IP slave**": Only available, if the IndraLogic L40 DP has been configured as Ethernet/IP slave, refer to fig. 4-3 "Entering the firmware version and the settings for the field bus interfaces" on page 18. This object allows configuration of the appropriate settings of the IndraLogic L40 DP, refer to chapter 4.8.1 "Configuring IndraLogic L40 as an Ethernet/IP Slave " on page 50.
- "**DeviceNet/M**": Only available, if a DeviceNet/M function module was configured, refer to fig. 4-3 "Entering the firmware version and the settings for the field bus interfaces" on page 18. This object allows parameterization of important Profibus operating data and integration of Profibus slaves, refer to chapter 4.9 "Configuring IndraLogic L40 as DeviceNet Master " on page 57.

Inserting an Profibus/M Object Subsequently

Drag the "Profibus/M..." object from the "FM" (function modules) library into the object tree of the IndraLogic L40 DP. Alternatively, you can also use the "Copy" and "Paste" functions of the particular context menu (right mouse button).

Inserting an EthernetIP Slave Object Subsequently

☞ An "EthernetIP slave" object can be inserted subsequently only if there isn't any "EtherNetIP Slave" in the object tree of the IndraLogic L40 DP yet!

Insert the "EtherNet IP Slave" object from the "FM" (function modules) library in the object tree of the IndraLogic L40 DP by drag-and-drop. Alternatively, you can also use the "Copy" and "Paste" functions of the particular context menu (right mouse button).

Inserting an DeviceNet/M Object Subsequently

Drag the "DeviceNet/M" object from the "FM" (function modules) library into the object tree of the IndraLogic L40 DP. Alternatively, you can also use the "Copy" and Paste functions of the particular context menu (right mouse button).

4.3.5 Modifying Basic Settings

Some of the settings generated with the help of the wizard can also be modified subsequently.

Modifying Device Settings

Use the "Properties" menu item in the context menu of the IndraLogic L40 DP device.

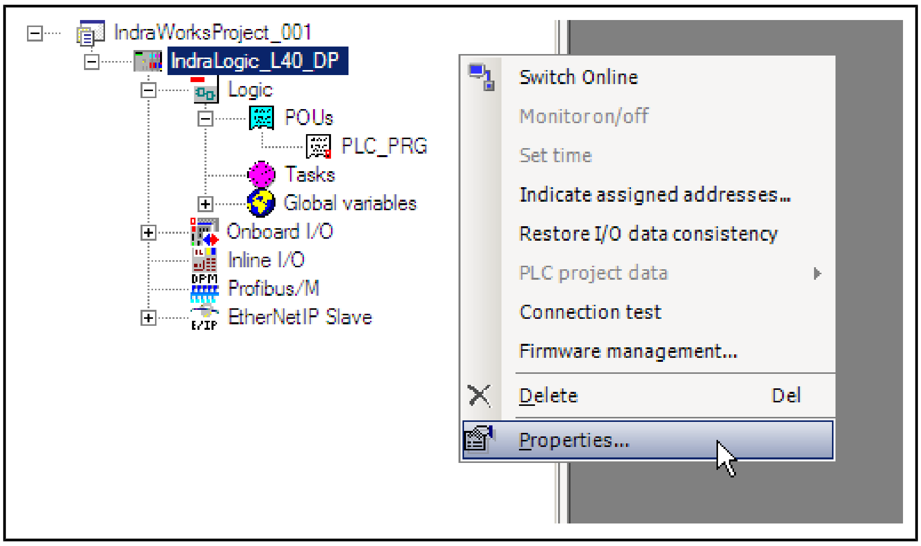


Fig.4-10: IndraLogic L40 context menu

This will open the dialog for modifying the device settings, refer to fig. 4-2 "Device Settings" on page 17.

Modifying Communication Settings

Use the "Properties" menu item in the context menu of the "Logic" object.

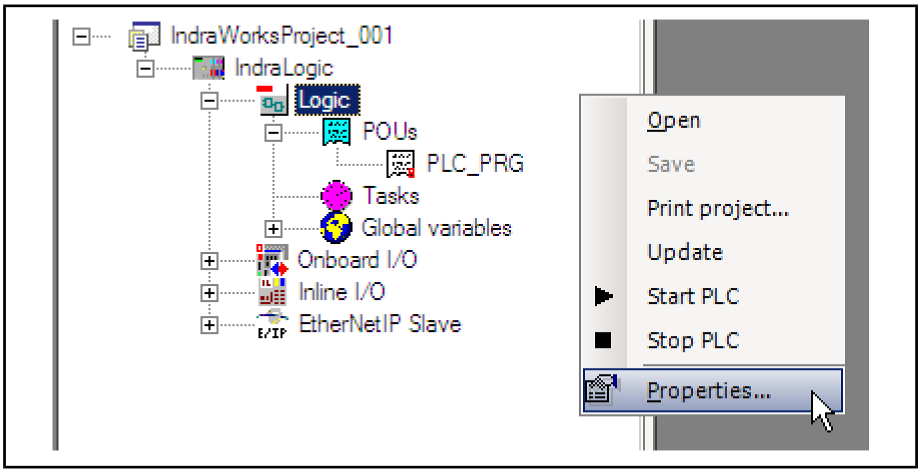
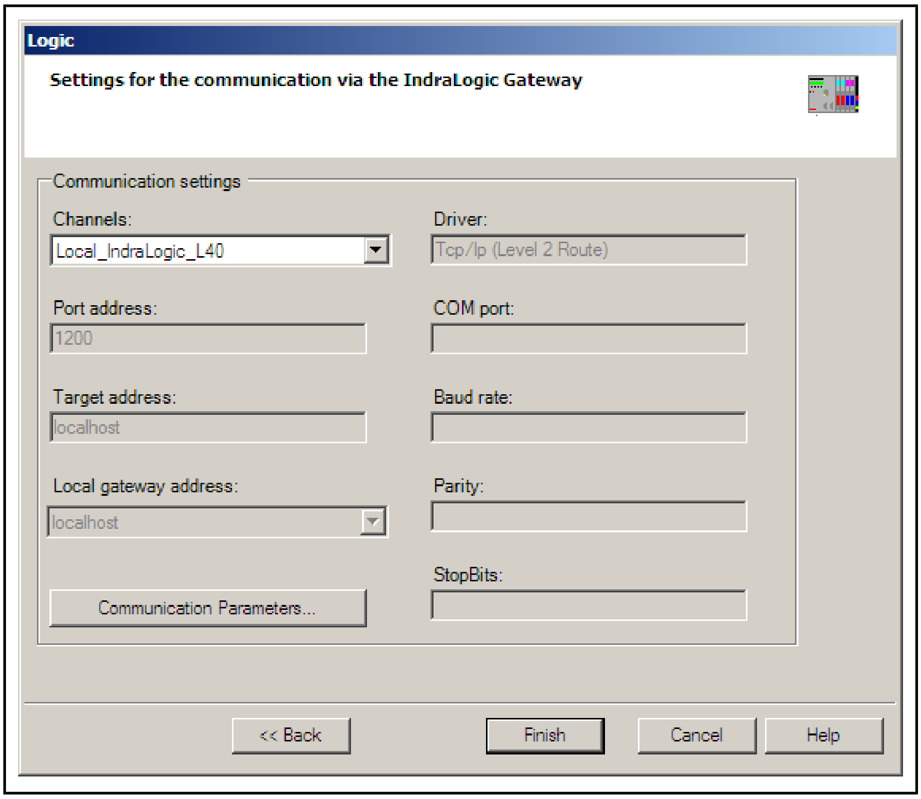


Fig.4-11: "Logic" object context menu

The following dialog opens:


Logic

Settings for the communication via the IndraLogic Gateway

Communication settings

Channels: Local_IndraLogic_L40

Driver: Tcp/Ip (Level 2 Route)

Port address: 1200

COM port:

Target address: localhost

Baud rate:

Local gateway address: localhost

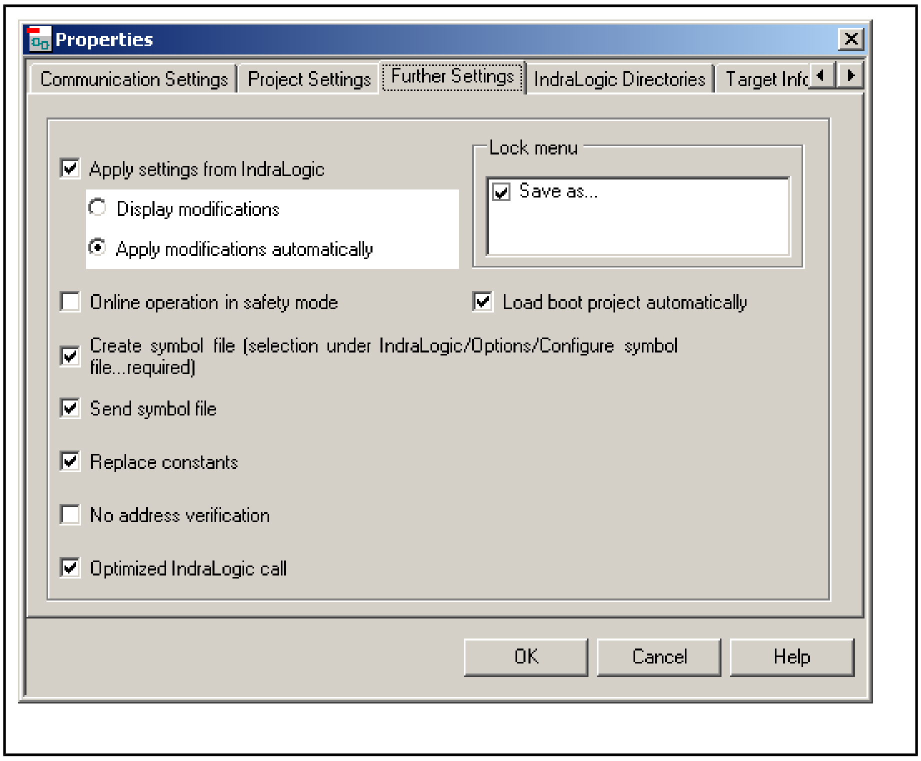
Parity:

StopBits:

Fig.4-12: "Properties" dialog of the "Logic" object

Please also note chapter 4.3.3 "Communication Settings" on page 19.

☞ The tabs of this dialog allow further settings to be made in addition to the communication settings (refer to 4-13). For more information about this subject, please refer to the IndraWorks documentation /5/ and/or online help.



☑ Apply settings from IndraLogic
○ Display modifications
● Apply modifications automatically

Lock menu
☑ Save as...

☐ Online operation in safety mode
☑ Load boot project automatically
☑ Create symbol file (selection under IndraLogic/Options/Configure symbol file...required)
☑ Send symbol file
☑ Replace constants
☐ No address verification
☑ Optimized IndraLogic call

Fig.4-13: "Further settings" tab in the "Properties" dialog

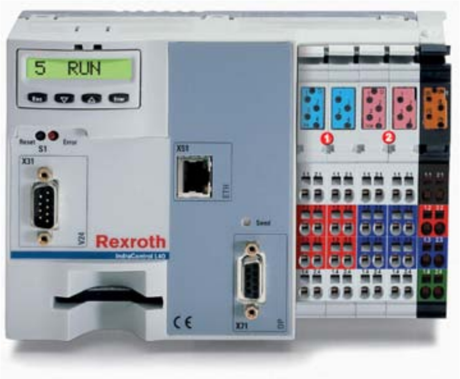
4.4 Configuring the Onboard I/O

4.4.1 Overview

The IndraLogic L40 DP is provided with 8 high-speed interrupt-compliant digital inputs and outputs each.

☞ With regard to the interrupt capability, please refer to chapter 4.10.3 "Task Configuration " on page 73.

☞ An I/O extension can be realized using Inline modules (see chapter 4.5 "Projecting Rexroth Inline Modules " on page 31) and/or coupling subordinated slaves (e.g. Profibus DP).



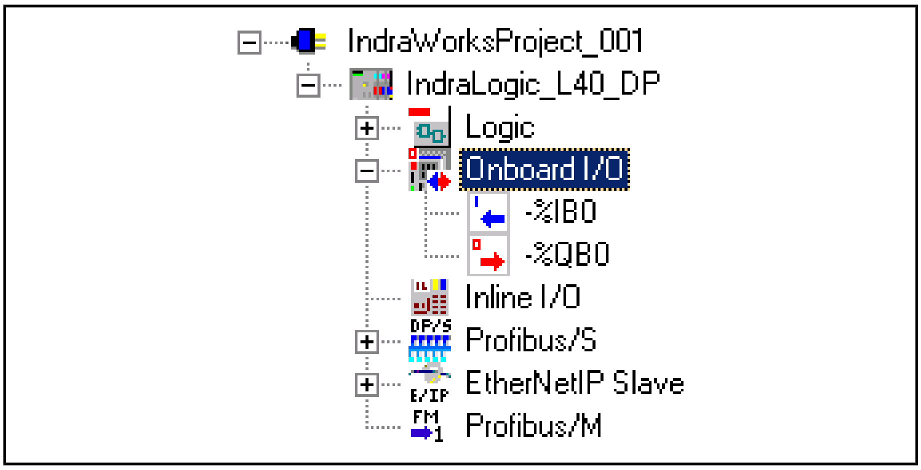
① Inputs
② Outputs
Fig.4-14: Onboard I/O of the IndraLogic L40

The digital inputs and outputs available on slots 1 to 4 (from left to right) are assigned to light-emitting diodes and bit addresses according to the following table:

		Inputs								Outputs							
	Slot	1				2				3				4			
	Status LED	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Byte-bit view	Bytes	IX0.0 – 0.7 (default)								QX0.0 – 0.7 (default)							
	Bit	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Module	Terminal point (signal)	1.1	2.1	1.4	2.4	1.1	2.1	1.4	2.4	1.1	2.1	1.4	2.4	1.1	2.1	1.4	2.4
	Terminal point (24 V)	1.2	2.2	1.3	2.3	1.2	2.2	1.3	2.3	–	–	–	–	–	–	–	–
	Terminal point (last ground)	–	–	–	–	–	–	–	–	1.2	2.2	1.3	2.3	1.2	2.2	1.3	2.3

Fig.4-15: Default address assignment of inputs and outputs

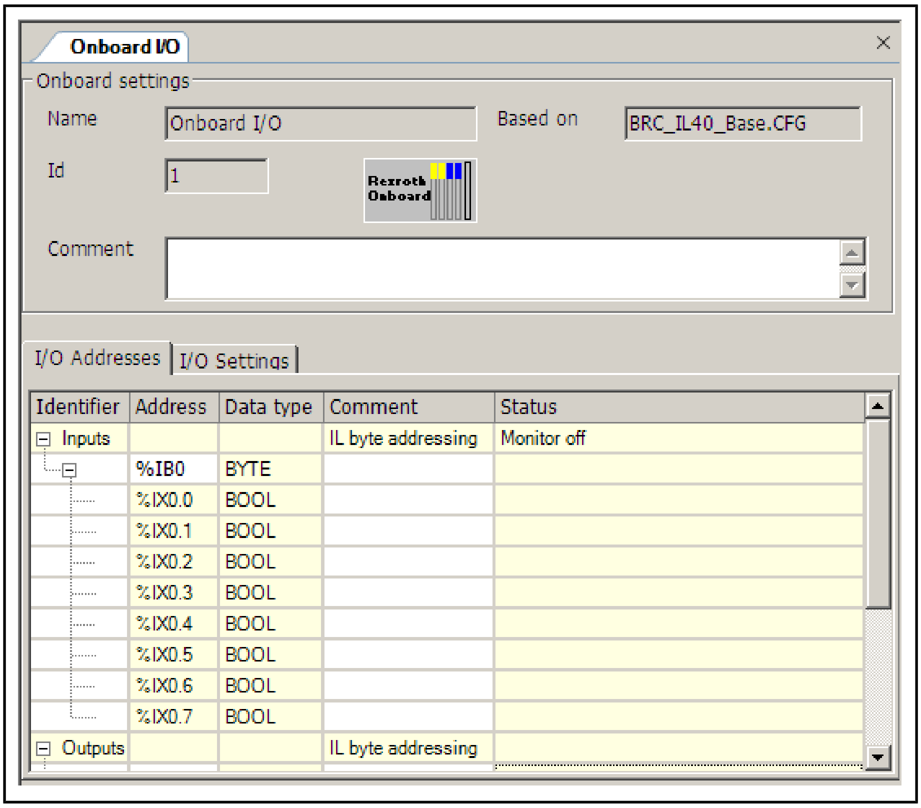
To configure the addresses, double-click on "Onboard I/O" in the project explorer.



IndraWorksProject_001
IndraLogic_L40_DP
Logic
Onboard I/O
-%IB0
-%QB0
Inline I/O
Profibus/S
EtherNetIP Slave
Profibus/M

Fig.4-16: "Onboard I/O" object

This opens a window in the workspace:



Onboard settings

Name: Onboard I/O

Based on: BRC_IL40_Base.CFG

Id: 1

Comment:

I/O Addresses | I/O Settings

Identifier	Address	Data type	Comment	Status
⊟ Inputs			IL byte addressing	Monitor off
⊟	%IB0	BYTE		
	%IX0.0	BOOL		
	%IX0.1	BOOL		
	%IX0.2	BOOL		
	%IX0.3	BOOL		
	%IX0.4	BOOL		
	%IX0.5	BOOL		
	%IX0.6	BOOL		
	%IX0.7	BOOL		
⊟ Outputs			IL byte addressing	

Fig.4-17: "Onboard I/O" window

"Onboard Settings"

Name: Internal name, specified by the installed target system.

Based on: Name of the basic device description file. The file is an integral part of the installed target system.

ID: Internal ID, specified by the installed target system.

Comment: Enter any comment to describe the onboard I/O in detail.

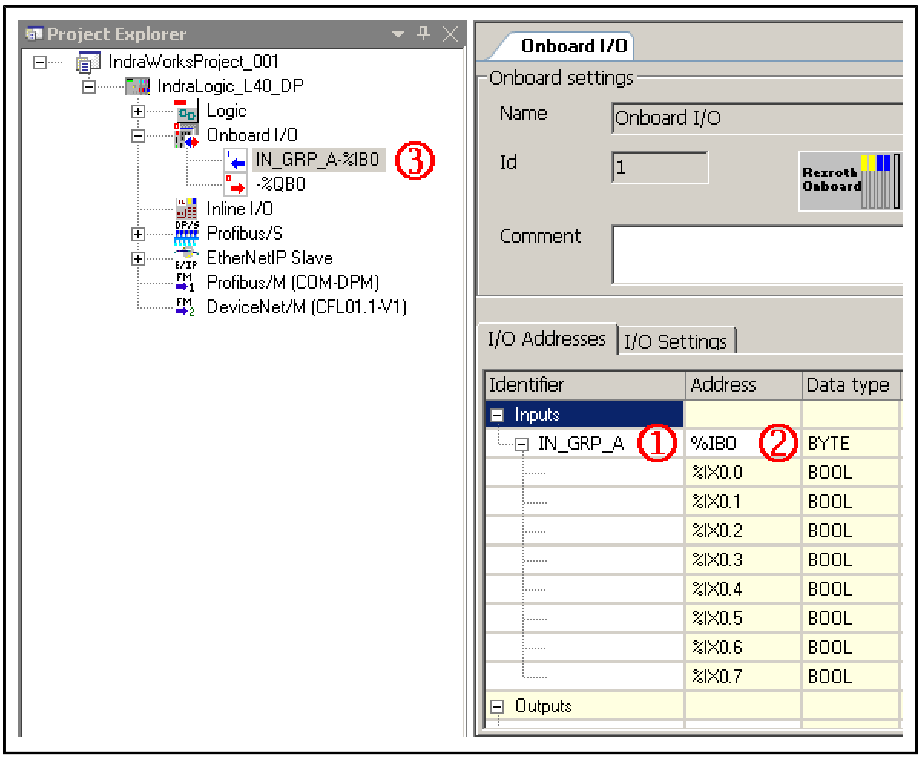
"I/O Addresses" Tab

This tab is for assigning the I/O areas of the onboard I/Os to the physical addresses of the control (I/O addresses of the PLC).

Identifier: This column shows the two input and output modules. Click on the plus or minus symbol to switch between the byte and bit views respectively.

You can also assign a symbolic address to each absolute address (double-click on the particular field). After it has been entered, the symbolic address is automatically created as a global variable in the PLC project.

The symbolic address of a node also appears in the project explorer:



Project Explorer

- IndraWorksProject_001
  - IndraLogic_L40_DP
    - Logic
    - Onboard I/O
      - IN_GRP_A-%IB0
      - -%QB0
    - Inline I/O
    - Profibus/S
    - EtherNetIP Slave
    - Profibus/M (COM-DPM)
    - DeviceNet/M (CFL01.1-V1)

Onboard I/O

Onboard settings

Name: Onboard I/O

Id: 1

Comment:

I/O Addresses | I/O Settings

Identifier	Address	Data type
Inputs		
IN_GRP_A	%IB0	BYTE
	%IX0.0	BOOL
	%IX0.1	BOOL
	%IX0.2	BOOL
	%IX0.3	BOOL
	%IX0.4	BOOL
	%IX0.5	BOOL
	%IX0.6	BOOL
	%IX0.7	BOOL
Outputs		

① Symbolic address

② Absolute address

③ Resulting entry in the project explorer

Fig.4-18: Symbolic and absolute addresses of I/O objects

Address: I/O address. Enter the desired I/O address as byte address (e.g. %IB10). Entries in italics serve only for display purposes and cannot be edited.

☞ Automatic readdressing is possible in the "I/O Settings" tab.

Data type: "BYTE" stands for byte addresses, and "BOOL" for bit addresses.

Comment: Enter any comment regarding an address in this column.

Status: Physical status of the input/output. The status is only indicated in diagnostic mode during communication between IndraWorks and IndraLogic L40 DP.

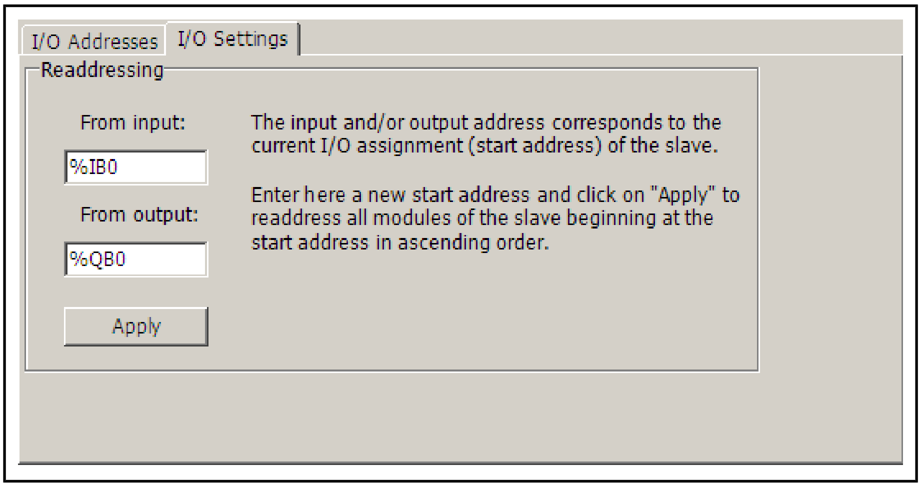
"I/O Settings" Tab

This tab is for starting the automatic assignment of I/O addresses.

From input/From output: Current or desired start addresses of the outputs and inputs.

"**Apply**": Renumbers automatically all outputs and inputs of the onboard I/O in ascending order, starting with the start addresses displayed (refer to "From input / From output"). Please note that any existing address gaps are closed during this process!

If the automatic numbering causes collisions with already assigned address areas, IndraWorks indicates the collision cause and automatically determines the next free address area.



From input: %IB0

From output: %QB0

The input and/or output address corresponds to the current I/O assignment (start address) of the slave.

Enter here a new start address and click on "Apply" to readdress all modules of the slave beginning at the start address in ascending order.

Fig.4-19: Readdressing the onboard I/O

4.5 Projecting Rexroth Inline Modules

4.5.1 Overview

The locally available I/O units can be extended by the Rexroth Inline I/O system simply by mounting the Inline modules to the right side of the IndraLogic L40 DP.

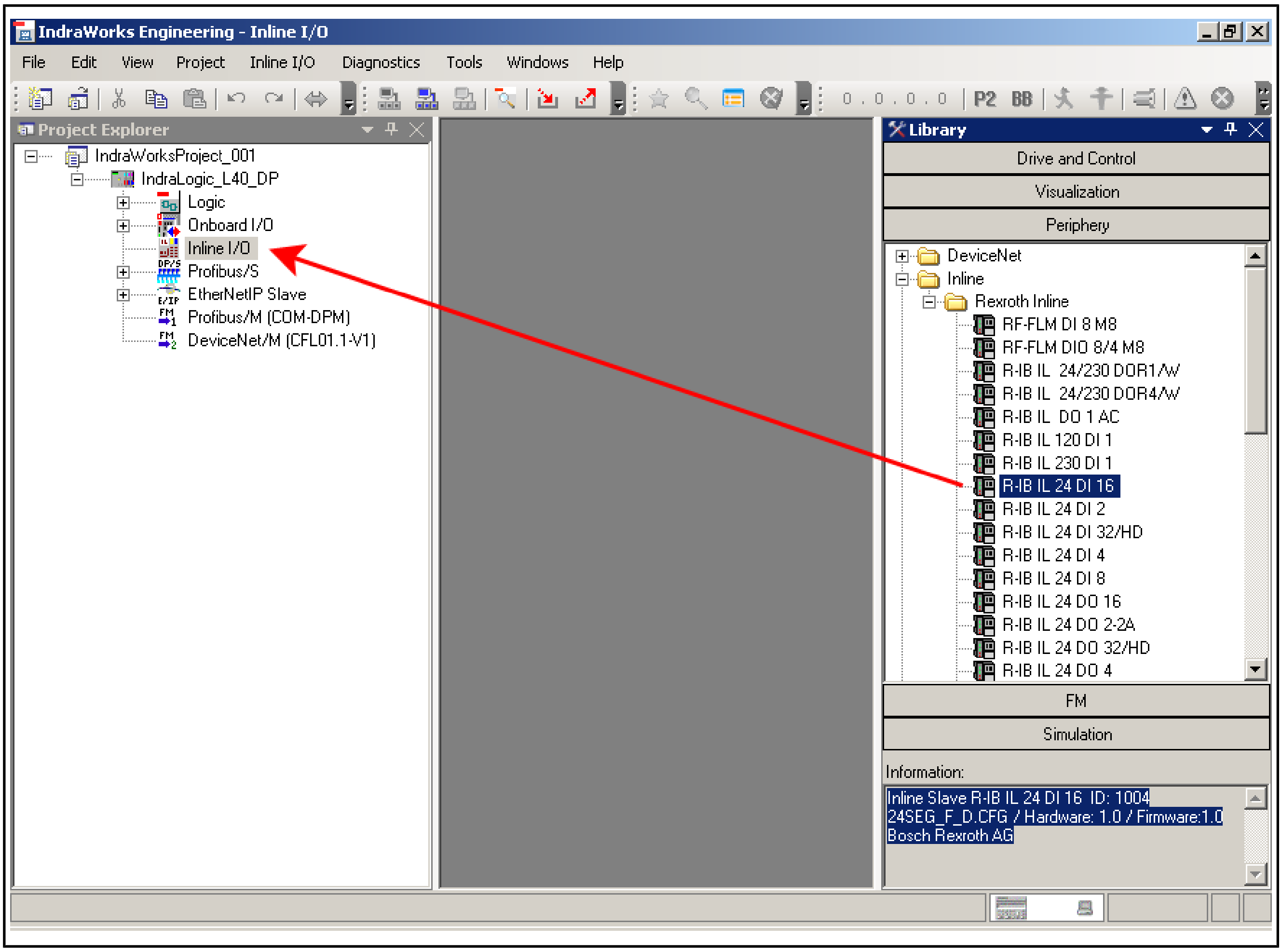
Main features of the IndraLogic L40 DP Inline module system:

- Extension of the local I/O field to up to 64 bytes (total of all inputs and outputs).
- Coupling up to 63 Inline modules to the IndraLogic L20 in any address-independent order. The actual usable number of Inline modules also depends on the current consumption of the individual modules, refer to documentation /1/.

4.5.2 Inserting Inline Modules

All Inline modules provided for the IndraLogic L40 DP can be found in the "Periphery" library under "Inline", "Rexroth Inline". Insert the required Inline modules from the library in the corresponding "Inline I/O" object by drag-and-drop. New Inline modules can also be inserted between already existing Inline modules in the project explorer.

Alternatively, you can also use the "Add module" function of the context menu of the "Inline I/O" object. The new module will be the last module under "Inline I/O".



Inline Slave R-IB IL 24 DI 16 ID: 1004
24SEG_F_D.CFG / Hardware: 1.0 / Firmware:1.0
Bosch Rexroth AG

Fig.4-20: Inserting an Inline module (example)

4.5.3 Configuring Inline Modules

Double-click on the desired Inline module in the project explorer.

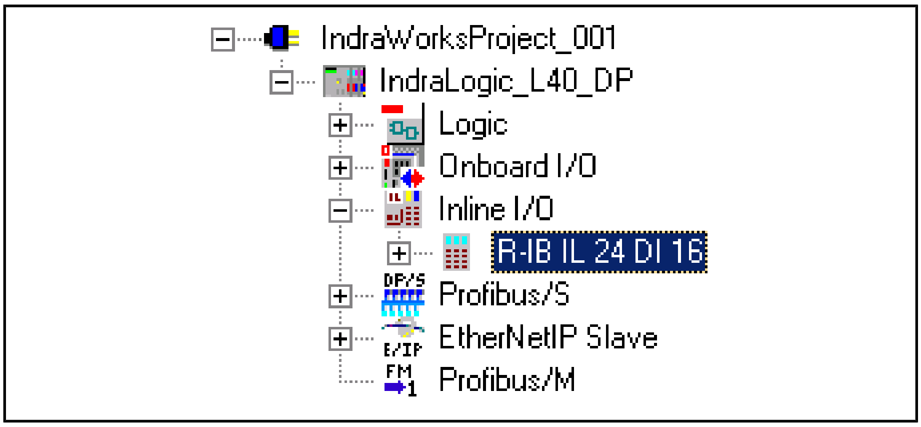
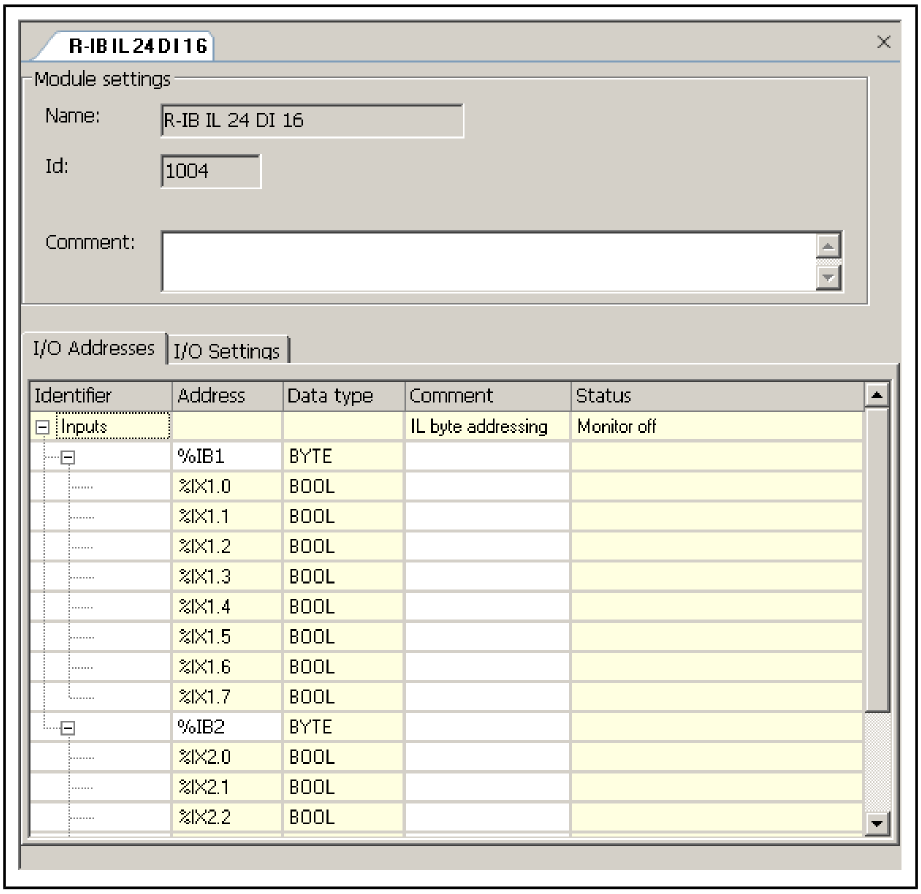


Fig.4-21: Inline module object (example)

This opens the following window in the workspace:



R-IB IL 24 DI 16

Module settings

Name: R-IB IL 24 DI 16

Id: 1004

Comment:

I/O Addresses | I/O Settings

Identifier	Address	Data type	Comment	Status
⊟ Inputs			IL byte addressing	Monitor off
⊟	%IB1	BYTE		
	%IX1.0	BOOL		
	%IX1.1	BOOL		
	%IX1.2	BOOL		
	%IX1.3	BOOL		
	%IX1.4	BOOL		
	%IX1.5	BOOL		
	%IX1.6	BOOL		
	%IX1.7	BOOL		
⊟	%IB2	BYTE		
	%IX2.0	BOOL		
	%IX2.1	BOOL		
	%IX2.2	BOOL		

Fig.4-22: Window of an Inline module (example)

"Module Settings"

Name: Internal name, defined through the installed target system.

ID: Internal ID, defined through the installed target system.

Comment: Enter any comment to describe the Inline module in detail.

"I/O Addresses" Tab

This tab is for assigning the I/O areas of the Inline modules to the physical addresses of the control (I/O addresses of the PLC).

Identifier: This column shows the input and output structure. Click on the plus or minus symbol to switch between the byte and bit views respectively.

You can also assign a symbolic address to each absolute address (double-click on the particular field). After it has been entered, the symbolic address is automatically created as a global variable in the PLC project.

The symbolic address of a node also appears in the project explorer. For an example of onboard I/O, refer to fig. 4-18 "Symbolic and absolute addresses of I/O objects" on page 30.

Address: I/O address. Enter the desired I/O address as byte address (e.g. %IB10). Entries in italics serve only for display purposes and cannot be edited.

☞ Automatic readdressing is possible in the "I/O Settings" tab.

Data type: "BYTE" stands for byte addresses, and "BOOL" for bit addresses.

Comment: Enter any comment regarding an address in this column.

Status: Physical status of the input/output. The status is only indicated in diagnostic mode during communication between IndraWorks and IndraLogic L40 DP.

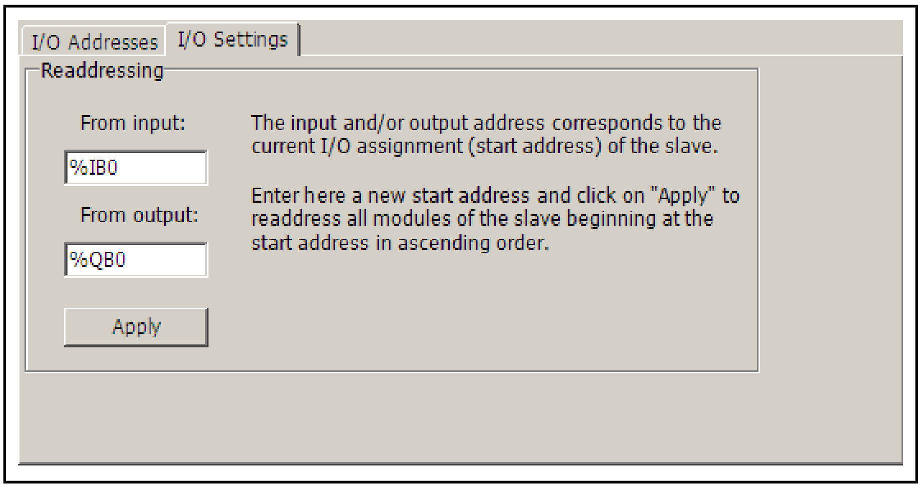
"I/O Settings" Tab

This tab is for starting the automatic assignment of I/O addresses.

From input/From output: Current or requested initial addresses of the inputs and output. Corresponding to the used Inline module functionality (module with inputs, module with outputs) only relevant input boxes are indicated. If, for example, you parameterize an Inline module having inputs only, there will be no "From output" input box.

"**Apply**": Renumbers automatically all inputs and outputs of the Inline module in ascending order, starting with the start addresses displayed (refer to "From input / From output"). Please note that any existing address gaps are closed during this process!

If the automatic numbering causes collisions with already assigned address areas, IndraWorks indicates the collision cause and automatically determines the next free address area.



From input: %IB0

From output: %QB0

The input and/or output address corresponds to the current I/O assignment (start address) of the slave.

Enter here a new start address and click on "Apply" to readdress all modules of the slave beginning at the start address in ascending order.

Fig.4-23: Readdressing the inline I/O

4.6 Configuring IndraLogic L40 as Profibus DP Master

4.6.1 Overview

The IndraLogic L40 DP is equipped with a Profibus DP interface (X7P) with bus master functionality according to DIN EN 50170, part 2. This allows connection of Profibus DP slaves and access to their I/O fields. A Profibus DP function module with bus master functionality can be coupled to IndraLogic L40 DP alternatively or additionally to the onboard Profibus DP interface.

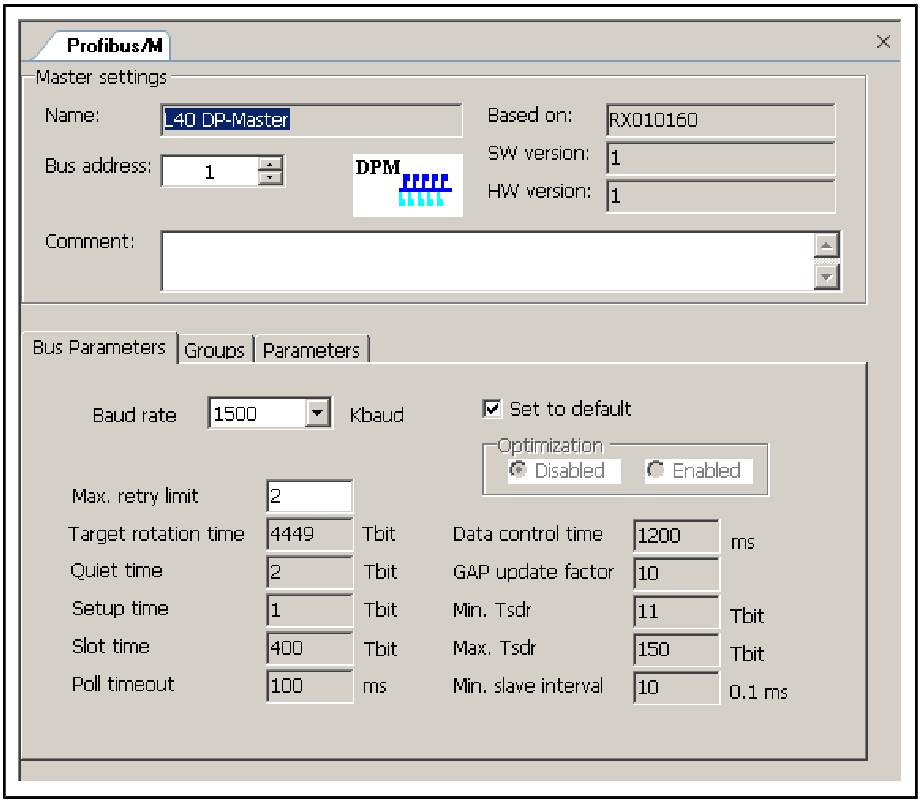
To use the IndraLogic L40 DP as Profibus DP master, proceed as follows:

1. Define the IndraLogic L40 DP as Profibus DP master, refer to chapter 4.3.2 "Device and Function Module Settings" on page 17 or chapter chapter "Inserting an Profibus/M Object Subsequently " on page 24. This is the only case where the required subordinate"Profibus/SM" object is available in the project explorer.
2. Make the appropriate master-specific settings.
3. Insert Profibus DP slaves in the "Profibus/M" object in the project explorer.
4. Configure the inserted Profibus DP slaves.

☞ The "IndraLogic L40 DP" can also be operated via the onboard Profibus DP interface (X7P) as subordinate Profibus DP slave. Please also refer to chapter 4.7 "Configuring IndraLogic L40 as Profibus DP Slave " on page 46.

4.6.2 Making Master-Specific Settings

Double-click on the appropriate "Profibus/M" object in the project explorer. This opens the following window in the workspace:



Profibus/M

Master settings

Name: L40 DP-Master

Bus address: 1

Based on: RX010160

SW version: 1

HW version: 1

Comment:

Bus Parameters | Groups | Parameters

Baud rate 1500 Kbaud

☑ Set to default

Optimization: ● Disabled ○ Enabled

Max. retry limit 2

Target rotation time 4449 Tbit

Quiet time 2 Tbit

Setup time 1 Tbit

Slot time 400 Tbit

Poll timeout 100 ms

Data control time 1200 ms

GAP update factor 10

Min. Tsdr 11 Tbit

Max. Tsdr 150 Tbit

Min. slave interval 10 0.1 ms

Fig.4-24: "Profibus/M" window

Master Settings

Name: Internal device name of the bus master

Bus address: Bus address of the bus master (FDL address: Fieldbus Data Link). Address "1" is entered automatically. If necessary, enter a different address in this box. Address "0" is reserved for engineering devices and cannot be used.

☞ The address values for the master should always be as low as possible. High address values deteriorate the bus performance!

Comment: Enter any comment to describe the bus master in detail.

Based on: File name of the basic ident number according to the GSD file. The GSD file contains the settings options of the device and is provided by the manufacturer of the device.

SW version: Software version according to the GSD file.

HW version: Hardware version according to the GSD file.

"Bus Parameters" Tag

The "Bus Parameters" tab contains the bus parameters required for operation of the Profibus DP. If the "Set to default" option is activated, only the "Baud rate" and "Max. retry limit" boxes can be edited. All other parameters are adjusted to the currently defined baud rate and are suitable for the majority of applications.

To change the values, deactivate the "Set to default" option and set the "Optimization" option to "Enabled". To disable the input boxes, you can reset the "Optimization" option to "Disabled". Once the "Set to default" option is activated, the values entered beforehand are overwritten by the adjusted values.

CAUTION

Modifications of the bus parameters might cause an unpredictable system behavior!

That is why the bus parameters may only be modified by skilled Profibus DP specialists, who are aware of the effects of such modifications!

Baud rate (transmission rate): Select the data transmission rate of the entire bus system from this list box. All connected slaves have to support the specified value. The maximum baud rate must not exceed the highest possible baud rate of the "worst" slave.

Max. retry limit: Maximum number of repetitions of a call telegram by the initiator, if the responder (receiver) does not send any answer. Possible settings: 1 to 15.

Target rotation time (Ttr): Command token rotation time for multi-master mode. Possible settings: 256 to 6647 Tbits (bit time units).

Quiet time (Tqui): Modulator quiet time or repeater rotation time. This is the time to elapse while the system waits for "quietness at the bus". Telegrams are neither sent nor received. Possible settings: 0 to 255 Tbits.

Setup time (Tset): Release time. Maximum time elapsing from the arrival of an event until execution of the required reaction. Possible settings: 1 to 255 Tbits.

Slot time (Tsl): "Waiting for reception" time. Maximum time that the initiator is waiting for the immediate acknowledgement or response after having emitted a call telegram (message cycle). Since Tid2 takes direct effect, the setting can be selected from a range from Tid2 + 15 to16383 Tbits.

Poll timeout: Time monitoring for an acyclic service (DPV1). Maximum time that the initiator is waiting for the reception of the response after emitting an acyclic request.

The setting can be selected in steps of 10 ms from a range from 10 to 655350 ms.

Data control time: Slave-related monitoring time in the bus master; within this time period at least one user data transfer with the corresponding slave must be completed.

Possible settings: 1 to 65535 ms.

GAP update factor: Factor allowing to control after how many bus cycles new masters can be identified.

Possible settings: 1 to 10.

Min. Tsdr: This is the minimum time a responder needs to answer a request telegram.

Possible settings: 11 to 255 Tbits.

Max. Tsdr: This is the maximum time a responder needs to answer a request telegram.

Possible settings: 35 to 1023 Tbits.

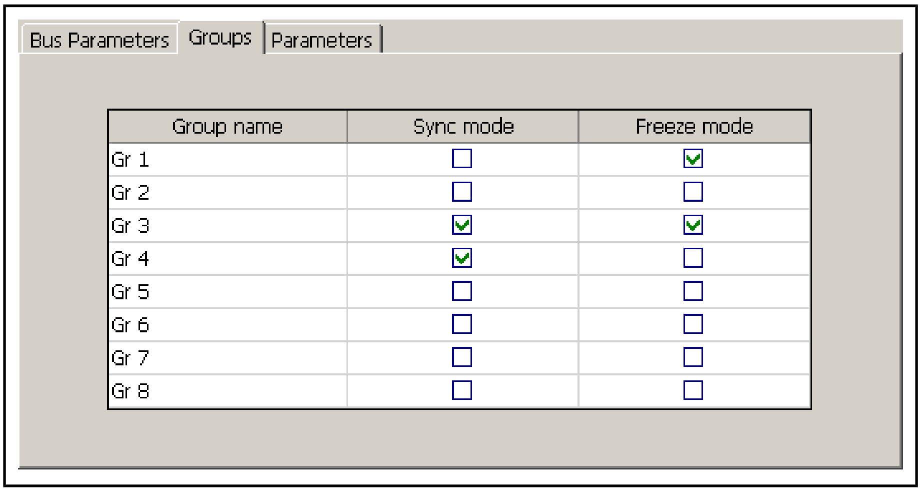
Min slave intervall: This is the minimum time having to elapse between two accesses of the bus master to the same slave.

Possible settings: 1 to 65535 (factor: 100 µs).

"Groups" Tag

Group assignment relates to the sync and freeze commands of the "global control" services provided by Profibus DP. Activate the group(s) to receive the sync and freeze commands sent by the master in the "Sync" and "Freeze" columns respectively.

Example: As can be seen from the group assignment shown in the following figure, the bus master is allowed to send the sync command to slaves of groups 3 and 4 and the freeze commands to slaves of groups 1 and 3.



Group name	Sync mode	Freeze mode
Gr 1	☐	☑
Gr 2	☐	☐
Gr 3	☑	☑
Gr 4	☑	☐
Gr 5	☐	☐
Gr 6	☐	☐
Gr 7	☐	☐
Gr 8	☐	☐

Fig.4-25: "Profibus/M" window, "Groups" tab

☞ For information on how to assign slaves to a particular group, refer to "Group Assignment Tab" on page 42.

☞ Control commands of the "sync" and "freeze" modes can be generated with the DP_SYCFR function block (refert to section "DP_SYCFR" on page 6-51). In this case, the master freezes the transfer of the commands to the relevant slaves if slaves of a "disabled" group are specified as receivers.

"Parameters" Tag

This tab is for displaying and editing possibly available vendor-specific bus master parameters. The integrated bus master of the IndraLogic L40 DP does not have any specific parameters. That is why this tab remains empty.

4.6.3 Inserting Profibus DP Slaves

All Profibus DP slaves provided for the IndraLogic L40 DP can be found in the "Periphery" library under "ProfibusDP". Move the required slaves from the library to the "Profibus/M" object by drag-and-drop. New slaves can also be inserted between existing slaves in the project explorer.

Alternatively, you can also use the "Add slave" function of the context menu of the "Profibus/M" object, refer to fig. 4-26 "Inserting a Profibus DP slave via the context menu of the "Profibus/M" object" on page 38. The new slave will be the last slave under "Profibus/M".

☞ If a required slave is not available in the library, it can be integrated in the library by importing its GSD file via the "Import GSD files..." function of the context menu of the "Profibus/M" object.

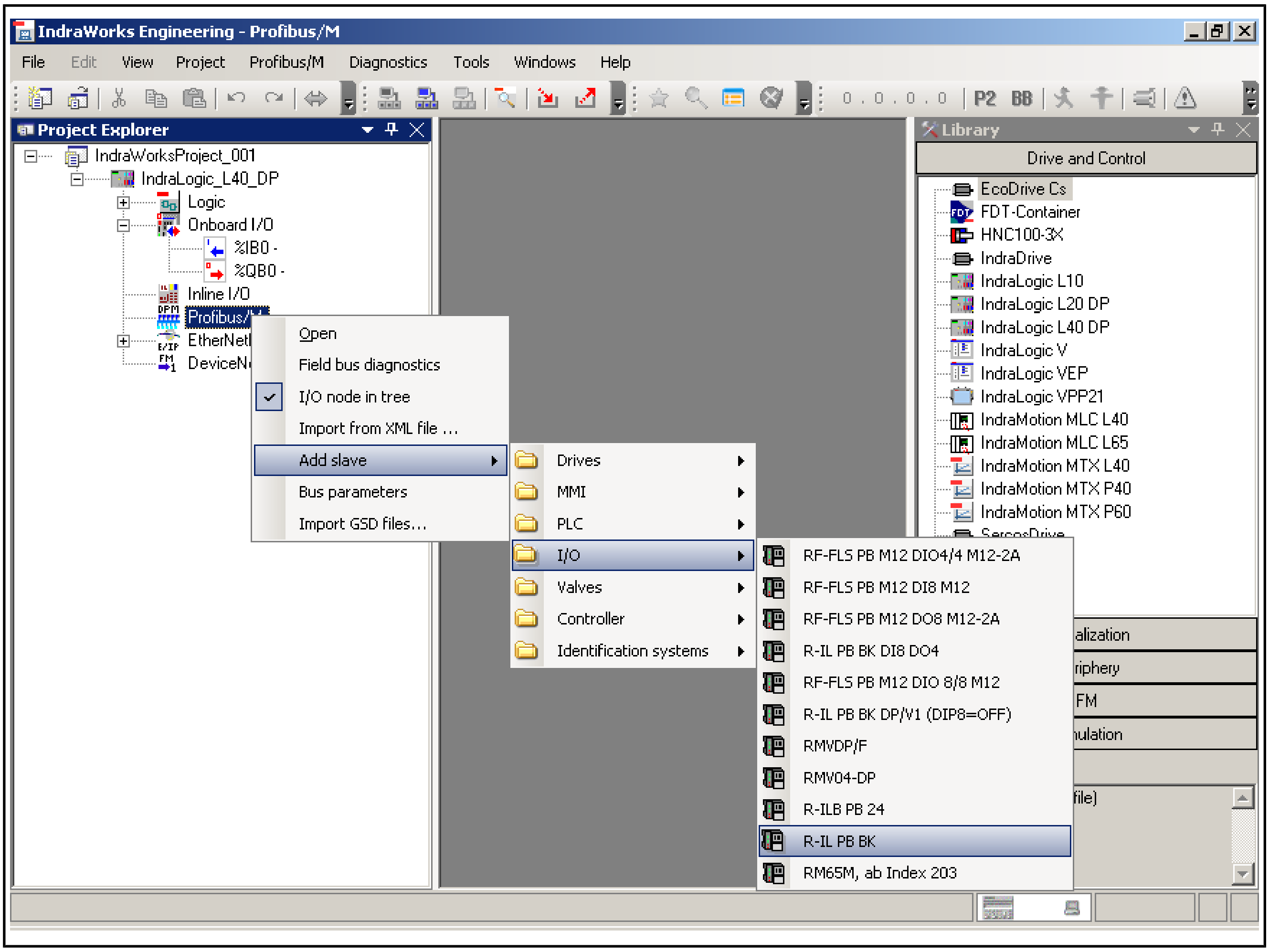
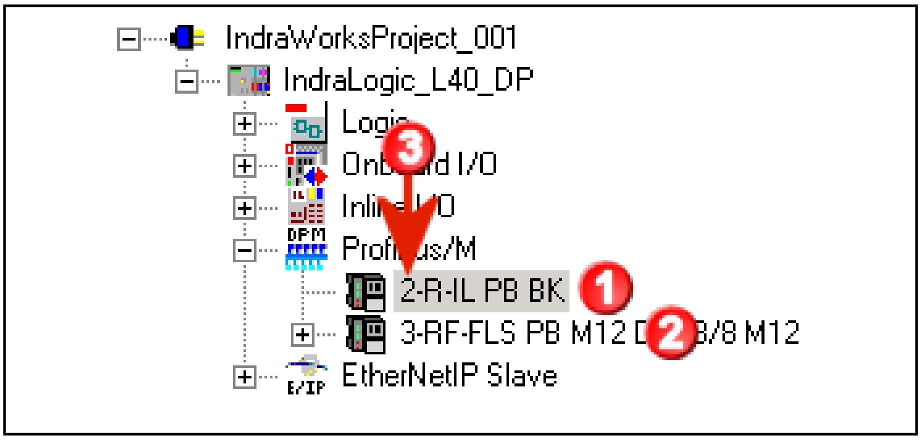


Fig.4-26: Inserting a Profibus DP slave via the context menu of the "Profibus/M" object

4.6.4 Configuring Profibus DP Slaves

Profibus DP distinguishes between two slave types:

- **Compact**: A compact slave has a defined module structure. After insertion of a slave in the project explorer, the modules below the slave object node are already completely available because of the compact design of the slaves.
- **Modular**: The module structure of the slave is variable. The modules can be arranged as desired, but according to the fitting specification. Subordinate (I/O) device levels of the slave are not yet available directly after insertion of the slave in the project explorer. Modules must be assigned manually to modular slaves. For information on how to insert modules, refer to chapter 4.6.5 "Inserting Modules in a Profibus DP Slave " on page 43.



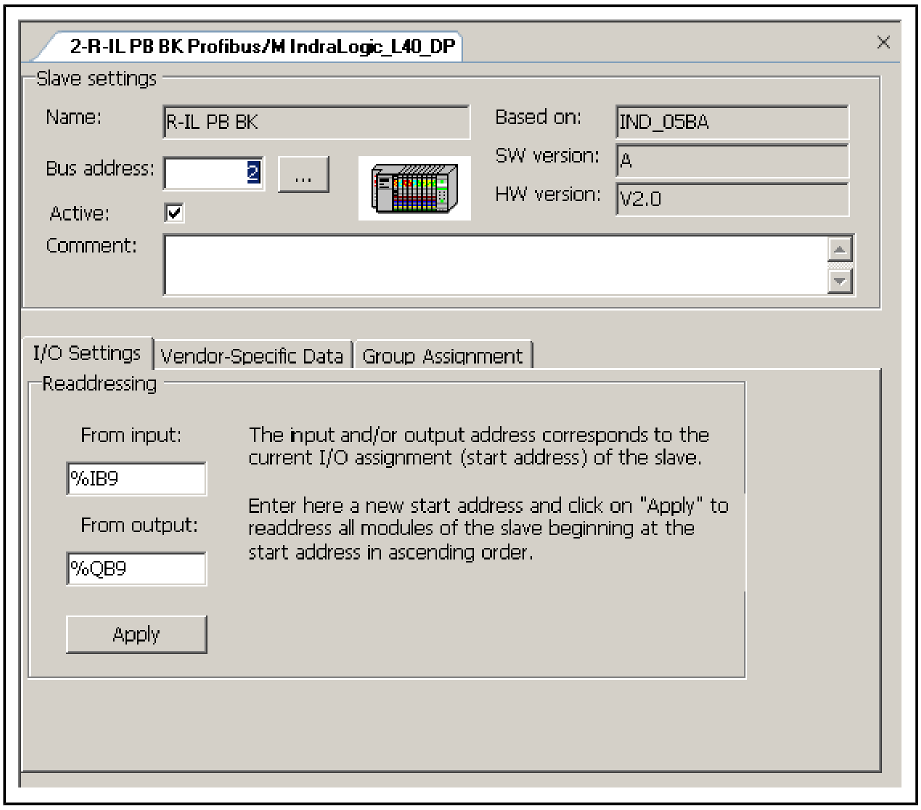
① Modular Profibus DP slave; still without subordinate modules
② Compact Profibus DP slave
③ Current bus addresses of the Profibus DP slaves

Fig.4-27: Profibus DP slave (example)

Select the "Bus address" menu item of the context menu. This will open the "DP Bus addresses" dialog where you can change the bus address of a slave. This dialog displays the complete address assignment of all Profibus DP devices. Double-click on a free field in the "Status" table column to apply the corresponding bus address to the currently selected slave.

☞ The address values for the master should always be as low as possible. High address values deteriorate the bus performance!

To configure a Profibus DP slave, double-click on the corresponding slave entry in the project explorer. This opens a window in the workspace, refer to the figure below:



Slave settings

Name: R-IL PB BK

Based on: IND_05BA

Bus address: 2

SW version: A

HW version: V2.0

Active: ☑

Comment:

Readdressing

From input: %IB9

From output: %QB9

The input and/or output address corresponds to the current I/O assignment (start address) of the slave.

Enter here a new start address and click on "Apply" to readdress all modules of the slave beginning at the start address in ascending order.

Fig.4-28: Window of a Profibus DP slave (example)

"Slave Settings"

Name: Device name according to the GSD file.

Bus address: Bus address of the slave (FDL address). Here, IndraWorks enters the next free bus address automatically. If necessary, enter a different address in this box. Alternatively, click on the "..." button to open the "DP bus addresses" dialog. This dialog displays the complete address assignment of all Profibus DP devices. Double-click on a free field in the "Status" or "Bus address" table column to apply the corresponding bus address to the currently selected slave.

Please note the following restrictions for the address selection:

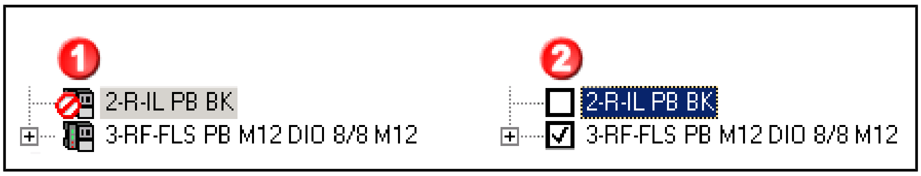
Address "0": reserved for engineering devices

Address "1": reserved for the Profibus DP master

The bus address of the slave also appears in the project explorer, refer to fig. 4-34 "Module of a Profibus DP slave (example)" on page 44.

Active: Activate this option, so that the slave can be commissioned at the Profibus after the next program download (refer to chapter 4.12 "Download and Commissioning" on page 81). Deactivate this option, if the slave is to be configured and archived, but not to be commissioned at the Profibus.

The setting (active/not active) can be also recognized and made in the project explorer. If the mouse pointer is positioned on the slave, the setting can be changed by a left mouse-click, see ② in the following figure.



① Display of the settings "not active" (first line) and "active" (second line).

② Display of the settings, if the mouse pointer is positioned on the slave.

Fig.4-29: Setting "active"/"not active" in the project explorer

Based on: File name of the basic ident number according to the GSD file.

SW version: Software version according to the GSD file.

HW version: Hardware version according to the GSD file.

Comment: Enter any comment to describe the slave in detail.

"I/O Settings" Tab

This tab is for starting the automatic assignment of I/O addresses.

From input/From output: Desired start addresses for automatic numbering of the inputs and outputs of all subordinate modules of the Profibus DP slaves.

Apply: Renumbers automatically all inputs and outputs of the subordinate Profibus DP slave modules in ascending order, starting with the start addresses displayed (see "From input / From output"). Please note that any existing address gaps are closed during this process!

If the automatic numbering causes collisions with already assigned address areas, IndraWorks indicates the collision cause and automatically determines the next free address area.

☞ Automatic numbering of subordinate modules can also be made in the module configuration. Also refer to chapter 4.6.5 "Inserting Modules in a Profibus DP Slave " on page 43.

"Vendor-Specific Data" Tab

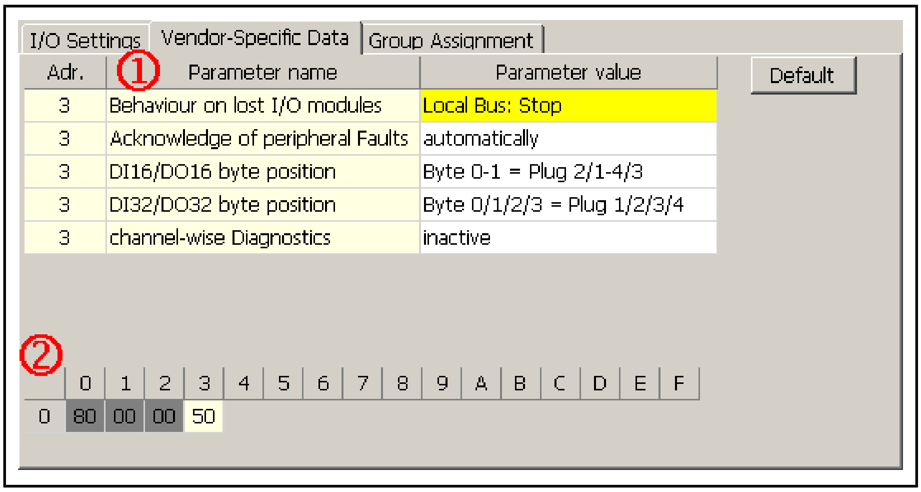
This tab is for displaying and editing possibly available vendor-specific slave parameters. This tab displays "vendor-specific data" with regard to the slave if such data is provided in the GSD file.

To change a parameter value, double-click on the "Parameter value" or "Value" field in the corresponding line of the table.

If you click on "Default", all modified values are overwritten by the original values from the GSD file.

Depending on the type provided in the GSD file, the data can be displayed in two different manners:

Display with address, parameter name and value:



Adr.	Parameter name	Parameter value
3	Behaviour on lost I/O modules	Local Bus: Stop
3	Acknowledge of peripheral Faults	automatically
3	DI16/DO16 byte position	Byte 0-1 = Plug 2/1-4/3
3	DI32/DO32 byte position	Byte 0/1/2/3 = Plug 1/2/3/4
3	channel-wise Diagnostics	inactive

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	80	00	00	50												

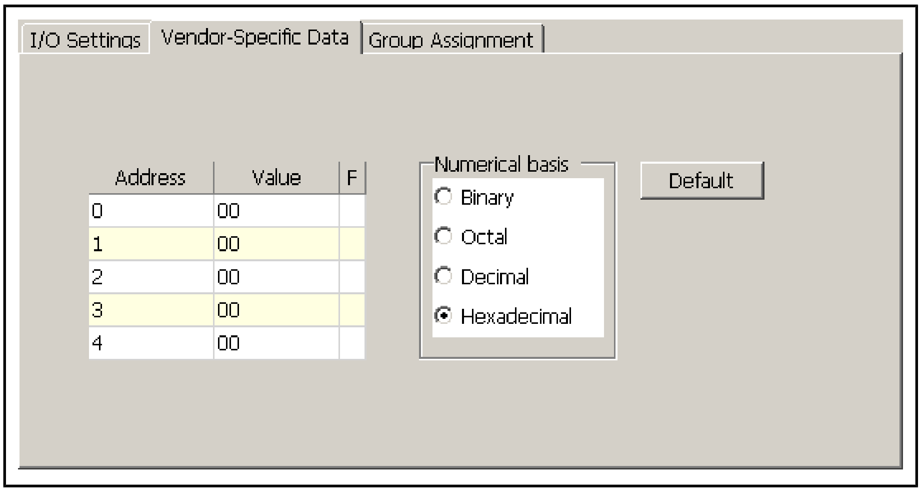
① List with parameter name and value

② Parameter data in byte view

Fig.4-30: Vendor-specific data with address, parameter name and value

In this type of display, a parameter value can be shown and edited according to its data type ①, e.g. "Active" and "Not active" for boolean values. Additionally, all resulting parameter data is displayed in the byte view ②.

Display of address and value:



Address	Value	F
0	00	
1	00	
2	00	
3	00	
4	00	

Numerical basis

○ Binary
○ Octal
○ Decimal
● Hexadecimal

Fig.4-31: Vendor-specific data with address and value

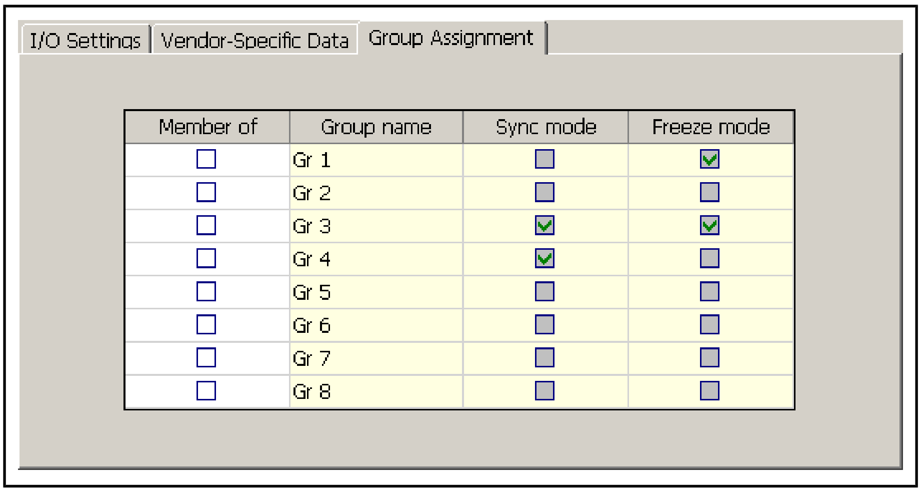
In this type of display, the values are subdivided in bytes. The **Numerical basis** can be set for display of byte values.

"Group Assignment" Tab

Group assignment relates to the sync and freeze commands of the "global control" services provided by Profibus DP. To assign the slave to one or several groups, activate the desired group in the "Member of" column.

The "Sync mode" and "Freeze mode" columns indicate the groups actually receiving the sync or freeze commands sent by the master. These assignments are set in the "Groups" tab in the "Master Settings", see chapter 4.6.2 "Making Master-Specific Settings" on page 35.

Example: As can be seen from the setting shown in the following figure, the master sends sync commands to groups 3 and 4 and freeze commands to groups 1 and 3. The outputs of the slave are frozen to (synchronized with) the current value as soon as the master sends the sync command to group 3. The states of the inputs are frozen to the current value, as soon as the master sends the freeze command to group 1 or group 3.



Member of	Group name	Sync mode	Freeze mode
□	Gr 1	□	☑
□	Gr 2	□	□
□	Gr 3	☑	☑
□	Gr 4	☑	□
□	Gr 5	□	□
□	Gr 6	□	□
□	Gr 7	□	□
□	Gr 8	□	□

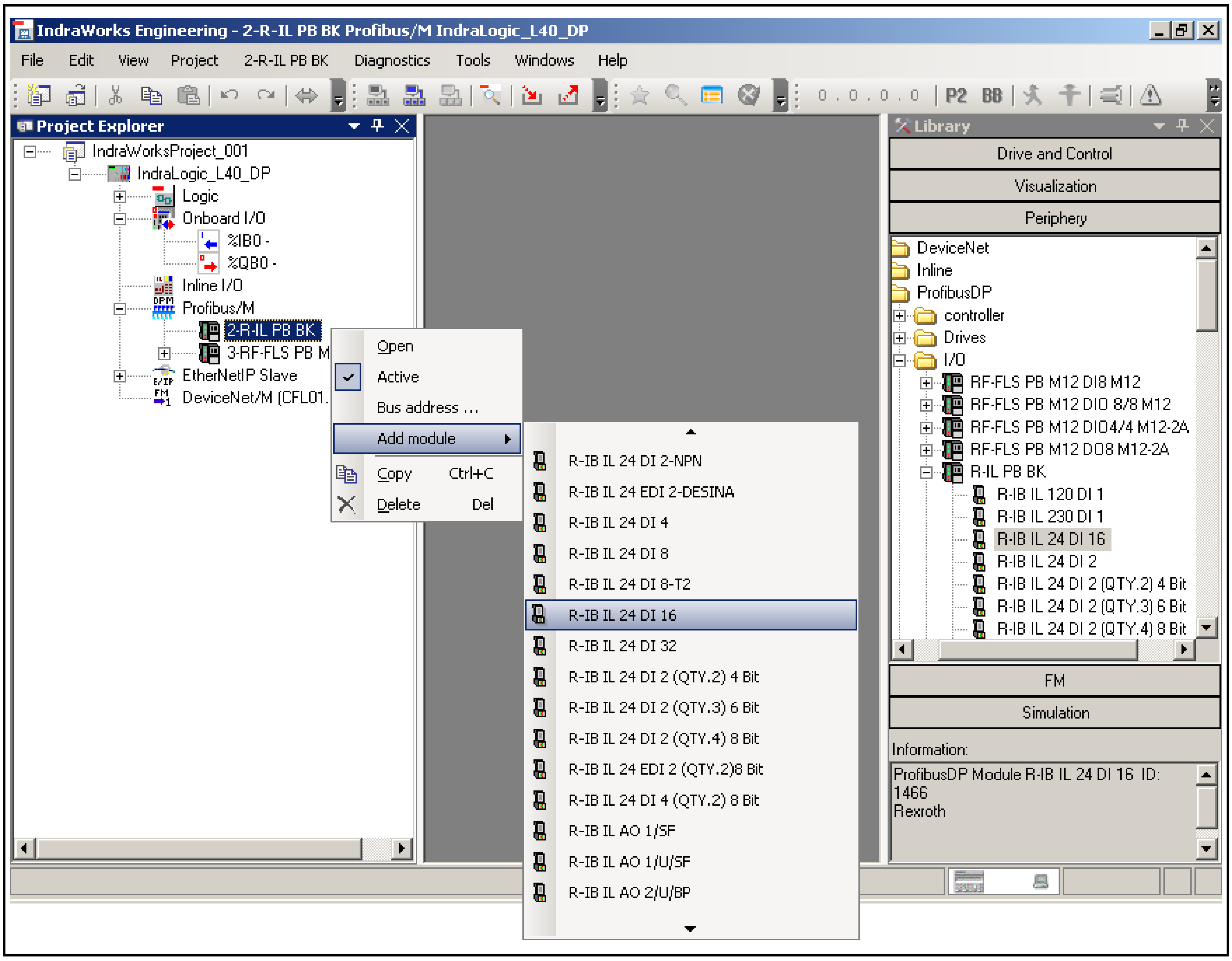
Fig.4-32: Group assignment of a Profibus DP slave (example)

4.6.5 Inserting Modules in a Profibus DP Slave

☞ Modules can only be inserted for Profibus DP slaves with modular structure, see chapter 4.6.4 "Configuring Profibus DP Slaves " on page 38.

The modules matching the particular Profibus DP slave can be found in the library **Periphery** ▸ **Profibus DP** below the particular Profibus DP slave. Drag the required modules from the library to the slave object. New modules can also be inserted between existing modules in the project explorer.

Alternatively, you can also use the "Add module" function of the context menu of the slave object, see fig. 4-33 "Inserting a module" on page 44. The new module will be the last module under the slave.



ProfibusDP Module R-IB IL 24 DI 16 ID: 1466
Rexroth

Fig.4-33: Inserting a module

4.6.6 Configuring Modules of a Profibus DP Slave

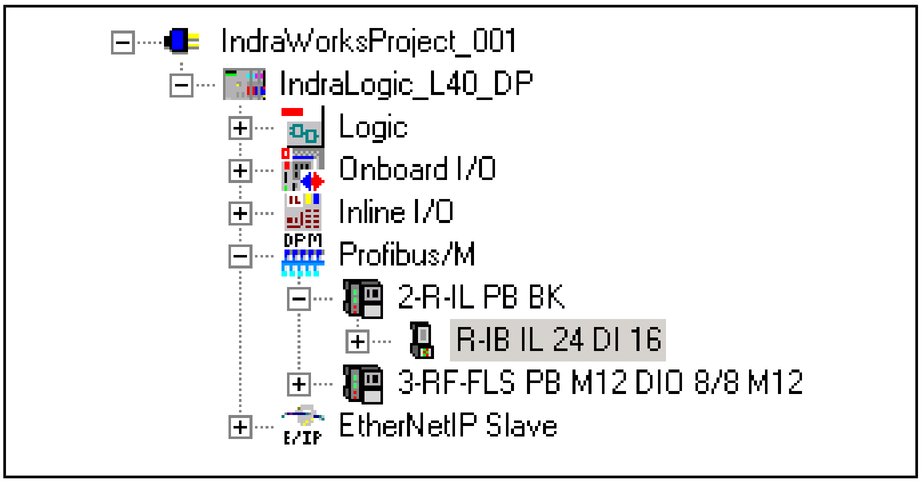
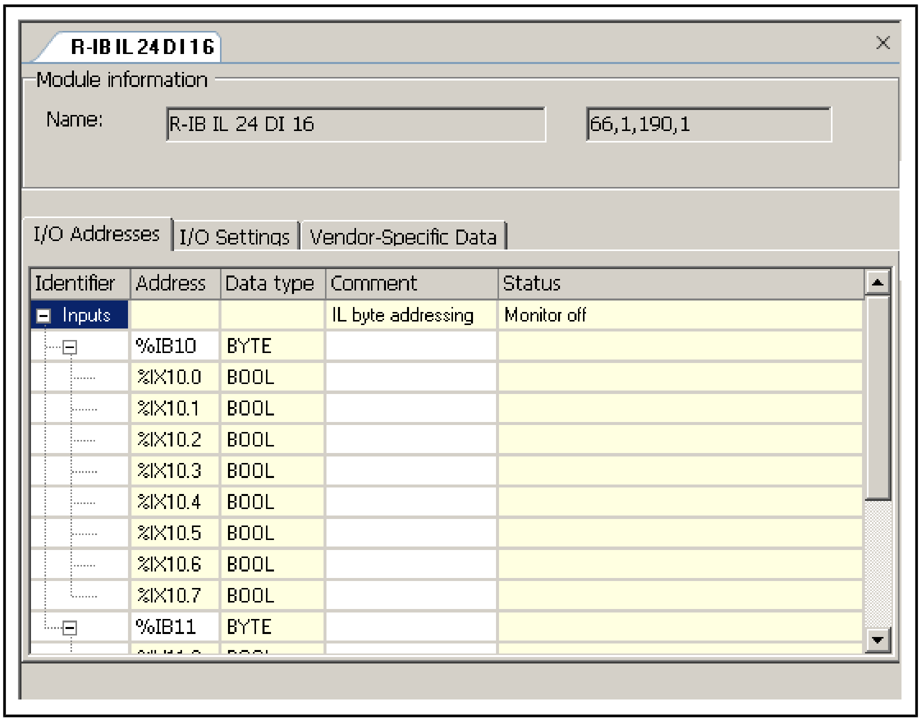


Fig.4-34: Module of a Profibus DP slave (example)

To configure a module, double-click on the corresponding module entry in the project explorer.

This opens the following window in the workspace:



R-IB IL 24 DI 16

Module information

Name: R-IB IL 24 DI 16 66,1,190,1

I/O Addresses | I/O Settings | Vendor-Specific Data

Identifier	Address	Data type	Comment	Status
⊟ Inputs			IL byte addressing	Monitor off
⊟	%IB10	BYTE		
	%IX10.0	BOOL		
	%IX10.1	BOOL		
	%IX10.2	BOOL		
	%IX10.3	BOOL		
	%IX10.4	BOOL		
	%IX10.5	BOOL		
	%IX10.6	BOOL		
	%IX10.7	BOOL		
⊟	%IB11	BYTE		

Fig.4-35: Window of a module (example)

"Module Information"

Name: Module name (box on the left) and internal module identification according to the GSD file (box on the right).

"I/O Addresses" Tab

This tab is for assigning the I/O areas of the modules to the physical addresses of the control (I/O addresses of the PLC).

Identifier: This column shows the input and output structure. Click on the plus or minus symbol to switch between the byte and bit views respectively.

You can also assign a symbolic address to each absolute address (double-click on the particular field). After it has been entered, the symbolic address is automatically created as a global variable in the PLC project.

The symbolic address of a node also appears in the project explorer. For an example of onboard I/O, refer to fig. 4-18 "Symbolic and absolute addresses of I/O objects" on page 30.

Address: I/O address. Enter the desired I/O address as byte address (e.g. %IB10). Entries in italics serve only for display purposes and cannot be edited.

☞ Automatic readdressing is possible in the "I/O Settings" tab.

Data type: "BYTE" stands for byte addresses, and "BOOL" for bit addresses.

Comment: Enter any comment regarding an address in this column.

Status: Physical status of the input/output. The status is only indicated in diagnostic mode during communication between IndraWorks and IndraLogic L40 DP.

"I/O Settings" Tab

This tab is for starting the automatic assignment of I/O addresses.

From input/From output: Current or requested initial addresses of the inputs and outputs. This tab provides only those input boxes that are relevant to the module funcitonality used (module with inputs, module with outputs). If, for example, you parameterize a module having inputs only, there will be no "From output" input box.

"**Apply**": Renumbers automatically all inputs and outputs of the module in ascending order, starting with the start addresses displayed (refer to "From input / From output"). Please note that any existing address gaps are closed during this process!

If the automatic numbering causes collisions with already assigned address areas, IndraWorks indicates the collision cause and automatically determines the next free address area.

"Vendor-Specific Data" Tab

This tab is for displaying and editing possibly available vendor-specific module parameters. This tab displays "vendor-specific data" with regard to the module if such data is provided in the GSD file.

For more information about "vendor-specific data," refer to section "Vendor-Specific Data Tab" on page 41.

4.7 Configuring IndraLogic L40 as Profibus DP Slave

4.7.1 Overview

The IndraLogic L40 DP can be addressed by a superordinate Profibus DP master as Profibus DP slave via the onboard Profibus DP interface (X7P).

The IndraLogic L40 DP must be defined as a Profibus DP slave, see chapter 4.3.2 "Device and Function Module Settings" on page 17. This is the only case where the required "Profibus/S" object is available in the project explorer.

4.7.2 Making Slave-Specific Basic Settings

Double-click on the "Profibus/**S**" object in the project explorer.

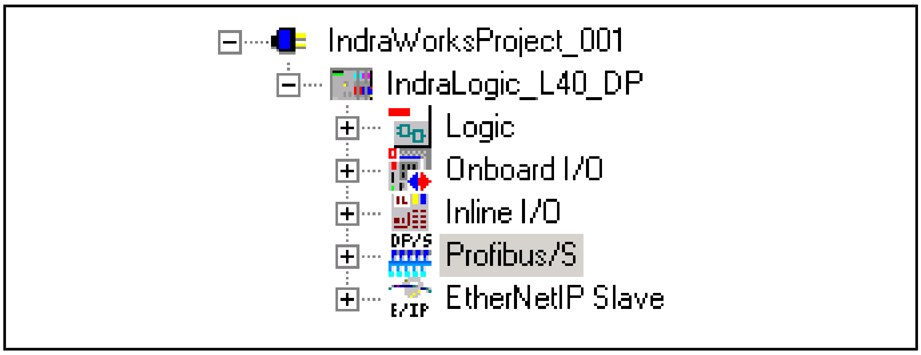
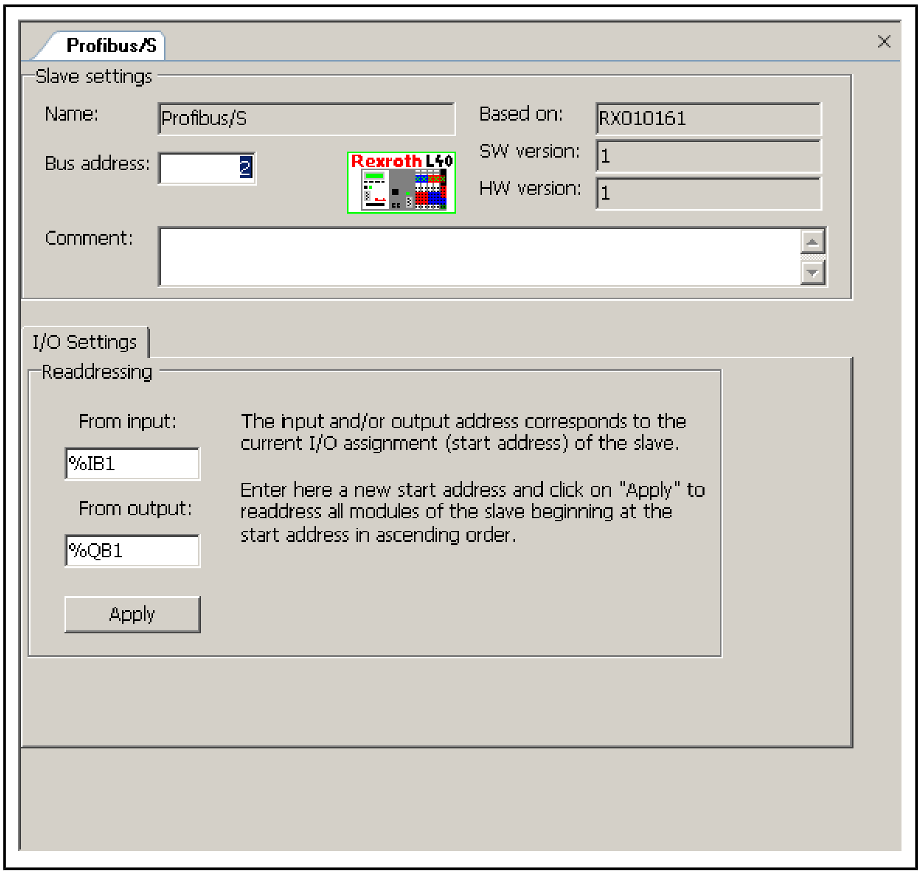


Fig.4-36: "Profibus/S" object

This opens the following window in the workspace:



Name: Profibus/S

Bus address: 2

Based on: RX010161

SW version: 1

HW version: 1

Comment:

From input: %IB1

From output: %QB1

The input and/or output address corresponds to the current I/O assignment (start address) of the slave.

Enter here a new start address and click on "Apply" to readdress all modules of the slave beginning at the start address in ascending order.

Fig.4-37: "Profibus/S" window

"Slave Settings"

Name: Internal device name of the slave

Bus address: Bus address of the slave (FDL address: Fieldbus Data Link). Address "2" is entered automatically. If necessary, enter a different address in this box.

Please note the following restrictions for the address selection:

Address "0": reserved for engineering devices

Address "1": reserved for the Profibus DP master

The bus address also appears in the project explorer, see fig. 4-27 "Profibus DP slave (example)" on page 39.

Based on: File name of the basic ident number according to the GSD file.

SW version: Software version according to the GSD file.

HW version: Hardware version according to the GSD file.

Comment: Enter any comment to describe the slave in detail.

"I/O Settings" Tab

This tab is for starting the automatic assignment of I/O addresses.

From input/From output: Desired start addresses for automatic numbering of the I/O coupling area (for a description see chapter 4.7.3 "Parameterizing the Coupling Area" on page 48).

"**Apply**": Renumbers automatically all inputs and outputs of the I/O coupling area in ascending order, starting with the start addresses displayed (see "From input / From output"). Please note that any existing address gaps are closed during this process!

If the automatic numbering causes collisions with already assigned address areas, IndraWorks indicates the collision cause and automatically determines the next free address area.

4.7.3 Parameterizing the Coupling Area

A coupling area is provided for data exchange between the IndraLogic L40 DP and the superordinate control if the IndraLogic L20 DP is configured as a Profibus DP slave. The coupling area is a reserved I/O area of the IndraLogic L40 DP which can, for example, be used to exchange status and diagnostic data.

The data capacity of the coupling area is preset to 16 bytes. The data capacity can be set separately for the input and output areas in increments of 8 bytes, ranging from 8 to 64 bytes. To achieve this, use the "Replace module" function of the context menu of the "Profibus/S" object, see fig. 4-38 "Selecting the data capacity for input/output area" on page 48.

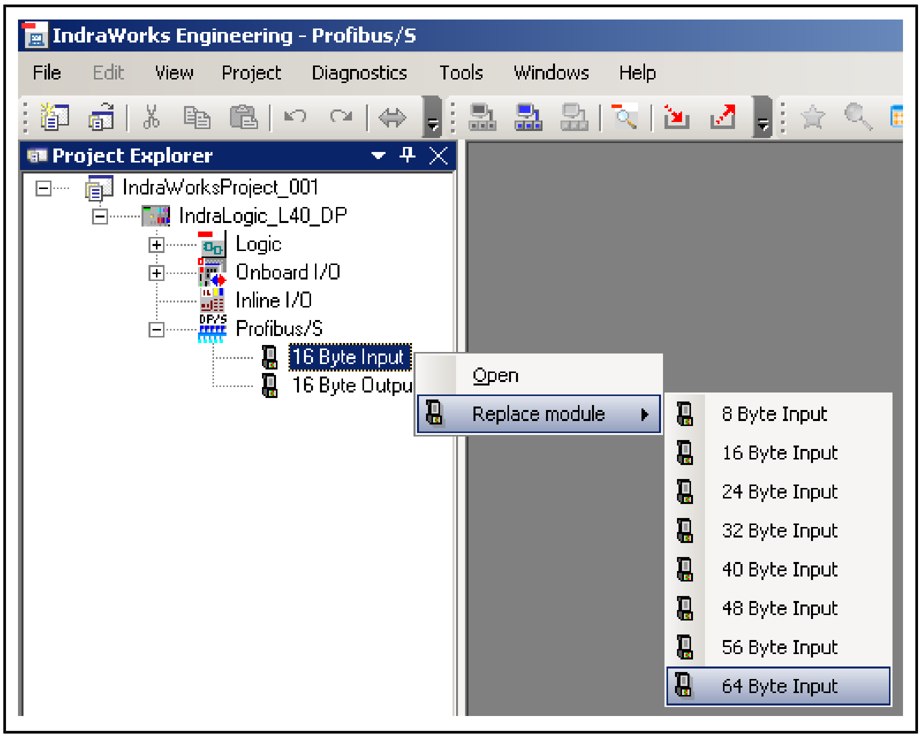
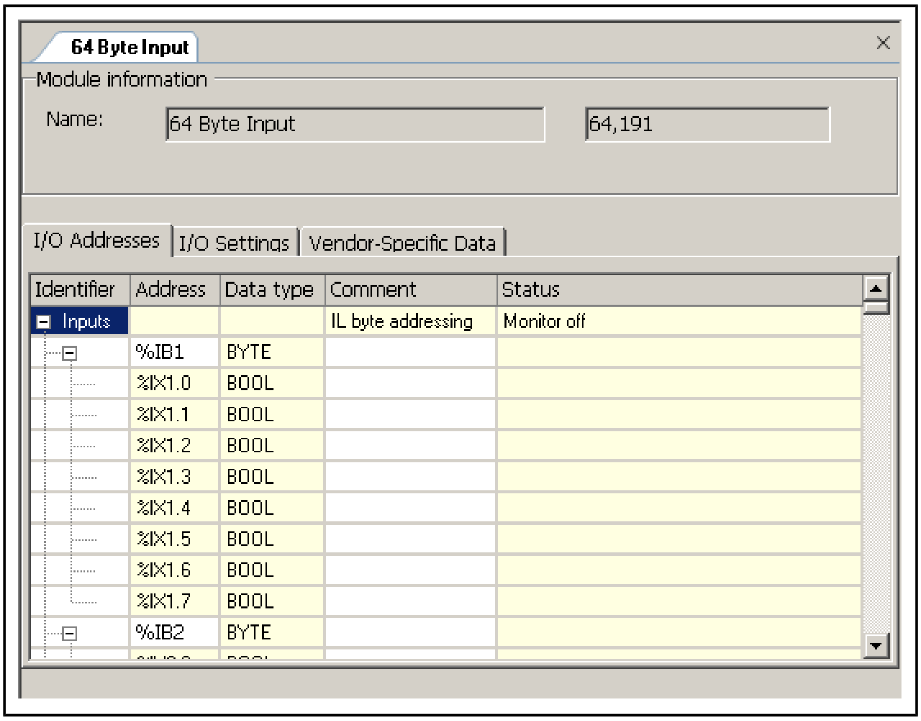


Fig.4-38: Selecting the data capacity for input/output area

To configure the inputs and outputs, double-click on the corresponding module entry in the project explorer.

This opens the following window in the workspace:



64 Byte Input

Module information

Name: 64 Byte Input 64,191

Identifier	Address	Data type	Comment	Status
Inputs			IL byte addressing	Monitor off
	%IB1	BYTE		
	%IX1.0	BOOL		
	%IX1.1	BOOL		
	%IX1.2	BOOL		
	%IX1.3	BOOL		
	%IX1.4	BOOL		
	%IX1.5	BOOL		
	%IX1.6	BOOL		
	%IX1.7	BOOL		
	%IB2	BYTE		

Fig.4-39: Coupling area of a "Profibus/S" object (example)

"Module Information"

Name: Module name (box on the left) and internal module identification (box on the right).

"I/O Addresses" Tab

This tab is for assigning the coupling area to the physical addresses of the control (I/O addresses of the PLC). The addresses are the local I/O addresses of the IndraLogic L40 DP.

Identifier: This column shows the input and output structure. Click on the plus or minus symbol to switch between the byte and bit views respectively.

You can also assign a symbolic address to each absolute address (double-click on the particular field). After it has been entered, the symbolic address is automatically created as a global variable in the PLC project.

The symbolic address of a node also appears in the project explorer. For an example of onboard I/O, refer to fig. 4-18 "Symbolic and absolute addresses of I/O objects" on page 30.

Address: I/O address. Enter the desired I/O address as byte address (e.g. %IB10). Entries in italics serve only for display purposes and cannot be edited.

☞ Automatic readdressing is possible in the "I/O Settings" tab.

Data type: "BYTE" stands for byte addresses, and "BOOL" for bit addresses.

Comment: Enter any comment regarding an address in this column.

Status: Physical status of the input/output. The status is only indicated in diagnostic mode during communication between IndraWorks and IndraLogic L40 DP.

"I/O Settings" Tab

This tab is for starting the automatic assignment of I/O addresses.

From input/From output: Current or desired start addresses for automatic numbering of the I/O coupling area.

"**Apply**": Renumbers automatically all inputs and outputs of the I/O coupling area in ascending order, starting with the start addresses displayed (see "From input / From output"). Please note that any existing address gaps are closed during this process!

If the automatic numbering causes collisions with already assigned address areas, IndraWorks indicates the collision cause and automatically determines the next free address area.

"Vendor-Specific Data" Tab This tab is of no relevance to coupling area configuration.

4.7.4 Projecting a Superordinate Control

To project the IndraLogic L40 DP as a Profibus DP slave for Profibus mode in the superordinate control, use the "RX010161" GSD file or the corresponding device in the "Periphery" library under **ProfibusDP ▸ PLC**, see the following figure:

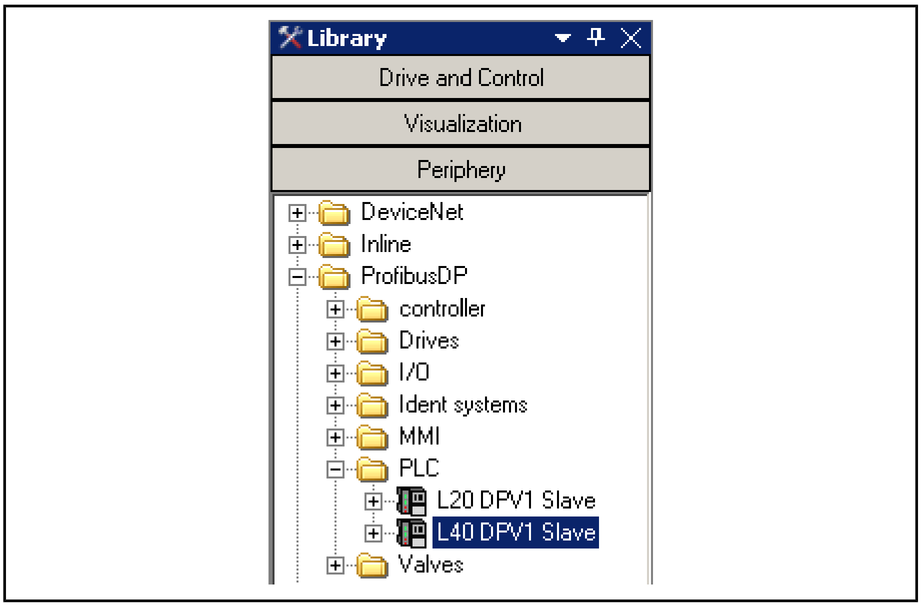


Fig.4-40: Projecting the IndraLogic L40 as slave in the superordinate control (device in the library)

While projecting the device, select those modules from the superordinated control, which correspond to the data capacity of the coupling area of the inputs and outputs. The following figure shows an example with a coupling area of 8 byte inputs and 8 byte outputs.

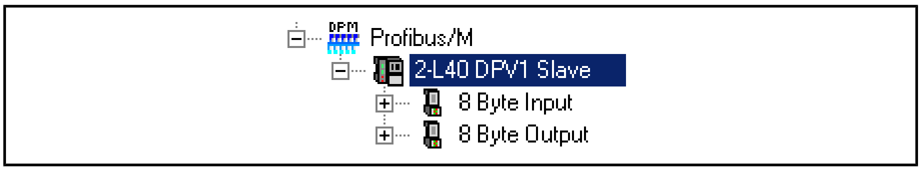


Fig.4-41: Defining the coupling area in the superordinate control

4.8 IndraLogic L40 as Ethernet/IP Slave

4.8.1 Configuring IndraLogic L40 as an Ethernet/IP Slave

Overview

The IndraLogic L40 DP can be addressed by a superordinate control (master/originator) as Ethernet/IP slave (target) via the Ethernet interface (X7E).

The IndraLogic L40 DP must be defined as an Ethernet/IP slave, refer to chapter 4.3.2 "Device and Function Module Settings" on page 17 and chapter "Inserting an EthernetIP Slave Object Subsequently " on page 24. This is the only case where the required subordinate "Ethernet/IP slave" object is available in the project explorer.

Making Slave-Specific Basic Settings

Double-click on the "Ethernet/IP slave" object in the project explorer.

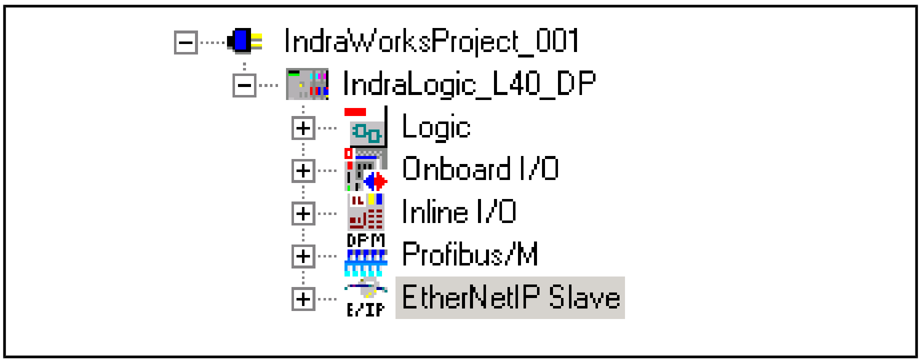
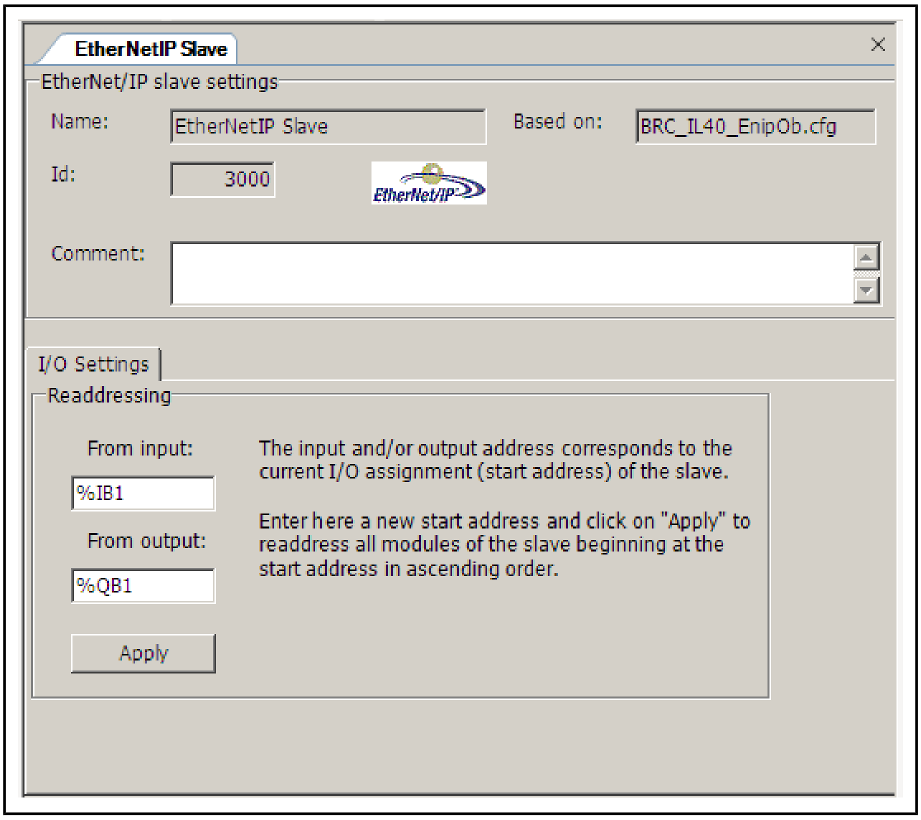


Fig.4-42: "Ethernet/IP slave" object

This opens the following window in the workspace:



Name: EtherNetIP Slave

Based on: BRC_IL40_EnipOb.cfg

Id: 3000

From input: %IB1

From output: %QB1

The input and/or output address corresponds to the current I/O assignment (start address) of the slave.

Enter here a new start address and click on "Apply" to readdress all modules of the slave beginning at the start address in ascending order.

Fig.4-43: "Ethernet/IP slave" window

"EtherNet/IP Slave Settings"

Name: Internal device name of the slave

Based on: Name of the basic device description file. The file is an integral part of the installed target system.

ID: Internal ID, specified by the installed target system.

Comment: Enter any comment to describe the slave in detail.

"I/O Settings" Tab

This tab is for starting the automatic assignment of I/O addresses.

From input/From output: Desired start addresses for automatic numbering of the I/O coupling area.

"**Apply**": Renumbers automatically all inputs and outputs of the I/O coupling area in ascending order, starting with the start addresses displayed (see "From input / From output"). Please note that any existing address gaps are closed during this process!

If the automatic numbering causes collisions with already assigned address areas, IndraWorks indicates the collision cause and automatically determines the next free address area.

Parameterizing the Coupling Area

A coupling area is provided for data exchange between the IndraLogic L40 DP and the superordinate control if the IndraLogic L20 DP is configured as an Ethernet/IP slave. The coupling area is a reserved I/O area of the IndraLogic L40 DP which can, for example, be used to exchange status and diagnostic data.

The data capacity of the coupling area is preset to 8 byte input and 8 byte output plus 4 byte Run Header. The data capacity can be set separately for the input and output areas in increments of 8 bytes, ranging from 0 to 128 bytes. To achieve this, use the "Replace module" function of the context menu of the "EtherNet/IP slave" object, see fig. 4-44 "Selecting the data capacity for the input and output fields of the Ethernet/IP slave" on page 52.

☞ If the input and output fields are parameterized with data capacities equal to 0 (module "0 byte input" and "0 byte output..."), cyclic communication is no longer possible.

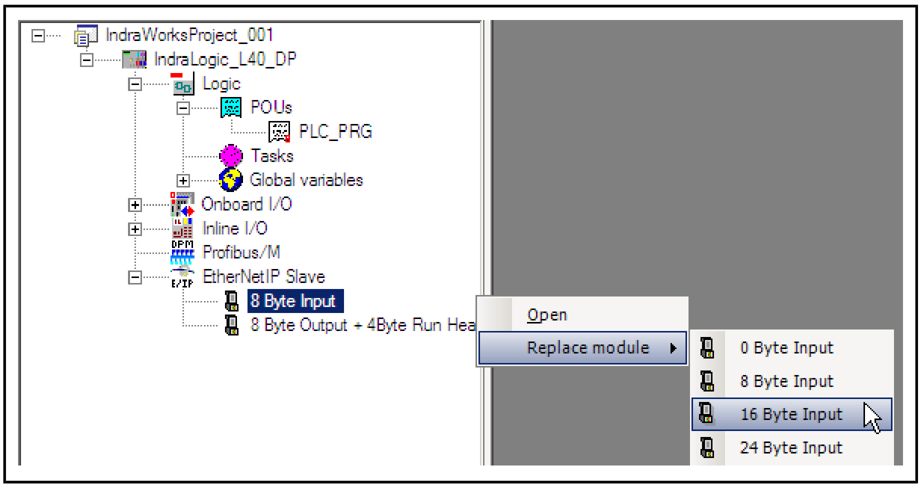
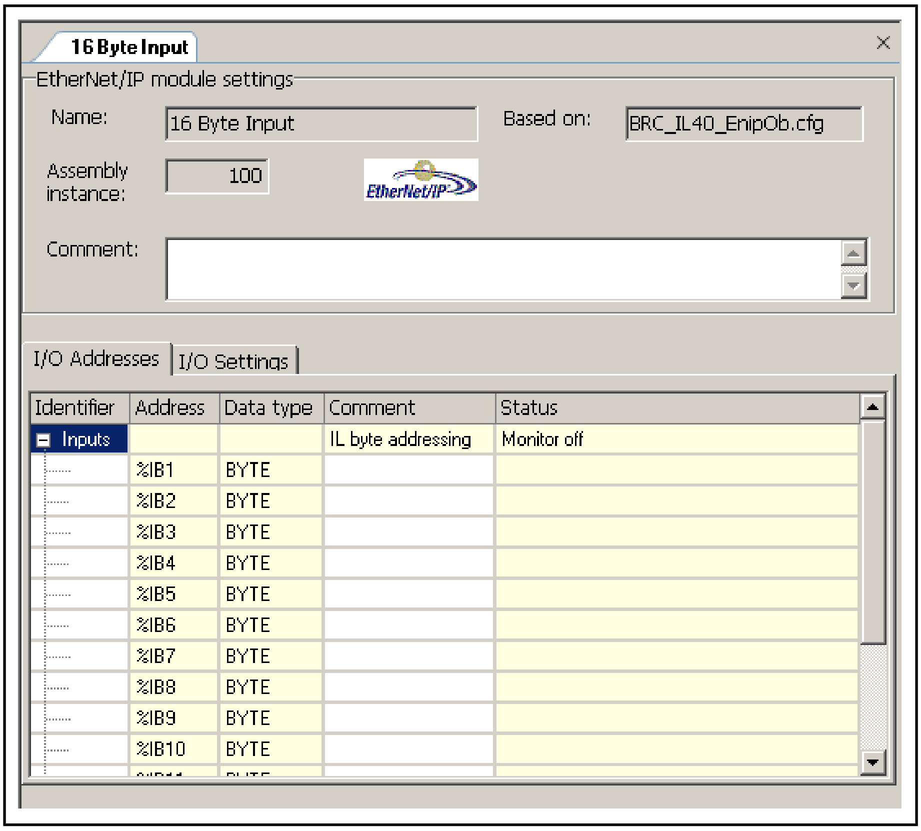


Fig.4-44: Selecting the data capacity for the input and output fields of the Ethernet/IP slave

To configure the inputs and outputs, double-click on the corresponding module entry in the project explorer.

This opens the following window in the workspace:



Name: 16 Byte Input

Based on: BRC_IL40_EnipOb.cfg

Assembly instance: 100

Comment:

Identifier	Address	Data type	Comment	Status
Inputs			IL byte addressing	Monitor off
	%IB1	BYTE		
	%IB2	BYTE		
	%IB3	BYTE		
	%IB4	BYTE		
	%IB5	BYTE		
	%IB6	BYTE		
	%IB7	BYTE		
	%IB8	BYTE		
	%IB9	BYTE		
	%IB10	BYTE		

Fig.4-45: Coupling area of an "EtherNet/IP slave" object (example)

"Ethernet/IP Module Settings"

Name: Internal name, specified by the installed target device.

Based on: File name of the used device description file. The file is an integral part of the installed target system.

Assembly instance: Internal identification.

Comment: Enter any text to describe the I/O field in detail.

"I/O Addresses" Tab

This tab is for assigning the coupling area to the physical addresses of the control (I/O addresses of the PLC). The addresses are the local I/O addresses of the IndraLogic L40 DP.

Identifier: This column shows the input and output structure. Click on the plus or minus symbol to switch between the byte and bit views respectively.

You can also assign a symbolic address to each absolute address (double-click on the particular field). After it has been entered, the symbolic address is automatically created as a global variable in the PLC project.

The symbolic address of a node also appears in the project explorer. For an example of onboard I/O, refer to fig. 4-18 "Symbolic and absolute addresses of I/O objects" on page 30.

Address: I/O address. Enter the desired I/O address as byte address (e.g. %IB10). Entries in italics serve only for display purposes and cannot be edited.

☞ Automatic readdressing is possible in the "I/O Settings" tab.

Data type: "BYTE" stands for byte addresses, and "BOOL" for bit addresses.

Comment: Enter any comment regarding an address in this column.

Status: Physical status of the input/output. The status is only indicated in diagnostic mode during communication between IndraWorks and IndraLogic L40 DP.

"I/O Settings" Tab

This tab is for starting the automatic assignment of I/O addresses.

From input/From output: Current or desired start addresses for automatic numbering of the I/O coupling area.

Apply: Renumbers automatically all inputs and outputs of the I/O coupling area in ascending order, starting with the start addresses displayed (see "From input / From output"). Please note that any existing address gaps are closed during this process!

If the automatic numbering causes collisions with already assigned address areas, IndraWorks indicates the collision cause and automatically determines the next free address area.

Ethernet/IP Connection Types

If the IndraLogic L40 DP is configured as an Ethernet/IP slave, data blocks can be cyclically transmitted ("implicit messaging") via an "Exclusive Owner connection" (transport class 1) and a "Listen Only connection" (transport class 1). If the "Listen Only connection" is established in addition to the existing "Exclusive Owner connection", a second master can read the Ethernet/IP output image of the IndraLogic L40 DP via multicast.

The input and output field sizes can each be parameterized in a range from 0 to 128 bytes in increments of 8 bytes.

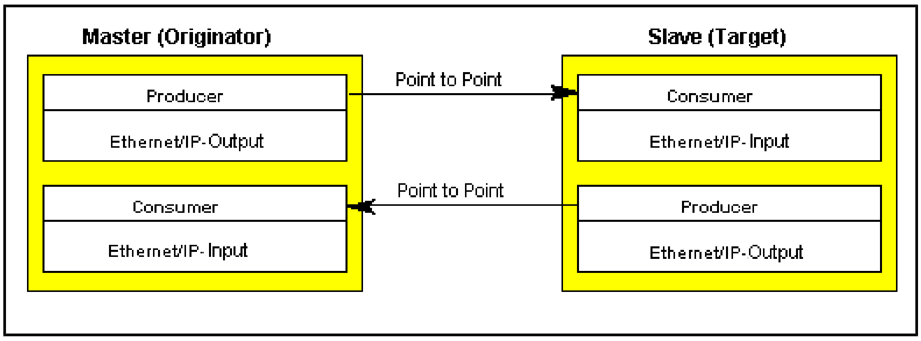


Fig.4-46: Ethernet/IP connection type: point to point

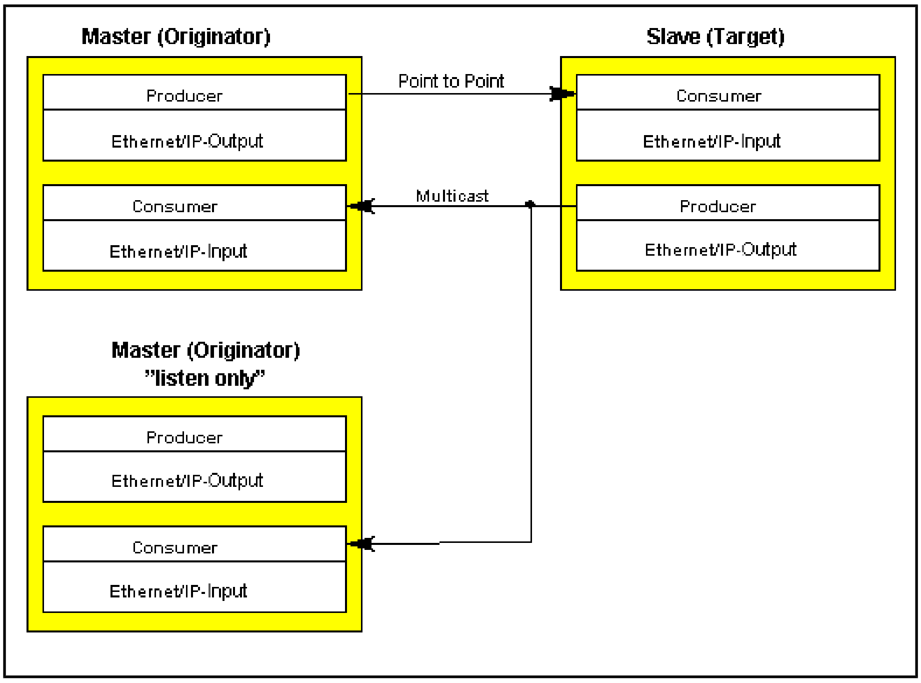


Fig.4-47: Ethernet/IP connection type: point to point and multicast

Features of Cyclic Data Transmission

- Cyclic data transmission requires that the Ethernet connection is in full-duplex mode. Otherwise, there might be transmission timeouts.
- The lowest cycle time RPI (Requested Packet Interval) supported by the IndraLogic L40 DP is 5 ms.
- The Ethernet/IP "Idle/Run" header (32 bits) is supported. The header is not visible in the cyclic I/O image of the IndraLogic L40 DP.

Reaction of the IndraLogic L40 DP:

- Idle state of the master (bit 0 in the Idle/Run header is not set): the input image is set to zero.
- Stop state of the IndraLogic L40 DP: the output image is set to zero while the input image is still updated.
- Transmission timeout: the input image is set to zero.

The consumer instance of the IndraLogic L40 DP monitors the cyclic transmission of the master output image by means of a monitoring time (timeout). To achieve this, the IndraLogic L40 DP receives the appropriate parameters while the cyclic data channel is initialized. The resulting monitoring time can be calculated from the following formulas:

$$TM = 2^{(TMV+2)}$$
$$t = TM * RPI$$

TMV:	Timeout Multiplier Value
TM:	Timeout Multiplier
RPI:	Requested Packet Interval (in µs)
t:	Monitoring time (in ms)

Fig.4-48: Formula for calculating the monitoring time t

Examples:

TMV	TM	RPI (in µs)	t (in ms)
0	4	5000	20
1	8	5000	40
2	16	5000	80
3	32	5000	160
4	64	5000	320
2	16	10000	160

TMV	Timeout Multiplier Value
TM	Timeout Multiplier
RPI	Requested Packet Interval
t	Monitoring time

Fig.4-49: Resulting monitoring times (examples)

4.8.2 Configuring IndraLogic L40 as CIP Data Server

Overview

If configured as CIP data server (CIP: Communication Industrial Protocol), the IndraLogic L40 DP provides enabled variables via the onboard Ethernet interface (X7E), using acyclic DataTable services ("unconnected explicit messaging" and "Class 3 connected messaging").

Projecting

To use the IndraLogic L40 DP as a CIP data server, proceed as follows:

1. Define the IndraLogic L40 DP as an Ethernet/IP slave, refer to chapter 4.3.2 "Device and Function Module Settings" on page 17. This is the only case where the required subordinate "Ethernet/IP slave" object is available in the project explorer.
2. If additional cyclic communication is not rquired, parameterize the data capacities of the I/O coupling area such that they are equal to 0. To achieve this, use the "Replace module" item of the particular contect menu and select the "0 Byte Input" or "0 Byte Output..." module, see fig. 4-44 "Selecting the data capacity for the input and output fields of the Ethernet/IP slave" on page 52.
3. Define the appropriate variables in IndraLogic.
4. Ensure that the "Dump symbol entries" and "Dump XML symbol table" options are activated in IndraLogic under **Project ▸ Options ▸ Symbol configuration**.
5. Open the "Set object attributes" dialog in IndraLogic by selecting **Project ▸ Options ▸ Symbol configuration ▸ Configure symbol file**. Select the desired variables and activate the "Export variables of the object" option.

☞ The "Write access" option affects the write protection of a variable. To avoid external overwriting of the variable value, deactivate the write access option.

Features of Acyclic Data Transmission

- The maximum number of connections to CIP data clients that can be established at the same time is six.
- The variables are addressed via their name.

- Access is possible to variables (INT, DINT, ...) and to one-dimensional variable arrays (up to 450 bytes).
- The CIP data server acknowledges the read and write accesses. Reasons for negative acknowledgements are listed in the following table:

Description	Error code	Additional error code
No symbol information for variable available	0x1F	0x0101
Variable is read-only	0x1F	0x0106
Too much data sent during the write access	0x1F	0x0107
Not enough data sent during the write access	0x1F	0x0108

Fig.4-50: Error codes of the CIP data server

4.8.3 Configuring IndraLogic L40 as CIP Data Client

If configured as CIP data client, the IndraLogic L40 DP can access appropriately enabled variables of a CIP data server via the Onboard Ethernet interface (X7E), using acyclic DataTable services.

To achieve this, the "RIL_EtherNetIP" library must be used.

To use the IndraLogic L40 DP as a CIP data client, proceed as follows:

1. Define the IndraLogic L40 DP as an Ethernet/IP slave, refer to chapter 4.3.2 "Device and Function Module Settings" on page 17. This is the only case where the required subordinate "Ethernet/IP slave" object is available in the project explorer.
2. If additional cyclic communication is not rquired, parameterize the data capacities of the I/O coupling area such that they are equal to 0. To achieve this, use the "Replace module" item of the particular contect menu and select the "0 Byte Input" or "0 Byte Output..." module, see fig. 4-44 "Selecting the data capacity for the input and output fields of the Ethernet/IP slave" on page 52.
3. Ensure that the CIP data server provides the relevant variables.
4. Define all variables required for the used function blocks of the "RIL_EtherNetIP" library in the PLC program. Call the necessary function blocks as appropriate.

4.9 Configuring IndraLogic L40 as DeviceNet Master

4.9.1 General Information

The DeviceNet master function module enables a connection of DeviceNet slaves and the access via the standardized DeviceNet protocol (EN 50325).

To use the IndraLogic L40 DP as DeviceNet master, proceed as follows:

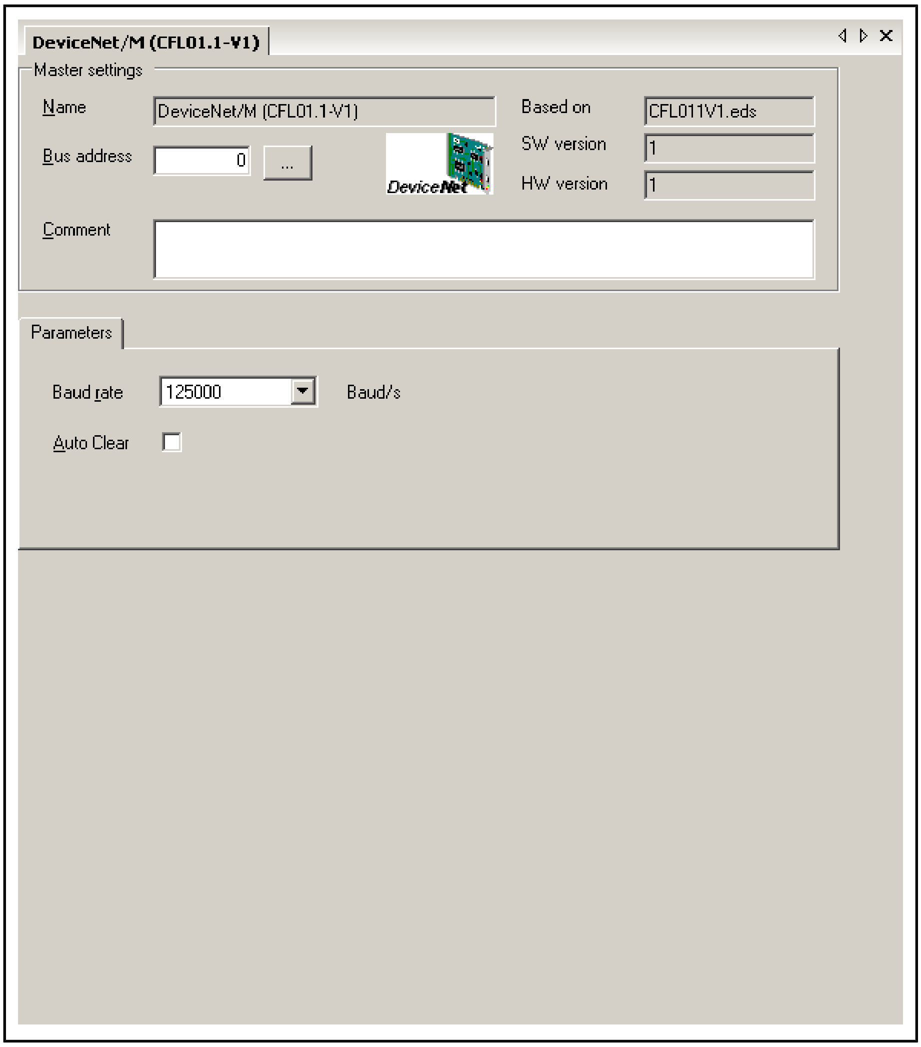
1. Define the IndraLogic L40 DP as DeviceNet master, refer to chapter 4.3.2 "Device and Function Module Settings" on page 17 and chapter "Inserting an DeviceNet/M Object Subsequently " on page 24. This is the only case where the required "DeviceNet/M" object is available in the project explorer.
2. Make the appropriate master-specific settings.
3. Insert DeviceNet slaves in the "DeviceNet/M" object in the project explorer.
4. Configure inserted DeviceNet slaves.

The maximum slave number is 63. 3.5 kBytes in total are available for IN and OUT.

4.9.2 Making Master-Specific Settings

Double-click on the appropriate "DeviceNet/M" object in the project explorer.

This opens the following window in the workspace:



DeviceNet/M (CFL01.1-V1)

Master settings

Name: DeviceNet/M (CFL01.1-V1)

Based on: CFL011V1.eds

Bus address: 0

SW version: 1

HW version: 1

Comment:

Parameters

Baud rate: 125000 Baud/s

Auto Clear: □

Fig.4-51: "DeviceNet/M" window

Master Settings

Name: Internal device name of the DeviceNet master

Bus address: Bus address of the DeviceNet master. Address "0" is entered automatically. If necessary, enter a different address in this box.

Comment: Enter any comment to describe the DeviceNet master in detail.

Based on: The EDS file (EDS: Electronic Data Sheet) contains the settings options of the module and is provided by the manufacturer of the device.

SW version: Software version according to the EDS file.

HW version: Hardware version according to the EDS file.

"Parameters" Tag

Contains the parameters required for the DeviceNet master operation.

⚠ CAUTION

Modifications of the parameters might cause an unpredictable system behavior!

Thus, the parameters may only be modified by instructed DeviceNet specialists, who are aware of the effects of this modifications!

Baud rate: Command data transfer rate of DeviceNet network in baud. All connected slaves have to support the specified value.

Possible settings: 125000, 250000, 500000

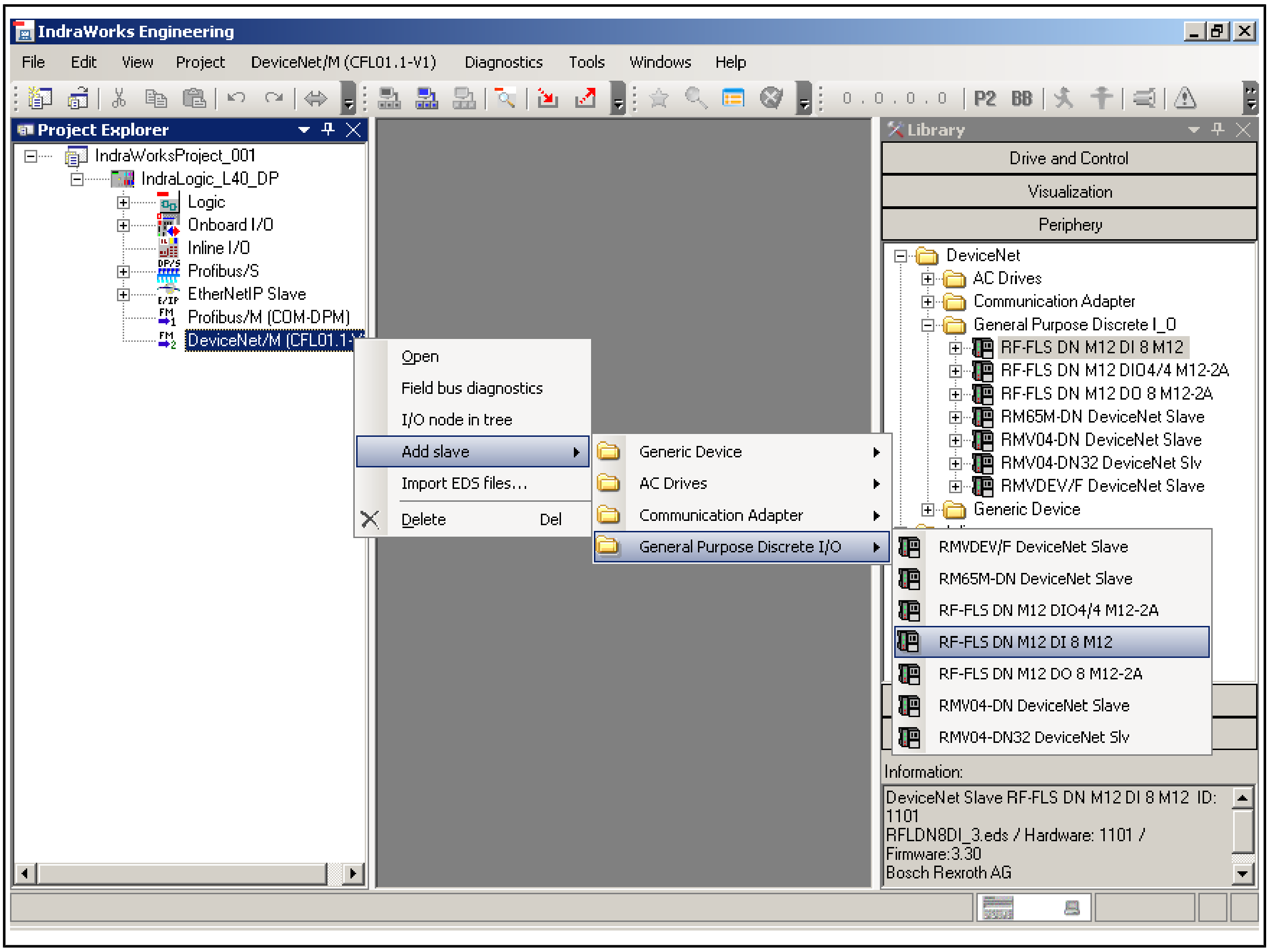
Auto Clear: Activate this option so that the master sets the outputs of all remaining slaves to safe status using a slave after a communication error.

4.9.3 Insert DeviceNet Slaves

All DeviceNet slaves provided for the IndraLogic L40 DP can be found in the "Periphery" library under "DeviceNet". Move the required slaves from the library to the "DeviceNet" object by drag-and-drop. New slaves can be also inserted in the project explorer between already existing slaves.

Alternatively, you can also use the "Add slave" function of the context menu of the "DeviceNet/M" object, refer to fig. 4-52 "Inserting a DeviceNet slave via the context menu of the "DeviceNet/M" object" on page 60. The new slave will be the last slave under "DeviceNet/M".

☞ If a required slave is not available in the library, it can be integrated in the library by importing its EDS file via the "Import EDS files..." function of the context menu of the "DeviceNet/M" object.

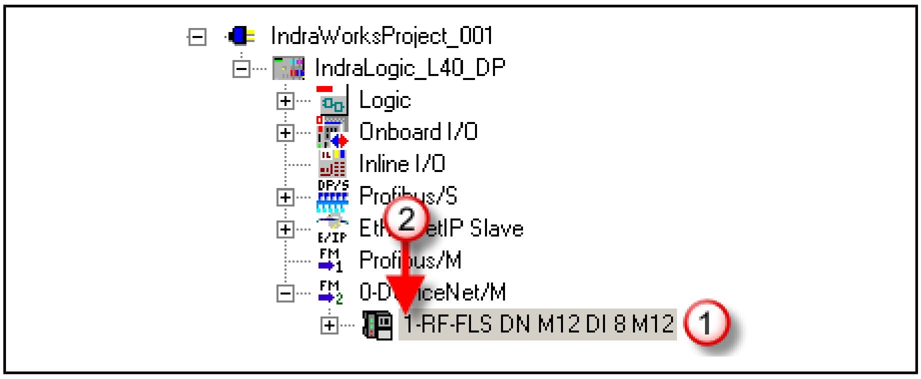


DeviceNet Slave RF-FLS DN M12 DI 8 M12 ID: 1101
RFLDN8DI_3.eds / Hardware: 1101 / Firmware:3.30
Bosch Rexroth AG

Fig.4-52: Inserting a DeviceNet slave via the context menu of the "DeviceNet/M" object

4.9.4 Configure DeviceNet Slaves

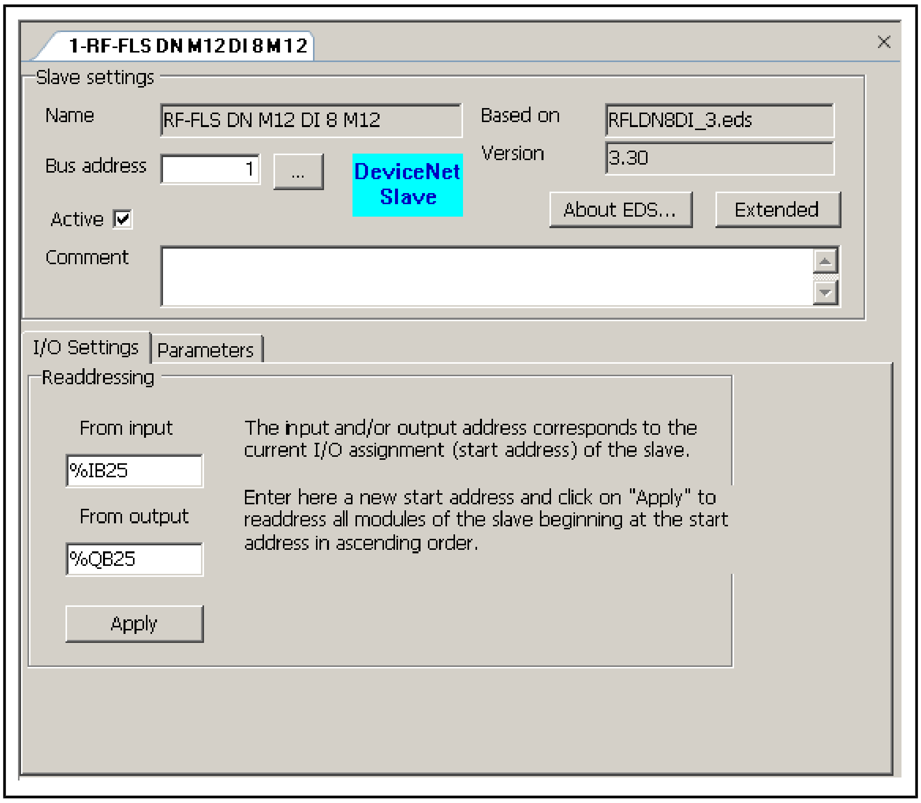
General Information



① DeviceNet slave
② current bus address

Fig.4-53: DeviceNet slave (example)

To configure a DeviceNet slave, double-click on the corresponding slave entry in the project explorer. This opens a window in the workspace:



Name: RF-FLS DN M12 DI 8 M12
Based on: RFLDN8DI_3.eds
Bus address: 1
Version: 3.30
Active ☑
Comment:

From input: %IB25
From output: %QB25

The input and/or output address corresponds to the current I/O assignment (start address) of the slave.

Enter here a new start address and click on "Apply" to readdress all modules of the slave beginning at the start address in ascending order.

Fig.4-54: Window of a DeviceNet slave (example)

"Slave Settings"

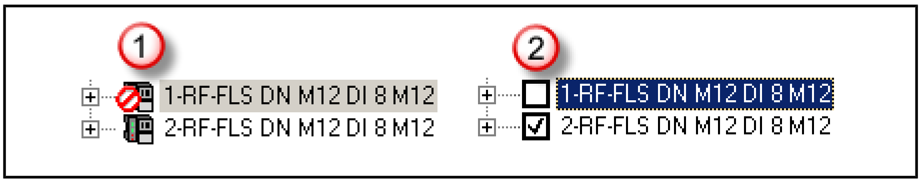
Name: Device name according to EDS file. The EDS file contains the settings options of the slave and is provided by the manufacturer of the device.

Bus address: Slave bus address. Here, IndraWorks enters the next free bus address automatically. If necessary, enter a different address in this box. Alternatively, click on the "..." button to open the "DeviceNet bus addresses" dialog. Here, the complete address assignment of all DeviceNet devices is indicated. Double-click on a free field in the "Status" table column to apply the corresponding bus address to the currently selected slave.

The bus address of the slave also appears in the project explorer, refer to chapter "General Information" on page 60.

Active: Activate this option, so that the slave can be commissioned at the DeviceNet master after the next program download (refer to chapter 4.12 "Download and Commissioning" on page 81). Deactivate this option, if the slave is to be configured and archived, but not to be commissioned at the DeviceNet.

The setting (active/not active) can be also recognized and made in the project explorer. If the mouse pointer is positioned on the slave, the setting can be changed by a left mouse-click, see the following figure (2).



① Display of the settings "not active" (first line) and "active" (second line).
② Display of the settings, if the mouse pointer is positioned on the slave.
Fig.4-55: Setting "active"/"not active" in the project explorer

Based on: File name of the used EDS file.

Version: Hardware version according to the EDS file.

Button "EDS info": Shows the EDS file content.

Button "Advanced":

- UCMM (Unconnected Message Manager for multiple connections): Activate this option so that the connection can be established via UCMM. If the slave does not recognize UCMM, the connection establishment is delayed by 2 seconds. Possible classifications: Group1 to Group5. Default setting: Activated, Group3
- Start: Specification of check, upon **start** of the network for the DeviceNet slave. During a check, the system compares the corresponding module data to the values contained in the EDS file.

Comment: Enter any comment to describe the slave in detail.

"I/O Settings" Tab

This tab is for starting the automatic assignment of I/O addresses.

From input/From output: Desired start addresses for automatic numbering of inputs and outputs of the DeviceNet slave.

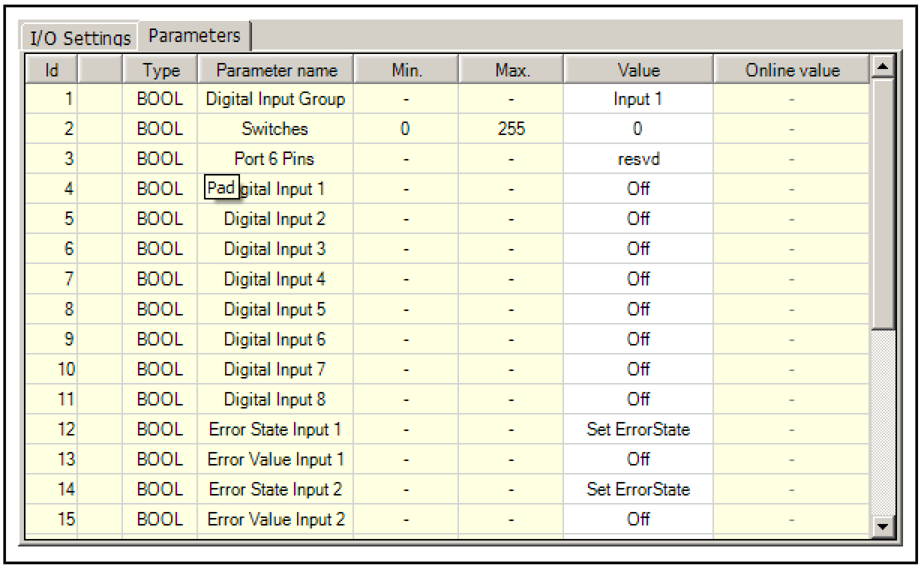
Renumbers automatically all inputs and outputs of the subordinate DeviceNet slave modules in ascending order, starting with the start addresses displayed (see "From input / From output"). Please note that any existing address gaps are closed during this process!

If the automatic numbering causes collisions with already assigned address areas, IndraWorks indicates the collision cause and automatically determines the next free address area.

☞ Automatic numbering of subordinate modules can also be made in the module configuration.

"Parameters" Tag

Displaying and processing of module-specific parameters of the DeviceNet slave. The parameters are defined in the EDS file.



Id		Type	Parameter name	Min.	Max.	Value	Online value
1		BOOL	Digital Input Group	-	-	Input 1	-
2		BOOL	Switches	0	255	0	-
3		BOOL	Port 6 Pins	-	-	resvd	-
4		BOOL	Pad gital Input 1	-	-	Off	-
5		BOOL	Digital Input 2	-	-	Off	-
6		BOOL	Digital Input 3	-	-	Off	-
7		BOOL	Digital Input 4	-	-	Off	-
8		BOOL	Digital Input 5	-	-	Off	-
9		BOOL	Digital Input 6	-	-	Off	-
10		BOOL	Digital Input 7	-	-	Off	-
11		BOOL	Digital Input 8	-	-	Off	-
12		BOOL	Error State Input 1	-	-	Set ErrorState	-
13		BOOL	Error Value Input 1	-	-	Off	-
14		BOOL	Error State Input 2	-	-	Set ErrorState	-
15		BOOL	Error Value Input 2	-	-	Off	-

Fig.4-56: Parameter table of a DeviceNet slave (example)

☞ The tooltip displays further information in the columns "ID", "Type", "Parameter", "Value" and "Online value":

The Connection Path is displayed in the "ID" column.

The object adress is entered in the "Type" column as class/instance/attribute.

The parameter short information is entered in the "Parameter name" column.

The parameter default values are entered in the "Value" column of the EDS file.

The slave value during online mode can be displayed in the "Online value" column.

For that, the mouse cursor has to be pointed on the desired field.

In order to change a parameter value, double-click on the corresponding table line on the "Value" field. Depending on the parameter type, the value can be entered directly or via a selection list within the permissible range limits (fields "Min" and "Max"). As soon as a value is changed, it is displayed in the second table column, see fig. 4-57 "Parameter value status: Flag in table column 2" on page 63.

☞ All changes only become active after the next program download! The settings have to be applied to IndraLogic. Use the context menu function "Apply changed parameters or parameters marked with default values to IndraLogic" in the "Value" column.

Flag	Meaning
(blank)	No change of value
🔒	**This parameter is blocked**: It is not possible to change the value.
✏	**Parameter is no default value**: The changed value is transferred to the control during the next download.
(blank)	**Transfer default value to control**: This means that the default value can be written in the device during each connection establishment to the device. Thus, a defective device can be replayed by a non-configured device. This is also called "Automatic Device Replacement" (ADR). This setting should be limited to relevant individual parameters in order to avoid memory space problems in the control!
☞	**Parameter with default value are highlighted for the transfer to IndraLogic**: This means that the default value is transferred to IndraLogic and subsequently to the control.

Fig.4-57: Parameter value status: Flag in table column 2

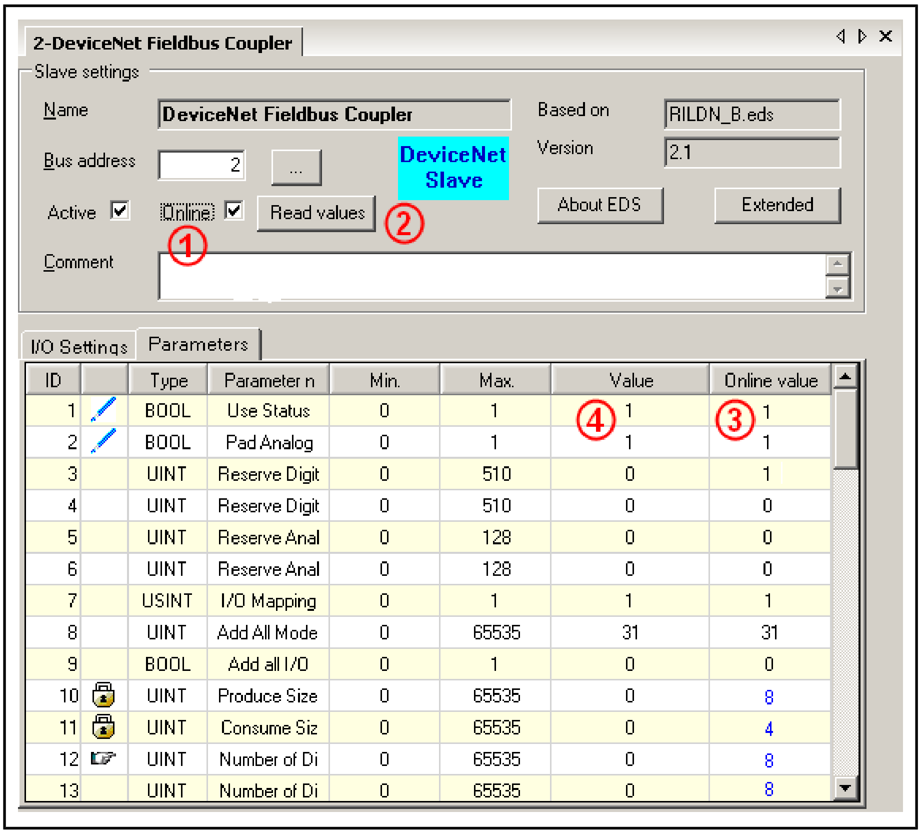
The parameter value statuses can be influenced via the context menu of the "Value" column. No context menu is displayed in case of write-protected parameters. The following options are available:

Context menu function	Effect
Hightlight parameter with default value for transfer to IndraLogic	The current parameter is set to ☞ state.
Hightlight parameter with default value for transfer to IndraLogic	All parameters are set to ☞ state. If parameters are changed (✏), the respective default value is not used for these parameters.

Context menu function	Effect
Cancel highlighting of parameter with default value	Highlighting ☞ of current parameter is reset.
Cancel highlighting of parameter with default value	All highlighted ☞ parameters are reset.
Reset parameters to default	The changed parameter (✎) is reset to the default value of the EDS file.
Reset all parameters to default	The changed parameter (✎) is reset to the default value of the EDS file.
Transfer changed parameters or parameters labelled with default values to IndraLogic	All highlighted and changed parameters are entered in a configuration file in the IndraLogic software. The changed data are thus loaded to the control upon the next download.
Tranfer parameter value to slave	The value of the currently highlighted parameter is directly transferred to the slave.

Fig.4-58: Context menu of the "Value" column

The **Online functions** are available if a connection from IndraWorks to the control is established and if a connection from the control via the DeviceNet to the slave is established. The field "Active" has to be checked in order to establish a connection. The online values are displayed in the tab parameter, see fig. 4-59 "Parameter table of a DeviceNet slave in online mode (example)" on page 65.



2-DeviceNet Fieldbus Coupler

Slave settings

Name: DeviceNet Fieldbus Coupler

Based on: RILDN_B.eds

Version: 2.1

Bus address: 2

Active ☑ Online ☑

Comment:

ID		Type	Parameter n	Min.	Max.	Value	Online value
1	✎	BOOL	Use Status	0	1	1	1
2	✎	BOOL	Pad Analog	0	1	1	1
3		UINT	Reserve Digit	0	510	0	1
4		UINT	Reserve Digit	0	510	0	0
5		UINT	Reserve Anal	0	128	0	0
6		UINT	Reserve Anal	0	128	0	0
7		USINT	I/O Mapping	0	1	1	1
8		UINT	Add All Mode	0	65535	31	31
9		BOOL	Add all I/O	0	1	0	0
10	🔒	UINT	Produce Size	0	65535	0	8
11	🔒	UINT	Consume Siz	0	65535	0	4
12	☞	UINT	Number of Di	0	65535	0	8
13		UINT	Number of Di	0	65535	0	8

① Switch display for value reading online
② Read values from slave
③ Previously read online values
④ Defined values.

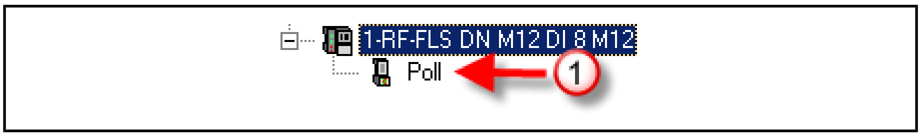
Fig.4-59: Parameter table of a DeviceNet slave in online mode (example)

Tick the checkbox "Online" (see ① fig. 4-59) and select "Read values" (see ② fig. 4-59). The online values are entered in the "Online values" column, see ③ fig. 4-59. These values are highlighted in blue as soon as they differ from the defined values in the "Values" column (see ④ fig. 4-59).

A parameter value change can only be made via the context menu of the "Value" column (see ④ fig. 4-59). Changed values can be tranferred via the menu entry "Transfer parameter values to the slave". A window displays the errors occurred during the transfer to the slave. The errors reported by the slave are defined in the DeviceNet specification of the ODVA definiert (CIP Specification Edition 3.0, Volume 1, Appendix B – Status Codes.)

Configuring I/O Connection of a DeviceNet Slave

Open the next level of the DeviceNet slave for configuring the I/O connection of a DeviceNet slave:



① I/O connection type

Fig.4-60: Used I/O connection of the DeviceNet slave

I/O connection types

- **Poll**: The slave data are cyclically queried by the master (Master-Slave procedure).
- **Bit Strobe**: The DeviceNet master sends a broadcast telegram to all slaves, requesting them to send their current data. The slaves respond in

ascending order of their bus addresses. Data which can be send from each device after a bit strobe command, are limited to 8 bytes.

- **Cos (Change of State)**: The slave automatically sends data after a change of its input.
- **Cyclic**: The slave automatically sends data after the cycle time expired ("heartbeat" function).

If another I/O connection type is required, delete the I/O connection type via the context menu item "Delete":

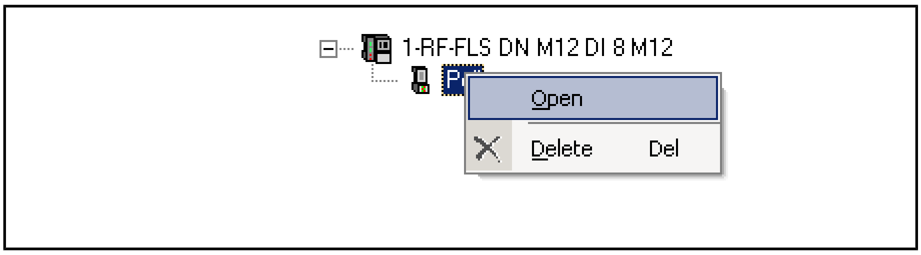
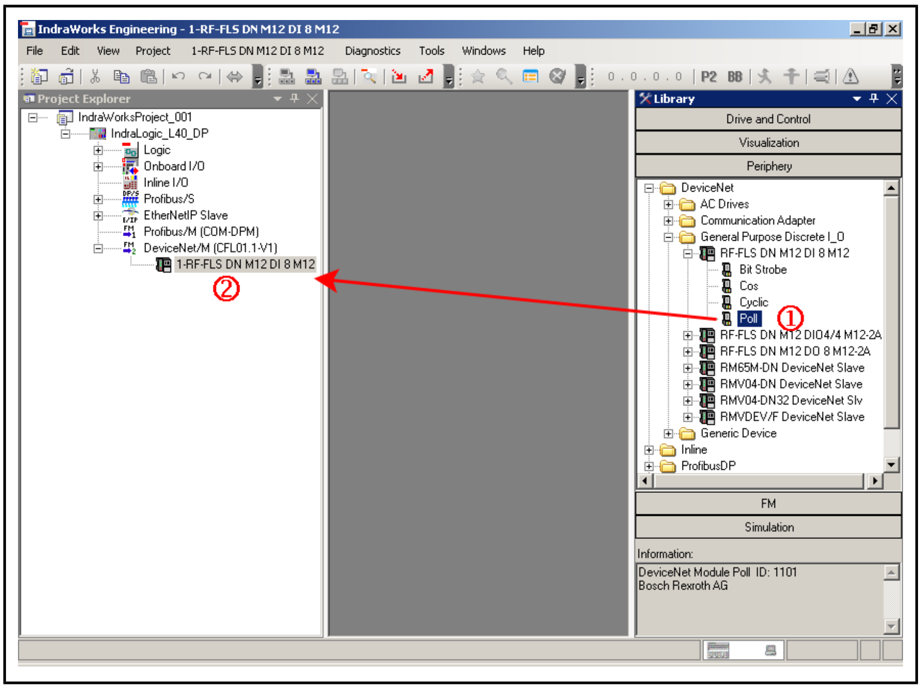


Fig.4-61: Deleting an I/O connection type of a DeviceNet slave

Select the desired I/O connection type from the library via **Periphery ▸ DeviceNet**. Only I/O connection types can be selected which are permissible for the current slave:



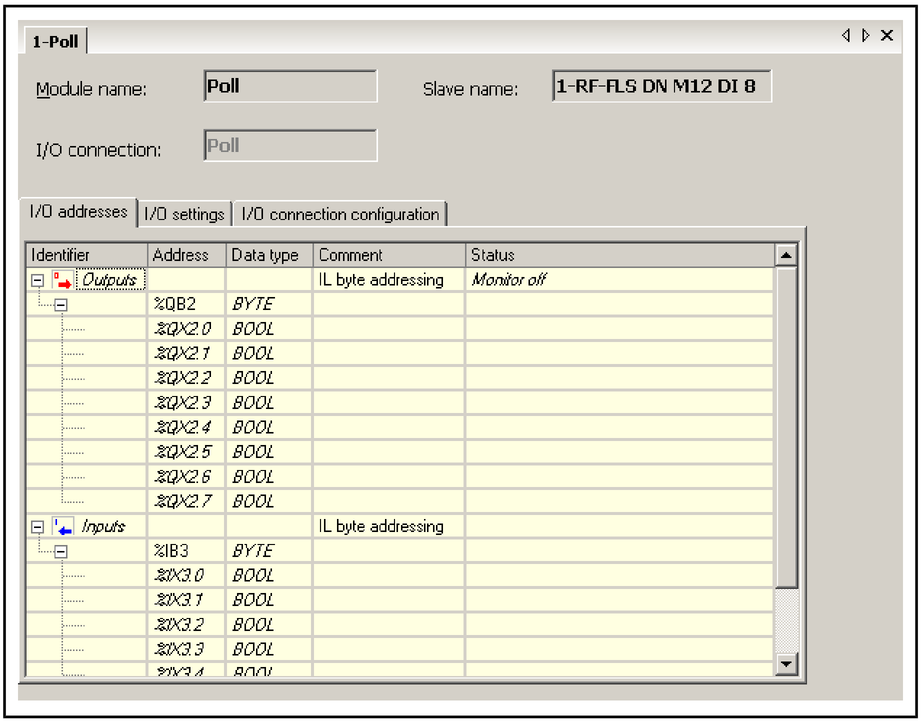
① I/O connection type

② DeviceNet slave

Fig.4-62: Inserting an I/O connection type of a DeviceNet slave

Double-click on the I/O connection type object in order to configure the I/O connection of the DeviceNet slave.

This opens the following window in the workspace:



Module name: Poll

Slave name: 1-RF-FLS DN M12 DI 8

I/O connection: Poll

Identifier	Address	Data type	Comment	Status
Outputs			IL byte addressing	*Monitor off*
	%QB2	*BYTE*		
	%QX2.0	*BOOL*		
	%QX2.1	*BOOL*		
	%QX2.2	*BOOL*		
	%QX2.3	*BOOL*		
	%QX2.4	*BOOL*		
	%QX2.5	*BOOL*		
	%QX2.6	*BOOL*		
	%QX2.7	*BOOL*		
Inputs			IL byte addressing	
	%IB3	*BYTE*		
	%IX3.0	*BOOL*		
	%IX3.1	*BOOL*		
	%IX3.2	*BOOL*		
	%IX3.3	*BOOL*		
	%IX3.4	*BOOL*		

Fig.4-63: Dialog I/O connection type, tab "I/O addresses"

General Displays

Module name: I/O connection type name.

I/O connection: I/O connection type of a DeviceNet slave.

Slave name: Bus address and DeviceNet slave name.

"I/O Addresses" Tab

This tab is for assigning the I/O areas of the modules to the physical addresses of the control (I/O addresses of the PLC).

Identifier: This column shows the input and output structure. Click on the plus or minus symbol to switch between the byte and bit views respectively.

You can also assign a symbolic address to each absolute address (double-click on the particular field). After it has been entered, the symbolic address is automatically created as a global variable in the PLC project.

The symbolic address of a node also appears in the project explorer. For an example of onboard I/O, refer to fig. 4-18 "Symbolic and absolute addresses of I/O objects" on page 30.

Address: I/O address. Enter the desired I/O address as byte address (e.g. %IB10). Entries in italics serve only for display purposes and cannot be edited.

☞ Automatic readdressing is possible in the "I/O Settings" tab.

Data type: "BYTE" stands for byte addresses, and "BOOL" for bit addresses.

Comment: Enter any comment regarding an address in this column.

Status: Physical status of the input/output. The status is only indicated in diagnostic mode during communication between IndraWorks and IndraLogic L40 DP.

"I/O Settings" Tab

This tab is for starting the automatic assignment of I/O addresses.

From input/From output: Current or requested initial addresses of the inputs and outputs. This tab provides only those input boxes that are relevant to the module funcitonality used (module with inputs, module with outputs). If, for example, you parameterize a module having inputs only, there will be no "From output" input box.

"**Apply**": Renumbers automatically all inputs and outputs of the module in ascending order, starting with the start addresses displayed (refer to "From input / From output"). Please note that any existing address gaps are closed during this process!

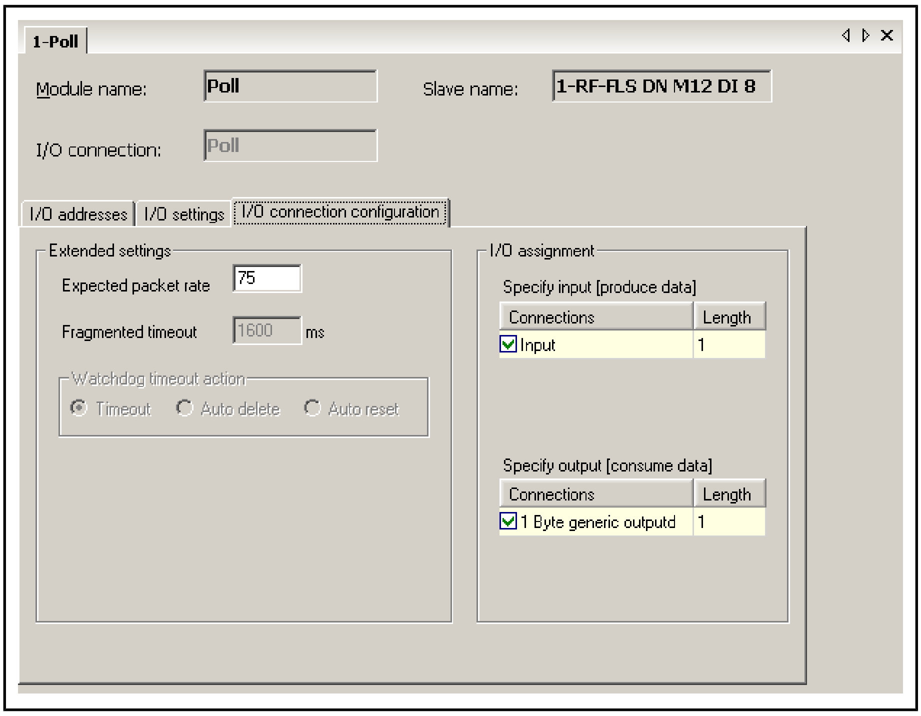
If the automatic numbering causes collisions with already assigned address areas, IndraWorks indicates the collision cause and automatically determines the next free address area.

"I/O Connection Configuration" Tab

DEfine the I/O field size. Depending on the I/O connection type, further data can be edited, see fig. 4-64 "Tab "I/O connection configuration" for I/O connection type "Poll"" on page 68 to fig. 4-67 "Tab "I/O connection configuration" for I/O connection type "Bit strobe"" on page 70.

The I/O assignment length has to entered in compliance with the actual module assembly. In the "I/O assignment" click on the field "Length" in order to change the setting. If several connections per data direction are displayed, max one connection can be selected.

☞ The tooltip displays the possible values in the "Length" column. For that, the mouse cursor has to be pointed on the "Length" field.



Module name: Poll

Slave name: 1-RF-FLS DN M12 DI 8

I/O connection: Poll

Expected packet rate 75

Fragmented timeout 1600 ms

● Timeout ○ Auto delete ○ Auto reset

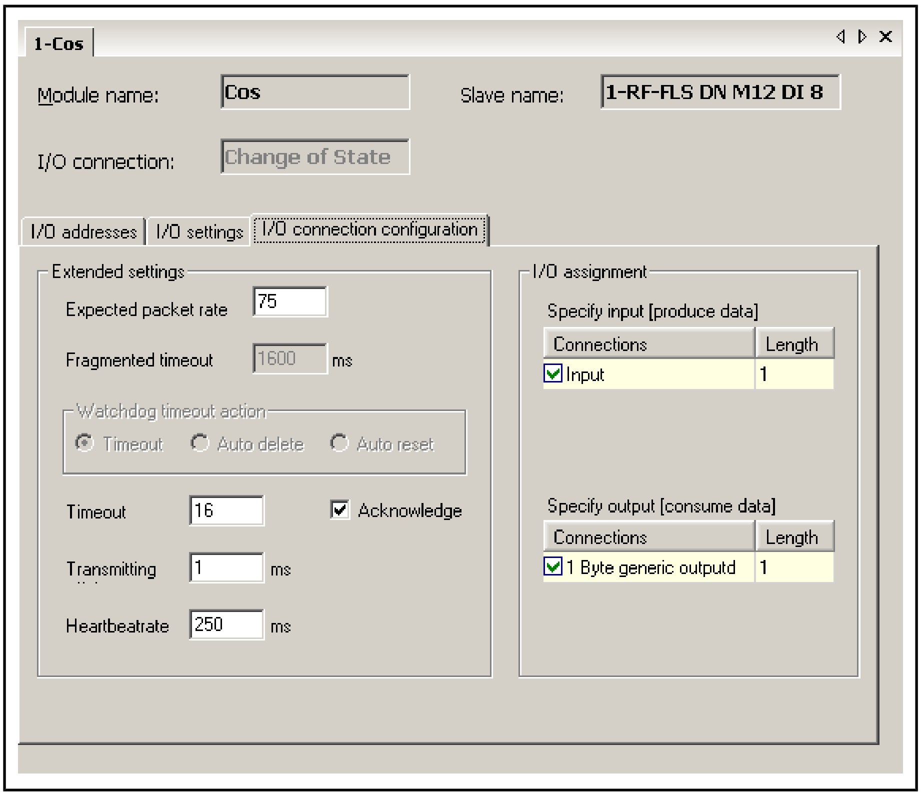
Specify input [produce data]

Connections	Length
☑ Input	1

Specify output [consume data]

Connections	Length
☑ 1 Byte generic outputd	1

Fig.4-64: Tab "I/O connection configuration" for I/O connection type "Poll"



1-Cos

Module name: Cos

Slave name: 1-RF-FLS DN M12 DI 8

I/O connection: Change of State

I/O addresses | I/O settings | I/O connection configuration

Extended settings

Expected packet rate: 75

Fragmented timeout: 1600 ms

Watchdog timeout action: ● Timeout ○ Auto delete ○ Auto reset

Timeout: 16

☑ Acknowledge

Transmitting: 1 ms

Heartbeatrate: 250 ms

I/O assignment

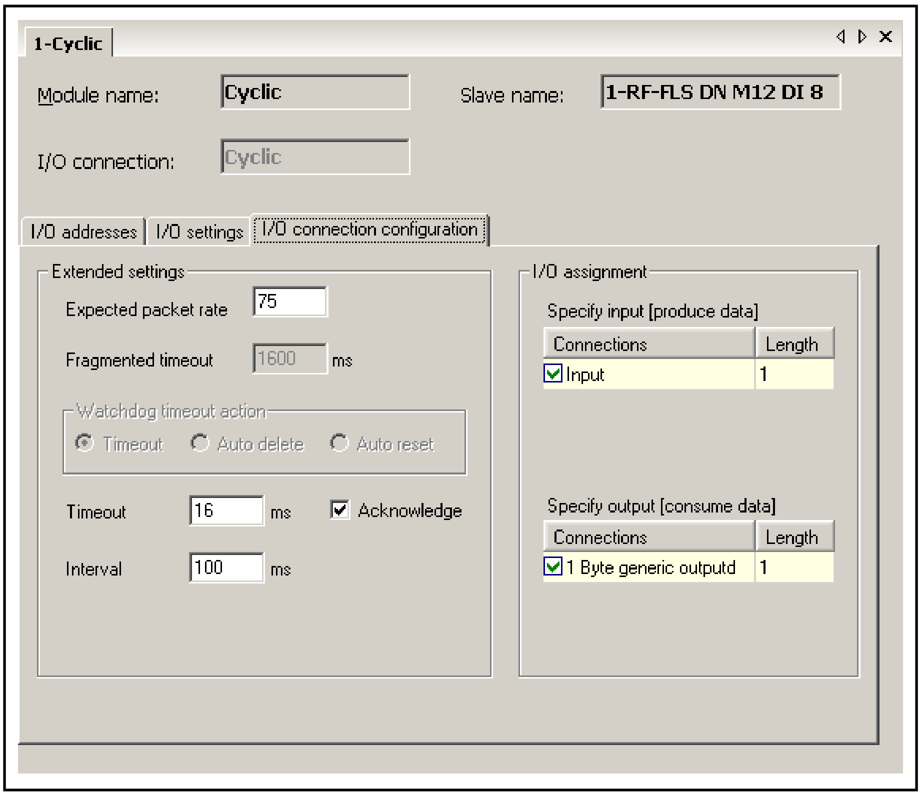
Specify input [produce data]

Connections	Length
☑ Input	1

Specify output [consume data]

Connections	Length
☑ 1 Byte generic outputd	1

Fig.4-65: Tab "I/O connection configuration" for I/O connection type "Change of state"



1-Cyclic

Module name: Cyclic

Slave name: 1-RF-FLS DN M12 DI 8

I/O connection: Cyclic

I/O addresses | I/O settings | I/O connection configuration

Extended settings

Expected packet rate: 75

Fragmented timeout: 1600 ms

Watchdog timeout action: ● Timeout ○ Auto delete ○ Auto reset

Timeout: 16 ms

☑ Acknowledge

Interval: 100 ms

I/O assignment

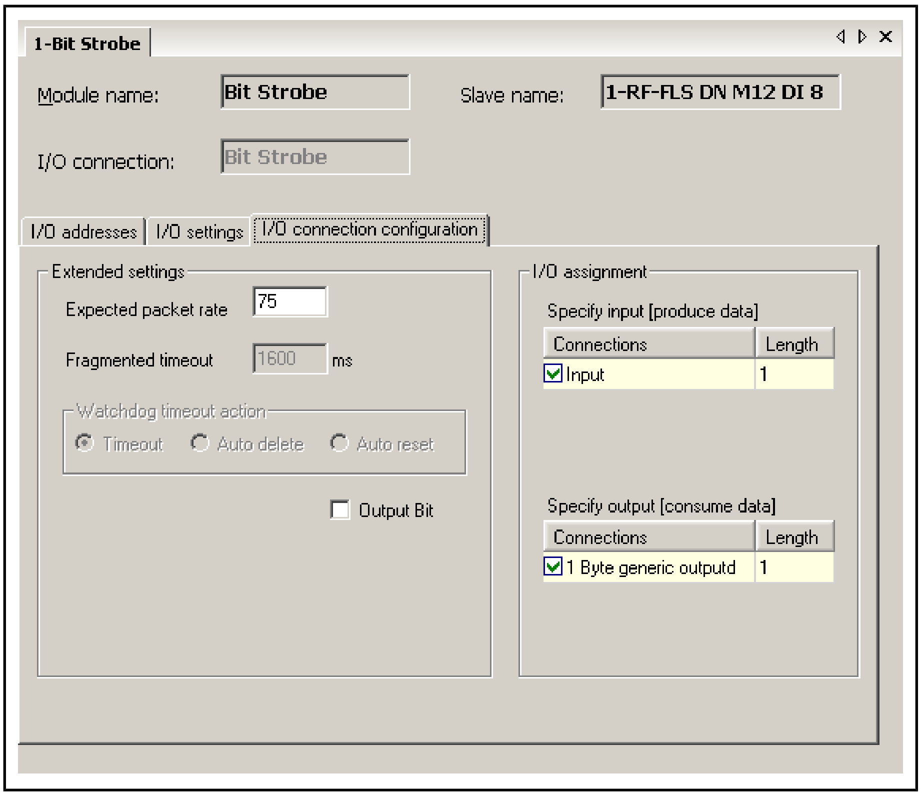
Specify input [produce data]

Connections	Length
☑ Input	1

Specify output [consume data]

Connections	Length
☑ 1 Byte generic outputd	1

Fig.4-66: Tab "I/O connection configuration" for I/O connection type "Cyclic"



1-Bit Strobe

Module name: Bit Strobe

Slave name: 1-RF-FLS DN M12 DI 8

I/O connection: Bit Strobe

Extended settings

Expected packet rate: 75

Fragmented timeout: 1600 ms

Watchdog timeout action: ● Timeout ○ Auto delete ○ Auto reset

□ Output Bit

I/O assignment

Specify input [produce data]

Connections	Length
☑ Input	1

Specify output [consume data]

Connections	Length
☑ 1 Byte generic outputd	1

Fig.4-67: Tab "I/O connection configuration" for I/O connection type "Bit strobe"

4.10 PLC Programming with IndraLogic

4.10.1 Overview

IndraLogic, a program integrated in IndraWorks, provides the following functions for PLC programming:

- Target settings: special settings for the IndraLogic L40 DP, see chapter 4.10.2 "Target Settings " on page 71.
- Task configuration: controls program processing, see chapter 4.10.3 "Task Configuration " on page 73.
- Library manager: manages the PLC module libraries, see chapter 4.10.4 "Library Manager " on page 74.
- Creation of a PLC program: creates function blocks in PLC programming languages, see chapter 4.10.5 "Creating a PLC Program " on page 76.

☞ The control is configured in IndraWorks, not in IndraLogic, see chapter 4.3 "Specifying Basic Settings (Wizard-Guided) " on page 16. That is why many input boxes of the IndraLogic PLC configuration are deactivated and cannot be edited.

PLC project planning with IndraLogic is activated via the "Logic" object in the project explorer.

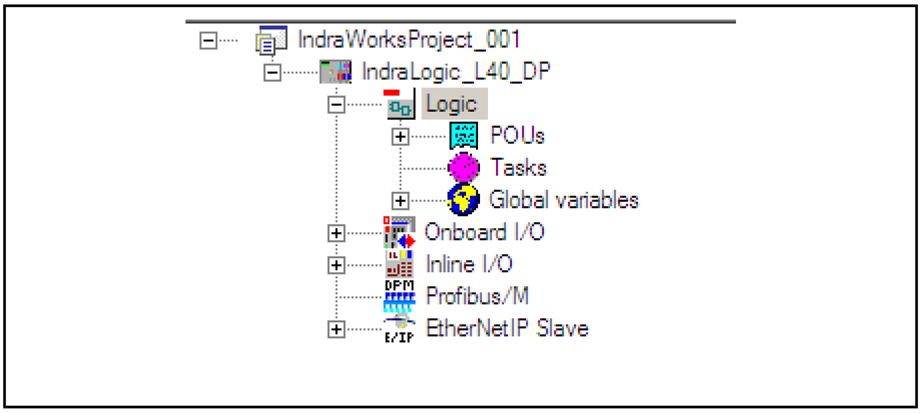


Fig.4-68: "Logic" object in the project explorer

- Double-click on the "Logic" object to start IndraLogic with the "IndraLogic L40 DP" as target system.
- Double-click on an entry under "POUs" (e.g. PLC_PRG) to open the corresponding module for further editing in IndraLogic, see chapter 4.10.5 "Creating a PLC Program " on page 76.
- Double-click on "Tasks" to start task configuration in IndraLogic, see chapter 4.10.3 "Task Configuration " on page 73.

Updating a Project

To apply all of the modules generated in IndraLogic as well as any edited settings to the IndraWorks project explorer, select the **Update** function from the context menu of the "Logic" object:

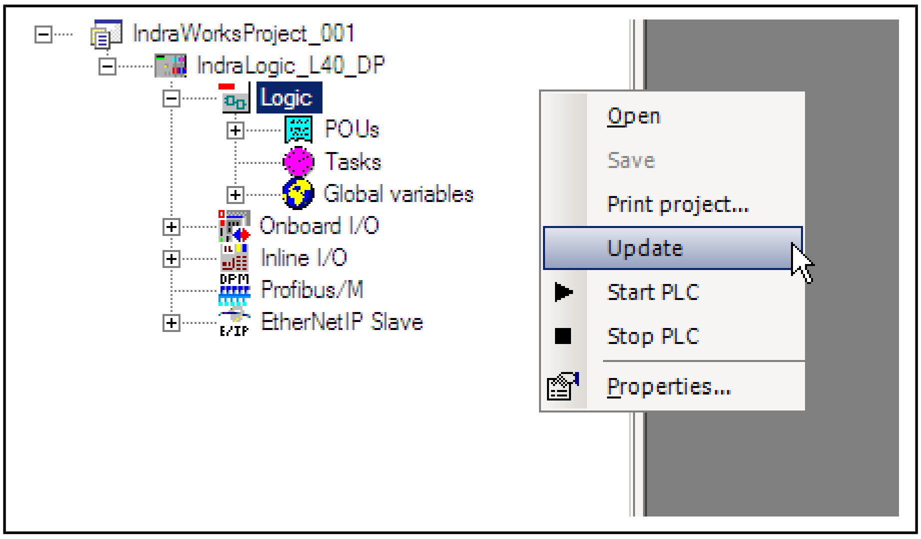


Fig.4-69: Updating a "Logic" object

4.10.2 Target Settings

CAUTION

Any modification of the preset target configuration can have serious effects on the behavior of the target system!

The target settings may only be modified by skilled specialists, who are aware of the effects of such modifications!

The target settings are optimized to comply with the most frequently used applications. Some of the settings can be adjusted to special cases of application. The target settings are modified via the properties dialog of the "Logic" object. IndraLogic must be closed.

Settings that are not available on the IndraWorks level (e.g. "Network functions") can be made in IndraLogic under "Target Settings". To open these settings in IndraLogic, click on "Target Settings" in the "Resources" tab, see fig. 4-70 "IndraLogic, resources, "Target Settings" object" on page 72.

☞ For more information about target settings, please refer to the IndraLogic documentation /2/ and/or online help.

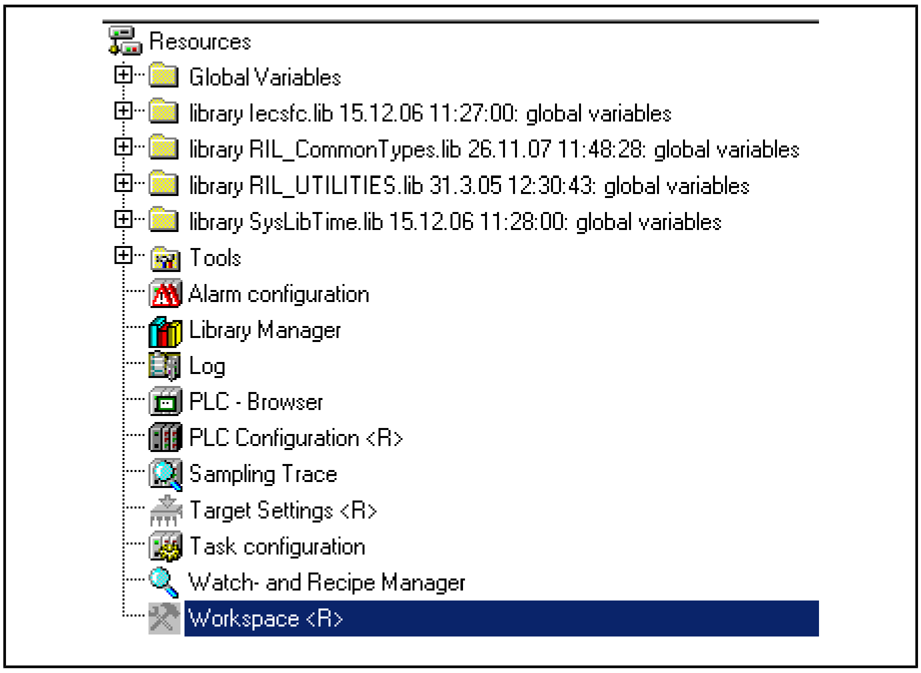

Resources
Global Variables
library Iecsfc.lib 15.12.06 11:27:00: global variables
library RIL_CommonTypes.lib 26.11.07 11:48:28: global variables
library RIL_UTILITIES.lib 31.3.05 12:30:43: global variables
library SysLibTime.lib 15.12.06 11:28:00: global variables
Tools
Alarm configuration
Library Manager
Log
PLC - Browser
PLC Configuration <R>
Sampling Trace
Target Settings <R>
Task configuration
Watch- and Recipe Manager
Workspace <R>

Fig.4-70: IndraLogic, resources, "Target Settings" object

Interrupting the Database Connection

Usually, there is always an active connection to the database server. Such an active connection is indicated by "<R>" appended to the name of the "Target Settings" object.

Target settings can be edited only if the connection to the database is interrupted. To achieve this, select the **Project database** ▸ **Check Out** item of the target settings context menu, see fig. 4-71 "Target settings: Checking out from the database" on page 73. After the target settings have been edited, the database connection can be reestablished via **Project database** ▸ **Check In**. After settings have been edited, it might be necessary to synchronize the IndraWorks and IndraLogic databases. This can be achieved with the **Project database** ▸ **Get Latest Version** menu item.

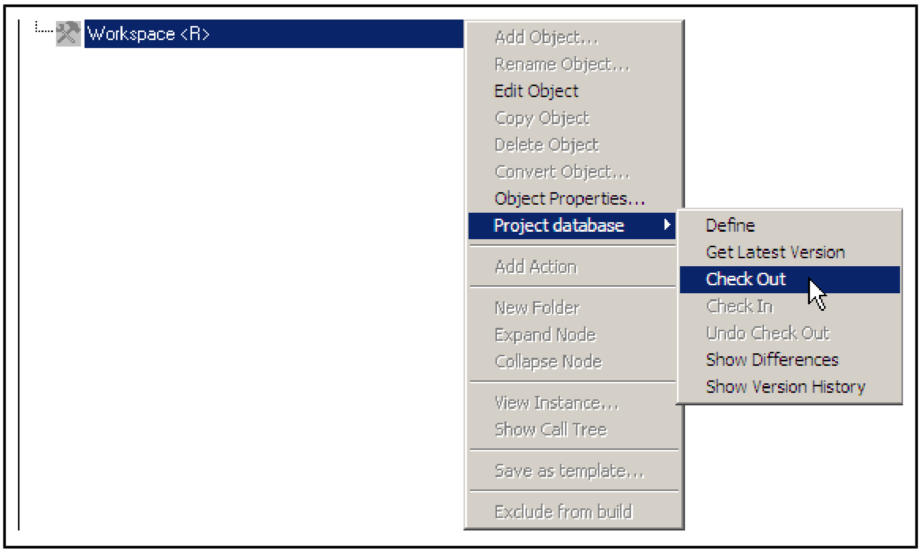


Fig.4-71: Target settings: Checking out from the database

4.10.3 Task Configuration

Many applications just require program execution without any special task management. In this case, the program is processed simply by executing the PLC_PRG "main program". PLC_PRG is automatically generated as a "Program" type module. It is called exactly one time in each control cycle.

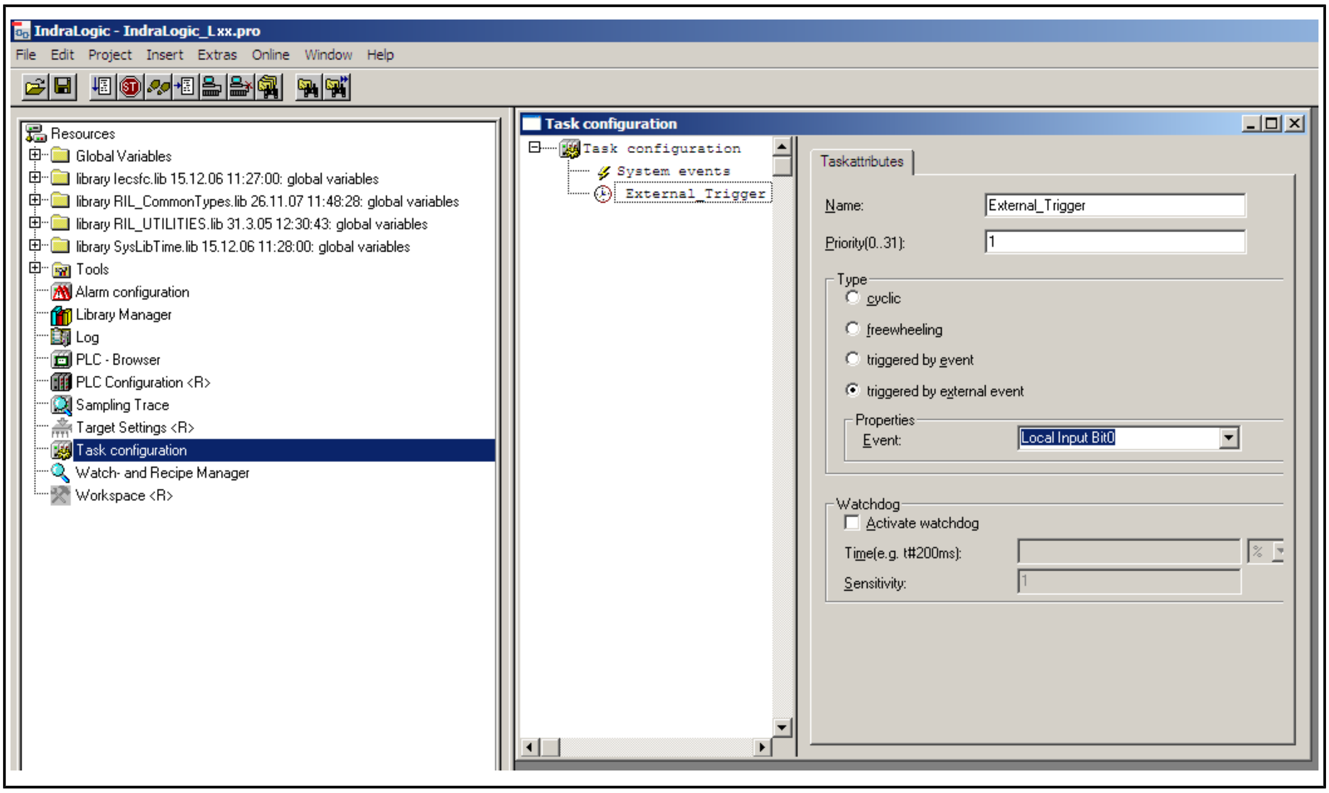
In special application cases, the processing of the tasks can be controlled. This requires the task configuration function.

Call the task configuration as follows:

- In the IndraWorks project explorer: Double-click on the "Tasks" object of the relevant "Logic" objects.

- OR -

- in the "Resources" tab of IndraLogic: Double-click on the "Task configuration" object.


IndraLogic - IndraLogic_Lxx.pro
File Edit Project Insert Extras Online Window Help
Resources
Global Variables
library Iecsfc.lib 15.12.06 11:27:00: global variables
library RIL_CommonTypes.lib 26.11.07 11:48:28: global variables
library RIL_UTILITIES.lib 31.3.05 12:30:43: global variables
library SysLibTime.lib 15.12.06 11:28:00: global variables
Tools
Alarm configuration
Library Manager
Log
PLC - Browser
PLC Configuration <R>
Sampling Trace
Target Settings <R>
Task configuration
Watch- and Recipe Manager
Workspace <R>
Task configuration
Task configuration
System events
External_Trigger
Taskattributes

Name: External_Trigger

Priority(0..31): 1

Type

○ cyclic

○ freewheeling

○ triggered by event

● triggered by external event

Properties

Event: Local Input Bit0

Watchdog

□ Activate watchdog

Time(e.g. t#200ms):

Sensitivity: 1

Fig.4-72: IndraLogic task configuration

For more information about task attributes, please refer to the IndraLogic documentation /2/ and/or online help.

The IndraLogic L40 DP provides a special task-controlled function:

Triggered by External Event

A task can be triggered by a rising edge or rising edges at one or several onboard inputs.

To achieve this, select the "triggered by external event" option in the "Task attributes" tab and the event required from the "Event" box:

- **Local Input Bit 0** ... **Local Input bit 7**: The task is started as soon as a rising edge is applied to the selected bit input of the onboard I/O.
- **Local input byte**: The task is started as soon as a rising edge is detected at at least one of the eight onboard inputs. Even if positive signal levels are already applied to the individual inputs, the task starts with each new rising edge of an input. Rising edges that are incoming simultaneously are detected as a single event and, thus, do not repeatedly start the task.

The independent task starts when the program is started. After having been executed, the task restarts without any waiting time. In order that other tasks can be started, the independent task **must** be the task that has the lowest priority among all of the programmed tasks. Only **one** independend task can be used.

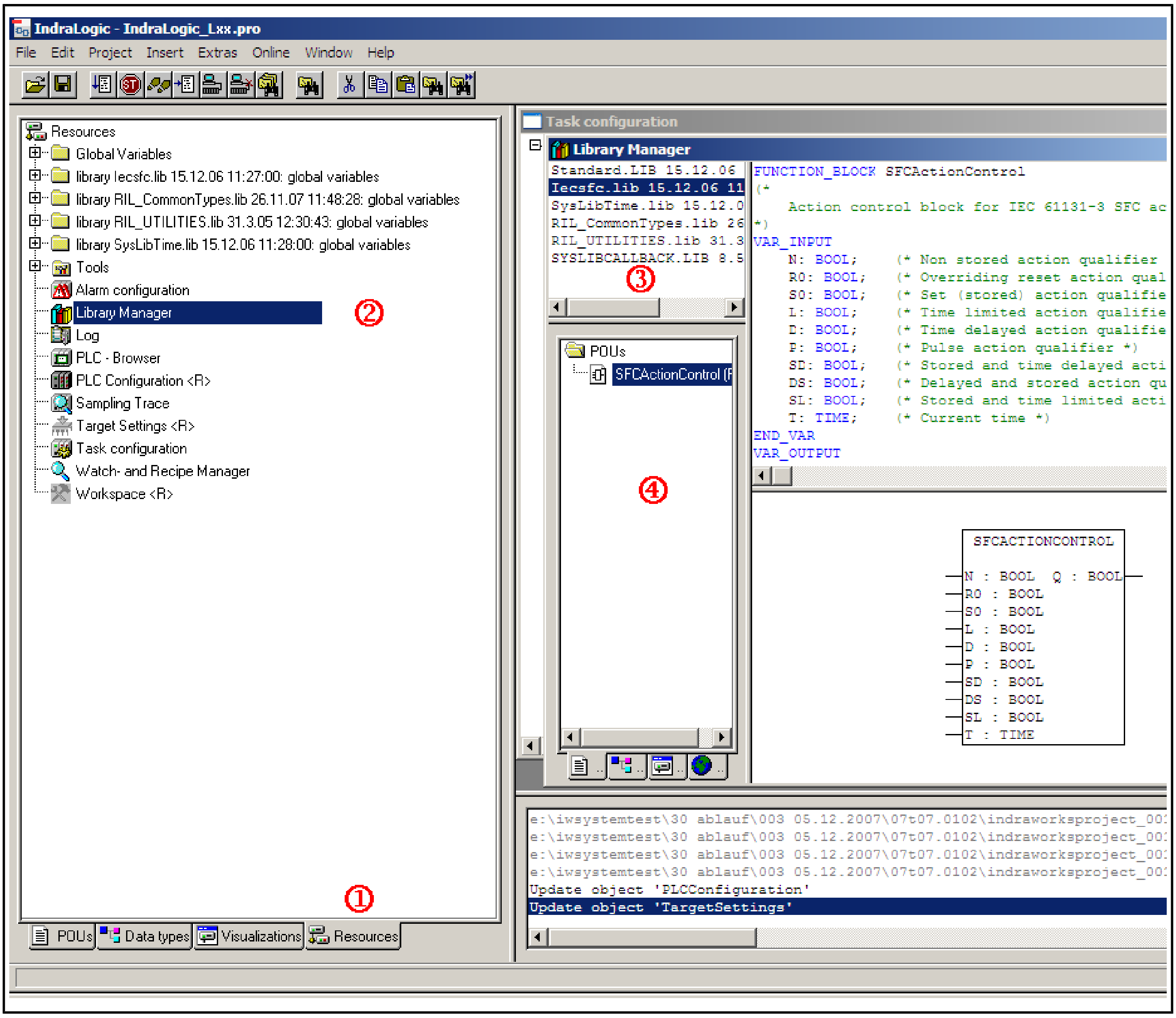
4.10.4 Library Manager

For more information on the library manager, please refer to the IndraLogic documentation /2/ and/or online help. For information about the IndraLogic L40 DP libraries, please refer to chapter 6 "Libraries".

The IndraLogic L40 DP already contains a few libraries that can be displayed in the "Resources" tab of IndraLogic.

☞ IndraWorks loads libraries for internal use (see chapter 6 "Libraries") automatically as soon as they are referenced. They do not have to be inserted manually using the library manager.

To link further libraries to the current project, double-click on the "Library Manger" object in the "Resoureces" tab. This opens the library manager:



```
Standard.LIB 15.12.06
Iecsfc.lib 15.12.06 11
SysLibTime.lib 15.12.0
RIL_CommonTypes.lib 26
RIL_UTILITIES.lib 31.3
SYSLIBCALLBACK.LIB 8.5
```

```
FUNCTION_BLOCK SFCActionControl
(*
   Action control block for IEC 61131-3 SFC ac
*)
VAR_INPUT
   N: BOOL;     (* Non stored action qualifier
   R0: BOOL;    (* Overriding reset action qual
   S0: BOOL;    (* Set (stored) action qualifie
   L: BOOL;     (* Time limited action qualifie
   D: BOOL;     (* Time delayed action qualifie
   P: BOOL;     (* Pulse action qualifier *)
   SD: BOOL;    (* Stored and time delayed acti
   DS: BOOL;    (* Delayed and stored action qu
   SL: BOOL;    (* Stored and time limited acti
   T: TIME;     (* Current time *)
END_VAR
VAR_OUTPUT
```

```
e:\iwsystemtest\30 ablauf\003 05.12.2007\07t07.0102\indraworksproject_00
e:\iwsystemtest\30 ablauf\003 05.12.2007\07t07.0102\indraworksproject_00
e:\iwsystemtest\30 ablauf\003 05.12.2007\07t07.0102\indraworksproject_00
e:\iwsystemtest\30 ablauf\003 05.12.2007\07t07.0102\indraworksproject_00
Update object 'PLCConfiguration'
Update object 'TargetSettings'
```

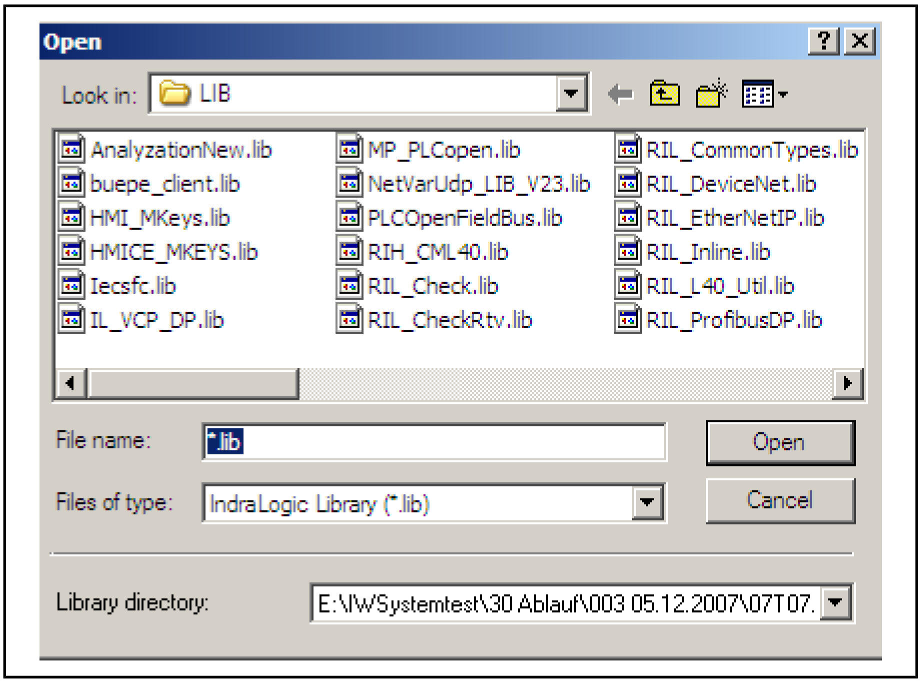
① "Resources" tab
② "Library Manager" object
③ List of loaded libraries
④ Display of all function blocks contained in a selected library

Fig.4-73: Library Manager

Select the **Insert ▸ Additional library** menu item or "Additional library..." from the context menu of area (3). This opens a dialog for library selection, see fig. 4-74 "Library folder of the IndraLogic L40 target" on page 76. Select the library folder of the installed IndraLogic L40 DP files (target files).

The data are residing in the following directory: "...\Rexroth\IndraWorks\IndraLogic\Targets\<TargetSystemName>\lib".

Example: Library folder of the IndraLogic L40 DP DPM 03 VRS: "C:\Programme \Rexroth\IndraWorks\IndraLogic\Targets\IndraLogic_L40_DP_03VRS\lib\"



Look in: LIB

AnalyzationNew.lib
buepe_client.lib
HMI_MKeys.lib
HMICE_MKEYS.lib
Iecsfc.lib
IL_VCP_DP.lib
MP_PLCopen.lib
NetVarUdp_LIB_V23.lib
PLCOpenFieldBus.lib
RIH_CML40.lib
RIL_Check.lib
RIL_CheckRtv.lib
RIL_CommonTypes.lib
RIL_DeviceNet.lib
RIL_EtherNetIP.lib
RIL_Inline.lib
RIL_L40_Util.lib
RIL_ProfibusDP.lib

File name: *.lib

Files of type: IndraLogic Library (*.lib)

Library directory: E:\IWSystemtest\30 Ablauf\003 05.12.2007\07T07.

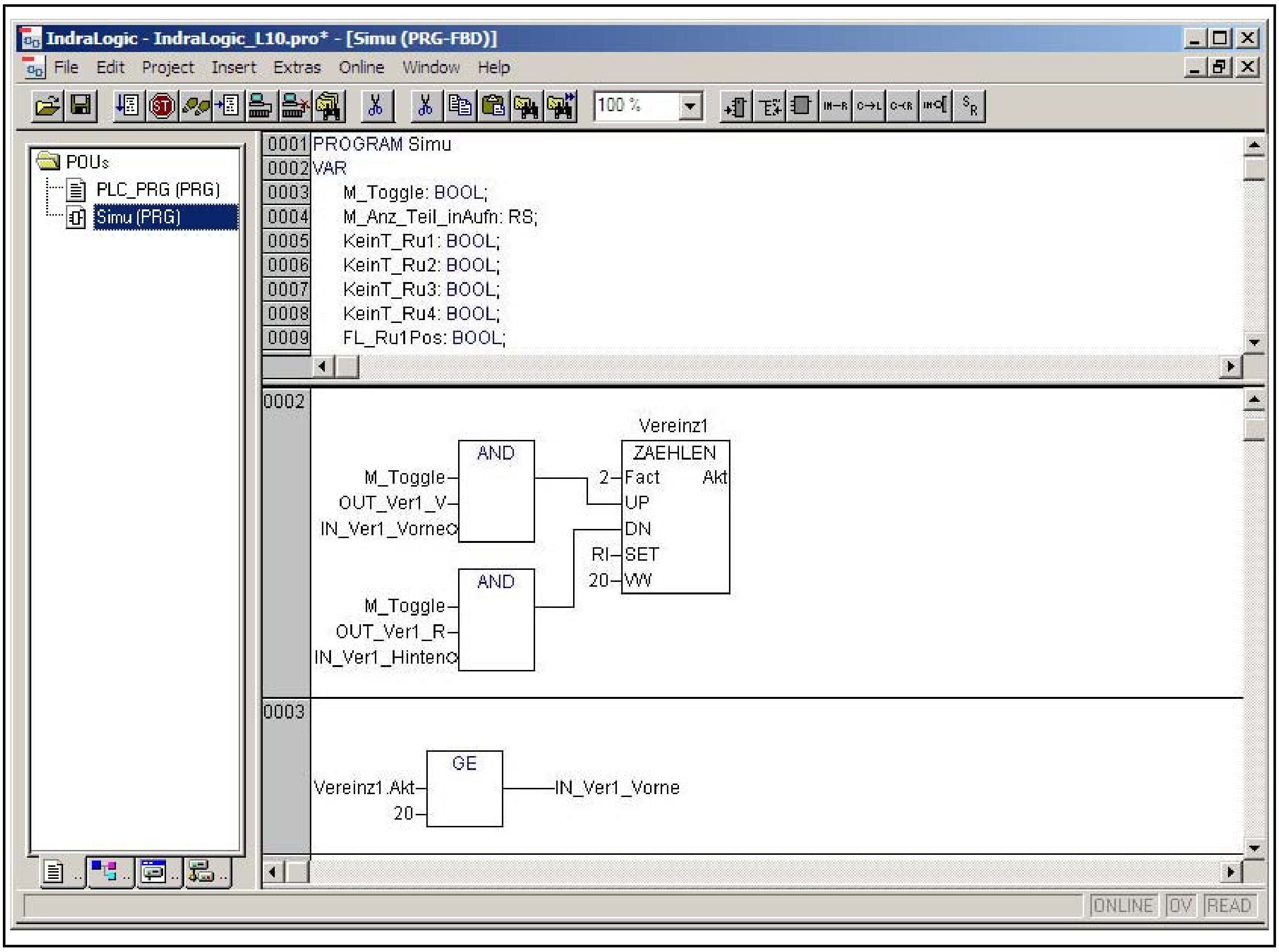
Fig.4-74: Library folder of the IndraLogic L40 target

Select the desired library and confirm your selection by clicking on "Open". The library is applied to area (3) (see fig. 4-73 "Library Manager" on page 75) and is ready for use.

4.10.5 Creating a PLC Program

☞ Create the PLC program in IndraLogic. For further information refer to the documentation or online help of IndraLogic /2/.

To create compatible IEC programs and to organize the memory of I/O addresses, please refer to chapter 4.11 " Compatible IEC Programming between different controls" on page 77.



```
PROGRAM Simu
VAR
    M_Toggle: BOOL;
    M_Anz_Teil_inAufn: RS;
    KeinT_Ru1: BOOL;
    KeinT_Ru2: BOOL;
    KeinT_Ru3: BOOL;
    KeinT_Ru4: BOOL;
    FL_Ru1Pos: BOOL;
```

Fig.4-75: PLC program example

4.10.6 Saving IndraLogic Project Data

All settings made in IndraLogic and all edited function blocks or modules can be saved by selecting the **File** ▸ **Save** menu function.

☞ Always save the IndraLogic project data first. Then edit the current project in IndraWorks.

4.11 Compatible IEC Programming between different controls

4.11.1 Overview

To ensure a compatible programming between the systems, consider the following features: It is described, which programming methods can be used, so that no incompatibilities may occur:

- Use of pointers within structures
- Packing Structures for IndraLogic L10 and L20
- Memory alignment for I/O addresses
- Assignment of structures to I/O addresses

4.11.2 Use of Pointers within Structures

If structures are created in the IEC program, the compiler maps this structures during the compilation process in the data memory of the control. The compiler recognizes exactly the possible restrictions of the used processor platform and archives the elements of the structure in the memory by using only addresses, that the processor can utilize for the corresponding data types.

If structures with elements containing different data types are created, the compiler inserts, e.g., for the IndraLogic L10/L20, filling bytes, that are not visible for the user.

Example

Structural differences between the controls

Program:

```
TYPE OutStruct :
STRUCT
    Out01 : BYTE;
    Out02 : WORD;
    Out03 : BYTE;
    Out04 : DWORD;
END_STRUCT
END_TYPE
```

Depending on the platform this structure is mapped to the memory as follows (here a comparison between the controls IndraLogic L40 DP and L20):

IndraLogic V, IndraLogic VEP, IndraLogic VPP, IndraLogic L40	IndraLogic L10/L20

ADR0 Out01 : BYTE;
ADR1 Out02 : WORD;
ADR3 Out03 : BYTE;
ADR4 Out04 : DWORD;
ADR8 :
	ADR0 Out01 : BYTE;
	ADR1 (filling byte)
	ADR2 Out02 : WORD;
	ADR4 Out03 : BYTE;
	ADR5 (filling byte)
	ADR6 (filling byte)
	ADR7 (filling byte)
	ADR8 Out04 : DWORD;
	ADR12 :

Fig.4-76: Resulting control-independent memory image

The structure elements are differently mapped, so that a compatible programming method being independent of the control's mapping must be used when accessing the structure elements.

Permissible Addressing

Thus, the compatible use of structures provides for a direct addressing of the elements via the point operator:

Structure name.Element name := Element value;

If the address of a structure is to be transferred to subfunctions via a pointer, the addressing of the elements is also only permissible by the offset calculation of the compiler:

Structure pointer^.Element name := Element value;

Program:

```
StructInst : OutStruct;             (* Declaration structure instance *)
pt : POINTER TO StructInst;         (* Declaration pointer to structure instance *)
```

```
StructInst.Out03 := 2#11110000;       (* Access to structure element *)

pt := ADR (StructInst);               (* Pointer initialization *)
pt^.Out02 := 16#FF00;                 (* Access to structure element by pointer *)
```

Impermissible Addressing

The addressing of a structure element by calculations in the code is **not allowed**:

Structure element pointer := ADR (Structure name.Element);

Structure element pointer := Structure element pointer + n;

Structure element pointer^ := value;

This is not allowed and causes incompatibilities, as it is not ensured, that the offset creation to address a structure element by calculation in the code is reliable. The number of filling bytes is different for the platforms.

4.11.3 Packing Structures for IndraLogic L10/20

Concerning the IndraLogic L40 structures are always packed, i. e. they are archived in the memory without filling bytes. Contrary to that, with an IndraLogic L10/20 the packing of structures must be forced by a corresponding compiler instruction (pragma "pack") during the structure declaration.

The pragma {pack} is supported as of IndraWorks Logic version 02V09.

The pragma {pack} is ignored by the compiler of the IndraLogic L40, i.e. it has no effect on the application's compilability or ability to run in the IndraLogic L40.

Example for the compiler instruction pack

Program:

```
TYPE structure name
STRUCT
{pack}
 Element_01 : USINT;
 Element_02 : DWORD;
 Element_03 : BYTE;
 Element_04 : INT;
END_STRUCT
END_TYPE
```

The structure of this programm listingis archived in the memory of the different systems as follows:

Byte address	IndraLogic V, IndraLogic VEP, IndraLogic VPP, IndraLogic L40	IndraLogic L10/L20 without pragma {pack}	IndraLogic L10/L20 with pragma {pack}
0	Element_01	Element_01	Element_01
1	Element_02 (byte 0)	Filling byte	Element_02 (byte 0)
2	Element_02 (byte 1)	Filling byte	Element_02 (byte 1)
3	Element_02 (byte 2)	Filling byte	Element_02 (byte 2)
4	Element_02 (byte 3)	Element_02 (byte 0)	Element_02 (byte 3)
5	Element_03	Element_02 (byte 1)	Element_03
6	Element_04 (byte 0)	Element_02 (byte 2)	Element_04 (byte 0)
7	Element_04 (byte 1)	Element_02 (byte 3)	Element_04 (byte 1)
8		Element_03	
9		Filling byte	

Byte address	IndraLogic V, IndraLogic VEP, IndraLogic VPP, IndraLogic L40	IndraLogic L10/L20 without pragma {pack}	IndraLogic L10/L20 with pragma {pack}
10		Element_04 (byte 0)	
11		Element_04 (byte 1)	

Fig.4-77: Memory assignment

By specifying the pragma {pack} a structure archive compatible to the IndraLogic L40 is created. The individual structure elements can be accessed like in the IndraLogic L40.

Example to pack structures

Program:

```
VAR str001:
 struct001;
 len_struct001:
 INT := 0;
 array001: ARRAY [0..15] OF BYTE;
 ps001: POINTER TO BYTE;
 pa001: POINTER TO BYTE;
i: INT := 0;
END_VAR
----------------------------------------------------------
len_struct001:=SIZEOF(str001);
str001.s001_byte1:=16#01;
str001.s001_DW:=16#05040302;
str001.s001_byte2:=16#06;
str001.s001_Word:=16#0807;
ps001:=ADR (str001);
pa001:=ADR (array001);
 FOR i :=0 TO len_struct001 BY 1 DO
pa001^:=ps001^;
pa001:=pa001+1;
ps001:=ps001+1;
END_FOR;
```

☞ To pack structures with the help of the compiler instruction {pack} causes a deceleration of the PLC program processing in the IndraLogic L10/20, as the access to word or double-word operands occurs bytewise.

4.11.4 Alignment for I/O Addresses

The processor of the IndraLogic L10/L20 supports a so-called "Natural Alignment". That is, the memory accesses are optimized in a manner, that the variables are archived on addresses corresponding to the data capacity of the data type.

The following table shows the factor for the optimum start address for the particular data type:

Data type	Factor = Data capacity (bytes)
BYTE	1
WORD	2
DWORD	4

Fig.4-78: Factor for start addresses

During the declaration of variables **without** assignment of an I/O address the IndraLogic compiler ensures an optimum alignment corresponding to the illustrated table.

Example Variable1: WORD → even address

Variable2: DWORD → address divisible by 4

During the declaration of variables **with** assignment of an I/O address the alignment in the memory depends on the assigned address.

Out01 AT %QB0 : DWORD; → even address

Out02 AT %QB5 : DWORD; → uneven address

☞ To assign word/byte variables to uneven I/O addresses causes a deceleration of the PLC program processing in the IndraLogic L10/L20, as the access to word and double-word operands occurs bytewise.

4.11.5 Assigning Structures to I/O Addresses

A structure declared with the pragma instruction {pack} (refer to chapter 4.11.3 "Packing Structures for IndraLogic L10/20" on page 79) can be directly mapped on a corresponding I/O area.

A non-packed structure causes a wrong addressing within the control because of the filling bytes.

Use of the data type LREAL Due to the processor architecture used for the IndraLogic L10/L20, the 64 bit data type LREAL is not mapped task and interrupt save. The processor type used contains only tabs with a 32 bits width. Operations executed with LREAL variables can be stopped during task change or interrupt processing so that consisitency errors occurr.

☞ Use the LREAL variables in **one** IEC task only to exclude consistency errors.

4.12 Download and Commissioning

4.12.1 Loading Configurations and the PLC Program to the Control

Touble-Shooting After having been programmed, the project can be trouble-shooted in IndraLogic by selecting the **Project** ▸ **Compile all** menu item.

Download and Online Mode Select the **Online** ▸ **Login** menu item to start communication between the programming system and the IndraLogic L40 DP and to enter the online mode.

If the current project has not been compiled since it was opened or edited, it will now be compiled (same as with **Project** ▸ **Compile**). IndraLogic will only enter the online mode if there are no compilation errors.

If the current project has been modified but not closed since it was downloaded to the control and the previous download information has not been deleted with the**Project** ▸ **Clean all** command, a dialog with the following request is opened:

"The program has been changed. Load changes? (Online Change)"

Click "Yes" to confirm that that the projects parts that have been changed are to be loaded to the control on login. Click "Load all" to reload the complete project to the control. Click "No" to log in without loading the changes made since the last download to the control.

☞ For this, refer also to the "online functions" in the IndraLogic documentation or help /2/.

4.12.2 Online Functions

The "online functions" of IndraWorks can be used to obtain information about the state of the control.

Connection Test To switch a control to online mode, the computer where IndraWorks is installed must be connected to the control. This connection can be checked via the context menu of the device, see fig. 4-79 "Checking the connection of a device" on page 82.

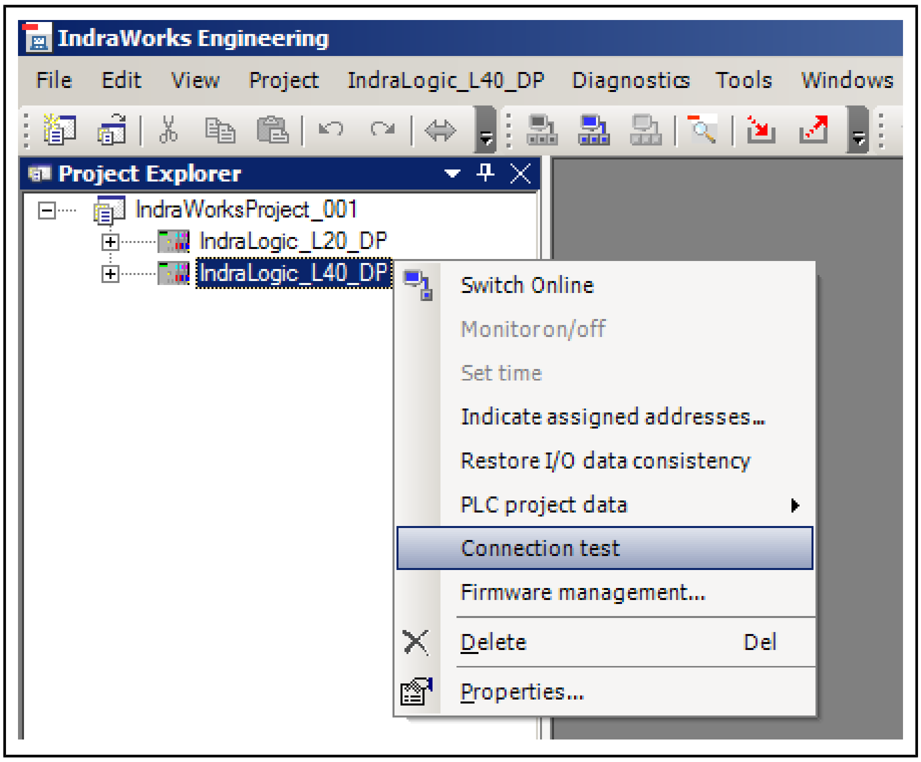


Fig.4-79: Checking the connection of a device

The result of the connection test is provided in the acknowledgement. If there is a connection to the device, the control can be switched to online mode.

Switching to Online Mode "Online switching" can be done via the context menu of the control nodeor the project node.

Only the selected control e.g. "IL_L40_b" is switched online via the context menu of the control node.

The menu with the configured controls is displayed as described via the project node context menu ("Switch devices online").

To switch the devices to online mode, select the **Project ▸ Switch Devices Online** menu item, see fig. 4-80 "Switching devices to online mode" on page 83.

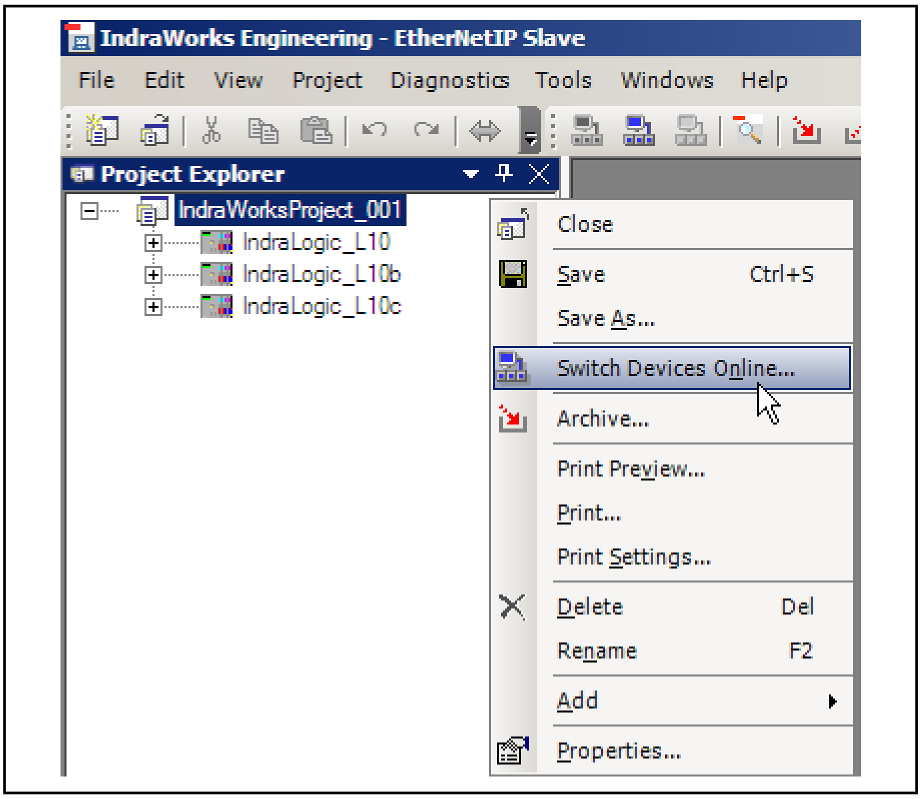
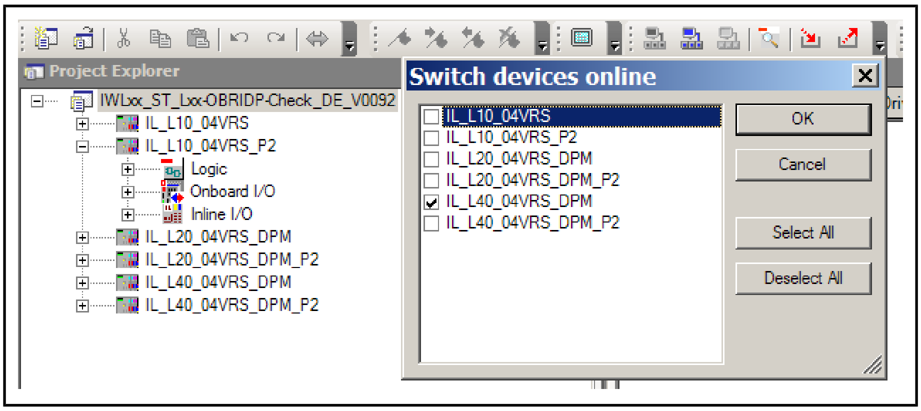


Fig.4-80: Switching devices to online mode

This opens a dialog where you can select the desired device, see fig. 4-81 "Selecting devices to be switched to online mode" on page 83.



□ IL_L10_04VRS
□ IL_L10_04VRS_P2
□ IL_L20_04VRS_DPM
□ IL_L20_04VRS_DPM_P2
☑ IL_L40_04VRS_DPM
□ IL_L40_04VRS_DPM_P2

Fig.4-81: Selecting devices to be switched to online mode

Individual devices can also be switched to online mode directly using the context menu of the device in the project explorer. The project explorer indicates devices in online mode through a small circle attached to the device icon: .

Activating the Monitor Function

Once a device is in online mode, the IndraWorks monitor function can be used. Provided it is on, the monitor displays the I/O status data in decimal format.

Switch the monitor on through the context menu of the device in online mode, see fig. 4-82 "Switching the monitor function on and off" on page 84.

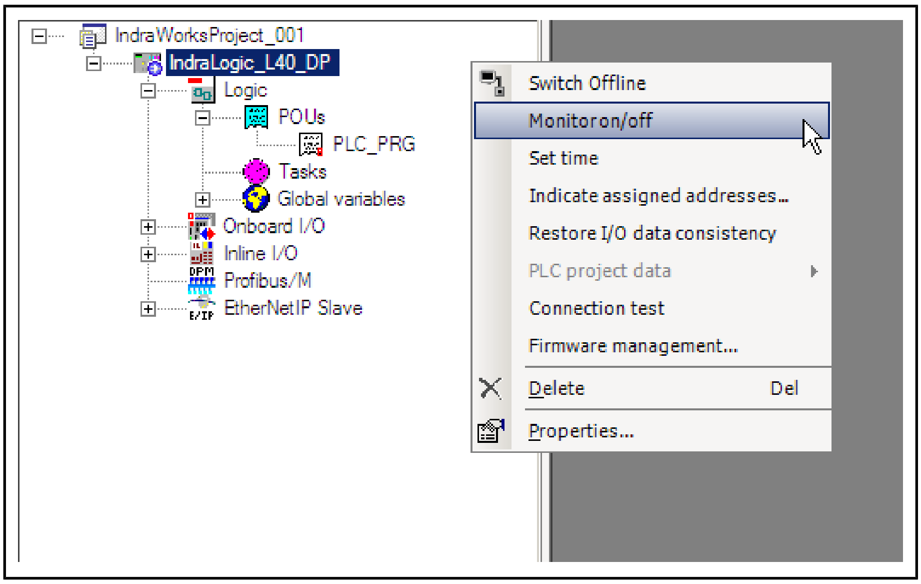
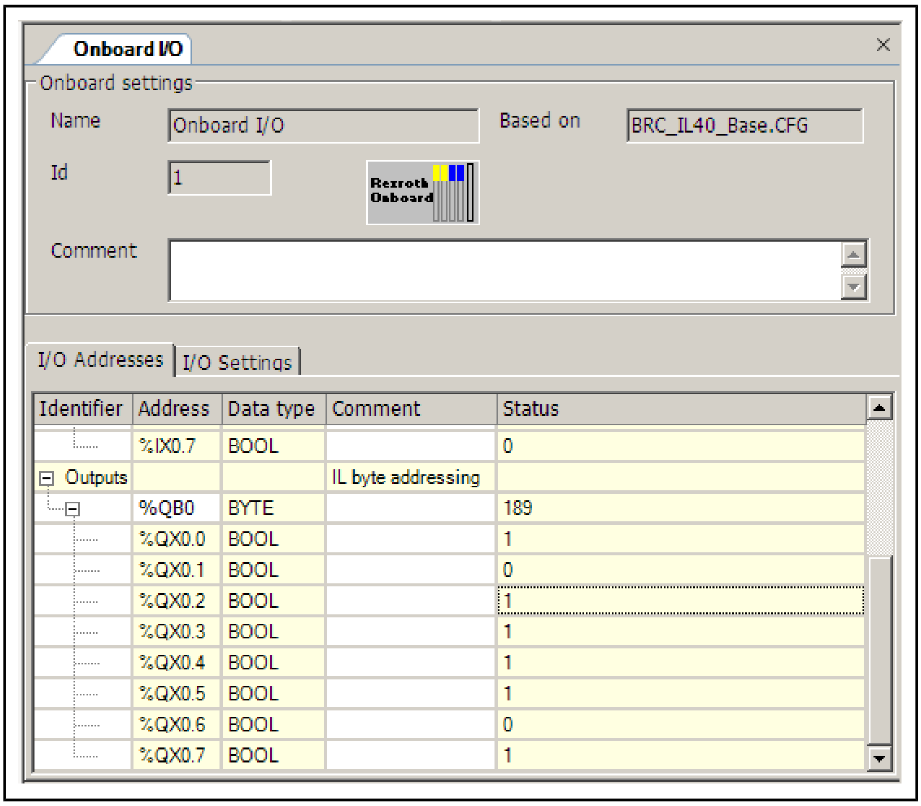


Fig.4-82: Switching the monitor function on and off

Provided the monitor function is activated, the monitor displays the I/O status data in decimal format. To view this data, double-click on "Onboard I/O" in the project explorer. The corresponding values are shown in the "Status" column, see fig. 4-83 "Output of I/O status data" on page 84.



Name: Onboard I/O

Based on: BRC_IL40_Base.CFG

Id: 1

Comment:

Identifier	Address	Data type	Comment	Status
	%IX0.7	BOOL		0
Outputs			IL byte addressing	
	%QB0	BYTE		189
	%QX0.0	BOOL		1
	%QX0.1	BOOL		0
	%QX0.2	BOOL		1
	%QX0.3	BOOL		1
	%QX0.4	BOOL		1
	%QX0.5	BOOL		1
	%QX0.6	BOOL		0
	%QX0.7	BOOL		1

Fig.4-83: Output of I/O status data

Setting the Time

To set the time of devices in online mode, select the "Set time" item of the context menu.

For more information about online functions, please refer to the IndraLogic documentation /2/ and/or help.

4.12.3 Control behavior during turnon

The behavior of the IndraLogic controls is always implemented as follows:

Function	Result
Control/soft panel is switched off / on	Last state (RUN/STOP/INIT)
Reset pushbutton (L-variants)	Last state (RUN/STOP/INIT)
Software reset	STOP
Software reset (cold)	STOP
Software reset (origin)	STOP (boot project deleted)

Fig.4-84: Control behavior during turnon

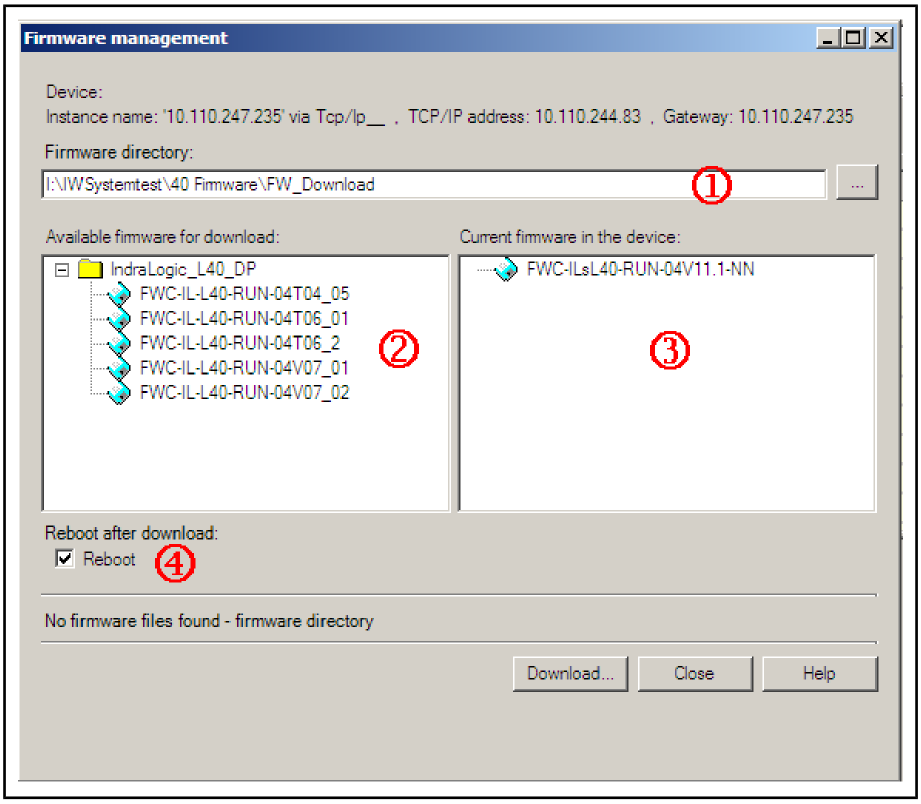
5 Additional Functions

5.1 Firmware Management

5.1.1 Updating the Control Firmware

☞ Firmware can only be downloaded if the control is in the stop state.

1. Copy the firmware files of the IndraLogic L40 DP (*.fw files) required for download to a local directory or to a directory in the network.
2. To open the firmware management dialog, select the "Firmware management" menu item in the context menu of the IndraLogic L40 DP device (in the project explorer). This opens the following dialog:



Firmware management

Device:
Instance name: '10.110.247.235' via Tcp/Ip__ , TCP/IP address: 10.110.244.83 , Gateway: 10.110.247.235

Firmware directory:
I:\IWSystemtest\40 Firmware\FW_Download

Available firmware for download:
- IndraLogic_L40_DP
  - FWC-IL-L40-RUN-04T04_05
  - FWC-IL-L40-RUN-04T06_01
  - FWC-IL-L40-RUN-04T06_2
  - FWC-IL-L40-RUN-04V07_01
  - FWC-IL-L40-RUN-04V07_02

Current firmware in the device:
- FWC-ILsL40-RUN-04V11.1-NN

Reboot after download:
☑ Reboot

No firmware files found - firmware directory

① Path of the new Firmware versions
② Available firmware versions
③ Currently installed firmware versions
④ Checkbox for activating or deactivating the option of rebooting after the download

Fig.5-1: "Firmware management" example dialog

3. Click the ... button to select the directory selected in step 1, see ① in figure 5-1.
4. Select the desired firmware on the left, see ② in figure 5-1. The list only shows firmware versions that are compatible to the current version of the device. Do not change the file names of the firmware updates.
5. The control must be restarted to activate the new firmware. Activate the "Reboot after download" checkbox (see ④ in figure 5-1) if you wish the control to restart immediately after the new firmware has been installed.
6. Click on the "Download" button. Follow the further screen instructions.

CAUTION

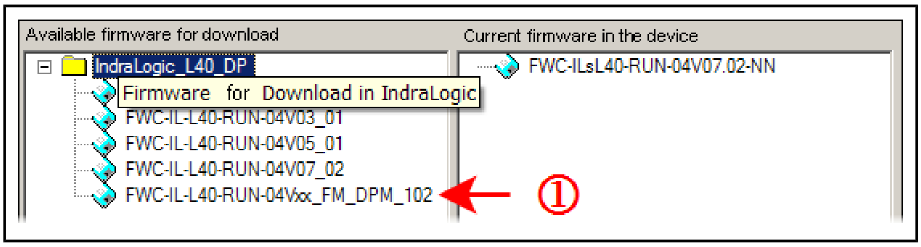
The device might be damaged by a voltage failure while the firmware is changed!

Ensure uninterrupted voltage supply to all devices involved while the firmware is changed!

☞ For more information, please refer to the IndraWorks /5/ documentation or online help.

5.1.2 Updating the Function Module Firmware

As is the case with control firmware, the firmware of function modules can also be updated. To update the firmware of function modules, proceed as described above in chapter 5.1.1 "Updating the Control Firmware" on page 87. The firmware updates available for function modules are listed on the left of the selection window, see ① in Fig. 5-2. Updates for function modules are identified by "FM" in their file name.


Available firmware for download
IndraLogic_L40_DP
Firmware for Download in IndraLogic
FWC-IL-L40-RUN-04V03_01
FWC-IL-L40-RUN-04V05_01
FWC-IL-L40-RUN-04V07_02
FWC-IL-L40-RUN-04Vxx_FM_DPM_102
①
Current firmware in the device
FWC-ILsL40-RUN-04V07.02-NN

① Example of a new firmware version for a function module

Fig.5-2: "Firmware management" example dialog with function module versions

The new firmware is transferred to all function modules of the same type.

5.2 Importing IndraLogic Project Data

The context menu of the device provides the option of importing project data (e.g. POUs, global data, control configurations) from an existing IndraLogic project file.

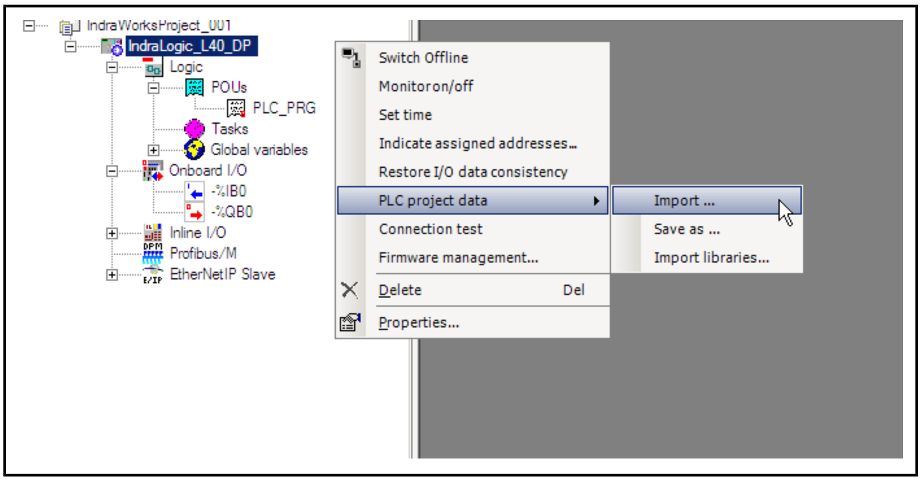


Fig.5-3: IndraLogic L40 context menu: importing PLC project data

☞ For more information, please refer to the IndraWorks documentation /5/ and/or online help.

5.3 IndraLogic Functions

The functions provided in the context menu of the "Logic" object in the project explorer depends on the current system state:

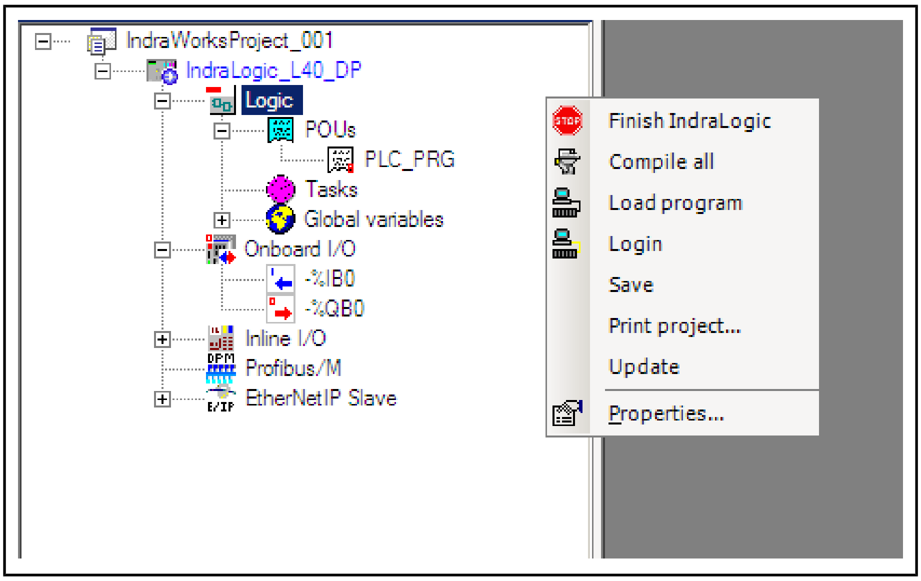


Fig.5-4: Context menu of the "Logic" object with activated IndraLogic

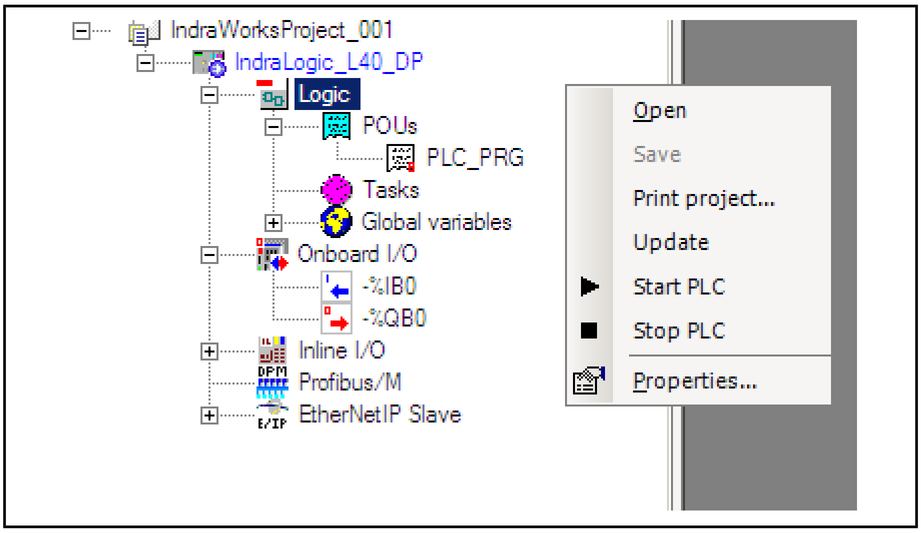


Fig.5-5: Context menu of the "Logic" object with deactivated IndraLogic

Available functions:

- Print, save and compile an IndraLogic project (**Compile all**)
- Start and exit IndraLogic.
- Log the control in or out.
- Update all modules created in IndraLogic and all modifications to configuration settings in the IndraWorks project explorer.

☞ The "Save" and "Properties" items of the context menu are only accessible after IndraLogic has been exited.

Further Settings

Further IndraLogic functions and settings are summarized in the "Further settings" tab:

- Apply settings from IndraLogic

- Safety mode
- Online operation in safety mode
- Create symbol file and send symbol file
- Replace constants
- No address verification
- Optimized IndraLogic call
- Lock menu
- Load boot project automatically

The properties can be called up via the context menu of the "Logic" object, while IndraLogic is deactivated.

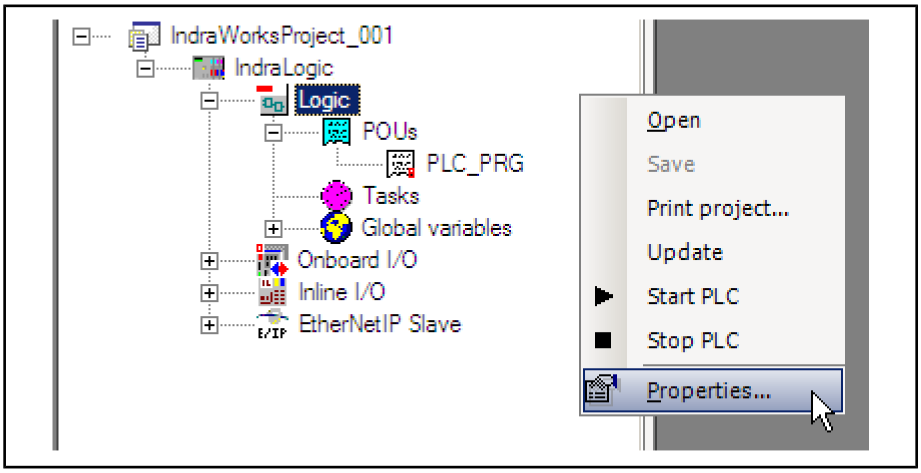


Fig.5-6: IndraLogic L40 context menu: Further Settings

☞ For more information, please refer to the IndraWorks documentation /5/ and/or online help.

6 Libraries

6.1 Overview

Libraries available for the IndraLogic L40 DP:

Library	Function	See
AnalyzationNew	Analysis of expressions	/2/
BuepE_Client	Communication between the IndraLogic L40 DP and the Bosch controls of the CL series	See chapter 6.2 "BuepE_Client " on page 92
HMI_Mkeys	M-key functionalities of the HMI (IndraWorks HMI)	/5/
HMICE_MKEYS	M-key functionality of the IW HMI for VEH30	/5/
Iecsfc	For internal use	–
IL_VCP_DP	Coupling of Rexroth VCP compact operator terminals with Profibus interface	See chapter 6.3 "IL_VCP_DP " on page 95
MP_PLCopen	MotionControl modules	/7/
NetVarUdp_LIB_V23	Application of network variables and of the parameter manager (for data exchange between two or more controls)	Networkfunctionality.pdf[1]
PLCOpenFieldBus	MotionControl based on PLCopen	/7/
ProViDiagnosis	Access to PLC diagnostics (ProVi)	/5/
RIH_CML40	Function blocks for the status of the IndraLogic L40 DP	See chapter 6.4 "RIH_CML40 " on page 97
RIL_Check	Automatic monitoring of internal over-ranges	See chapter 6.5 "RIL_Check.lib" on page 101
RIL_CheckRtv	Same as RIL_Check	See chapter 6.5 "RIL_Check.lib" on page 101
RIL_CommonTypes	For internal use	–
RIL_DeviceNet	Functions in connection with DeviceNet	See chapter 6.6 "RIL_DeviceNet " on page 103
RIL_EthernetIP	Functions in connection with Ethernet/IP	See chapter 6.7 "RIL_EtherNetIP" on page 115
RIL_L40_Util	Function blocks for the status of the IndraLogic L40 DP (no longer applicable; comparable functions in RIH_CML40)	–
RIL_Inline	Diagnostic functions for Inline modules	See chapter 6.9 "RIL_Inline" on page 121
RIL_ProfibusDP	Profibus DPV1 services, diagnostic interface between Profibus master and PLC program, sync and freeze	See chapter 6.10 "RIL_ProfibusDP " on page 137
RIL_ProfibusDP_02	Same functionality as RIL_ProfibusDP, however, with additionally available multimaster extension	See chapter 6.11 "RIL_ProfibusDP_02" on page 154
RIL_Utilities	General IL functions and function blocks	See chapter 6.12 "RIL_Utilities " on page 176

[1] *Documentation about the system libraries (SysLibXXX.pdf) can be found in a subdirectory of the IndraLogic installation, e.g.: ".../Rexroth/IndraWorks/IndraLogic/Documents/English"*

Library	Function	See
RIL_VExUtil	Safe key transmission to an HMI device	See chapter 6.13 "RIL_VExUtil.lib" on page 183
Standard	Standard FBs and functions according to IEC 61131-3	/2/
SysLibCom	Serial communication with the IndraLogic L40 DP	SysLibCom.pdf[1]
SysLibFile	File system support on the IndraLogic L40 DP	SysLibFile.pdf[1]
SysLibFileAsync	Asynchronous file accesses from the IEC application	SysLibFileAsync.pdf[1]
SysLibIecTasks	Management of IEC tasks	SysLibIecTasks.pdf[1]
SysLibMem	Memory management	SysLibMem.pdf[1]
SysLibPlcCtrl	Start, stop and reset of the control	SysLibPlcCtrl.pdf[1]
SysLibRtc	Access to the real-time clock and the battery state of the IndraLogic L40 DP	SysLibRtc.pdf[1]
SysLibSockets	Access to sockets for communication via TCP/IP and UDP	SysLibSockets.pdf[1]
SysLibSocketsAsync	Asynchronous access to sockets for communication via TCP/IP and UDP	SysLibSocketsAsync.pdf
SysLibStr	Character string functions	SysLibStr.pdf[1]
SysLibTime	Retrieve time and date	SysLibTime.pdf[1]
Util	As supplement to the Standard.lib: BCD conversion, bit/byte functions, mathematical help functions, controllers, signal generators, function manipulators, and analog value processing	/2/

Fig.6-1: Overview of IndraLogic L40 libraries

For information on how to link libraries to the current project, please refer to chapter 4.10.4 "Library Manager " on page 74 under PLC Programming with IndraLogic .

☞ IndraWorks automatically downloads libraries for internal use as soon as they are referenced. They do not have to be inserted manually using the library manager.

💡 For more information about the system and firmware libraries, please also refer to the IndraLogic online help.

6.2 BuepE_Client

6.2.1 General Information

The library BuepE_Client provides the communication between the IndraLogic L40 DP and the Bosch controls of the CL series. The IndraLogic L40 DP maps the client functionality. The CL control represents always the server.

6.2.2 BuepE_Client

Brief Description

The BuepE_Client function block can be simultaneously used several times. A new instance is required for every order.

Interface Description

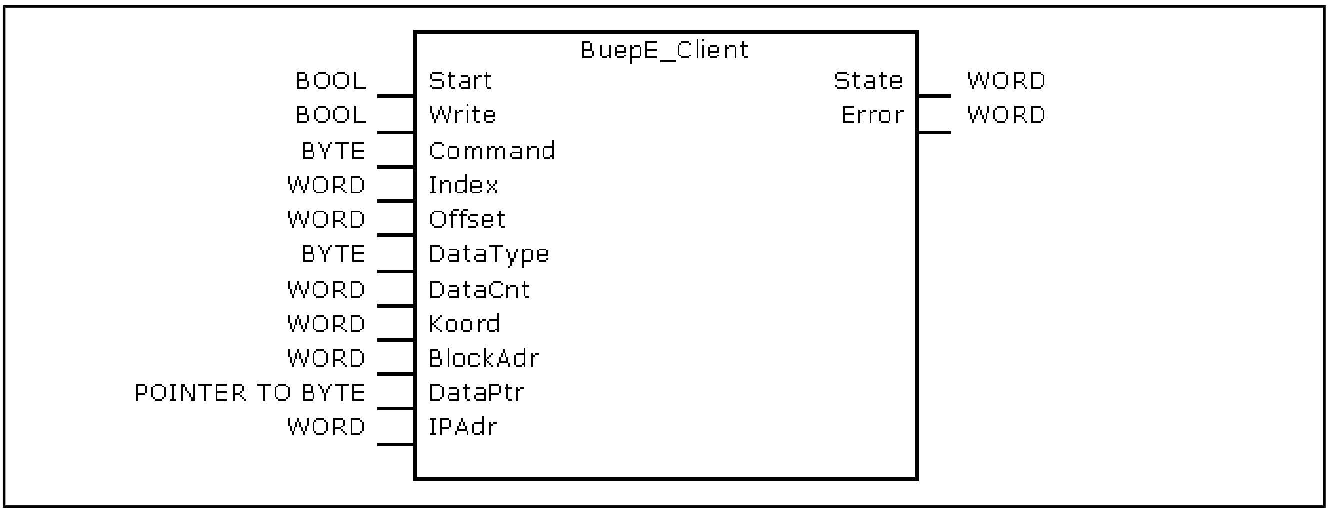


Fig.6-2: BuepE_Client

	Name	Type	Comment
VAR_INPUT	Start	BOOL	TRUE: The transmission is started
	Write	BOOL	TRUE: Writing access FALSE: Reading access
	Command	BYTE	Field type of the Bosch control: 16#44: Data block 16#43: Data field 16#4D: Flag
	Index	WORD	Number of the data block (only for the "Data block" field type)
	Offset	WORD	Byte offset address within the selected data field or data block
	DataType	BYTE	Data type: 0: BYTE 1: WORD
	DataCnt	WORD	Number of the data = f(DataType), depending on the setting in DataType
	Coord	WORD	Coordination flag 0: Uncoordinated
	BlockAdr	WORD	Module block address, only for CL200, CL400 and CL500 16#FFFF: without block address
	DataPtr	POINTER TO BYTE	Pointer to the source data (for "Write"=TRUE) or pointer to the target data (for "Write"=FALSE)
	IPAdr	DWORD	IP address of the Bosch CL control

	Name	Type	Comment
VAR_OUTPUT	State	WORD	Status: 0: Without errors 16#0100: Order in process 16#0305: Order faulty 16#0405: Order not started Further states about the communication functionality of the particular control are described in the documentation .
	Error	WORD	Error: 0: Without errors 16#FF04: No order in process 16#FF20: No UDP socket free (e.g. too many active instances at the same time) 16#FF21: Error during "bind" (e.g. too many active instances at the same time) 16#FF22: Error during "sendto" (e.g. wrong IP address or partner not activated) 16#FF23: Error during "UDP-receive" 16#FF24: Errors in UDP-receive length 16#FF25: Too many repetitions (partner does not respond, e.g. incorrect IP address) Further states about the communication functionality of the particular control are described in the documentation .

Fig.6-3: Interface of BuepE_Client

Example The following program extract shows an example of the BuepE_Client call.

Program:

```
VAR
 Bclient : BuepE_Client;         (* Declare instance *)
 MyDB : ARRAY [0..511] OF BYTE;  (* Data block data array*)
END_VAR
(* example: Read a DB from a CL500-ZS1 *)
Bclient( Start := TRUE,          (* Start transmission *)
         Write := FALSE,         (* Reading access)
         Command := 16#44,       (* Data block *)
         index := 0,             (* DB number *)
         offset := 0,            (* No address offset *)
         DataType := 0,          (* Byte *)
         DataCnt := 512,         (* 512 bytes *)
         Coord := 0,             (* Uncoordinated *)
         BlockAdr := 16#0008,    (* ZS 1 *)
         DataPtr := ADR(MyDB),   (* Pointer to data block-data array *)
         IPAdr := SysSockInetAddr('10.110.244.46'),
      (* state=> ,*)
      (* error=>*)
);
```

6.3 IL_VCP_DP

6.3.1 Overview

This library is used to couple Rexroth VCP compact operator terminals with Profibus interface to the PLC. There are three alternative function blocks to achieve this:

- VCP_PBS16_A4096
- VCP_PBS32_A4096
- VCP_PBS32_A65536

6.3.2 VCP_PBS16_A4096

Brief Description

This function block (FB) activates the Profibus DP protocol for the compact operator terminals VCPxx. Additionally, the I/O image of the physical addresses is transmitted between PLC and operator terminal.

The data capacity for data transmission is 16 bytes. The size of the address range available through an ARRAY is 4096 bytes (inputs and outputs in total).

Interface Description

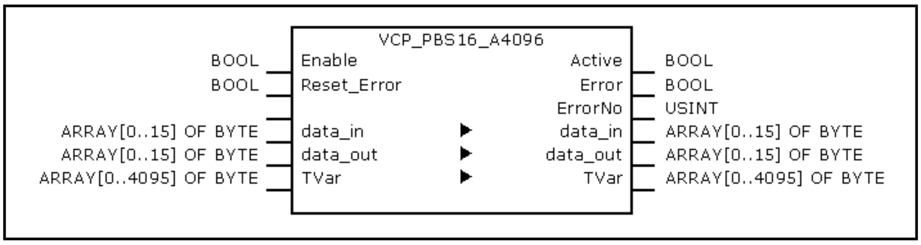


Fig.6-4: VCP_PBS16_A4096

	Name	Type	Comment
VAR_INPUT	Enable	BOOL	TRUE: FB is processed FALSE: FB is not processed
	Reset_Error	BOOL	TRUE: Resets "Error" (to FALSE) and sets "ErrorNo" to 0
VAR_IN_OUT	Data_in	ARRAY [0..15] OF BYTE	Data to connect the physical inputs of the compact operator terminal
	Data_out	ARRAY [0..15] OF BYTE	Data to connect the physical outputs of the compact operator terminal
	TVar	ARRAY [0..4095] OF BYTE	Array to read from and write to the compact operator terminal
VAR_OUTPUT	Active	BOOL	TRUE, as long as "Enable" is also TRUE
	Error	BOOL	TRUE, as soon as there is an error Can be reset with "Reset-Error".
	ErrorNo	USINT	Error type: 2: InputRangeError4: calculation error

Fig.6-5: VCP_PBS16_A4096 interface

VI-Composer

If the Rexroth VI-Composer is used for configuration, the addresses in the variables list refer to the appropriate byte in the "TVar" array which the PLC program uses for data exchange.

Error Handling

As soon as an error occurs, communication is interrupted and the display of the compact operator terminal shows COMMUNICATION ERROR, ERROR CODE 110.

The error type (ErrorNo) shows, whether the error is an address calculation error (CalculationError), or whether the data capacity selected for Data_in or Data_out is too big (InputRangeError).

6.3.3 VCP_PBS32_A4096

Brief Description This function block (FB) activates the Profibus DP protocol for the compact operator terminals VCPxx. Additionally, the I/O image of the physical addresses is transmitted between PLC and operator terminal.

The data capacity for data transmission is 32 bytes. The size of the address range available through an ARRAY is 4096 bytes (inputs and outputs in total).

Interface Description

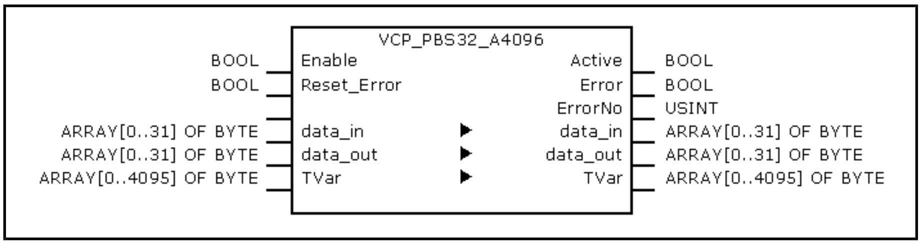


Fig.6-6: VCP_PBS32_A4096

	Name	Type	Comment
VAR_INPUT	Enable	BOOL	TRUE: FB is processed FALSE: FB is not processed
	Reset_Error	BOOL	TRUE: Resets "Error" (to FALSE) and sets "ErrorNo" to 0
VAR_IN_OUT	Data_in	ARRAY [0..31] OF BYTE	Data to connect the physical inputs of the compact operator terminal
	Data_out	ARRAY [0..31] OF BYTE	Data to connect the physical outputs of the compact operator terminal
	TVar	ARRAY [0..4095] OF BYTE	Array to read from and write to the compact operator terminal
VAR_OUTPUT	Active	BOOL	TRUE, as long as "Enable" is also TRUE
	Error	BOOL	TRUE, as soon as there is an error Can be reset with "Reset-Error".
	ErrorNo	USINT	Error type: 4: calculation error

Fig.6-7: VCP_PBS32_A4096 interface

VI-Composer If the Rexroth VI-Composer is used for configuration, the addresses in the variables list refer to the appropriate byte in the "TVar" array which the PLC program uses for data exchange.

Error Handling As soon as an error occurs, communication is interrupted and the display of the compact operator terminal shows COMMUNICATION ERROR, ERROR CODE 110.

The error type (ErrorNo) indicates that the error is an address calculation error (CalculationError).

6.3.4 VCP_PBS32_A65536

Brief Description This function block (FB) activates the Profibus DP protocol for the compact operator terminals VCPxx. Additionally, the I/O image of the physical addresses is transmitted between PLC and operator terminal.

The data capacity for data transmission is 32 bytes. The size of the address range available through an ARRAY is 65536 bytes (inputs and outputs in total).

Interface Description

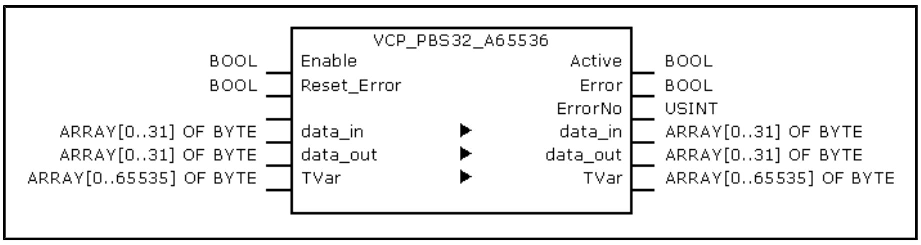


Fig.6-8: VCP_PBS32_A65536

	Name	Type	Comment
VAR_INPUT	Enable	BOOL	TRUE: FB is processed FALSE: FB is not processed
	Reset_Error	BOOL	TRUE: Resets "Error" (to FALSE) and sets "ErrorNo" to 0
VAR_IN_OUT	Data_in	ARRAY [0..31] OF BYTE	Data to connect the physical inputs of the compact operator terminal
	Data_out	ARRAY [0..31] OF BYTE	Data to connect the physical outputs of the compact operator terminal
	TVar	ARRAY [0..65535] OF BYTE	Array to read from and write to the compact operator terminal
VAR_OUTPUT	Active	BOOL	TRUE, as long as "Enable" is also TRUE
	Error	BOOL	TRUE, as soon as there is an error Can be reset with "Reset-Error".
	ErrorNo	USINT	Error type: 4: calculation error

Fig.6-9: VCP_PBS32_A65536 interface

VI-Composer

If the Rexroth VI-Composer is used for configuration, the addresses in the variables list refer to the appropriate byte in the "TVar" array which the PLC program uses for data exchange.

Error Handling

As soon as an error occurs, communication is interrupted and the display of the compact operator terminal shows COMMUNICATION ERROR, ERROR CODE 110.

The error type (ErrorNo) indicates that the error is an address calculation error (CalculationError).

6.4 RIH_CML40

6.4.1 Overview

- IH_GetOhcCtrl: Read operating hours counter of the control
- IH_GetOhcFan: Read fan operating hours counter
- IH_ResetOhcFan: Reset fan operating hours counter
- IH_SetDisplay: Show user displays
- IH_FanStatus: Determine fan status (on/off)
- IH_Temperature: Determine current internal temperature of the control
- IH_TempWarning: Check for over-temperature warning

6.4.2 IH_GetOhcCtrl

Brief Description Supplies the operating time of the control in hours.

☞ It is not possible to reset the operating hours meter.

Interface Description

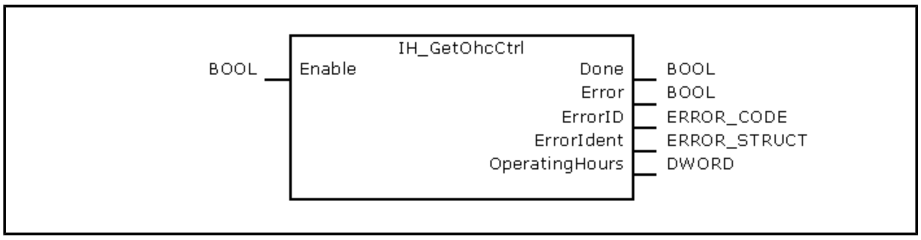


Fig.6-10: IH_GetOhcCtrl

	Name	Type	Comment
VAR_INPUT	Enable	BOOL	TRUE: FB is processed FALSE: FB is not executed
VAR_OUTPUT	Done	BOOL	TRUE: OperatingHours relevant FALSE: OperatingHours in process
	Error	BOOL	Always 0
	ErrorID	ERROR_CODE	Undefined and cannot be evaluated
	ErrorIdent	ERROR_STRUCT	Undefined and cannot be evaluated
	OperatingHours	DWORD	Operating time of the control in hours

Fig.6-11: Interface of IH_GetOhcCtrl

6.4.3 IH_GetOhcFan

Brief Description Supplies the operating time of the fan in hours.

Interface Description

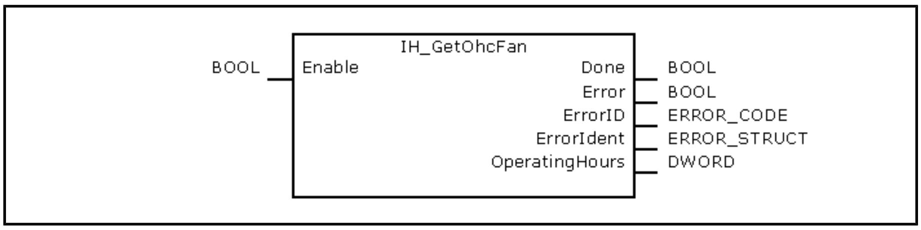


Fig.6-12: IH_GetOhcFan:

	Name	Type	Comment
VAR_INPUT	Enable	BOOL	TRUE: FB is processed FALSE: FB is not processed
VAR_OUTPUT	Done	BOOL	TRUE: OperatingHours relevant FALSE: OperatingHours in process
	Error	BOOL	Always 0
	ErrorID	ERROR_CODE	Undefined; evaluation not possible

	Name	Type	Comment
	ErrorIdent	ERROR_STRUCT	Undefined; evaluation not possible
	OperatingHours	DWORD	Operating time of the fan in hours

Fig.6-13: Interface of IH_GetOhcFan

6.4.4 IH_ResetOhcFan

Brief Description Sets the operating hours counter of the fan to zero.

Interface Description

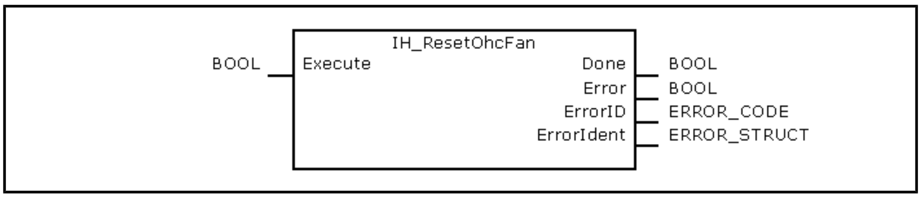


Fig.6-14: IH_ResetOhcFan

	Name	Type	Comment
VAR_INPUT	Execute	BOOL	Positive edge resets the fan operating hours counter (=0).
VAR_OUTPUT	Done	BOOL	TRUE: Resetting was successful FALSE: Resetting was not successfull or is not yet completed
	Error	BOOL	Always 0
	ErrorID	ERROR_CODE	Undefined; evaluation not possible
	ErrorIdent	ERROR_STRUCT	Undefined; evaluation not possible

Fig.6-15: Interface of IH_ResetOhcFan

6.4.5 IH_SetDisplay

Brief Description Outputs a user message of up to 80 characters on the LCD display of the control.

Interface Description

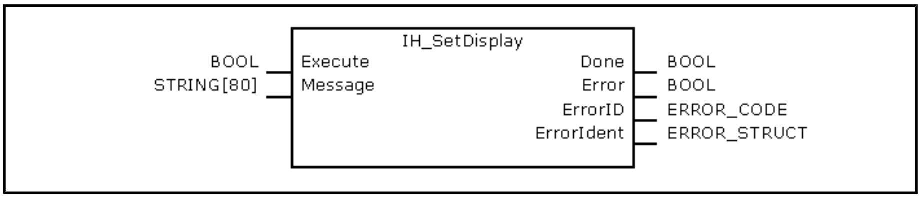


Fig.6-16: IH_SetDisplay

	Name	Type	Comment
VAR_INPUT	Execute	BOOL	A positive edge starts the output of the string transmitted at the "Message" input.
	Message	STRING[80]	String with up to 80 characters to be output on the display. Empty string: Delete the pending display.
VAR_OUTPUT	Done	BOOL	TRUE: The string is output to the display FALSE: Processing not yet completed
	Error	BOOL	Always 0

Libraries

	Name	Type	Comment
	ErrorID	ERROR_CODE	Undefined; evaluation not possible
	ErrorIdent	ERROR_STRUCT	Undefined; evaluation not possible

Fig.6-17: IH_SetDisplay interface

As long as there is no error message, a flashing "IL:" on the default display (Stop/Run) indicates that there is an active message.

To read the message, use the <up> or <down> key to open the "IL display", see Chapter "Display and Operating Keys". The message, preceded by "IL:", will then be output on the control display in a rotating manner.

After an empty string has been transmitted, the flashing "IL:" is cleared.

6.4.6 IH_FanStatus

Brief Description

Determines, whether the fan is currently switched on or switched off (replaces function "IL_GetStateFan" from library "RIL_L40_Util").

Interface Description

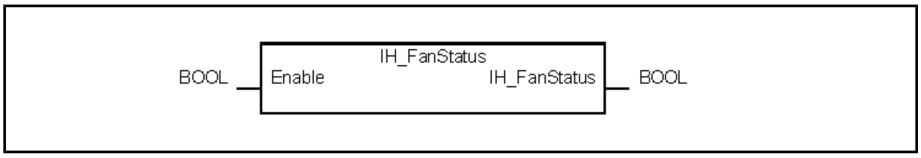


Fig.6-18: IH_FanStatus

	Name	Type	Comment
VAR_INPUT	Enable	BOOL	TRUE: Output value is valid. FALSE: Output value is not representative.
Function value	IH_FanStatus	BOOL	TRUE: Fan is switched on FALSE: Fan is switched off

Fig.6-19: IH_FanStatus interface

6.4.7 IH_Temperature

Brief Description

Determines the internal temperature of the control (replaces the "IL_GetTemp" function from the "RIL_L40_Util" library).

Interface Description

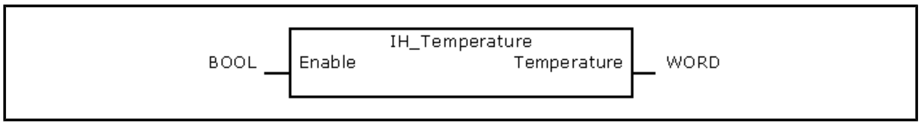


Fig.6-20: IH_Temperature

	Name	Type	Comment
VAR_INPUT	Enable	BOOL	TRUE: Output of the current temperature FALSE: No output
Function value	Temperature	WORD	Temperature: Bit 15: 0 = positive temperature 1 = negative temperature Bits 14 – 8: temperature in degrees Celsius Bit 7: 1 = decimal place (half degree Celsius) 0 = no decimal place

Fig.6-21: IH_Temperature interface

Examples

Function value	Temperature
2#00010100_10000000	+20.5 °C
2#01001011_00000000	+75 °C
2#10000101_10000000	-5.5 °C

Fig.6-22: Examples of IH_Temperature

6.4.8 IH_TempWarning

Brief Description

Determines, if the internal temperature of the control has exceeded the critical value of **70 °C** (replaces the IL_TempWarning function from the RIL_L40_Util library).

Interface Description

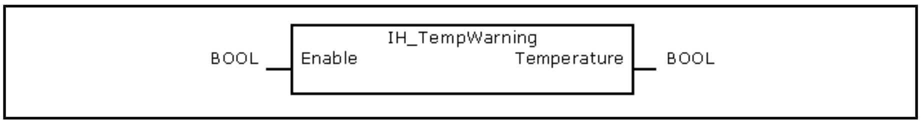


Fig.6-23: IH_TempWarning

	Name	Type	Comment
VAR_INPUT	Enable	BOOL	TRUE: Output value is valid. FALSE: Output value is not representative.
Function value	Temperature	BOOL	TRUE: Temperature warning FALSE: No temperature warning

Fig.6-24: IH_TempWarning interface

☞ If the internal temperature reaches **80 °C**, the control automatically switches to the "Stop" mode. The outputs enter the safe status, and the "Temp !!!" warning is emitted on the display. This mode can only be exited by switching the power supply off/on.

6.5 RIL_Check.lib

Faulty accesses (accidentally) programmed outside of the ranges of arrays and subrange types of variables, as well as the division by zero, are not compensated by the IndraLogic compiler and/or the runtime system and cause partly unpredictable errors during program processing. When implementing the RIL_Check.libm, such over-ranges are monitored and prevented. Thereby, the access for arrays and subrange types are limited to the smallest or highest possible value. Thus, e. g. concerning an array, the element with the highest indices is accessed, even if a higher index value was specified in the PLC program. For a division by zero the divisor is replaced by "1".

☞ If the RIL_Check.lib library is integrated in the PLC project, before each testable operation the corresponding test function is automatically inserted (invisible for the user).

Further function calls are not necessary!

☞ If library RIL_Check.lib is integrated in the PLC project, the PLC cycle time is charged, as each testable operation is automatically monitored. With respect to the robustness of a PLC project and the safety of the whole system we recommend to use the RIL_Check.lib in the PLC project.

When using the MP_PLCOpen.lib, the RIL_Check.lib is required. In this case the RIL_Check.lib must be integrated in the PLC project.

Functions All functions contained in the RIL_Check library (see fig. 6-25 "Functions of RIL_Check" on page 102) are automatically integrated in the PLC program and must not be explicitly called.

Designation	Description
CheckBounds	Automatic verification, if the permissible indices of the access to array elements is exceeded or fallen below the minimum value.
CheckDivByte	Automatic verification on division by zero (BYTE access).
CheckDivDWord	Automatic verification on division by zero (DWORD access).
CheckDivReal	Automatic verification on division by zero (REAL access).
CheckDivWord	Automatic verification on division by zero (WORD access).
CheckRangeSigned	Automatic verification, if the permissible value range of a signed variable is exceeded or fallen below the minimum value.
CheckRangeUnsigned	Automatic verification, if the permissible value range of an unsigned variable is exceeded or fallen below the minimum value.

Fig.6-25: Functions of RIL_Check

CheckExceedingOccurred The error cause can be determined exactly by requesting the bits in the global variable "CheckExceedingOccurred" directly. All error accesses are prevented by RIL_Check, so that the permissible value range is not exceeded or fallen below the minimum value and that no division by zero occurs. The single bits have the following meaning:

Bit variable	Value	Meaning
CheckExceedingOccurred.0	16#01	CheckBoundsLowerLimitation: The permissible indices of the access to array elements are fallen below the minimum value.
CheckExceedingOccurred.1	16#02	CheckBoundsUpperLimitation: The permissible indices of the access to array elements are exceeded.
CheckExceedingOccurred.2	16#04	CheckBoundsExceedingLimitation: The permissible indices of the access to array elements are exceeded or fallen below the minimum value.

Bit variable	Value	Meaning
CheckExceedingOccurred.3	16#08	CheckRangeLowerLimitation: The permissible value range of a variable is fallen below the minimum value.
CheckExceedingOccurred.4	16#16	CheckRangeUpperLimitation: The permissible value range of a variable is exceeded.
CheckExceedingOccurred.5	16#32	CheckRangeExceedingLimitation: The permissible value range of a variable is exceeded or fallen below the minimum value.
CheckExceedingOccurred.6	16#64	DivisionByZeroPrevention: Division by zero.

Fig.6-26: CheckExceedingOccurred

Sample Program

The following PLC sample program shows the use of variable CheckExceedingOccurred. The limitation of arrays with CheckBounds always sets the respective bit in error case, when the permissible value range is exceeded or fallen below the minimum value (CheckBoundsLowerLimitation or CheckBoundsUpperLimitation) and the general bit of the access violation (CheckBoundsExceedingLimitation). Thus, a general or detailed verification, if a limit value was exceeded, can be realized.

Program:

```
CheckExceedingOccurred := 16#00;     (* Resetting the variables *)

IF Axis_Data[AxisNo].bCheckAccessOK    (* Array access *)
THEN
                ...
 IF CheckExceedingOccurred.2 (* Array access violation detected? *)
THEN
 IF CheckExceedingOccurred.0 (* Bit set, if value below specified range? *)
 THEN String := 'Access below the possible array range'
                ...
END_IF
IF CheckExceedingOccurred.0  (* Bit set, if value above specified range? *)
THEN String := 'Access above the possible array range'
                ...
END_IF
ELSE String := 'Access successful'
END_IF
```

6.6 RIL_DeviceNet

6.6.1 Overview

The DeviceNet master function module enables a connection of DeviceNet slaves and the access via the standardized DeviceNet protocol (EN 50325). Important DeviceNet services can be accessed, controlled by the program, using this library.

Designation	Type	Description
IL_BusDiagDN	FB	Function block for DeviceNet bus diagnosis
IL_BusStateDN	FB	Function block to determine the DeviceNet operating status
IL_DevDiagDN	FB	Function block for DeviceNet slave diagnosis
IL_ExplicitMsgDN	FB	General function block for acyclic master-slave communication (Explicit Messaging)

Designation	Type	Description
IL_GetAttribSingleDN	FB	Function block for DeviceNet communication service SINGLE_GET_ATTRIBUTE (read operation)
IL_SetAttribSingleDN	FB	Function block for DeviceNet communication service SINGLE_SET_ATTRIBUTE (write operation)
Version_RIL_DeviceNet_02V01	FN	Internal version function for library RIL_DeviceNet.lib

Fig.6-27: RIL_DeviceNet.lib functions and function blocks

Designation	Type	Description
IL_FB_STATE_DN	ENUM	Internal state machine for DeviceNet function blocks whose data are provided by the message interface of the DeviceNet master.
IL_MSG_ERROR_DN	ENUM	Definition: DeviceNet communication error
IL_BUS_DIAG_FLAGS_DN	STRUCT	Status bit of a DeviceNet slave
IL_BUS_DIAG_GLOBAL_BITS_DN	STRUCT	Status bits of the global bus diagnostic bit field within the DeviceNet master
IL_BUS_DIAG_STATE_DN	ENUM	Definition: Operation states of the DeviceNet master
IL_DEV_STATUS_DN	STRUCT	Bit field for slave diagnosis
IL_EXPL_MSG_SERVICE_DN	ENUM	List of the general acyclic DeviceNet specification (see DeviceNet specification Vol.1, Appendix G*)

Fig.6-28: Used data types for RIL_DeviceNet.lib

6.6.2 Data Types

IL_MSG_ERROR_DN

Brief Description "IL_MSG_ERROR_DN" maps errors which can occur during the use of a DeviceNet function block.

Library	Area
RIL_DeviceNet.lib	Common

Fig.6-29: Library assignment: IL_MSG_ERROR_DN

Type Enumeration type ENUM

Definition The following errors are defined:

Identifier	Description
NO_ERROR_DN_MSG	No data exchange active
TIMEOUT_WAIT_FOR_ACCESS_DN	Timeout during the message interface is requested
RESOURCES_UNAVAILABLE_DN	Resource not available
SERVICE_NOT_AVAILABLE_DN	DeviceNet service not available
INVALID_ATTRIBUTE_VALUE_DN	Attribute invalid
ALREADY_IN_REQUEST_MODE_DN	Slave already executes the request
OBJECT_STATE_CONFLICT_DN	Object state conflict
ATTRIBUTE_NOT_SETTABLE_DN	Write access to attribute not allowed
PERMISSION_CHECK_FAILED_DN	No access rights

Identifier	Description
STATE_CONFLICT_DN	Command non-executable
TIME_OUT_ERROR_DN	Timeout slave response telegram
NOT_ENOUGH_DATA_RECEIVED_DN	Transmitted data incomplete
ATTRIBUTE_NOT_SUPPORTED_DN	Attribute not available on slave
OBJECT_NOT_EXISTS_DN	Addressed object not available on slave
INTERNAL_BUFFER_OVERFLOW_DN	Internal error
VENDOR_SPECIFIC_ERROR_CODE	Vendor-specific ErrorCode
FORMAT_ERROR_RESPONSE_DN	Response data faulty
RESET_COMMAND_STILL_ACTIVE_DN	Reset command active
MAC_ID_OUT_OF_RANGE_DN	MAC Id invalid
SEQ_ERR_IN_FRAG_RESP_SEQUENCE_DN	Fragmentation error during slave response occurred
MASTER_NOT_CONFIGURED_DN	DeviceNet master not configured

Fig.6-30: Description of IL_MSG_ERROR_DN elements

IL_BUS_DIAG_FLAGS_DN

Brief Description Communication state of a slave.

Library	Area
RIL_DeviceNet.lib	Diagnostics

Fig.6-31: Library assignment: IL_BUS_DIAG_FLAGS_DN

Type STRUCT

Definition The following communication states are defined:

Identifier	Description
Configured	This bit indicates, that the bus participant does not exist in the configuration
IO_Connection	This bit indicates that the bus participant is available
Diagnostics	A diagnosis is pending in the queue of the bus device. The flag is reset as soon as the diagnostics has been retrieved.
ExplicitConnection	This bit indicates, that a connection via Explicit Messaging exists

Fig.6-32: Description of IL_BUS_DiAG_FLAGS_DN elements

IL_BUS_DIAG_GLOBAL_BITS_DN

Brief Description Bus error summary which are detected by the master.

Library	Area
RIL_DeviceNet.lib	Diagnostics

Fig.6-33: Library assignment: IL_BUS_DIAG_GLOBAL_BITS_DN

Type STRUCT

Definition The following bus errors are defined:

Identifier	Description
ControlError	This bit indicates that there is a master parameterization error.
AutoClearError	The device interrupted the communication to all other devices (nodes) and has reached the Auto Clear end status.
NonExchangeError	This bit indicates that at least one device has not reached the data exchange status and no process data is exchanged with the device.
FatalError	This bit indicates that the communication is not possible due to a severe bus error
EventError	The used CAN chip has recognized transmission errors. The bit is set after an error occurs for the first time and is not reset again.
HostNotReady	This bit indicates that the PLC is not ready to receive data.
DuplicateMacId	This bit indicates that a device witth identical MAC ID was detected at the bus.
DuplMacIdCheck	The DeviceNet master sets the bit as long as it carries out the test if there's a double MAC ID.

Fig.6-34: Description of IL_BUS_DIAG_GLOBAL_BITS_DN elements

IL_BUS_DIAG_STATE_DN

Brief Description Possible states of DeviceNet master switch-on

Library	Area
RIL_DeviceNet.lib	Diagnostics

Fig.6-35: Library assignment: IL_BUS_DIAG_STATE_DN

Type Enumeration type ENUM

Definition The following master state bits are defined:

Identifier	Description
OFFLINE_MASTER_DN	Master Offline
STOP_MASTER_DN	Master stopped
CLEAR_MASTER_DN	Master in Clear Mode
OPERATE_MASTER_DN	Bus operation active

Fig.6-36: Description of IL_MSG_ERROR_DN elements

IL_DEV_STATUS_DN

Brief Description Structure with diagnostics bits of a DeviceNet slave.

Library	Area
RIL_DeviceNet.lib	Diagnostics

Fig.6-37: Library assignment: IL_DEV_STATUS_DN

Type STRUCT

Definition The following diagnostic bits are defined:

Identifier	Description
NoResponse	The bus participant is configured, however, does not respond.
BufferOverflow	Error buffer overflow of the bus participant

Identifier	Description
PrmFault	The bus participant was faultily parameterized.
CfgFault	Configuration error: The configured I/O data width doesn not correspond to the actual I/O data width.
SupportUCMM	Bus participant supports connectionless services (UCMM)
Deactivated	Bus participant not activated in current configuration.

Fig.6-38: Description of IL_DEV_STATUS_DN elements

IL_FB_STATE_DN

Brief Description

The access is possible via the message interface of the DeviceNet master via special function blocks (Slave diagnostics, Explicit Messaging). As the data access is not always ensured immediately, individual steps of the edge-controlled function blocks are mapped using the state machine and are reported via IL_FB_STATE_DN.

Library	Area
RIL_DeviceNet.lib	Common

Fig.6-39: Library assignment: IL_FB_STATE_DN

Type

Enumeration type ENUM

Definition

The following steps are defined:

Identifier	Description
WAIT_START_MSG_DN	Function block waits for input Execute:=TRUE
CHK_DATA_MSG_DN	Function block verifies the validity of the input data
SND_DATA_MSG_DN	Function block writes request data in the message interface.
GET_DATA_MSG_DN	Function block polls response data / acknowledgement
WAIT_RESET_MSG_DN	Function block is waiting for negative edge at input Execute.

Fig.6-40: Decritpion of IL_FB_STATE_DN elements

6.6.3 Version_RIL_DeviceNet_02V01

Brief Description

For versioning of the "RIL_DeviceNet.lib" library.

Library	Area
RIL_DeviceNet.lib	Version

Fig.6-41: Library assignment: Version_RIL_DeviceNet_02V01

Interface Description

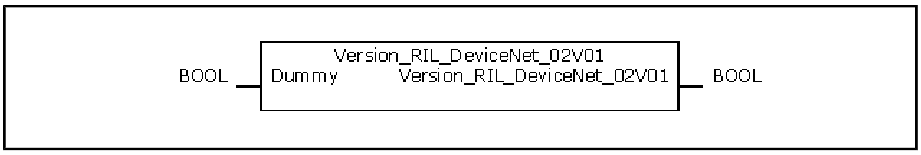


Fig.6-42: Structure: Version_RIL_DeviceNet_02V01

Libraries

	Name	Type	Comment
VAR_INPUT	Dummy	BOOL	Dummy
VAR_OUTPUT	Version_RIL_DeviceNet_02V01	BOOL	TRUE: Library is valid

Fig.6-43: Interface signals: Version_RIL_DeviceNet_02V01

Functional description

The function should only be used for internal check mechanisms. If library "RIL_DeviceNet.lib" is contained in the current PLC project, it is verified during the download of the PLC program whether the library functionality of the selected target system's firmware is supported. If this is not the case, a corresponding error message is generated during the download.

☞ The verification of the system and the releases is also active, if function Version_RIL_DeviceNet_02V01 is not used.

Error Handling

Function generates no error at run time.

6.6.4 IL_BusStateDN

Brief Description

Determines the operating status of the selected DeviceNet master.

Library	Area
RIL_DeviceNet.lib	Diagnostics

Fig.6-44: Library assignment: IL_BusStateDN

Interface Description

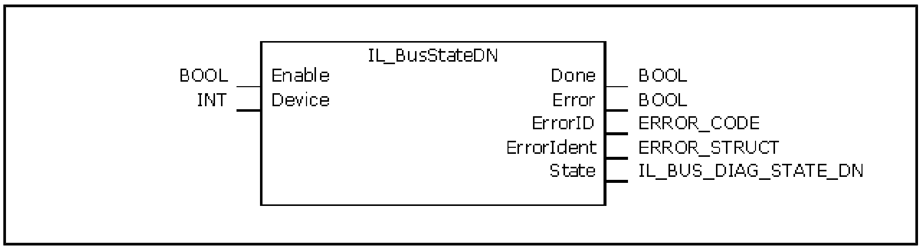


Fig.6-45: Structure: IL_BusStateDN

	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Processing enable of the function block.
	Device	INT	Selected DeviceNet master (for parallel operation of several field buses). The value always has to be set to 0.
VAR_OUTPUT	Done	BOOL	TRUE: The service is stopped or an error has occurred.
	Error	BOOL	TRUE: An error has occurred.
	ErrorID	ERROR_CODE	Error short description.
	ErrorIdent	ERROR_STRUCT	Detailled error description according to error table.
	State	IL_BUS_DIAG_STATE_DN	Operating state of the selected DeviceNet master.

Fig.6-46: Interface signals: IL_BusStateDN

Functional description

The function block "IL_BusStateDN" cyclically determines the operating state of the DeviceNet master as soon as the "Enable" input is set.

Error Handling

An error is signalized during the processing of the function block by setting the "Error" output. Information regarding the error can be obtained via "ErrorID" and "ErrorIdent", see chapter 6.6.10 "Error Messages of DeviceNet Function Blocks" on page 114.

6.6.5 IL_BusDiagDN

Brief Description

The DeviceNet bus diagnostics contains the communication states of all bus participants. They are retrieved via the function block and provided at the corresponding function block outputs.

Library	Area
RIL_DeviceNet.lib	Diagnostics

Fig.6-47: Library assignment: IL_BusDiagDN

Interface Description

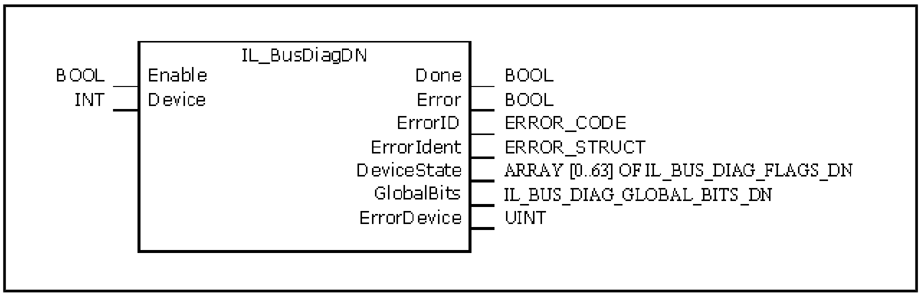


Fig.6-48: Structure: IL_BusDiagDN

	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Processing enable of the function block. As long as the input is set to TRUE, the bus diagnosis is immediately refreshed after every processing of the function block.
	Device	INT	Selected DeviceNet master (for parallel operation of several field buses). The value always has to be set to 0.
VAR_OUTPUT	Done	BOOL	Master status information at output State is valid.
	Error	BOOL	TRUE: An error has occurred.
	ErrorID	ERROR_CODE	Error short description.
	ErrorIdent	ERROR_STRUCT	Detailled error description according to error table.
	DeviceState	ARRAY [0..63] OF IL_BUS_DIAG_FLAGS_DN	Status flags of all slave bus participants.
	GlobalBits	IL_BUS_DIAG_GLOBAL_BITS_DN	Bus error flags
	ErrorDevice	UINT	If one of the bus error flags "NonExchangeError", "AutoClearError" or "ControlError" is set, the MAC ID of the faulty bus device is mapped. If the error cause is due to the master, the "ErrorDevice" is set to 255. Otherwise the MAC ID of the faulty slave is indicated.

Fig.6-49: Interface signals: IL_BusDiagDN

Functional description

As soon as the "Enable" input is set, the function block cyclically performs a bus diagnostics at the selected DeviceNet master.

Error Handling

An error is signalized during the processing of the function block by setting the "Error" output. Information regarding the error can be obtained via "ErrorID" and "ErrorIdent", see chapter 6.6.10 "Error Messages of DeviceNet Function Blocks" on page 114.

6.6.6 IL_DevDiagDN

Brief Description

Determines diagnostic data of the selected slave.

Library	Area
RIL_DeviceNet.lib	Diagnostics

Fig.6-50: Library assignment: IL_DevDiagDN

Interface Description

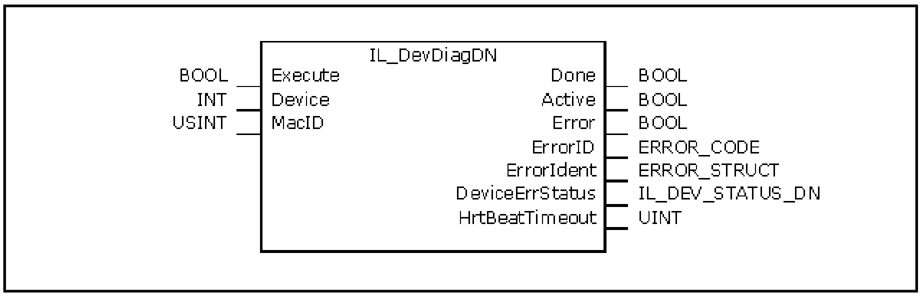


Fig.6-51: Structure: IL_DevDiagDN

	Name	Type	Comment
VAR_INPUT	Execute	BOOL	Processing enable of the function block.
	Device	INT	Selected DeviceNet master (for parallel operation of several field buses). The value always has to be set to 0.
	MacID	USINT	MAC ID of the selected slave.
VAR_OUTPUT	Done	BOOL	Diagnostics available. The outputs are valid.
	Active	BOOL	Diagnostics is requested.
	Error	BOOL	TRUE: An error has occurred.
	ErrorID	ERROR_CODE	Error short description.
	ErrorIdent	ERROR_STRUCT	Detailled error description according to error table.
	DeviceErrStatus	IL_DEV__STATUS_DN	Diagnostic flags of the selected DeviceNet slave.
	HrtBeatTimeout	UINT	Timeout counter: The number of the occurred timeouts gives information about the transmission quality.

Fig.6-52: Interface signals: IL_DevDiagDN

Functional description

If a slave indicates a diagnosis, its diagnostic data can be requested by "IL_DevDiagDN" via the message interface of the DeviceNet master. As soon as a positive edge is recognized at input "Execute", the current diagnostic data of the addressed slave is determined and written on the corresponding outputs of the function block.

☞ To request a new diagnostics, the input "Execute" must initially be set to FALSE and the function block must be called one time in the internal State = WAIT_RESET_MSG_DN (edge detection).

Error Handling

An error is signalized during the processing of the function block by setting the "Error" output. Information regarding the error can be obtained via "ErrorID" and "ErrorIdent", see chapter 6.6.10 "Error Messages of DeviceNet Function Blocks" on page 114.

6.6.7 IL_ExplicitMsgDN

Brief Description

Function block for acyclic communication via Explicit Messaging.

Library	Area
RIL_DeviceNet.lib	Explicit Messaging

Fig.6-53: Library assignment: IL_ExplicitMsgDN

Interface Description

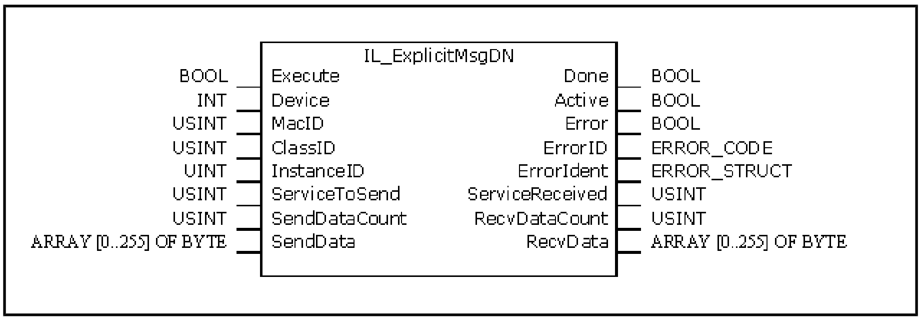


Fig.6-54: Structure: IL_ExplicitMsgDN

	Name	Type	Comment
VAR_INPUT	Execute	BOOL	Processing enable of the function block.
	Device	INT	Selected DeviceNet master (for parallel operation of several field buses). The value always has to be set to 0.
	MacID	USINT	MAC ID of the selected slave.
	ClassID	USINT	Class ID of the selected object.
	InstanceID	UINT	Instance ID of the selected object.
	ServiceToSend	USINT	Explicit Messaging Service Code (see DeviceNet Spec Vol1, Appendix G)
	SendDataCount	USINT	Number of the data to be send from SendData
	SendData	ARRAY [0.0.255] OF BYTE	Byte array for data to be send.
VAR_OUTPUT	Done	BOOL	Communikation ended.
	Active	BOOL	Explicit Messaging active.
	Error	BOOL	TRUE: An error has occurred.
	ErrorID	ERROR_CODE	Error short description.
	ErrorIdent	ERROR_STRUCT	Detailled error description according to error table.
	ServiceReceived	USINT	Explicit Messaging Service Code (see DeviceNet Spec Vol1, Appendix G)
	RecvDataCount	USINT	Number of received response data.
	RecvData	ARRAY [0.0.255] OF BYTE	Byte array for received response data.

Fig.6-55: Interface signals: IL_ExplicitMsgDN

Functional description

This function block "IL_ExplicitMsgDN" enables to send any Explicit Message to a slave via the message interface of the DeviceNet master. The addressing of the desired object results from the "MacID", "ClassID" and "InstanceID". The desired Explicit Messaging Code is determined via "ServiceToSend". Further requested data are transferred in array "SendData". The byte length of this data has to be specified in "SendData". As soon as a positive edge is recognized at input "Execute", the data to be send is verified as far as possible and written in the message interface of the DeviceNet master. It transfers the request to the addressed slave. As soon as output "Done" is set to TRUE, the response data is available. "ServiceReceived" specifies the service code of the response data. Via "RecvData" and "RecvDataCount", the response data and their byte length are transferred.

☞ To send a new Explicit Message with the function block, input "Execute" must initially be set to FALSE and the function block must be called one time in the State = WAIT_RESET_MSG_DN (edge detection).

Error Handling An error is signalized during the processing of the function block by setting the "Error" output. Information regarding the error can be obtained via "ErrorID" and "ErrorIdent", see chapter 6.6.10 "Error Messages of DeviceNet Function Blocks" on page 114.

6.6.8 IL_GetAttribSingleDN

Brief Description Implementing the DeviceNet service "Get_Attribute_Single".

Library	Area
RIL_DeviceNet.lib	Explicit Messaging

Fig.6-56: Library assignment: IL_GetAttribSingleDN

Interface Description

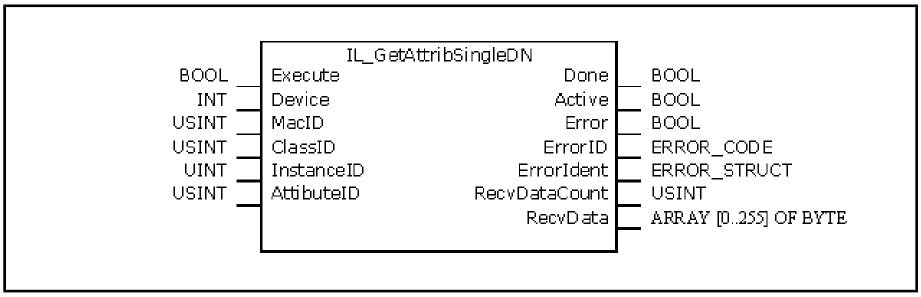


Fig.6-57: Structure: IL_GetAttribSingleDN

	Name	Type	Comment
VAR_INPUT	Execute	BOOL	Processing enable of the function block.
	Device	INT	Selected DeviceNet master (for parallel operation of several field buses). The value always has to be set to 0.
	MacID	USINT	MAC ID of the selected slave.
	ClassID	USINT	Class ID of the selected object.
	InstanceID	UINT	Instance ID of the selected object.
	AttributeID	USINT	Attribute ID of the selected object.
VAR_OUTPUT	Done	BOOL	Attribute data was received.
	Active	BOOL	Attribute data is received.
	Error	BOOL	TRUE: An error has occurred.
	ErrorID	ERROR_CODE	Error short description.
	ErrorIdent	ERROR_STRUCT	Detailled error description according to error table.
	RecvDataCount	USINT	Byte length of the received attribute data.
	RecvData	ARRAY [0.0.255] OF BYTE	Byte array for received attribute data.

Fig.6-58: Interface signals: IL_GetAttribSingleDN

Functional description

This function block enables the reading of the addressed attribute "IL_GetAttribSingleDN". The addressing of the desired object results from the "MacID", "ClassID", "InstanceID" and "AttributeID". As soon as a positive edge is recognized at input "Execute", the requested data is verified and written in the message interface of the DeviceNet master. It transfers the request to the addressed slave. As soon as output "Done" is set to TRUE, the response data of the addressed attribute is available. Via "RecvData" and "RecvDataCount", the attribute data and their byte length are provided.

> ☞ To request new data, input "Execute" must initially be set to FALSE and the function block must be called once in the internal State = WAIT_RESET_MSG_DN (edge detection).

Error Handling

An error is signalized during the processing of the function block by setting the "Error" output. Information regarding the error can be obtained via "ErrorID" and "ErrorIdent", see chapter 6.6.10 "Error Messages of DeviceNet Function Blocks" on page 114.

6.6.9 IL_GetAttribSingleDN

Brief Description

Implementing the DeviceNet service "Set_Attribute_Single".

Library	Area
RIL_DeviceNet.lib	Explicit Messaging

Fig.6-59: Library assignment: IL_SetAttribSingleDN

Interface Description

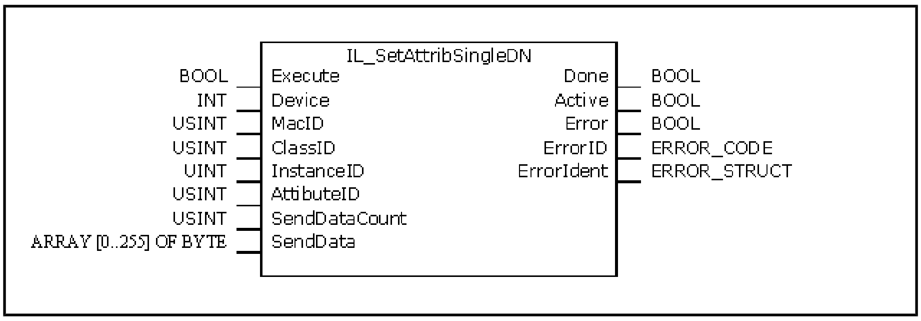


Fig.6-60: Structure: IL_SetAttribSingleDN

	Name	Type	Comment
VAR_INPUT	Execute	BOOL	Processing enable of the function block.
	Device	INT	Selected DeviceNet master (for parallel operation of several field buses). The value always has to be set to 0.
	MacID	USINT	MAC ID of the selected slave.
	ClassID	USINT	Class ID of the selected object.
	InstanceID	UINT	Instance ID of the selected object.
	AttributeID	USINT	Attribute ID of the selected object.
	SendDataCount	USINT	Byte length of the attribute data to be send in SendData
	SendData	ARRAY [0.0.255] OF BYTE	Byte array for attribute data to be send.
VAR_OUTPUT	Done	BOOL	Attribute data have been transferred.
	Active	BOOL	Attribute data are transferred.
	Error	BOOL	TRUE: An error has occurred.

	Name	Type	Comment
	ErrorID	ERROR_CODE	Error short description.
	ErrorIdent	ERROR_STRUCT	Detailled error description according to error table.

Fig.6-61: Interface signals: IL_SetAttribSingleDN

Functional description

This function block enables the changing of the addressed attribute "IL_SetAttribSingleDN". The addressing of the desired object results from the "MacID", "ClassID", "InstanceID" and "AttributeID". Via "SendData" and "SendDataCount", the attribute data and their byte length are transferred. As soon as a positive edge is recognized at input "Execute", the requested data is verified and written in the message interface of the DeviceNet master. It transfers the request to the addressed slave. As soon as output "Done" is set to TRUE, the response data of the addressed attribute is transferred successfully.

☞ To write new attribute data, input "Execute" must initially be set to FALSE and the function block must be called once in the internal State = WAIT_RESET_MSG_DN (edge detection).

Error Handling

An error is signalized during the processing of the function block by setting the "Error" output. Information regarding the error can be obtained via "ErrorID" and "ErrorIdent", see chapter 6.6.10 "Error Messages of DeviceNet Function Blocks" on page 114.

6.6.10 Error Messages of DeviceNet Function Blocks

The DeviceNet function blocks supply a detailed diagnostics in the error case. The diagnostics comprises the "Error", "ErrorID" and "ErrorIdent" outputs.

Additional 1	Additional 2	Description
16#00000000	16#00000000	No error
16#00000001	16#00000000	Timeout during the message interface is requested
16#00000002	16#00000000	Resource not available
16#00000008	16#00000000	DeviceNet service not available
16#00000009	16#00000000	Attribute invalid
16#0000000B	16#00000000	Slave already executes the request
16#0000000C	16#00000000	Object status conflict
16#0000000E	16#00000000	Write access to attribute not allowed
16#0000000F	16#00000000	No access rights
16#00000010	16#00000000	Command non-executable
16#00000011	16#00000000	Timeout slave response telegram

Additional 1	Additional 2	Description
16#00000013	16#00000000	Transmitted data incomplete
16#00000014	16#00000000	Attribute not available on slave
16#00000015	16#00000000	Too many data received
16#00000016	16#00000000	Addressed object not available on slave
16#00000017	16#00000000	Internal error
16#0000001F	16#00000000	Vendor-specific ErrorCode
16#00000032	16#00000000	Faulty response telegram
16#00000036	16#00000000	Reset command active
16#00000037	16#00000000	MAC Id invalid
16#00000039	16#00000000	Fragmentation error during slave response occurred
16#000000C8	16#00000000	DeviceNet master not configured

Fig.6-62: Error table DEVICENET_TABLE

6.7 RIL_EtherNetIP

6.7.1 Overview

A control configured as CIP data client is able to read and – if write access is allowed – to describe variables of a CIP data server.

IL_ReadDataTable	FB	Read variables provided by the CIP data server
IL_WriteDataTable	FB	Write on variables provided by the CIP data server
IL_Status	FB	For diagnostics of the cyclic communication

Fig.6-63: Function blocks contained in the RIL_EtherNetIP.lib

6.7.2 IL_ReadDataTable

Brief Description Reads variables provided by a CIP data server.

Interface Description

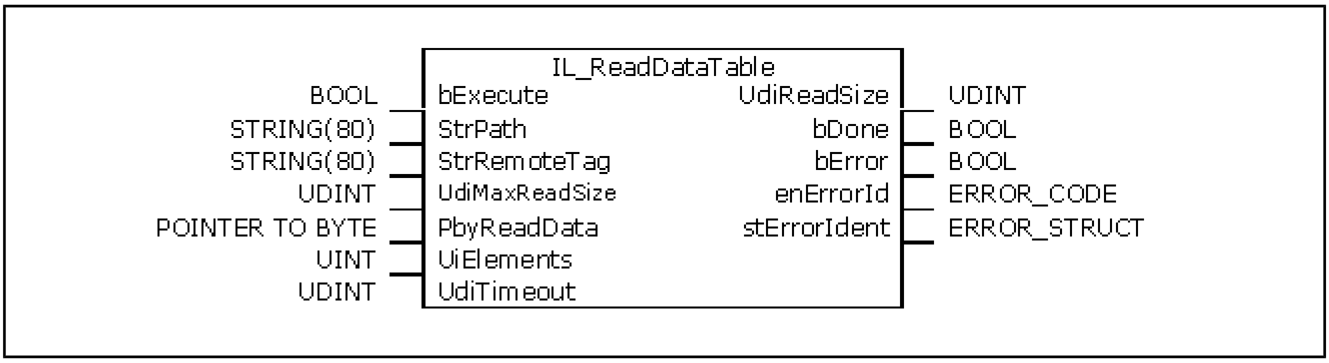


Fig.6-64: Structure of IL_ReadDataTable

	Name	Type	Comment
VAR_INPUT	bExecute	BOOL	Positive edge starts the service.
	StrPath	STRING(80)	IP address of the CIP data server (e.g. "192.168.73.105")
	StrRemoteTag	STRING(80)	Name of the variable to be read (e.g. "Test1").
	UdiMaxReadSize	UDINT	Size of the receiver array "PbyReadData".
	PbyReadData	POINTER TO BYTE	Pointer to the array, in which the data are to be written. The array has to be provided by the user of the FB.
	UiElements	UINT	Number of array elements to be read. For atomic data types (SINT, DINT...) UiElements has to be set to value 1.
	UdiTimeout	UDINT	Specifies the maximum time in ms that the CIP data server may need to respond.
VAR_OUTPUT	UdiReadSize	UDINT	Number of received data bytes.
	bDone	BOOL	TRUE: The service is stopped or an error occurred.
	bError	BOOL	TRUE: An error occurred.
	enErrorID	ERROR_CODE	See error messages.
	stErrorIdent	ERROR_STRUCT	

Fig.6-65: Interface signals of IL_ReadDataTable

Example Read the DINT variable "diCounter" from the CIP data server with the IP address "192.168.73.105". As "diCounter" is used in the PLC program "PLC_PRG", the user has to enter "PLC_PRG.diCounter" as name of the variable to be read.

The reading process can be started by a positive edge change of variable "Read2Fb_bExecute" (FALSE -> TRUE). Status TRUE of variable "bDone" confirms the execution of the service.

Program:

```
PROGRAM PLC_PRG
VAR
(**************************************************
** Variables for IL_ReadDataTable
**************************************************)
 Read2In1:                  IL_ReadDataTable;
 Read2Fb_bExecute:          BOOL;
 Read2Fb_strPath:           STRING;
 Read2Fb_strRemoteTag:      STRING;
 Read2Fb_udiMaxReadSize:    UDINT;
 Read2Fb_diReadData:       DINT;
 Read2Fb_uiElements:       UINT;
 Read2Fb_udiTimeout:       UDINT;
 Read2Fb_udiReadSize:      UDINT;
 Read2Fb_bDone:            BOOL := FALSE;
 Read2Fb_bError:           BOOL;
 Read2Fb_enErrorID:        ERROR_CODE;        (*Type: RIL_CommonTypes.lib*)
 Read2Fb_stErrorIdent:     ERROR_STRUCT;      (*Type: RIL_CommonTypes.lib*)
(*Diagnose*)
 Read2Fb_udiCtrErr:        UDINT := 0;
 Read2Fb_udiCtrGood:       UDINT := 0;
 Read2Fb_CtrValidInput:    UDINT := 0;
(*for automation test*)
 Read2Fb_bAutoTest:        BOOL := FALSE;
```

Program:

```
(******************************************************************
** Explicit message: Read data table - 2 DINT
*******************************************************************)
 Read2Fb_strPath      := '192.168.73.105';      (*IP-Adresse des Data-Table-Servers*)
 Read2Fb_strRemoteTag := 'PLC_PRG.diCounter';   (*Name of the variables to be read*)
 Read2Fb_uiElements   := 1;                     (*Number of the elements to be read*)
 Read2Fb_udiTimeout   := 100;                   (*Timeout in ms*)
 Read2In1(
  bExecute       := Read2Fb_bExecute,
  strPath        := Read2Fb_strPath,
  strRemoteTag   := Read2Fb_strRemoteTag,
  udiMaxReadSize := SIZEOF(Read2Fb_diReadData),
  pbyReadData    := ADR(Read2Fb_diReadData),
  uiElements     := Read2Fb_uiElements,
  udiTimeout     := Read2Fb_udiTimeout,
  udiReadSize    => Read2Fb_udiReadSize,
  bDone          => Read2Fb_bDone,
  bError         => Read2Fb_bError,
  enErrorID      => Read2Fb_enErrorID,
  stErrorIdent   => Read2Fb_stErrorIdent
 );
(*Automation test*)
IF Read2Fb_bAutoTest = TRUE THEN
    IF Read2Fb_bDone = TRUE THEN
        IF Read2Fb_bError = TRUE THEN
           Read2Fb_udiCtrErr  := Read2Fb_udiCtrErr + 1;
        ELSE
           Read2Fb_udiCtrGood := Read2Fb_udiCtrGood + 1;
        END_IF Read2Fb_bExecute := FALSE;
    ELSE
       Read2Fb_bExecute := TRUE;
    END_IF
END_IF
```

Error Messages

Error ID	Error table	Additional1	Additional2	Error name	Description
COMMUNICATION_ERROR (2)	ETHERNET_IP(151)	0x10000001	0x00000000	NoCyclicCom	Master has not yet established the cyclic communication with the IndraLogic EtherNet/IP slave.
COMMUNICATION_ERROR (2)	ETHERNET_IP(151)	0x10000002	0x00000000	Idle	Ethernet/IP master has started cyclic communication, but set the idle flag in the cyclic channel. Thus, the master indicates that its output image is not valid.

Error ID	Error table	Additional1	Additional2	Error name	Description
COMMUNICATION_ERROR (2)	ETHERNET_IP(151)	0x10000003	0x00000000	Timeout	Time monitoring of the cyclic communication indicates errors.
COMMUNICATION_ERROR (2)	ETHERNET_IP(151)	0x10000004	0x00000000	Closed	Cyclic communication was actively finished by the Ethernet/IP master.

Fig.6-66: Error Messages IL_ReadDataTable and IL_WriteDataTable

6.7.3 IL_WriteDataTable

Brief Description

Writes values in variables provided by a CIP data server.

☞ Write access at the CIP data server has to be allowed for the relevant variables.

Interface Description

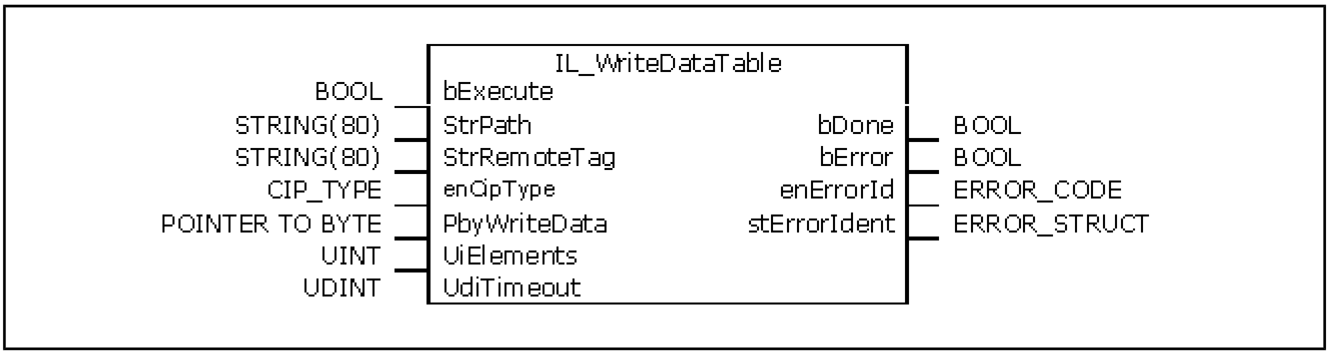


Fig.6-67: Structure of IL_WriteDataTable

	Name	Type	Comment
VAR_INPUT	bExecute	BOOL	Positive edge starts the service.
	StrPath	STRING(80)	IP address of the CIP data server (e.g. "192.168.73.105")
	StrRemoteTag	STRING(80)	Name of the variable to be written (e.g. "Test1").
	EnCipType	CIP_TYPE	Type of the written data. Is verified by some CIP data server.
	PbyWriteData	POINTER TO BYTE	Pointer to the array that contains the data to be written on the CIP data client. The array has to be provided by the user of the FB.
	UiElements	UINT	Number of array elements to be written. For atomic data types (SINT, DINT...) UiElements has to be set to value 1.
	UdiTimeout	UDINT	Specifies the maximum time in ms that the CIP data server may need to respond.
VAR_OUTPUT	bDone	BOOL	TRUE: The service is stopped or an error occurred.
	bError	BOOL	TRUE: An error occurred.
	EnErrorID	ERROR_CODE	See error messages.
	StErrorIdent	ERROR_STRUCT	

Fig.6-68: Interface signals of IL_WriteDataTable

Example Write the SINT variable "siWriteData" on a CIP data server with the IP address "192.168.73.105". As "siWriteData" is used in the PLC program "PLC_PRG", "PLC_PRG.siWriteData" is to be entered as name of the variable to be written.

The writing process can be started by a positive edge change of variable "WriteFb_bExecute" (FALSE -> TRUE). Status TRUE of variable "bDone" confirms the execution of the service.

Program:

```
(*************************************************
** Variables for writing data SINT
**************************************************)
 WriteIn1:                 IL_WriteDataTable;
 WriteFb_bExecute:         BOOL;
 WriteFb_strPath:          STRING;
 WriteFb_strRemoteTag:     STRING;
 WriteFb_enCipType:        CIP_TYPE;
 WriteFb_uiElements:       UINT;
 WriteFb_siWriteData:      SINT;
 WriteFb_udiTimeout:       UDINT;
 WriteFb_bDone:            BOOL := FALSE;
 WriteFb_bError:           BOOL;
 WriteFb_enErrorID:        ERROR_CODE;
 WriteFb_stErrorIdent:     ERROR_STRUCT;
(*Diagnostics*)
 WriteFb_udiCtrErr:        UDINT := 0;
 WriteFb_udiCtrGood:       UDINT := 0;
 WriteFb_udiCtrValidInput: UDINT := 0;
(*for automation test*)
 WriteFb_bAutoTest:        BOOL := FALSE;
```

Program:

```
(******************************************************************
** Explicit message: Write data table - 1 SINT
*******************************************************************)
 WriteFb_strPath        := '192.168.73.105';
 WriteFb_strRemoteTag   := 'PLC_PRG.siWriteData';
 WriteFb_uiElements     := 1;
 WriteFb_enCipType      := CIPTYPE_SINT;
 WriteFb_udiTimeout     := 1000;  (*ms*)
 WriteIn1(
  bExecute := WriteFb_bExecute,         (*FALSE after TRUE starts FB*)
  strPath := WriteFb_strPath,           (*IP address of the server*)
  strRemoteTag := WriteFb_strRemoteTag, (*Name of the variable to be written*)
  enCipType := WriteFb_enCipType,       (*Data type of the variables*)
  uiElements := WriteFb_uiElements,     (*Number of the elements to be written.*)
                                        (* For atomic data types *)
                                        (* (SINT, DINT...)always 1.*)
  pbyWriteData    := ADR(WriteFb_siWriteData),  (*pointer to the data, you want to write*)
  udiTimeout := WriteFb_udiTimeout,          (*Timeout der Expicit Message*)
  bDone => WriteFb_bDone,               (*TRUE => FB is ready with order.*)
  bError => WriteFb_bError,             (*TRUE => Error*)
  enErrorID => WriteFb_enErrorID,
  stErrorIdent => WriteFb_stErrorIdent
 );
(*automatic test*)
IF WriteFb_bAutoTest = TRUE THEN
    IF WriteFb_bDone = TRUE THEN
        IF WriteFb_bError = TRUE THEN
            WriteFb_udiCtrErr    := WriteFb_udiCtrErr + 1;
        ELSE
            WriteFb_udiCtrGood   := WriteFb_udiCtrGood + 1;
            WriteFb_siWriteData  := WriteFb_siWriteData + 1;
        END_IF
        WriteFb_bExecute := FALSE;
    ELSE
        WriteFb_bExecute := TRUE;
    END_IF
END_IF
```

Error Messages

Error ID	Error table	Additional1	Additional2	Error name	Description
COMMUNICATION_ERROR (2)	ETHERNET_IP(151)	0x10000001	0x00000000	NoCyclicCom	Master has not yet established the cyclic communication with the IndraLogic EtherNet/IP slave.
COMMUNICATION_ERROR (2)	ETHERNET_IP(151)	0x10000002	0x00000000	Idle	Ethernet/IP master has started cyclic communication, but set the idle flag in the cyclic channel. Thus, the master indicates that its output image is not valid.
COMMUNICATION_ERROR (2)	ETHERNET_IP(151)	0x10000003	0x00000000	Timeout	Time monitoring of the cyclic communication indicates errors.
COMMUNICATION_ERROR (2)	ETHERNET_IP(151)	0x10000004	0x00000000	Closed	Cyclic communication was actively finished by the Ethernet/IP master.

Fig.6-69: Error Messages IL_ReadDataTable and IL_WriteDataTable

6.7.4 IL_Status

Brief Description The FB IL_Status is used for the diagnostics of the cyclic communication ("Implicit Messaging").

Interface Description

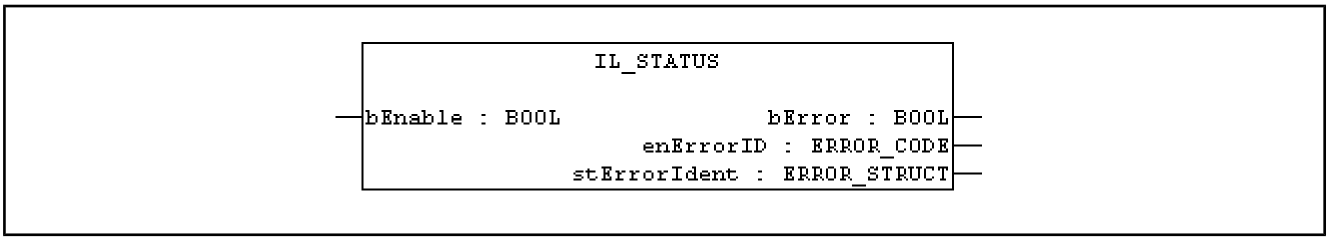


Fig.6-70: IL_Status

	Name	Type	Comment
VAR_INPUT	bEnable	BOOL	If the status is positive (TRUE), the status of the cyclic Ethernet/IP communication is requested with each call of the FB.
VAR_OUTPUT	bError	BOOL	TRUE: An error occurred.
	EnErrorID	ERROR_CODE	See error messages.
	StErrorIdent	ERROR_STRUCT	

Fig.6-71: Interface of IL_WriteDataTable

Example The following example shows how to use FB IL_Status.

Program:

```
(***************************************************
** Variables for IL_Status
***************************************************)
PROGRAM PLC_PRG
VAR
 Diag:         IL_Status;
 CtrValidInput: DINT
 ...
END_VAR;  ... ... ...
(************************************************
** Program
*************************************************)
Diag(bEnable:=TRUE);
   IF(Diag.bError = TRUE) THEN
      (*Insert error treatment here.*)
      ...
      Diag(bEnable:=FALSE );
      (*Reset error (bEnable)*)
   ELSE
      (*valid data: Insert input and output data processing here *)
      CtrValidInput := CtrValidInput +1;
   END_IF
   ...
```

Error Messages

Error ID	Error table	Additional1	Additional2	Error name	Description
COMMUNICATION_ERROR (2)	ETHERNET_IP(151)	0x10000001	0x00000000	NoCyclicCom	Master has not yet established the cyclic communication with the IndraLogic EtherNet/IP slave.
COMMUNICATION_ERROR (2)	ETHERNET_IP(151)	0x10000002	0x00000000	Idle	Ethernet/IP master has started cyclic communication, but set the idle flag in the cyclic channel. Thus, the master indicates that its output image is not valid.
COMMUNICATION_ERROR (2)	ETHERNET_IP(151)	0x10000003	0x00000000	Timeout	Time monitoring of the cyclic communication indicates errors.
COMMUNICATION_ERROR (2)	ETHERNET_IP(151)	0x10000004	0x00000000	Closed	Cyclic communication was actively finished by the Ethernet/IP master.

Fig.6-72: Error Messages IL_ReadDataTable and IL_WriteDataTable

6.8 RIL_L40_Util

The functions "IL_GetStateFan", "IL_GetTemp" and "IL_TempWarning" provided in the previous library RIL_L40_Util were replaced by the functions "IH_FanStatus", "IH_Temperature" and "IH_TempWarning" in the library RIH_CML40 (see chapter 6.4 "RIH_CML40 " on page 97).

6.9 RIL_Inline

6.9.1 FB Overview

Structure

The library comprises the following folders:

_Version

The RIL_Inline.library is used to set the version number to 01V01.

RIL Inline-Services

Overview of Functions

Designation	Type	Description
Version_RIL_Inline_01V01	BOOL	Version ID
IL_InlineDiagModulImage		Diagnostics module

Designation	Type	Description
IL_InlineCfgDiagData		Determine system configuration
IL_InlineDiagCfgChk		System configuration diagnostics
IL_InlineDiagModulPosi		Diagnostics of a defined module
IL_InlineDiagModulNext		Diagnostics of the next module
IL_InlineDiagMasterBus		Bus error diagnostics
IL_InlineDiagMasterFatal		Stack error diagnostics
IL_InlineDiagGlobalStatus		Global system diagnostics

Fig.6-73: Overview of the function blocks and functions contained in the library

6.9.2 Data Types

Configuration ID Description

Program:

```
TYPE  CfgIdDesc :
 STRUCT
  ModuleIdCFG :WORD;
  ModuleIdSCN :WORD;
 END_STRUCT
END_TYPE
```

Offset	Type	Designation	Description
0	WORD	ModuleIdCFG	Module ID of configured modules
1	WORD	ModuleIdSCN	Module ID of scanned modules

Fig.6-74: Configuration ID

Module ID	
High byte	**Low byte**
ID code	Length code

Fig.6-75: Module ID

Rexroth Inline modules	Part number	ID code Hexadecimal	Length code Hexadecimal	ID code Decimal
Digital input				
R-IB IL 24 DI 16	R911289290	BE	01	190
R-IB IL 24 DI 16-PAC	R911170752	BE	01	190
R-IB IL 24 DI 16-2MBD-PAC	R911170408	BE	01	190
R-IB IL 24 DI 32/HD	R911297188	BE	02	190
R-IB IL 24 DI 32/HD-PAC	R911170753	BE	02	190
R-IB IL 24 DI 32/HD-NPN-PAC	R911170405	BE	02	190
R-IB IL 24 DI 4	R911289287	BE	41	190
R-IB IL 24 DI 4-PAC	R911170750	BE	41	190
R-IB IL 24 EDI 2-DES	R911289292	BE	41	190
R-IB IL 24 DI 16-NPN-PAC	R911170404	BE	41	190

Rexroth Inline modules	Part number	ID code Hexadecimal	Length code Hexadecimal	ID code Decimal
R-IB IL 24 DI 8	R911289288	BE	81	190
R-IB IL 24 DI 8-PAC	R911170751	BE	81	190
R-IB IL 24 DI 8-2MBD-PAC	R911170407	BE	81	190
R-IB IL 24 DI 2	R911289286	BE	C2	190
R-IB IL 24 DI 2-PAC	R911170767	BE	C2	190
R-IB IL 24 DI 2-NPN-PAC	R911170403	BE	C2	190
Feeder and segment terminals				
IB IL 24 SEG/F-D-PAC	R911170710	BE	C2	190
R-IB IL 24 PWR IN/2F-D-2MBD -PAC	R911170447	BE	C2	190
R-IB IL 24 SEG/F-D-2MBD -PAC	R911170448	BE	C2	190
Digital output				
R-IB IL 24 DO 16	R911289299	BD	01	189
R-IB IL 24 DO 16-PAC	R911170757	BD	01	189
R-IB IL 24 DO 16-2MBD-PAC	R911170415	BD	01	189
R-IB IL 24 DO 32/HD	R911297191	BD	02	189
R-IB IL 24 DO 32/HD-PAC	R911170768	BD	02	189
R-IB IL 24 DO 32/HD-NPN-PAC	R911170411	BD	02	189
R-IB IL 24 DO 4	R911289295	BD	41	189
R-IB IL 24 DO 4-PAC	R911170755	BD	41	189
R-IB IL 24 DO 4-2MBD-PAC	R911170413	BD	41	189
R-IB IL 24 DO 8	R911289297	BD	81	189
R-IB IL 24 DO 8-PAC	R911170756	BD	81	189
R-IB IL 24 DO 8-2A	R911289298	BD	81	189
R-IB IL 24 DO 8-2A-PAC	R911170759	BD	81	189
R-IB IL 24 DO 8-NPN-PAC	R911170410	BD	81	189
R-IB IL 24 DO 8-2MBD-PAC	R911170414	BD	81	189
R-IB IL 24 DO 2-2A	R911289294	BD	C2	189
R-IB IL 24 DO 2-2A-PAC	R911170754	BD	C2	189
R-IB IL 24 DO 2-NPN-PAC	R911170409	BD	C2	189
R-IB IL 24 DO 2-2A-2MBD-PAC	R911170412	BD	C2	189
Relay terminals				
R-IB IL 24/230 DOR 4/W	R911289302	BD	41	189
R-IB IL 24/230 DOR 4/W-PAC	R911170758	BD	41	189
R-IB IL 24/230 DOR 4/W-2MBD-PAC	R911170417	BD	41	189
R-IB IL 24/230 DOR 1/W	R911289301	BD	C2	189
R-IB IL 24/230 DOR 1/W-PAC	R911170769	BD	C2	189

Libraries

Rexroth Inline modules	Part number	ID code Hexadecimal	Length code Hexadecimal	ID code Decimal
Analog input				
R-IB IL AI 8/SF-PAC	R911308493	5F	02	95
R-IB IL AI 8/IS-PAC	R911308494	5F	02	95
R-IB IL AI 8/SF-2MBD-PAC	R911170430	5F	02	95
R-IB IL AI 2/SF	R911289306	7F	02	127
R-IB IL AI 2/SF-PAC	R911170784	7F	02	127
R-IB IL AI 2/SF-230-PAC	R911170425	7F	02	127
R-IB IL TEMP 2 RTD	R911289305	7F	02	127
R-IB IL TEMP 2 RTD-PAC	R911170785	7F	02	127
R-IB IL TEMP 2 UTH-PAC	R911170431	7F	02	127
R-IB IL SGI 2/F-PAC	R911170432	DF	03	223
R-IB IL SGI 2/F-2MBD-PAC	R911170433	DF	03	223
R-IB IL AI 4/EF-PAC	R911170426	DF	05	223
R-IB IL AI 4/EF-2MBD-PAC	R911170427	DF	05	223
R-IB IL TEMP 4/8 RTD-PAC	R911170428	DF	05	223
R-IB IL TEMP 4/8 RTD-2MBD-PAC	R911170429	DF	05	223
R-IB IL SGI 2/P-PAC	R911170434			
R-IB IL SGI 2/P-2MBD-PAC	R911170435			
Analog output				
R-IB IL AO 2/U/BP	R911289381	5B	02	91
R-IB IL AO 2/U/BP-PAC	R911170786	5B	02	91
R-IB IL AO 2/SF-PAC	R911170436	5B	02	91
R-IB IL AO 2/SF-2MBD-PAC	R911170437	5B	02	91
R-IB IL AO 1/SF	R911289303	7D	01	125
R-IB IL AO 1/SF-PAC	R911170787	7D	01	125
R-IB IL AO 4/8/U/BP-2MBD-PAC	R911170438	DF	05	223
Function terminals				
R-IB IL INC-IN-PAC	R911308491	7F	02	127
R-IB IL CNT	R911289315	BF	02	191
R-IB IL CNT-PAC	R911170788	BF	02	191
R-IB IL CNT-2MBD-PAC	R911170439	BF	02	191
R-IB IL INC-PAC	R911308492	BF	02	191
R-IB IL SSI-PAC	R911308594	BF	02	191
R-IB IL PWM/2-PAC	R911170444	BF	02	191
R-IB IL TEMPCON UTH-PAC	R911308596	BF	02	191
R-IL BK DDL	BRP-Podukt	BF	04	191

Rexroth Inline modules	Part number	ID code Hexadecimal	Length code Hexadecimal	ID code Decimal
R-IB IL RS232-PRO-PAC	R911170440	BF	06	191
R-IB IL RS232-PRO-2MBD-PAC	R911170441	BF	06	191
R-IB IL RS485/422-PRO-PAC	R911170442	BF	06	191
R-IB IL RS485/422-PRO-2MBD-PAC	R911170443	BF	06	191
Fieldline Modular M8				
RF-FLM DI 8 M8	R911170449	B2	81	178
RF-FLM DIO 8/4 M8	R911170450	B3	81	179

Error Struct

Program:

```
TYPE  ERROR_STRUCT :
 STRUCT
  Table       :WORD;
  Additional1 :DWORD;
  Additional2 :DWORD;
 END_STRUCT
END_TYPE
```

Offset	Type	Designation	Description
0	WORD	Table	Identifies the "error table" from which the error numbers are entered in ErrorAdditional.
1	DWORD	Additional1	Assignment varying according to ErrorTable, e.g. RIL_INLINE error...
2	DWORD	Additional2	Potential additional error information, depending on ErrorTable

Fig.6-76: Error Struct

ErrorCode		ErrorIdent			Description
Symbol	Code	ErrorTable	Additional1	Additional2	
------- User error -------					
RIL_INPUT_INVALID_ERROR	1	400 dec / 0190 hex	1311	[module position]	Invalid module position
RIL_INPUT_INVALID_ERROR	1	400 dec / 0190 hex	2000	0	Parameter error FB
------- Internal errors -------					
RIL_NONE_ERROR	0	400 dec / 0190 hex	0	0	No error
RIL_ACCESS_ERROR	4	400 dec / 0190 hex	1310	0	Internal access error (Diag module not ready)
RIL_ACCESS_ERROR	4	400 dec / 0190 hex	1410	0	Internal access error (Config module not ready)
RIL_ACCESS_ERROR	4	400 dec / 0190 hex	1411	0	Internal access error (invalid module list)

ErrorCode		ErrorIdent			Description
Symbol	Code	ErrorTable	Additional1	Additional2	
RIL_ACCESS_ERROR	4	400 dec / 0190 hex	1412	0	Internal access error (invalid module scan)
RIL_ACCESS_ERROR	4	400 dec / 0190 hex	1413	0	Internal access error (invalid module position)
RIL_ACCESS_ERROR	4	400 dec / 0190 hex	1414	0	Internal access error (no module entries available)
RIL_ACCESS_ERROR	4	400 dec / 0190 hex	1420	[call position]	Internal access error (driver not ready)
RIL_ACCESS_ERROR	4	400 dec / 0190 hex	1421	[call position]	Internal access error (execution error)
RIL_DEVICE_ERROR	8	400 dec / 0190 hex	1330	0	Internal error (invalid Diag type)
RIL_DEVICE_ERROR	8	400 dec / 0190 hex	1331	0	Internal error (invalid list index)
RIL_DEVICE_ERROR	8	400 dec / 0190 hex	1332	0	Internal error (ring buffer overflow)
RIL_DEVICE_ERROR	8	400 dec / 0190 hex	1333	0	Internal error (module state conflict)
RIL_DEVICE_ERROR	8	400 dec / 0190 hex	1334	0	Internal error (invalid buffer index)
RIL_OTHER_ERROR	7FFE hex	400 dec / 0190 hex	1188	0	Undefined error
RIL_OTHER_ERROR	7FFE hex	400 dec / 0190 hex	1199	0	Undefined error
RIL_SYSTEM_ERROR	7FFF hex	400 dec / 0190 hex	1177	0	Internal system error
RIL_SYSTEM_ERROR	7FFF hex	400 dec / 0190 hex	2000	0	Internal system error FB

6.9.3 Version

Version_RIL_Inline_01V01

Brief Description

To ensure that the firmware version is compatible with the library, a version ID is carried along. If the names of the version functions fail to be equal, downloading of the application program will be rejected.

Library	Area
RIL_Inline.library	_Version

Fig.6-77: Version_RIL_Inline_01V01 library assignment

Interface Description

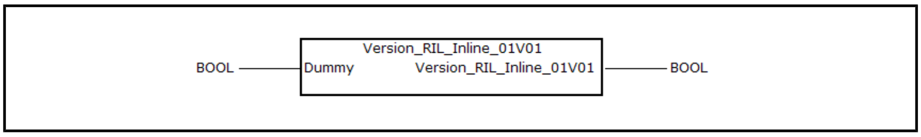


Fig.6-78: Version_RIL_Inline_01V01 structure

	Name	Type	Description
VAR_INPUT	Dummy	BOOL	–
VAR_OUTPUT	Version_RIL_Inline_01V01	BOOL	Version ID

Fig.6-79: Version_RIL_Inline_01V01 interface

6.9.4 Diagnostics

IL_InlineDiagGlobalStatus

Brief Description

The "IL_InlineDiagGlobalStatus" function block is used to determine whether there is a diagnose in the system. The returned counters each indicate the active and total numbers of diagnostics.

Library	Area
RIL_Inline.library	RIL Inline-Services

Fig.6-80: IL_InlineDiagGlobalStatus library assignment

Interface Description

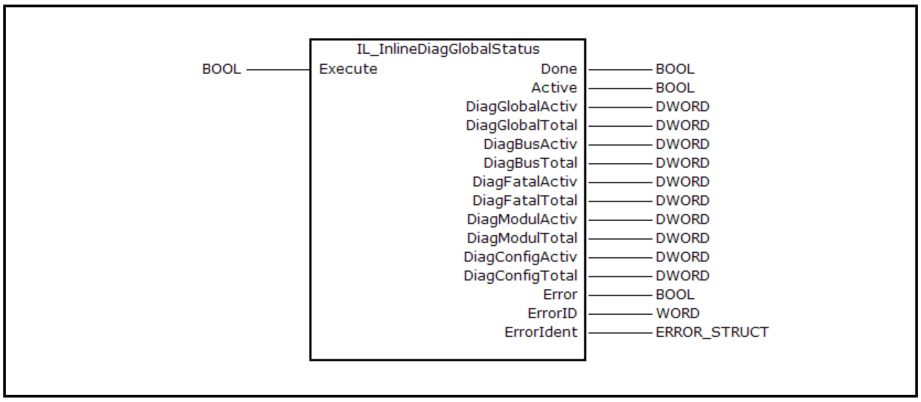


Fig.6-81: IL_InlineDiagGlobalStatus

	Name	Type	Description
VAR_INPUT	Execute	BOOL	Activates the service
VAR_OUTPUT	Done	BOOL	Done message (successful)
	Active	BOOL	Activity display
	DiagGlobalActiv	DWORD	Total number of active diagnostics
	DiagGlobalTotal	DWORD	Total number of diagnostics since the PLC program was loaded
	DiagBusActiv	DWORD	Active bus diagnostics
	DiagBusTotal	DWORD	Total number of bus diagnostics since the PLC program was loaded
	DiagFatalActiv	DWORD	Active fatal diagnostics
	DiagFatalTotal	DWORD	Number of active fatal diagnostics since the PLC program was loaded

	Name	Type	Description
	DiagModulActiv	DWORD	Active module diagnostics
	DiagModulTotal	DWORD	Total number of module diagnostics since the PLC program was loaded
	DiagConfigActiv	DWORD	Active configuration diagnostics
	DiagConfigTotal	DWORD	Total number of configuration diagnostics since the PLC program was loaded
	Error	BOOL	Done message (unsuccessful)
	ErrorID	ERROR_CODE	See chapter "ErrorID" on page 171
	ErrorIdent	ERROR_STRUCT	Error structure from PB_DP_TABLE ErrorTable

Fig.6-82: IL_InlineDiagGlobalStatus interface

Signal Time Diagram

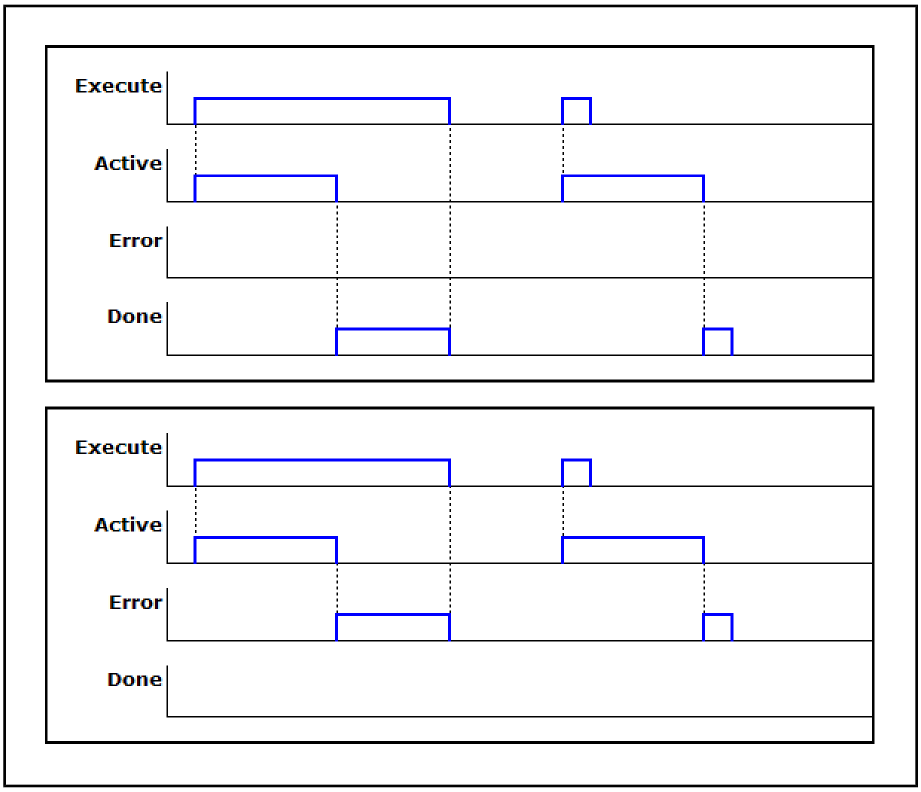


Fig.6-83: IL_InlineDiagGlobalStatus signal time diagram

IL_InlineDiagModulImage

Brief Description

The "IL_InlineDiagModulImage" function block is used to determine whether a module signals a diagnose in the system. Each bit in the returned bit string is assigned to a module in the system in relation to its position and indicates whether this module signals a diagnose.

Library	Area
RIL_Inline.library	RIL Inline-Services

Fig.6-84: IL_InlineDiagModulImage library assignment

Interface Description

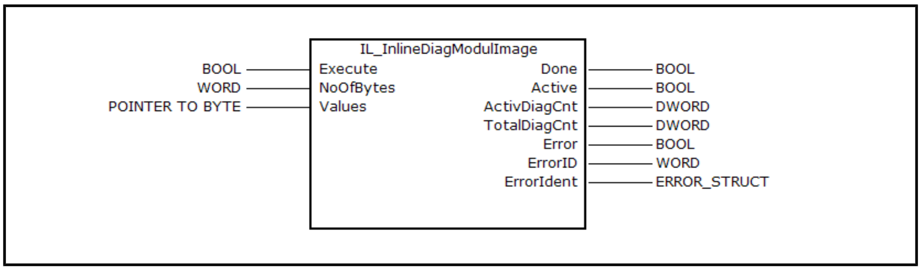


Fig.6-85: IL_InlineDiagModulImage

	Name	Type	Description
VAR_INPUT	Execute	BOOL	Activates the service
	NoOfBytes	BYTE	Maximum length of the data to be read; number of bytes available on the "Values" pointer
	Values	POINTER TO BYTE	Pointer to Diag bits: Byte0/Bit0 indicates the diagnostics for Module1 (from left to right)
VAR_OUTPUT	Done	BOOL	Done message (successful)
	Active	BOOL	Activity display
	ActivDiagCnt	DWORD	Number of active diagnostics bits
	TotalDiagCnt	DWORD	Total number of diagnostics bits
	Error	BOOL	Done message (unsuccessful)
	ErrorID	ERROR_CODE	See chapter "ErrorID" on page 171
	ErrorIdent	ERROR_STRUCT	Error structure from PB_DP_TABLE ErrorTable

Fig.6-86: IL_InlineDiagModulImage interface

Signal Time Diagram

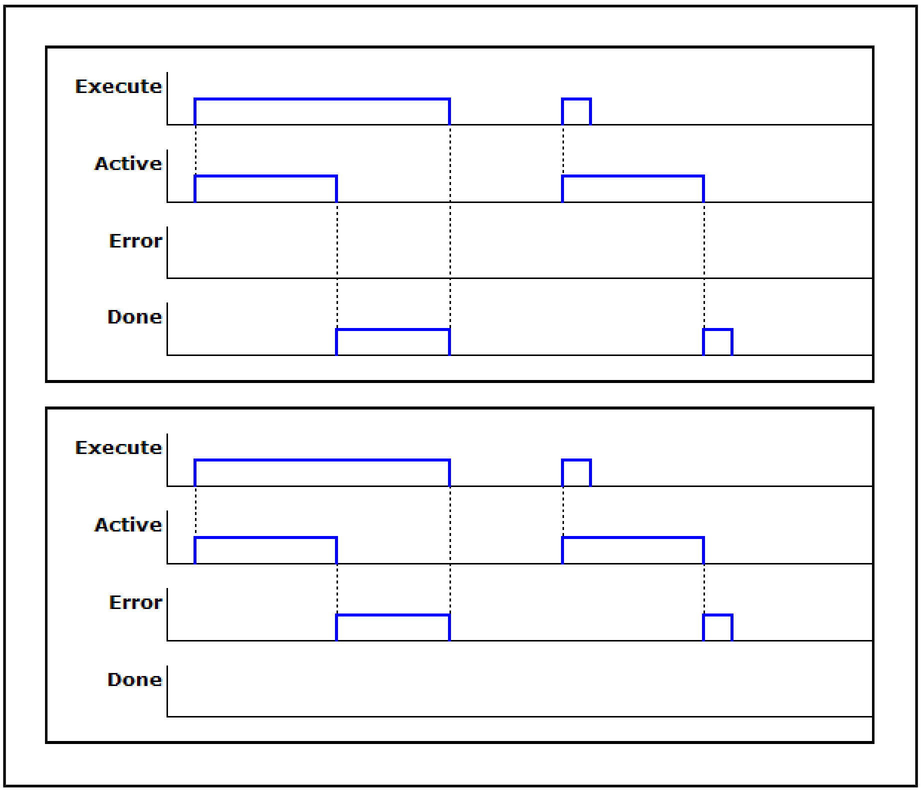


Fig.6-87: IL_InlineDiagModulImage signal time diagram

IL_InlineCfgDiagData

Brief Description

The "IL_InlineCfgDiagData" function block is used to determine the system configuration. The return parameters indicate the modules configured/scanned/compared. In addition, the module IDs of the modules configured/scanned are positioned in one field.

Library	Area
RIL_Inline.library	RIL Inline-Services

Fig.6-88: IL_InlineCfgDiagData library assignment

Interface Description

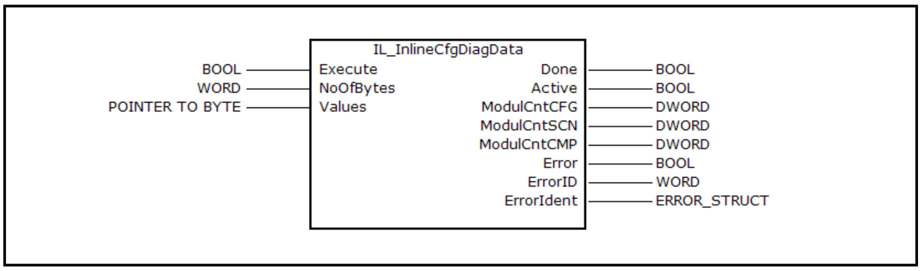


Fig.6-89: IL_InlineCfgDiagData library assignment

	Name	Type	Description
VAR_INPUT	Execute	BOOL	Activates the service
	NoOfBytes	BYTE	Maximum length of the data to be read; number of bytes available on the Values pointer
	Values	POINTER TO BYTE	Pointer to configuration data; see data type 1.3.1 CfgIdDesc
VAR_OUTPUT	Done	BOOL	Done message (successful)
	Active	BOOL	Activity display
	ModulCntCFG	DWORD	Number of configured modules
	ModulCntSCN	DWORD	Number of scanned modules
	ModulCntCMP	DWORD	Number of compared modules
	Error	BOOL	Done message (unsuccessful)
	ErrorID	ERROR_CODE	See chapter "ErrorID" on page 171
	ErrorIdent	ERROR_STRUCT	Error structure from PB_DP_TABLE ErrorTable

Fig.6-90: IL_InlineCfgDiagData interface

Signal Time Diagram

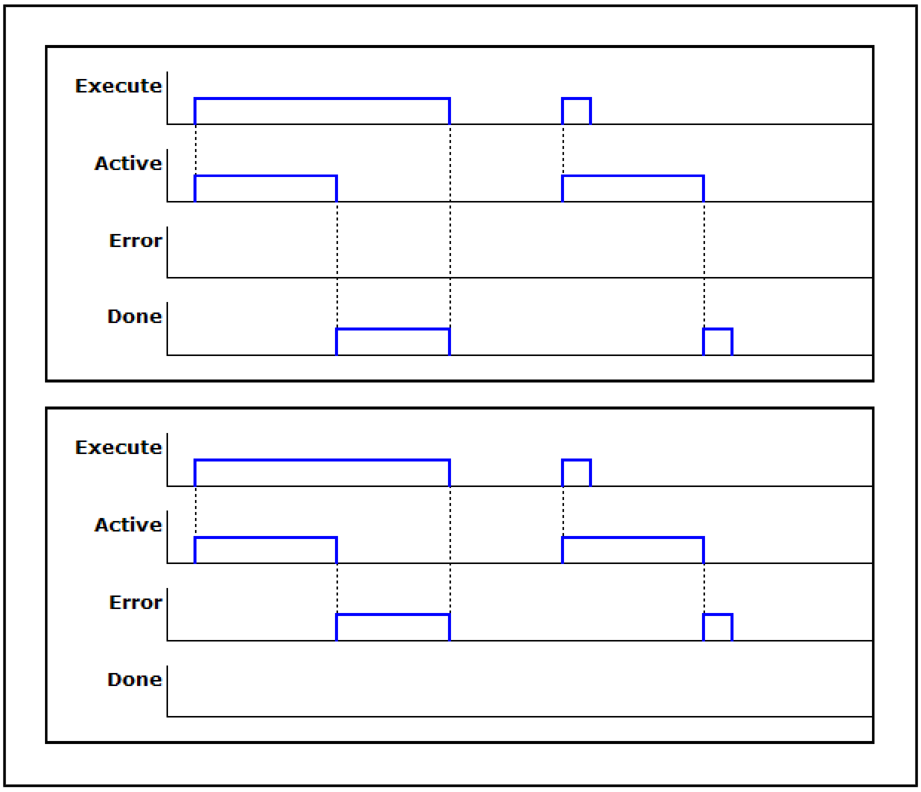


Fig.6-91: IL_InlineCfgDiagData signal time diagram

IL_InlineDiagCfgChk

Brief Description

The "IL_InlineDiagCfgChk" function block is used to check the system configuration for errors and to prepare the information diagnostically.

Library	Area
RIL_Inline.library	RIL Inline-Services

Fig.6-92: IL_InlineDiagCfgChk library assignment

Interface Description

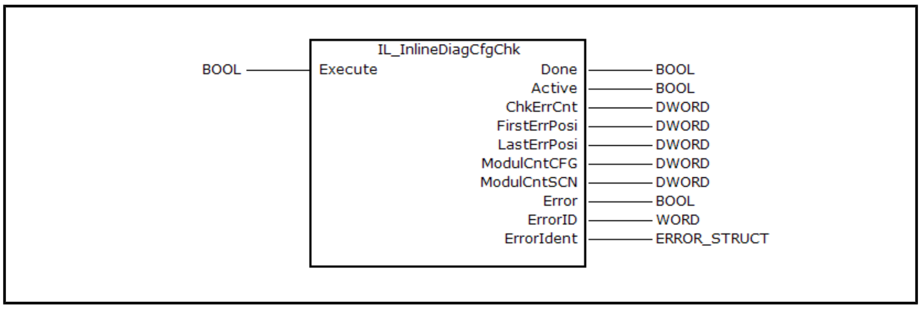


Fig.6-93: IL_InlineDiagCfgChk

	Name	Type	Description
VAR_INPUT	Execute	BOOL	Activates the service
VAR_OUTPUT	Done	BOOL	Done message (successful)
	Active	BOOL	Activity display
	ChkErrCnt	DWORD	Number of modules with errors
	FirstErrPosi	DWORD	First module with errors

	Name	Type	Description
	LastErrPosi	DWORD	Last module with errors
	ModulCntCFG	DWORD	Number of configured modules (target configuration)
	ModulCntSCN	DWORD	Number of scanned modules (target configuration)
	Error	BOOL	Done message (unsuccessful)
	ErrorID	ERROR_CODE	See chapter "ErrorID" on page 171
	ErrorIdent	ERROR_STRUCT	Error structure from PB_DP_TABLE ErrorTable

Fig.6-94: IL_InlineDiagCfgChk interface

Signal Time Diagram

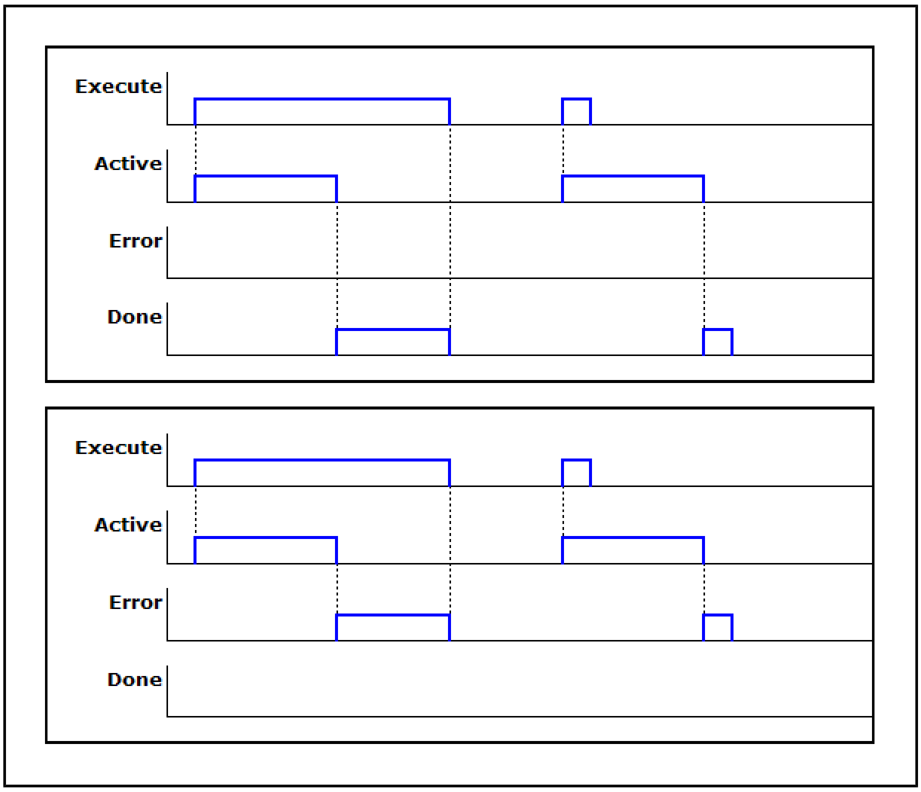


Fig.6-95: IL_InlineDiagCfgChk signal time diagram

IL_InlineDiagModulPosi

Brief Description

The "IL_InlineDiagModulPosi" function block is used to check a module at a defined position for diagnostics.

Library	Area
RIL_Inline.library	RIL Inline-Services

Fig.6-96: IL_InlineDiagModulPosi library assignment

Interface Description

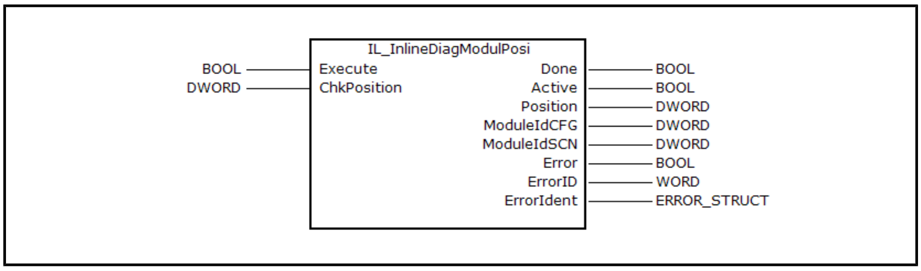


Fig.6-97: IL_InlineDiagModulPosi

	Name	Type	Description
VAR_INPUT	Execute	BOOL	Activates the service
	ChkPosition	DWORD	Position to be checked "0" causes the next faulty position to be determined
VAR_OUTPUT	Done	BOOL	Done message (successful)
	Active	BOOL	Activity display
	Position	DWORD	Position checked
	ModuleIdCFG	WORD	Configured module ID
	ModuleIdSCN	WORD	Scanned module ID
	Error	BOOL	Done message (unsuccessful)
	ErrorID	ERROR_CODE	See chapter "ErrorID" on page 171
	ErrorIdent	ERROR_STRUCT	Error structure from PB_DP_TABLE ErrorTable

Fig.6-98: IL_InlineDiagModulPosi interface

Signal Time Diagram

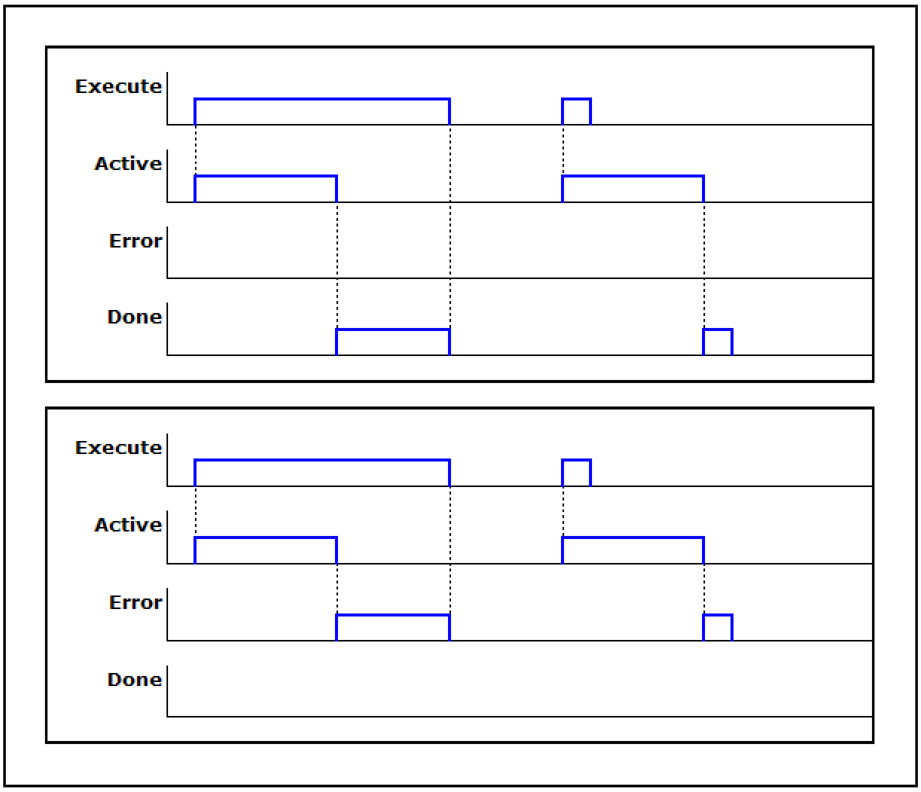


Fig.6-99: IL_InlineDiagModulPosi signal time diagram

IL_InlineDiagModulNext

Brief Description

The "IL_InlineDiagModulNext" function block is used to determine the next module with a diagnostics request.

Library	Area
RIL_Inline.library	RIL Inline-Services

Fig.6-100: IL_InlineDiagModulNext library assignment

Interface Description

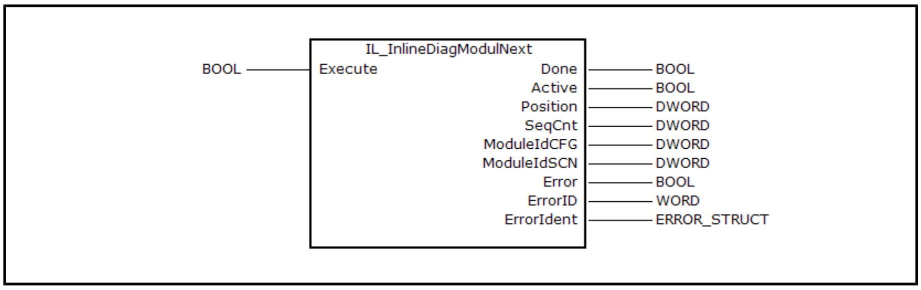


Fig.6-101: IL_InlineDiagModulNext

	Name	Type	Description
VAR_INPUT	Execute	BOOL	Activates the service
VAR_OUTPUT	Done	BOOL	Done message (successful)
	Active	BOOL	Activity display
	Position	DWORD	Next active Diag position
	SeqCnt	DWORD	Residual number of diagnostics
	ModuleIdCFG	WORD	–
	ModuleIdSCN	WORD	–
	Error	BOOL	Done message (unsuccessful)
	ErrorID	ERROR_CODE	See chapter "ErrorID" on page 171
	ErrorIdent	ERROR_STRUCT	Error structure from PB_DP_TABLE ErrorTable

Fig.6-102: IL_InlineDiagModulNext interface

Signal Time Diagram

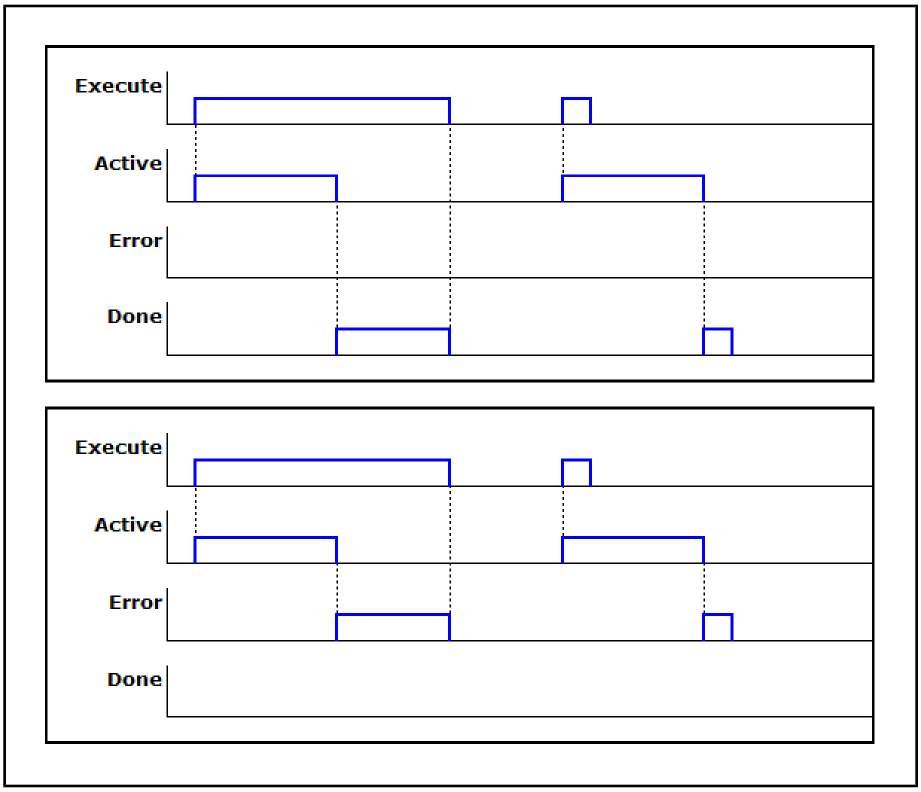


Fig.6-103: IL_InlineDiagModulNext signal time diagram

IL_InlineDiagMasterBus

Brief Description

The "IL_InlineDiagMasterBus" function block is used to determine whether there is a bus error.

Library	Area
RIL_Inline.library	RIL Inline-Services

Fig.6-104: IL_InlineDiagMasterBus library assignment

Interface Description

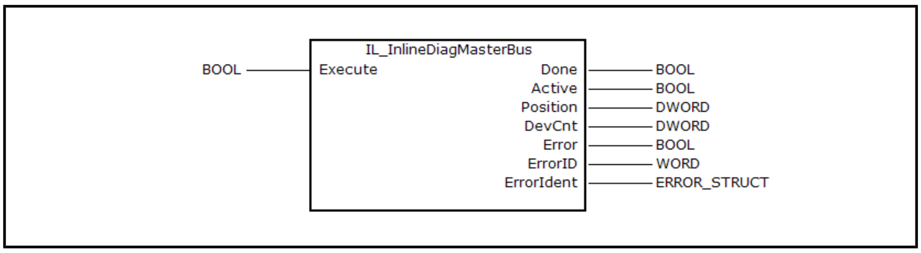


Fig.6-105: IL_InlineDiagMasterBus

	Name	Type	Description
VAR_INPUT	Execute	BOOL	Activates the service
VAR_OUTPUT	Done	BOOL	Done message (successful)
	Active	BOOL	Activity display
	Position	DWORD	Position of the bus error 0: no error 1...n: module number
	DevCnt	DWORD	Number of modules available
	Error	BOOL	Done message (unsuccessful)
	ErrorID	ERROR_CODE	See chapter "ErrorID" on page 171
	ErrorIdent	ERROR_STRUCT	Error structure from PB_DP_TABLE ErrorTable

Fig.6-106: IL_InlineDiagMasterBus interface

Signal Time Diagram

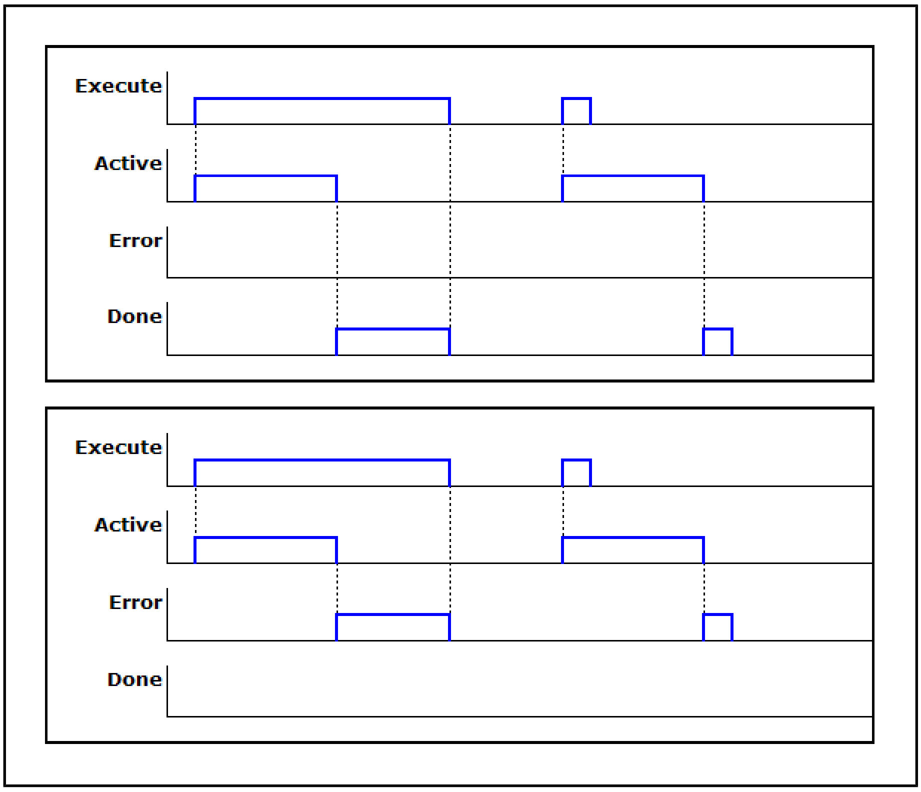


Fig.6-107: IL_InlineDiagMasterBus signal time diagram

IL_InlineDiagMasterFatal

Brief Description

The "IL_InlineDiagMasterFatal" function block returns a present system error from the stack error list.

Library	Area
RIL_Inline.library	RIL Inline-Services

Fig.6-108: IL_InlineDiagMasterFatal library assignment

Interface Description

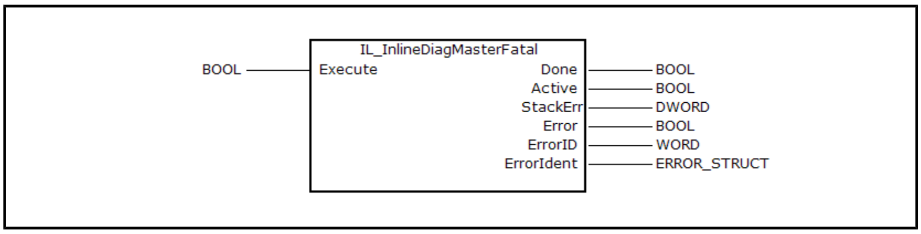


Fig.6-109: IL_InlineDiagMasterFatal

	Name	Type	Description
VAR_INPUT	Execute	BOOL	Activates the service
VAR_OUTPUT	Done	BOOL	Done message (successful)
	Active	BOOL	Activity display
	StackErr	DWORD	System error (report error number to BRC Service)
	Error	BOOL	Done message (unsuccessful)

	Name	Type	Description
	ErrorID	ERROR_CODE	See chapter "ErrorID" on page 171
	ErrorIdent	ERROR_STRUCT	Error structure from PB_DP_TABLE ErrorTable

Fig.6-110: IL_InlineDiagMasterFatal interface

Signal Time Diagram

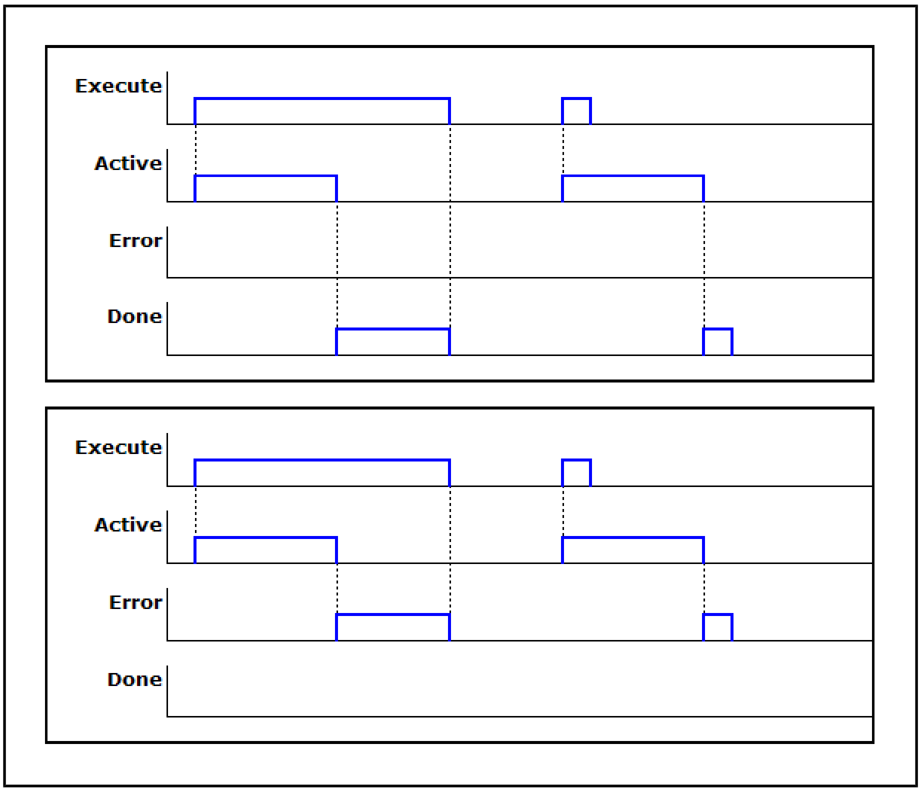


Fig.6-111: IL_InlineDiagMasterFatal signal time diagram

6.10 RIL_ProfibusDP

6.10.1 Overview

General Information

This library is used to provide DPV1 services of the Profibus Masters (DP master class 1) and also to establish a diagnostic interface between Profibus master and PLC program. Moreover, the library can be used to implement sync and freeze control commands.

DPV1 Services

Function Blocks

The function blocks serve to provide read and write access for the acyclic data exchange (DPV1):

- DP_RDREC
- DP_WRREC

Functions

Moreover, help functions are available for addressing:

- DP_ADDR
- DP_ID
- DP_SLOT

Diagnostic Information

Diagnostic information can be determined via function blocks. There are the following diagnostics:

- Slave Diagnostic Data According to the Profibus DP Standard : "DP_RDIAG " and "DP_RDIAG_EXT "
- General Field Bus Diagnostics : "fbd..."

Data Types

A part of this data is managed in special data types (arrays, structures):

- tFBD_BM_INFO
- tFBD_BIT_LIST
- tFBD_KSD_LIST

Function Blocks

- DP_RDIAG
- DP_RDIAG_EXT
- fbdBaudrateGet
- fbdBmErrorGet
- fbdBmInfoGet
- fbdBmStateGet
- fbdKsdListGet
- fbdPdTypeGet
- fbdPrjSlaveListGet
- fbdSlaveDiagListGet

Functions

Moreover, help functions are available for addressing:

- DP_ADDR
- DP_ID
- DP_SLOT

Sync and Freeze

Control commands to synchronize inputs and outputs.

- DP_SYCFR

Functions

Moreover, help functions are available for addressing:

- DP_ADDR
- DP_ID
- DP_SLOT

6.10.2 Slave Diagnostic Data According to the Profibus DP Standard

Profibus-specific diagnostic information according to the Profibus DP standard can be read via the following function blocks:

- DP_RDIAG
- DP_RDIAG_EXT

The slave diagnostic data is available through the DINFO or DINFO_PTR parameters. While for "DINFO" all diagnostic data is filed to an array, "DINFO_PTR" is a pointer to the diagnostic data.

The slave diagnostic data is subdivided in two parts:

- a general part with a defined length of 6 bytes,
- extended diagnostics (slave-specific, with variable length).

Offset	Type	Designation	Description
0	BYTE	Station status_1	See below
1	BYTE	Station status_2	See below
2	BYTE	Station status_3	See below
3	BYTE	Master_Add	Bus address of the master having parameterized the slave
4	WORD	Ident_Number	Ident_Number of the slave
6 – 243		Ext_Diag_Data	Extended diagnostics according to Profibus DP standard

Fig.6-112: Slave diagnostic data

The following description of station statuses 1 to 3 is an extract from the Profibus DP standard.

Bit	Designation	Description
7	Master_Lock	The DP slave was parameterized by a different master. This bit is set by the DP master (class 1), if the address in octet 4 is not equal to 255 and not equal to its own address. The DP slave sets this bit permanently to zero.
6	Prm_Fault	This bit is set by the DP slave, if the last parameter telegram was faulty, e.g. wrong length, wrong Ident_Number, invalid parameters.
5	Invalid_Slave_Response	This bit is set by the DP master as soon as one of the addressed DP slaves receives an implausible response. The DP slave sets this bit permanently to zero.
4	Not_Supported	This bit is set by the DP slave as soon as a function was requested that is not supported by this DP slave.
3	Ext_Diag	This bit is set by the DP slave. If the bit is set, a diagnostic entry must be available in the slave-specific diagnostic area (Ext_Diag_Data). If the bit is not set, a status message might be present in the slave-specific diagnostic area (Ext_Diag_Data). The meaning of this status message must be defined according to the respective application.
2	Cfg_Fault	This bit is set by the DP slave as soon as the configuration data previously received by the DP master does not correspond to the data determined by the DP slave.
1	Station_Not_Ready	This bit is set by the DP slave, if the DP slave is not yet ready for data exchange.
0	Station_Non_Existent	This bit is set by the DP master, if this DP slave cannot be reached via the bus. If this bit is set, the diagnostic bits contain the status of the last diagnostic message or the initial value. The DP slave sets this bit permanently to zero.

Fig.6-113: Station status_1

Bit	Designation	Description
7	Deactivated	This bit is set by the DP master as soon as the DP slave in the DP slave parameter set is not indicated as active and was removed from cyclic processing. The DP slave sets this bit permanently to zero.
6	Reserved	-
5	Sync_Mode	This bit is set by the DP slave as soon as it has received the sync control command. A change in these bits does not initiate a diagnostic message. That is why these bits usually do not reflect the current state.

Bit	Designation	Description
4	Freeze_Mode	This bit is set by the DP slave as soon as it has received the freeze control command. A change in these bits does not initiate a diagnostic message. That is why these bits usually do not reflect the current state.
3	WD_On (Watchdog)	This bit is set by the DP slave as soon as its response monitoring is activated.
2	1	The DP slave sets this bit permanently to zero.
1	Stat_Diag (static diagnostics)	If the DP slave sets this bit, the DP master has to retrieve diagnostic information until this bit is deleted again. The DP slave sets this bit, for example, if it cannot provide any valid user data.
0	Prm_Req	If the DP slave sets this bit, it must be reparameterized and reconfigured. The bit remains set until parameterization is completed. This bit is set by the DP slave.

Fig.6-114: Station status_2

Bit	Designation	Description
7	Ext_Diag_Overflow	If this bit is set, the amount of diagnostic information exceeds that specified in Ext_Diag_Data. The DP slave sets this bit, for example, if the number of channel diagnostics exceeds that which the DP slave can enter in its transmitter buffer; or the DP master sets this bit, if the DP slave sends more diagnostic information than the DP master is able to enter in its diagnostic buffer.
6	Reserved	-
5	Reserved	-
4	Reserved	-
3	Reserved	-
2	Reserved	-
1	Reserved	-
0	Reserved	-

Fig.6-115: Station status_3

For more information about the diagnostic functions, please refer to the description of the Function Blocks on page 144 et seq.

6.10.3 General Field Bus Diagnostics

The general field bus diagnostics provides the following data areas:

- BmState: Bus master status word, see "tFBD_BM_INFO "
- BmError: Bus master error word, see "tFBD_BM_INFO "
- SD: Slave diagnostic list, see "tFBD_BIT_LIST "
- KSD: Classified slave diagnostics, see "tFBD_KSD_LIST "
- Projected slave list, see "tFBD_BIT_LIST "

Bus Master Status Word "BmState" provides an overview of the status of the bus master and the slaves at the field bus. It indicates, for example, whether there is at least one slave with pending diagnostics.

Bus Master Error Word "BmError" indicates fatal errors rendering operation at the field bus impossible.

Slave Diagnostic List SD indicates the slaves which are signalling diagnostics.

Classified Slave Diagnostics KSD provides detailed information. It indicates the slaves which are signalling diagnostics and subdivides the diagnostics in error classes.

Projected Slave List

The projected slave list contains all available slaves according to the master configuration file. For example, this list can be used to compare the projected slaves with the slaves actually available at the field bus.

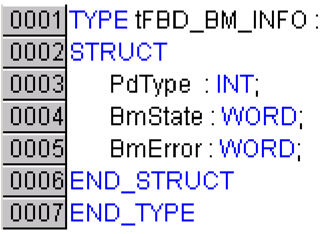
For more information about data areas, please refer to the description of the Data Types on page 141 et seq. and, about diagnostic functions, refer to the description of the Function Blocks on page 144 et seq.

6.10.4 Data Types

tFBD_BM_INFO

Brief Description

This structure comprises the "PdType", "BmStatus" and "BmError" variables. It is used in the "fbdBmInfoGet " function block.



```
TYPE tFBD_BM_INFO :
STRUCT
    PdType  : INT;
    BmState : WORD;
    BmError : WORD;
END_STRUCT
END_TYPE
```

Fig.6-116: tFBD_BM_INFO

PdType: Peripheral Driver

The "fbdBmInfoGet " function block determines the type of the installed peripheral driver and indicates it in "PdType":

Driver	Description
FBD_PDT_NONE	No peripheral driver installed
FBD_PDT_PCI_BMDP	Peripheral driver: Profibus DP
FBD_PDT_PCI_BMCAN	Peripheral driver: CANopen (presently not available)
FBD_PDT_PCI_BMIBS	Peripheral driver: Interbus (presently not available)

Fig.6-117: Installed peripheral driver in "PdType"

BmState: Bus Master Status Word

The bus master status word "BmState " provides an overview of the status of the bus master and the slaves at the field bus. It indicates, for example, whether there is at least one slave with pending diagnostics. Each set bit (TRUE) in "BmState" represents a status:

Bit	Status	Description
0	BMS_BMF	Bus master error: This bit indicates that there is a bus master error. In this case, the bus master error word contains more detailed information.
1	BMS_KSD	Classified slave diagnostics: If this bit is set, at least one slave indicates a classified diagnostics. Which classified diagnosis/diagnoses is/are set, can be determined through bits 8 to 13.
2	BMS_SD	Slave diagnostics: If this bit is set, at least one slave indicates a slave diagnostics.
3	-	- Reserved -
4	-	- Reserved -
5	-	- Reserved -
6	-	- Reserved -

Bit	Status	Description
7	BMS_AKTIV	Active identification: The value of this bit must always be 1. If this is not the case, there is a fatal error in the software of the bus master.
8	BMS_SNE	One or more slaves are not accessible via the bus.
9	BMS_SKF	One or more slaves indicate configuration errors.
10	BMS_DPS	One or more slaves indicate static diagnostics.
11	BMS_EXD	One or more slaves indicate extended diagnostics.
12	BMS_SNB	One or more slaves are not ready for cyclic data exchange.
13	BMS_SF	One or more slaves indicate a miscellaneous error.
14	-	- Reserved -
15	-	- Reserved -

Fig.6-118: Status coding in "BmState"

BmState is used in the following function blocks:

- fbdBmInfoGet
- fbdBmStateGet

BmError: Bus Master Error Word

The "BmError" bus master error word indicates fatal errors rendering operation at the field bus impossible. Each set bit (TRUE) in "BmError" represents an error:

Bit	Error *	Description
0	BMF_HW_ERR	Hardware error
1	BMF_MPS_ERR	Master parameter set (field bus configuration file) is missing or faulty
2	BMF_BUS_ERR	Error at the field bus (e.g. short-circuit, ...)
3	BMF_SW_ERR	System error in the peripheral driver (i.e. the driver software has detected a fatal error)

* "BMF_OK" indicates that there is no error

Fig.6-119: Error coding in "BmError"

BmError is used in the following function blocks:

- fbdBmErrorGet
- fbdBmInfoGet

tFBD_BIT_LIST

Brief Description

The bit list "tFBD_BIT_LIST" has a defined length of 16 bytes (128 bits).

```
TYPE tFBD_BIT_LIST :
    ARRAY [0..15] OF BYTE;
END_TYPE
```

Fig.6-120: tFBD_BIT_LIST

Bit List Coding

Each bit of the bit list is assigned to a bus address of the slave (Profibus: FDL address). For example, the lowest-order bit in the first array element (ARRAY[0]) is assigned to the Profibus user with address 0:

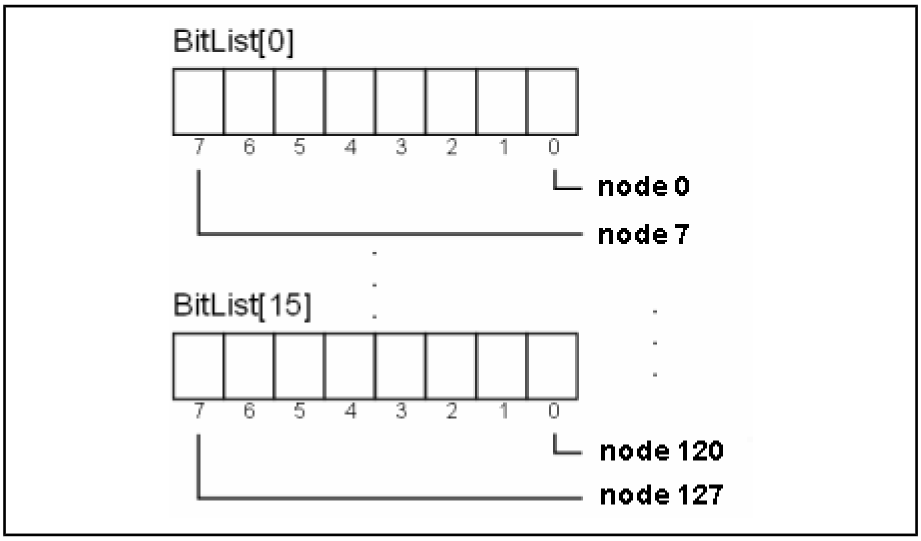


Fig.6-121: Bit list coding

Example

The bit list is used in the following function blocks:

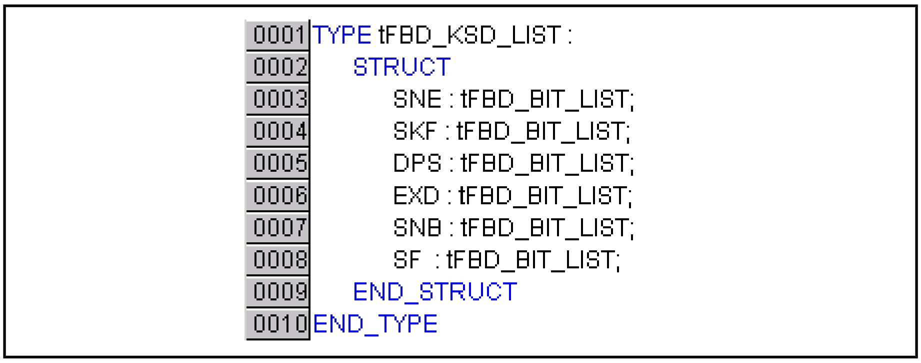
- fbdPrjSlaveListGet
- fbdSlaveDiagListGet

The bit list is also used in the tFBD_KSD_LIST (classified slave diagnostics).

tFBD_KSD_LIST

Brief Description

The list of the classified slave diagnostics (KSD list) consists of six bit lists. That means that there is a bit list for each error type.



```
TYPE tFBD_KSD_LIST :
    STRUCT
        SNE : tFBD_BIT_LIST;
        SKF : tFBD_BIT_LIST;
        DPS : tFBD_BIT_LIST;
        EXD : tFBD_BIT_LIST;
        SNB : tFBD_BIT_LIST;
        SF  : tFBD_BIT_LIST;
    END_STRUCT
END_TYPE
```

Fig.6-122: tFBD_KSD_LIST

Classified Slave Diagnostics (KSD)

The classified slave diagnostics provides the following error types..

SNE	Slave not accessible. The slave is not accessible at the bus. Possible causes: ● Slave not available ● Voltage at slave turned off ● Faulty bus installation ● Physical interferences
SKF	Slave configuration error. The slave type or the I/O configuration of the slave does not correspond to the projected values in the field bus configuration file of the master.
DPS	Slave indicates static diagnostics: The slave cannot provide valid user data. The application layer of the slave is not ready for data exchange with the master.

EXD	Slave indicates extended diagnostics. The extended diagnostics is slave-specific and can be found in the description of the slave. Possible causes: ● Load voltage missing/switched off (e.g. with E-STOP) ● Short-circuit at one output ● Overload ● Over-temperature ● Line break
SNB	Slave is not ready. The slave is not ready for data exchange, as it is not yet put into operation by the master (message from the protocol layer of the slave).
SF	Slave indicates a miscellaneous error.

Fig.6-123: Classified slave diagnostics

Bit List Coding

Each bit of a bit list is assigned to a bus address of the slave (Profibus: FDL address). For example, the lowest-order bit in the first array element (ARRAY[0]) is assigned to the Profibus device with address 0:

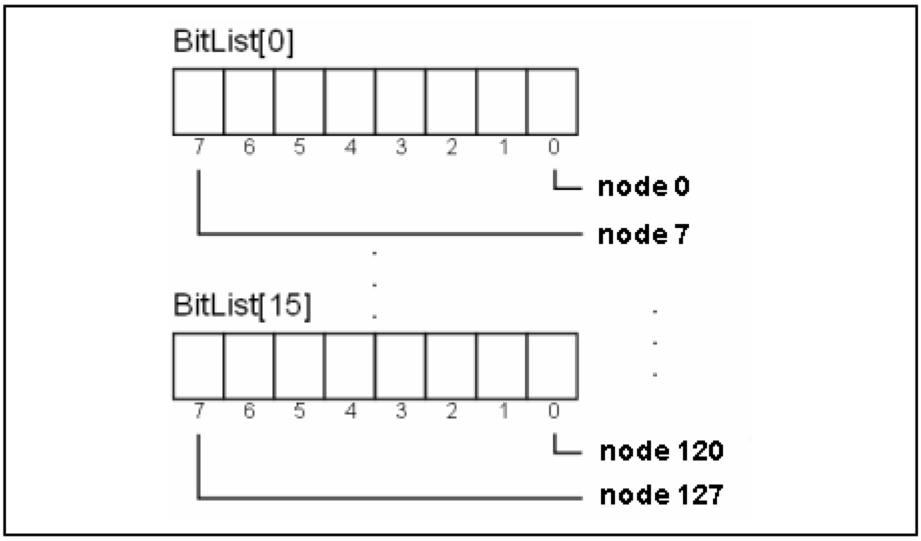


Fig.6-124: Bit list coding

Example

Bit[x] = FALSE	Slave[x] has no diagnostics
Bit[x] = TRUE	Slave[x] has diagnostics

Fig.6-125: Bit list: Classified slave diagnostics

6.10.5 Function Blocks

DP_RDIAG

Brief Description

The "DP_RDIAG" function block is used by the DP master (DPM1) to read the diagnostic data of a slave. The data buffer of the diagnostic data must be provided to address it via a POINTER.

Interface Description

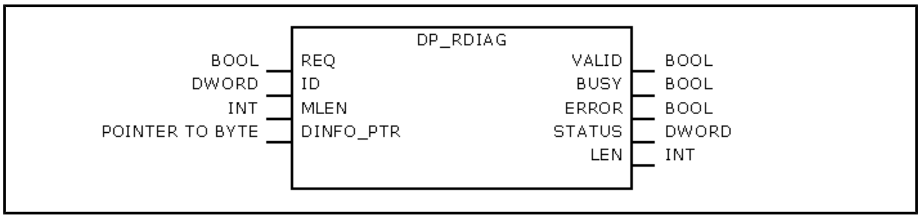


Fig.6-126: DP_RDIAG

	Name	Type	Comment
VAR_INPUT	REQ	BOOL	TRUE: Start reception
	ID	DWORD	Slot handle, see the following table
	MLEN	INT	Maximum length of the data to be read
	DINFO_PTR	POINTER TO BYTE	Pointer to the data buffer of the Slave Diagnostic Data According to the Profibus DP Standard
VAR_OUTPUT	VALID	BOOL	TRUE: New, valid diagnostic data available
	ERROR	BOOL	TRUE: Error/s occurred
	BUSY	BOOL	TRUE: The function block is busy. As long as BUSY = TRUE, the data cannot be evaluated yet.
	STATUS	DWORD	Previously determined status
	LEN	INT	Length of the diagnostic data in bytes

Fig.6-127: DP_RDIAG interface

Byte	Contents	Description
0	MASTER	ID of the DP system: DP master identification
1	SEGMENT	Number of the DP segment
2	STATION	Number of the DP slave (bus address)
3	SLOT	Number of the slot within the slave

Fig.6-128: Slot handle: "ID" parameter

Example To address the slave with bus address 12, the value of the ID is 16#000C0000.

The ID is formed from the various components by means of the DP_ID function.

DP_RDIAG_EXT

Brief Description The "DP_RDIAG_EXT" function block is used by the DP master (DPM1) to read the diagnostic data of a slave. The diagnostic data is stored in an ARRAY.

Interface Description

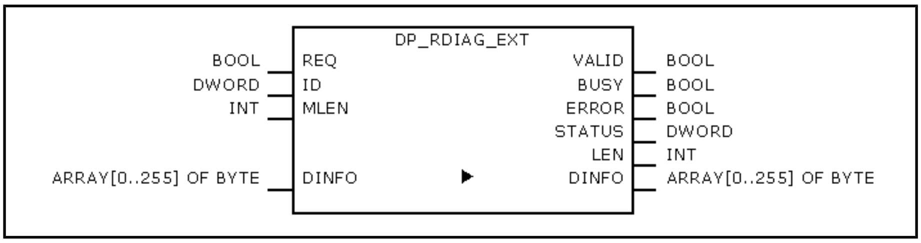


Fig.6-129: DP_RDIAG_EXT

	Name	Type	Comment
VAR_INPUT	REQ	BOOL	TRUE: Start reception
	ID	DWORD	Slot handle, see the following table
	MLEN	INT	Maximum length of the data to be read
VAR_IN_OUT	DINFO	ARRAY [0..255] OF BYTE	Slave Diagnostic Data According to the Profibus DP Standard
VAR_OUTPUT	VALID	BOOL	TRUE: New, valid diagnostic data available
	ERROR	BOOL	TRUE: Error/s occurred
	BUSY	BOOL	TRUE: The function block is busy. As long as BUSY = TRUE, the data cannot be evaluated yet.
	STATUS	DWORD	Previously determined status
	LEN	INT	Length of the diagnostic data in bytes

Fig.6-130: DP_RDIAG_EXT interface

Byte	Contents	Description
0	MASTER	ID of the DP system: DP master identification
1	SEGMENT	Number of the DP segment
2	STATION	Number of the DP slave (bus address)
3	SLOT	Number of the slot within the slave

Fig.6-131: Slot handle: "ID" parameter

Example

To address the slave with bus address 12, the value of the ID is 16#000C0000.

The ID is formed from the various components by means of the DP_ID function.

DP_RDREC

Brief Description

The "DP_RDREC" function block serves to provide read access for acyclic data exchange (DPV1). A pointer (POINTER) must be addressed to define a target area for the process data to be read.

Interface Description

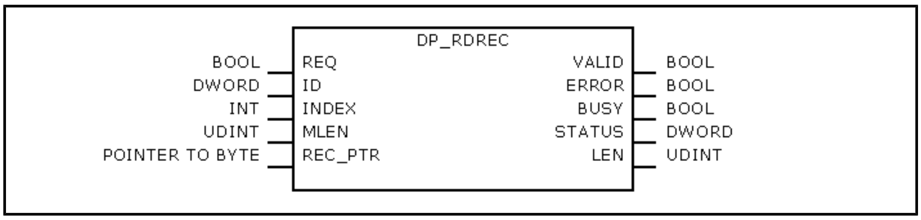


Fig.6-132: DP_RDREC

	Name	Type	Comment
VAR_INPUT	REQ	BOOL	TRUE: Start reading
	ID	DWORD	Slot handle, see the following table
	INDEX	INT	Index of the process data (offset)
	MLEN	UDINT	Maximum length of the process data in bytes
	REC_PTR	POINTER TO BYTE	Pointer to the target area

	Name	Type	Comment
VAR_OUTPUT	VALID	BOOL	TRUE: New, valid data available
	ERROR	BOOL	TRUE: Error/s occurred
	BUSY	BOOL	TRUE: The function block is busy. As long as BUSY = TRUE, the data cannot be evaluated yet.
	STATUS	DWORD	Previously determined status
	LEN	UDINT	Length of the process data in bytes

Fig.6-133: DP_RDREC interface

Byte	Contents	Description
0	MASTER	ID of the DP system: DP master identification
1	SEGMENT	Number of the DP segment
2	STATION	Number of the DP slave (bus address)
3	SLOT	Number of the slot within the slave

Fig.6-134: Slot handle: "ID" parameter

Example

To address the slave with bus address 12, the value of the ID is 16#000C0000.

The ID is formed from the various components by means of the DP_ID function.

DP_WRREC

Brief Description

The DP_WRREC function block serves to provide write access for acyclic data exchange (DPV1). The process data to be written must be provided by addressing a pointer (POINTER).

Interface Description

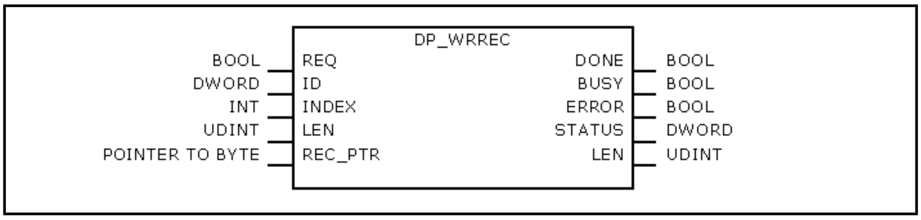


Fig.6-135: DP_WRREC

	Name	Type	Comment
VAR_INPUT	REQ	BOOL	TRUE: Start writing
	ID	DWORD	Slot handle, see the following table
	INDEX	INT	Index of the process data (offset)
	LEN	UDINT	Length of the process data in bytes
	REC_PTR	POINTER TO BYTE	Pointer to the process data to be written
VAR_OUTPUT	DONE	BOOL	TRUE: Call completed
	ERROR	BOOL	TRUE: Error/s occurred

	Name	Type	Comment
	BUSY	BOOL	TRUE: The function block is busy. As long as BUSY = TRUE, the data cannot be evaluated yet.
	STATUS	DWORD	Previously determined status

Fig.6-136: DP_WRREC interface

Byte	Contents	Description
0	MASTER	ID of the DP system: DP master identification
1	SEGMENT	Number of the DP segment
2	STATION	Number of the DP slave (bus address)
3	SLOT	Number of the slot within the slave

Fig.6-137: Slot handle: "ID" parameter

Example To address the slave with bus address 12, the value of the ID is 16#000C0000.

The ID is formed from the various components by means of the DP_ID function.

fbdBaudrateGet

Brief Description The "fbdBaudrateGet" function block reads the baud rate of the connected field bus. The baud rate is specified in bits per second.

Interface Description

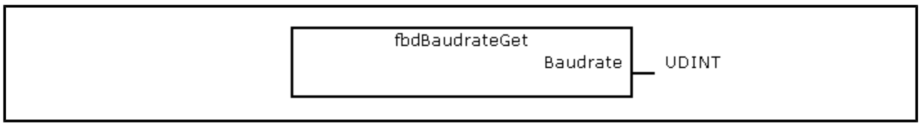


Fig.6-138: fbdBaudrateGet

	Name	Type	Comment
VAR_OUTPUT	Baud rate	UDINT	Baud rate at the field bus (bit/s)

Fig.6-139: fbdBaudrateGet interface

fbdBmErrorGet

Brief Description The "fbdBmErrorGet" function block reads the current bus master error word.

Interface Description

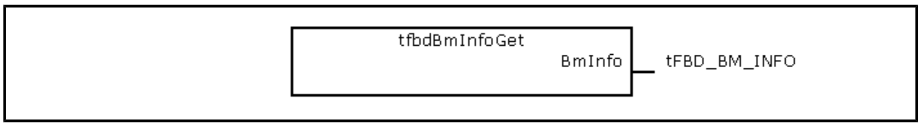


Fig.6-140: fbdBmErrorGet

	Name	Type	Comment
VAR_OUTPUT	BmError	WORD	See tFBD_BM_INFO

Fig.6-141: fbdBmErrorGet interface

fbdBmInfoGet

Brief Description The "fbdBmInfoGet" function block returns the "tFBD_BM_INFO" structure. This structure contains the "PdType", "BmStatus" and "BmError" variables. Thus, the information is provided by one single call, rather than by calling several different functions.

Interface Description

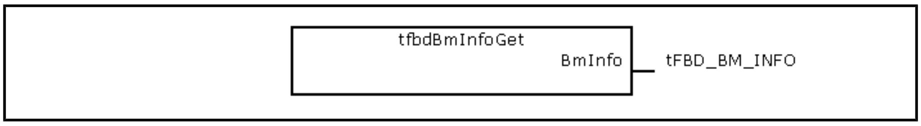


Fig.6-142: fbdBmInfoGet

	Name	Type	Comment
VAR_OUTPUT	BmInfo	fbdBmInfoGet	fbdBmInfoGet

Fig.6-143: fbdBmInfoGet interface

fbdBmStateGet

Brief Description The "fbdBmStateGet" function block reads the current bus master status word.

Interface Description

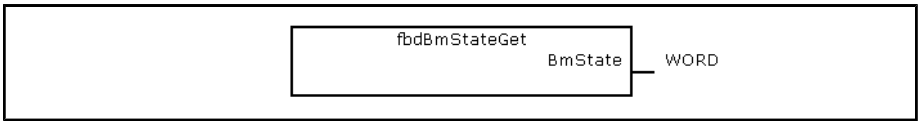


Fig.6-144: fbdBmStateGet

	Name	Type	Comment
VAR_OUTPUT	BmState	WORD	See fbdBmInfoGet

Fig.6-145: fbdBmStateGet interface

fbdKsdListGet

Brief Description The "fbdKsdListGet" function block reads the current KSD list.

Interface Description

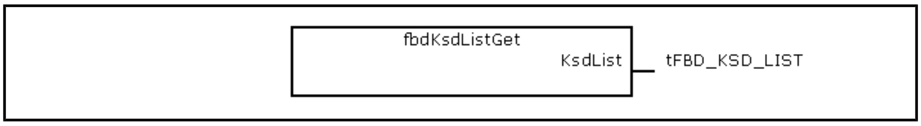


Fig.6-146: fbdKsdListGet

	Name	Type	Comment
VAR_OUTPUT	KsdList	tFBD_KSD_LIST	See tFBD_KSD_LIST

Fig.6-147: fbdKsdListGet interface

fbdPdTypeGet

Brief Description The "fbdPdTypeGet" function block determines the type of the installed peripheral driver.

Interface Description

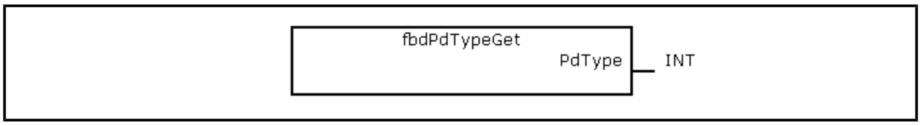


Fig.6-148: fbdPdTypeGet

	Name	Type	Comment
VAR_OUTPUT	PdType	INT	Peripheral driver type: PDT_NONE: No peripheral driver installed. PDT_PCI_BMDP: Profibus DP. PDT_PCI_BMCAN: CAN-Open. PDT_PCI_BMIBS: Interbus S.

Fig.6-149: fbdPdTypeGet interface

fbdPrjSlaveListGet

Brief Description

The "fbdPrjSlaveListGet" function block reads the list of the projected slaves. The list contains all available slaves according to the master configuration file.

Interface Description

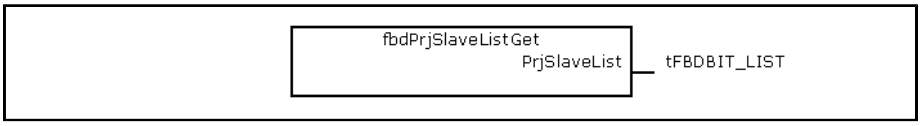


Fig.6-150: fbdPrjSlaveListGet

	Name	Type	Comment
VAR_OUTPUT	PrjSlaveList	tFBD_BIT_LIST	See tFBD_BIT_LIST

Fig.6-151: fbdPrjSlaveListGet interface

Example

Each bit of the bit list is assigned to a bus address of the slave:

Bit[x] = TRUE	Slave[x] is projected
Bit[x] = FALSE	Slave[x] is not projected

Fig.6-152: Bit list: projected slaves

fbdSlaveDiagListGet

Brief Description

The "fbdSlaveDiagListGet" function block reads the current slave diagnostic list.

Interface Description

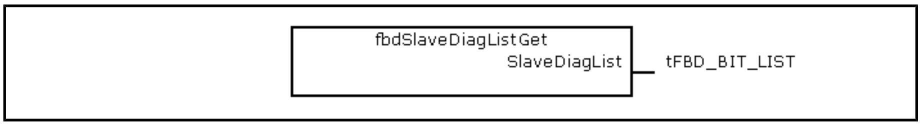


Fig.6-153: fbdSlaveDiagListGet

	Name	Type	Comment
VAR_OUTPUT	SlaveDiagList	tFBD_BIT_LIST	See tFBD_BIT_LIST

Fig.6-154: fbdSlaveDiagListGet interface

Example

Each bit of the bit list is assigned to a bus address of the slave:

Bit[x] = TRUE	Slave[x] has diagnostics
Bit[x] = FALSE	Slave[x] has no diagnostics

Fig.6-155: Bit list: Slave diagnostics

DP_SYCFR

Brief Description

The "DP_SYCFR" function block can be used to implement control commands for synchronization of inputs and outputs. For example, the outputs of several drives can be synchronized (e.g. Rexroth EcoDrive, IndraDrive), several axes can start at the same time, etc.

This is based on the possibility that a Profibus DP master can send a "global control telegram" to a defined slave or to entire slave groups. The global control telegram can be used to transmit one of the following control commands:

- **Freeze** (control command code: **16#08**): causes all addressed slaves (see table below, "ID" and "GROUP" input parameters) to switch to freeze mode. On the transition to this state, a slave "freezes" the current status of its **input data**. The input data will again be updated once when the next freeze command arrives.

- **Unfreeze** (control command code: **16#04**): causes all addressed slaves to exit the freeze mode.
- **Sync** (control command code: **16#20**): causes all addressed slaves to switch to sync mode. On the transition to this state, a slave "freezes" the current status of its **outputs** corresponding to its current internal output image. The outputs will again be updated once when the next sync command arrives, according to the current internal output image.
- **Unsync** (control command code: **16#10**): causes all addressed slaves to update their outputs according to the current internal output image and to exit the sync mode.

☞ In order that the control commands of the sync and freeze modes are actually transmitted from the master to all addressed slaves, the following requirements must be met:

- Each slave is assigned to a slave group, see Fig. 4-32.
- The relevant mode (sync, freeze) is enabled for the corresponding slave group, see Fig. 4-25.

Interface Description

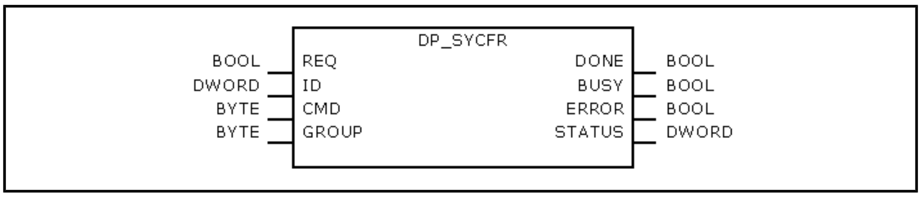


Fig.6-156: DP_SYCFR

	Name	Type	Comment
VAR_INPUT	REQ	BOOL	TRUE: Execute function
	ID	DWORD	Slot handle, see fig. 6-158 "Slot handle: parameter ID" on page 151.
	CMD	BYTE	Control command
	GROUP	BYTE	Selects one or more groups to which the command refers. Each bit is assigned to a group.
VAR_OUTPUT	DONE	BOOL	TRUE: Call completed
	BUSY	BOOL	TRUE: The function block is busy. As long as BUSY = TRUE, the data cannot be evaluated yet.
	ERROR	BOOL	TRUE: Error/s occurred
	STATUS	DWORD	Previously determined status

Fig.6-157: DP_SYCFR interface

Byte	Contents	Description
0	MASTER	ID of the DP system: DP master identification
1	SEGMENT	Number of the DP segment

Byte	Contents	Description
2	STATION	Number of the DP slave (bus address): If the command is to apply for only one special slave, then the bus address of the slave must be entered here (0..125). However, if the command is to be entered for all slaves of a group, the global address (= 127) must be entered here.
3	SLOT	Number of the slot within the slave

Fig.6-158: Slot handle: parameter ID

Example

Slot handle

Address all slaves of a group: ID = 16#007F0000.

The ID is formed from the various components by means of the DP_ID function.

☞ Outputs can only be synchronized if all slaves received the current output data before having received the sync or unsync control command! To achieve this, call the "DP_SYCFR" function block from the same PLC task from which the output data of the slave/s is written.

In a PLC task, the output data for the synchronized slaves is written first. Then, the sync or unsync command is started with "DP_SYCFR". As long as the function block is BUSY, the output must not be modified.

Example

Start several axes at the same time.

1. Send the sync control command to the participating Profibus DP devices. This freezes their outputs.
2. Transmit the command which starts the axes to the participating devices (e.g. "MoveAbsolut" for the drives).
3. Send the unsync control command to the participating Profibus DP devices. As a result, the devices update their outputs according to the current internal output image, simultaneously start the axis movements and exit the sync mode.

6.10.6 Functions

DP_ADDR

Brief Description

This function is not implemented. This function can be called for reasons of compatibility with the Profibus Guideline 2182 (see http://www.profibus.com/celummdb/doc/PROFIBUS/Downloads/Specifications%20&%20Standards/Comm-Func-Block_2182_V20_Nov05.pdf) but it transfers a handle without any changes.

Interface Description

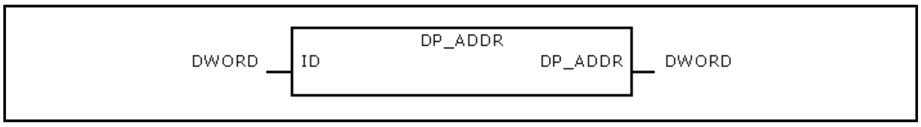


Fig.6-159: DP_ADDR

	Name	Type	Comment
VAR_INPUT	ID	DWORD	Slot handle
Function value		DWORD	

Fig.6-160: DP_ADDR interface

DP_ID

Brief Description

This function is based on the Profibus Guideline 2182 (see http://www.profibus.com/celummdb/doc/PROFIBUS/Downloads/Specifications%20&%20Standards/Comm-Func-Block_2182_V20_Nov05.pdf). It provides the handle for a physical address of a slot.

Interface Description

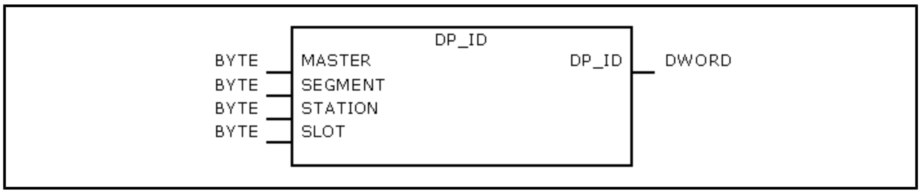


Fig.6-161: DP_ID

	Name	Type	Comment
VAR_INPUT	MASTER	BYTE	ID of the DP system: DP master identification
	SEGMENT	BYTE	Number of the DP segment
	STATION	BYTE	Number of the DP slave (bus address): If the command is to apply for only one special slave, then the bus address of the slave must be entered here (0..125). Only for DP_SYCFR: However, if the command is to be entered for all slaves of a group, the global address (= 127) must be entered here.
	SLOT	BYTE	Number of the slot within the slave
Function value		DWORD	Slot handle

Fig.6-162: DP_ID interface

DP_SLOT

Brief Description

This function is based on the Profibus Guideline 2182 (see http://www.profibus.com/celummdb/doc/PROFIBUS/Downloads/Specifications%20&%20Standards/Comm-Func-Block_2182_V20_Nov05.pdf). It sets the specified slot number in the slot handle.

Interface Description

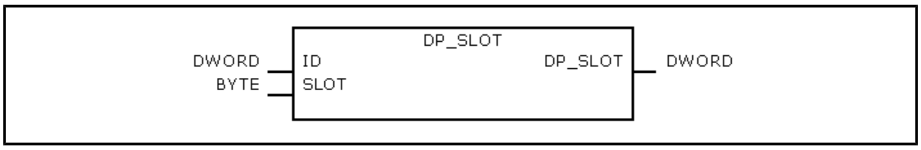


Fig.6-163: DP_SLOT

	Name	Type	Comment
VAR_INPUT	ID	DWORD	Slot handle
	SLOT	BYTE	Slot number
Function value		DWORD	Slot handle

Fig.6-164: DP_SLOT interface

6.11 RIL_ProfibusDP_02

6.11.1 Overview

History

For the first time, the CML40 now provides one system for several DP masters. This is achieved by using the DP master function module and the existing onboard master.

The RIL_ProfibusDP.lib version 01V01 is not suitable for use on systems with more DP masters than one. This library will be expanded to version 01V02 to ensure compatibility.

In addition, a RIL_ProfibusDP_02.lib will be created. This library is able to distinguish several DP masters.

Target Systems

The library can be used in the following systems.

Target assembly	Remark
CML40	Onboard / 4 COM-BM-DP
CML20	Compatible (onboard master/slave only)

6.11.2 FB Overview

Structure

The library comprises the following folders:

_Version

The RIL_ProfibusDP_02.lib is used to set the version number to 01V00.

RIL DP-Services

DP and diagnostic functions

Overview of Functions

Designation	**Type**	**Description**
Version_RIL_ProfibusDP_02_01V00	FUW	Version ID
IL_DPIdent	FUW	Assembly of Profibus IDs
IL_DPV1Read	FB	Reading V1 service
IL_DPV1Write	FB	Writing V1 service
IL_DPReadDiag	FB	Read diagnostic data of a slave according to DP standard
IL_DPSyncFr	FB	Synchronize/freeze control command
IL_DPDeviceListGet	FB	List of active DP devices in the system
IL_DPBaudrateGet	FB	Read the current baud rate
IL_DPDevInfoGet	FB	Read the bus master information structure
IL_DPPrjSlaveListGet	FB	List of projected slaves
IL_DPSlaveDiagListGet	FB	List of current slave diagnostics

Fig.6-165: Overview of the function blocks and functions contained in the library

6.11.3 Data Types

Slave Diagnostic Data According to the Profibus DP Standard

Overview

Structure of the Profibus-specific diagnostic information according to the Profibus DP standard.

The slave diagnostic data is subdivided in two parts:

- A general part with a defined length of 6 bytes.
- Extended diagnostics (slave-specific, with variable length).

Offset	Type	Designation	Description
0	BYTE	Station status_1	See below
1	BYTE	Station status_2	See below
2	BYTE	Station status_3	See below
3	BYTE	Master_Add	Bus address of the master having parameterized the slave
4	WORD	Ident_Number	Ident_Number of the slave
6-243		Ext_Diag_Data	Extended diagnostics according to Profibus DP standard

Fig.6-166: Slave diagnostic data

Station Status

The following description of station statuses 1 to 3 is an extract from the Profibus DP standard.

Bit	Designation	Description
7	Master_Lock	The DP slave was parameterized by a different master. This bit is set by the DP master (class 1), if the address in octet 4 is not equal to 255 and not equal to its own address. The DP slave sets this bit permanently to zero.
6	Prm_Fault	This bit is set by the DP slave, if the last parameter telegram was faulty, e.g. wrong length, wrong Ident_Number, invalid parameters.
5	Invalid_Slave_Response	This bit is set by the DP master as soon as one of the addressed DP slaves receives an implausible response. The DP slave sets this bit permanently to zero.
4	Not_Supported	This bit is set by the DP slave as soon as a function was requested that is not supported by this DP slave.
3	Ext_Diag	This bit is set by the DP slave. If the bit is set, a diagnostic entry must be available in the slave-specific diagnostic area (Ext_Diag_Data). If the bit is not set, a status message might be present in the slave-specific diagnostic area (Ext_Diag_Data). The meaning of this status message must be defined according to the respective application.
2	Cfg_Fault	This bit is set by the DP slave as soon as the configuration data previously received by the DP master does not correspond to the data determined by the DP slave.
1	Station_Not_Ready	This bit is set by the DP slave, if the DP slave is not yet ready for data exchange.
0	Station_Non_Existent	This bit is set by the DP master, if this DP slave cannot be reached via the bus. If this bit is set, the diagnostic bits contain the status of the last diagnostic message or the initial value. The DP slave sets this bit permanently to zero.

Fig.6-167: Station status_1

Bit	Designation	Description
7	Deactivated	This bit is set by the DP master as soon as the DP slave in the DP slave parameter set is not indicated as active and was removed from cyclic processing. The DP slave sets this bit permanently to zero.
6	Reserved	-
5	Sync_Mode	This bit is set by the DP slave as soon as it has received the sync control command. A change in these bits does not initiate a diagnostic message. That is why these bits usually do not reflect the current state.
4	Freeze_Mode	This bit is set by the DP slave as soon as it has received the freeze control command. A change in these bits does not initiate a diagnostic message. That is why these bits usually do not reflect the current state.
3	WD_On (Watchdog)	This bit is set by the DP slave as soon as its response monitoring is activated.
2	1	The DP slave sets this bit permanently to zero.
1	Stat_Diag (static diagnostics)	If the DP slave sets this bit, the DP master has to retrieve diagnostic information until this bit is deleted again. The DP slave sets this bit, for example, if it cannot provide any valid user data.
0	Prm_Req	If the DP slave sets this bit, it must be reparameterized and reconfigured. The bit remains set until parameterization is completed. This bit is set by the DP slave.

Fig.6-168: Station status_2

Bit	Designation	Description
7	Ext_Diag_Overflow	If this bit is set, the amount of diagnostic information exceeds that specified in Ext_Diag_Data. The DP slave sets this bit, for example, if the number of channel diagnostics exceeds that which the DP slave can enter in its transmitter buffer; or the DP master sets this bit, if the DP slave sends more diagnostic information than the DP master is able to enter in its diagnostic buffer.
6	Reserved	-
5	Reserved	-
4	Reserved	-
3	Reserved	-
2	Reserved	-
1	Reserved	-
0	Reserved	-

Fig.6-169: Station status_3

Bus Master Status Word

The "BmState" bus master status word provides an overview of the status of the bus master and the slaves at the field bus. It indicates, for example, whether there is at least one slave with pending diagnostics. Each set bit (TRUE) in "BmState" represents a status:

Bit	Status	Description
0	BMS_BMF	Bus master error: This bit indicates that there is a bus master error. In this case, the bus master error word contains more detailed information.
1	BMS_KSD	Classified slave diagnostics: If this bit is set, at least one slave indicates a classified diagnostics. Which classified diagnostics is/are set, can be determined by bits 8 to 13.
2	BMS_SD	Slave diagnostics: If this bit is set, at least one slave indicates a slave diagnosis.
3	-	Reserved
4	-	Reserved
5	-	Reserved
6	-	Reserved
7	BMS_AKTIV	Active identification: The value of this bit must always be 1. If this is not the case, there is a fatal error in the software of the bus master.
8	BMS_SNE	One or more slaves are not accessible via the bus.
9	BMS_SKF	One or more slaves indicate configuration errors.
10	BMS_DPS	One or more slaves indicate static diagnostics.
11	BMS_EXD	One or more slaves indicate extended diagnostics.
12	BMS_SNB	One or more slaves are not ready for cyclic data exchange.
13	BMS_SF	One or more slaves indicate a miscellaneous error.
14	-	Reserved
15	-	Reserved

Fig.6-170: Status coding in "BmState"

Bus Master Error Word

The "BmError" bus master error word indicates fatal errors rendering operation at the field bus impossible. Each set bit (TRUE) in "BmError" represents an error:

Bit	Error[2]	Description
0	IL_BMF_HW_ERR	Hardware error
1	IL_BMF_MPS_ERR	Master parameter set (field bus configuration file) is missing or faulty

[2] *"IL_BMF_OK" indicates that there is no error*

Bit	Error2)	Description
2	IL_BMF_BUS_ERR	Error at the field bus (e.g. short-circuit)
3	IL_BMF_SW_ERR	System error in the peripheral driver (i.e. the driver software has detected a fatal error)

Fig.6-171: Error coding in "BmError"

DP_SLAVELIST Structure

Brief Description

The "DP_SLAVELIST" bit list (DP_BITLIST) has a defined length of 16 bytes (128 bits).

Program:

```
TYPE  DP_SLAVELIST : ARRAY  [0..15] OF BYTE; END_TYPE
```

Bit List Coding

Each bit of the bit list is assigned to a bus address of the slave (Profibus: FDL address). For example, the lowest-order bit in the first array element (ARRAY[0]) is assigned to the Profibus user with address 0:

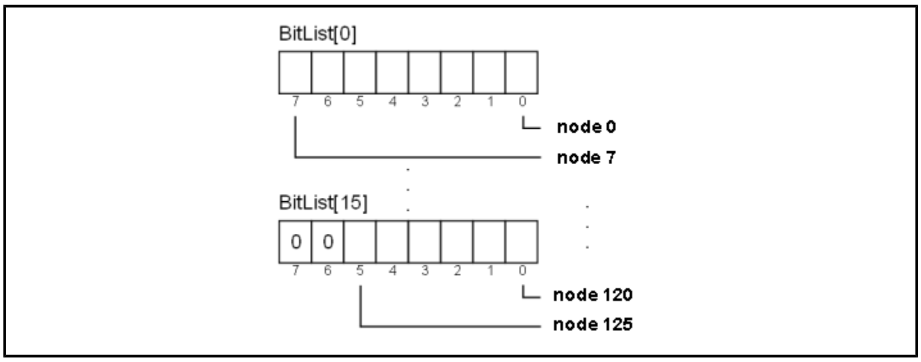


Fig.6-172: Bit list coding

DP_DEVICELIST Structure

Brief Description

This data type comprises information with regard to a DP master:

Example

Program:

```
TYPE  DP_DEVICELIST:
 STRUCT
 bMasterAdr         :  BYTE;  (* master addressing *)
 bMasterBusAdr      :  BYTE;  (* master bus address *)
 wMasterError       :  WORD;  (* cf. IL_DPDevInfoGet *)
 wMasterState       :  WORD;  (* cf. IL_DPDevInfoGet *)
 udBaudrate         :  UDINT; (* cf. IL_DPBaudrateGet *)
 dFirmwareVersion   :  DINT;  (* driver firmware version *)
 dHardwareVersion   :  DINT;  (* hardware version *)
 dAddInfo1          :  DINT;  (* 3S module identification *)
 dAddInfo2          :  DINT;  (* res *)
 END_STRUCT
END_TYPE

TYPE  tDP_MasterList: ARRAY [0..5] of tDP_MasterListTyp;
  (* list for 6 masters *)
END_TYPE
```

Note: Any possible onboard slave is also included in this list.

DP_MASTERINFO Structure

Brief Description

This data type comprises information about the state of the bus master.

Example

Program:

```
TYPE  DP_MASTERINFO: (*DP_DEVICEINFO:*)
 STRUCT
 BmState    :  WORD;      (* bus master status *)
 BmError    :  WORD;      (* bus master error word *)
 END_STRUCT
END_TYPE
```

6.11.4 Selecting the DP Master

Addressing

DP masters are distinguished according to their order in the DP configuration.

0 .. n(5) Number of the DP master in ascending configuration order. Only DP masters are counted.

Application

The functions and function blocks having to access a certain instance of a master have the type BYTE "Master" input parameter assigned to them.

The functions and function blocks complying with Profibus Guideline 2182 have an "ID" DWORD parameter which is interpreted as slot handle. A byte for selecting the master is reserved therein. The DP_SLOT function can be used to generate the ID parameter.

The functions and function blocks using the "Ident" DWORD parameter can generate this parameter by means of the IL_DPIdent function.

Byte	Contents	Description
0	MASTER	ID of the DP system: DP master identification (or onboard slave identification)
1	SEGMENT	Number of the DP segment (0)
2	STATION	Number of the DP slave (bus address)
3	SLOT	Number of the slot within the slave

Fig.6-173: Slot handle: "ID" parameter

6.11.5 Version_RIL_ProfibusDP_02_01V00

Brief Description

To ensure that the firmware version is compatible with the library, a version ID is carried along. If the names of the version functions fail to be equal, downloading of the application program will be rejected.

Interface Description

Version_RIL_ProfibusDP_02_01V00

6.11.6 IL_DPIdent

Brief Description

This function assembles an ident handle from various components.

Library	Area
RIL_ProfibusDP_02.lib	RIL DPV1-Services

Fig.6-174: IL_DPIdent library assignment

Interface Description

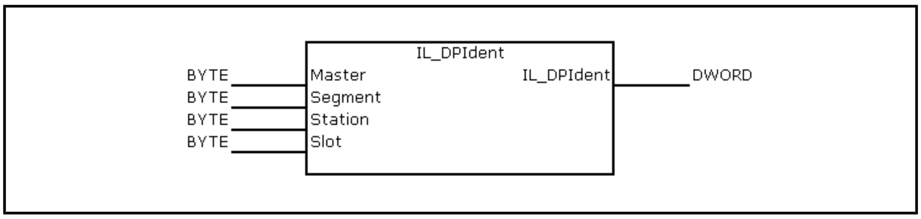


Fig.6-175: IL_DPIdent structure

	Name	Type	Description
VAR_INPUT	Master	BYTE	DP master identification (see above)
	Segment	BYTE	Number of the DP segment (0)

	Name	Type	Description
	Station	BYTE	Number of the DP slave (bus address): If the command is to apply for only one special slave, then the bus address of the slave must be entered here (0..125). Applicable to IL_DPSycFr only: However, if the command is to be entered for all slaves of a group, the global address (= 127) must be entered here.
	Slot	BYTE	Number of the slot within the slave (according to the slave specification) (value range: 0...254).
Function value		DWORD	Ident handle

Fig.6-176: IL_DPIdent interface

Functional Description

The 4 byte values are used to form the "Ident" DWORD. This DWORD is required as an input parameter for the following function blocks.

6.11.7 IL_DPV1Read

Brief Description

The "IL_DPV1Read" function block is used for DPV1 read access. Data exchange on the Profibus DP is acyclic. A pointer (POINTER) must be addressed to define a target area for the process data to be read.

Library	Area
RIL_ProfibusDP_02.lib	RIL DPV1-Services

Fig.6-177: IL_DPV1Read library assignment

Interface Description

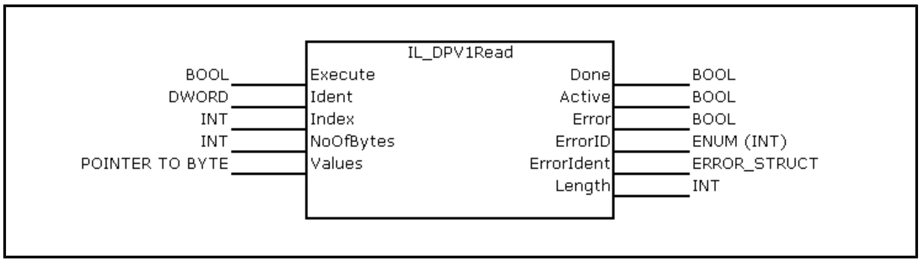


Fig.6-178: IL_DPV1Read structure

	Name	Type	Description
VAR_INPUT	Execute	BOOL	Function activation; interruption of an activated function block is not possible.
	Ident	DWORD	Ident handle (see IL_DPIdent function)
	Index	INT	Process data index (field number)
	NoOfBytes	INT	Maximum length of the data to be read; number of bytes available on the "Values" pointer
	Values	POINTER TO BYTE	Pointer to the data buffer for the target data
VAR_OUTPUT	Done	BOOL	Done message (successful)
	Active	BOOL	Activity display
	Error	BOOL	Done message (unsuccessful)
	ErrorID	ERROR_CODE	See chapter "ErrorID" on page 171

	Name	Type	Description
	ErrorIdent	ERROR_STRUCT	Error structure from PB_DP_TABLE ErrorTable
	Length	INT	Length of the data read in bytes

Fig.6-179: IL_DPV1Read interface

Signal Time Diagram

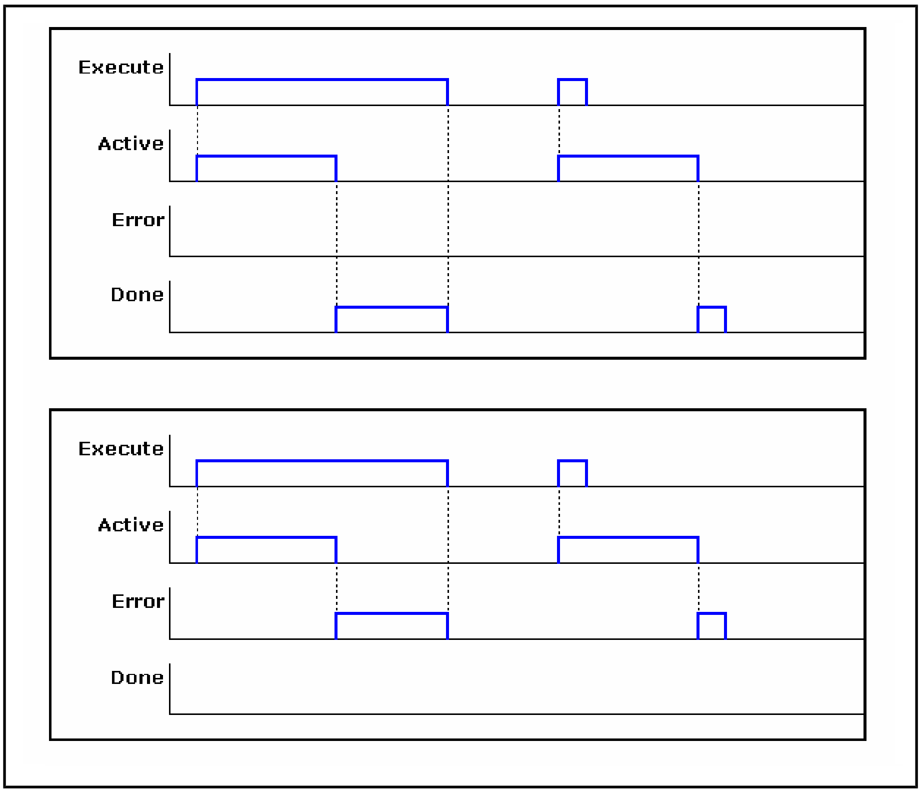


Fig.6-180: IL_DPV1Read signal time diagram

Functional Description

The master (class 1) accesses a DP-V1 slave. It reads the data record of the slave. This data record is addressed through the slave address, the slot and the index. Addressing with slot and index and data interpretation are slave-specific and can be found in the documentation of the particular slave. The function execution time is dependent on the bus load and the set baud rate, among other factors.

This function is only available for slaves participating in the DP bus cycle.

6.11.8 IL_DPV1Write

Brief Description

The "IL_DPV1Write" function block is used for DPV1 write access. Data exchange on the Profibus DP is acyclic. A pointer (POINTER) must be addressed to deliver the process data to be written.

Library	Area
RIL_ProfibusDP.lib	RIL DPV1-Services

Fig.6-181: IL_DPV1Write library assignment

Interface Description

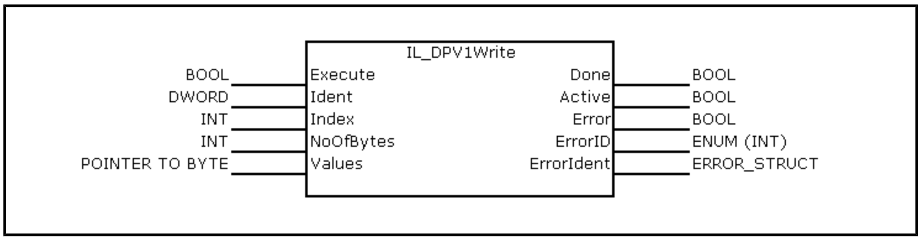


Fig.6-182: IL_DPV1Write structure

	Name	Type	Description
VAR_INPUT	Execute	BOOL	Function activation; interruption of an activated function block is not possible.
	Ident	DWORD	Ident handle (see IL_DPIdent function)
	Index	INT	Process data index (field number)
	NoOfBytes	INT	Maximum length of the data to be read; number of bytes available on the "Values" pointer
	Values	POINTER TO BYTE	Pointer to the data buffer for the data
VAR_OUTPUT	Done	BOOL	Done message (successful)
	Active	BOOL	Activity display
	Error	BOOL	Done message (unsuccessful)
	ErrorID	ERROR_CODE	See chapter "ErrorID" on page 171
	ErrorIdent	ERROR_STRUCT	Error structure from PB_DP_TABLE ErrorTable

Fig.6-183: Interface signals: IL_DPV1Write

Signal Time Diagram

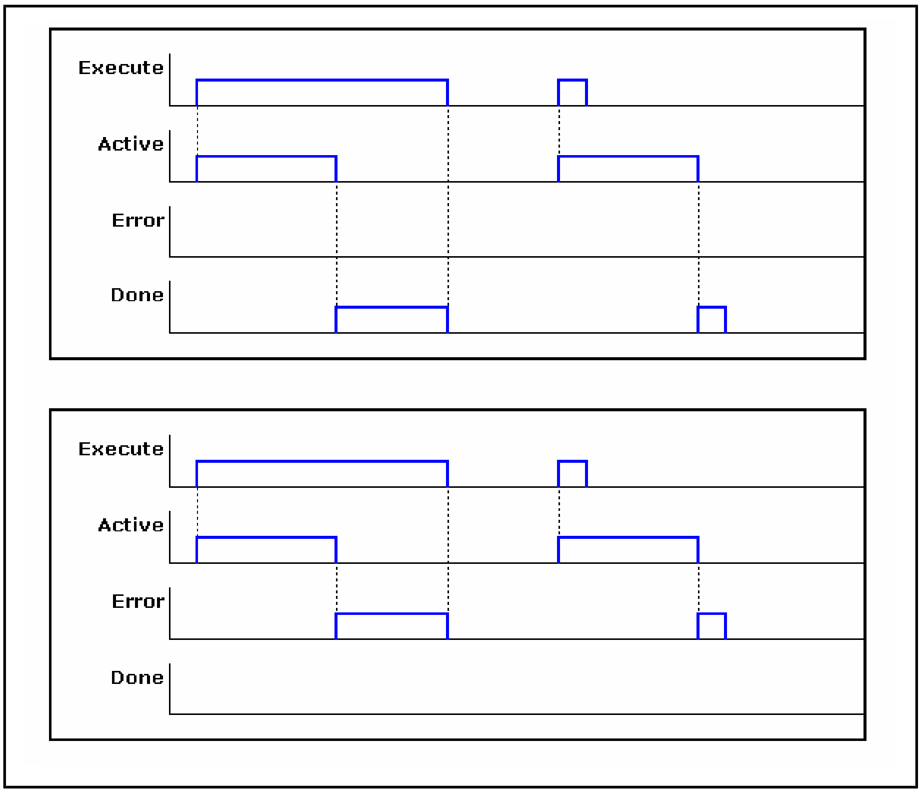


Fig.6-184: IL_DPV1Write signal time diagram

Functional Description

The master (class 1) accesses a DP-V1 slave. It reads the data record of the slave. This data record is addressed through the slave address, the slot and the index. Addressing with slot and index and data interpretation are slave-specific and can be found in the documentation of the particular slave. The function execution time is dependent on the bus load and the set baud rate, among other factors.

This function is only available for slaves participating in the DP bus cycle.

6.11.9 IL_DPReadDiag

Brief Description

The "IL_DPReadDiag" function block is used by the DP master (DPM1) to read the diagnostic data of a slave. The data buffer of the diagnostic data must be provided to address it via a pointer (POINTER).

Library	Area
RIL_ProfibusDP.lib	RIL DPV1-Services

Fig.6-185: IL_DPReadDiag library assignment

Interface Description

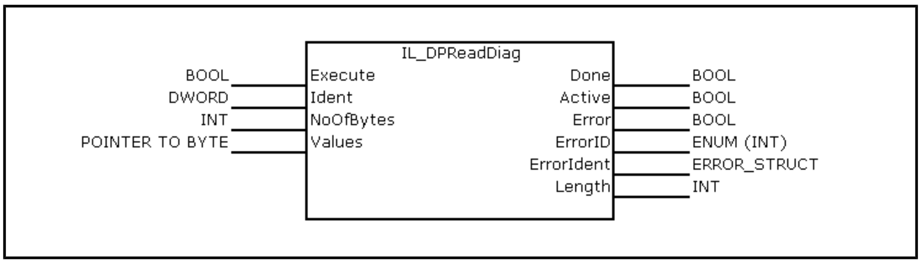


Fig.6-186: IL_DPReadDiag structure

	Name	Type	Description
VAR_INPUT	Execute	BOOL	Function activation; interruption of an activated function block is not possible.
	Ident	DWORD	Ident handle (see IL_DPIdent function)
	NoOfBytes	INT	Maximum length of the data to be read; number of bytes available on the "Values" pointer
	Values	POINTER TO BYTE	Pointer to data buffers of slave diagnostics data according to Profibus DP standard
VAR_OUTPUT	Done	BOOL	Done message (successful)
	Active	BOOL	Activity display
	Error	BOOL	Done message (unsuccessful)
	ErrorID	ERROR_CODE	See chapter "ErrorID" on page 171
	ErrorIdent	ERROR_STRUCT	Error structure from PB_DP_TABLE ErrorTable
	Length	INT	Length of the diagnostic data in bytes

Fig.6-187: IL_DPReadDiag interface

Signal Time Diagram

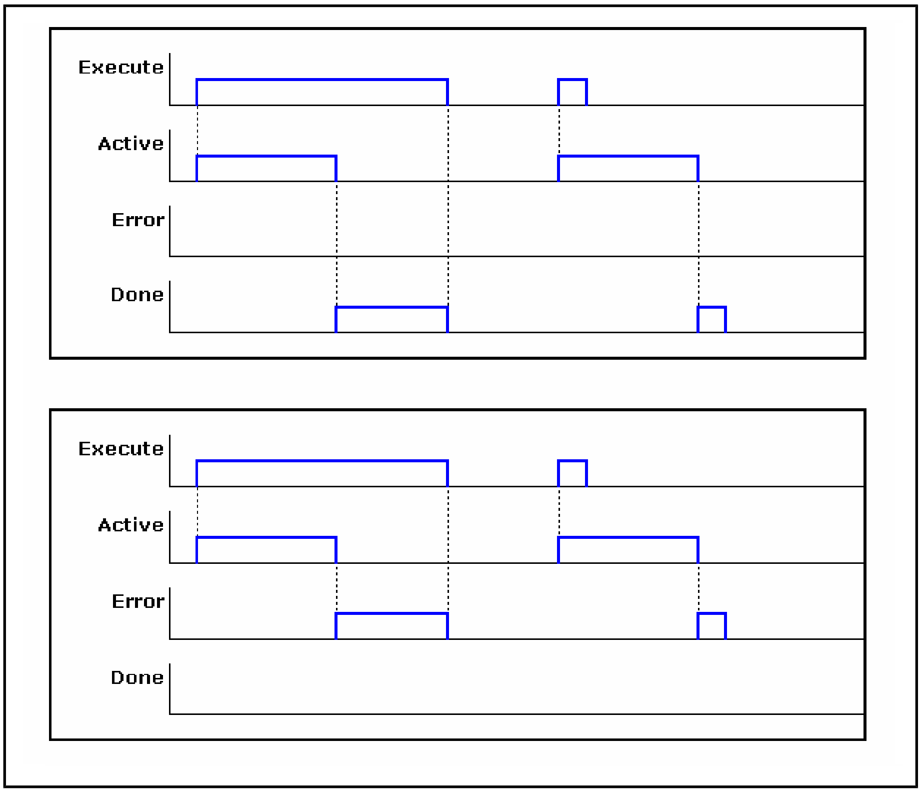


Fig.6-188: IL_DPReadDiag signal time diagram

Functional Description

The master reads the diagnostic information from the slaves and saves it in relation to the particular slave. Using request bits, the slave triggers the diagnostic request of the master in cyclic telegram traffic. The slave is responsible for the content of the diagnostic data. The present function does not trigger any telegram traffic at the Profibus. It just accesses the diagnostic information provided by the master.

6.11.10 IL_DPSycFr

Brief Description

The "IL_DPSycFr" function block can be used to implement control commands for synchronization of inputs and outputs.

Profibus DP provides the possibility that a master sends what is called a "global control telegram" to a group of slaves. The global control telegram contains a control command.

- Using the **freeze** control command, all slaves of the addressed group are storing the current **input data** at the same time (synchronize inputs).
- Using the **sync** control command, all slaves of the addressed group are applying the current **output data** at the same time (synchronize outputs).

The sync-freeze functionality is supported only with the onboard master.

Library	Area
RIL_ProfibusDP.lib	RIL DPV1-Services

Fig.6-189: IL_DPSycF library assignment

Interface Description

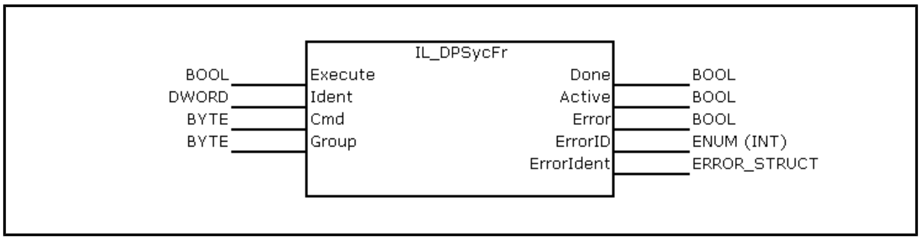


Fig.6-190: DPSycFr structure

	Name	Type	Description
VAR_INPUT	Execute	BOOL	Function activation; interruption of an activated function block is not possible.
	Ident	DWORD	Ident handle; see IL_DPIdent (slot is irrelevant and should be 0)
	Cmd	BYTE	Control command (see above)
	Group	BYTE	Selects one or more groups to which the command refers. Each bit is assigned to a group.
VAR_OUTPUT	Done	BOOL	Done message (successful)
	Active	BOOL	Activity display
	Error	BOOL	Done message (unsuccessful)
	ErrorID	ERROR_CODE	See chapter "ErrorID" on page 171
	ErrorIdent	ERROR_STRUCT	Error structure from PB_DP_TABLE ErrorTable

Fig.6-191: IL_DPSycF interface

Signal Time Diagram

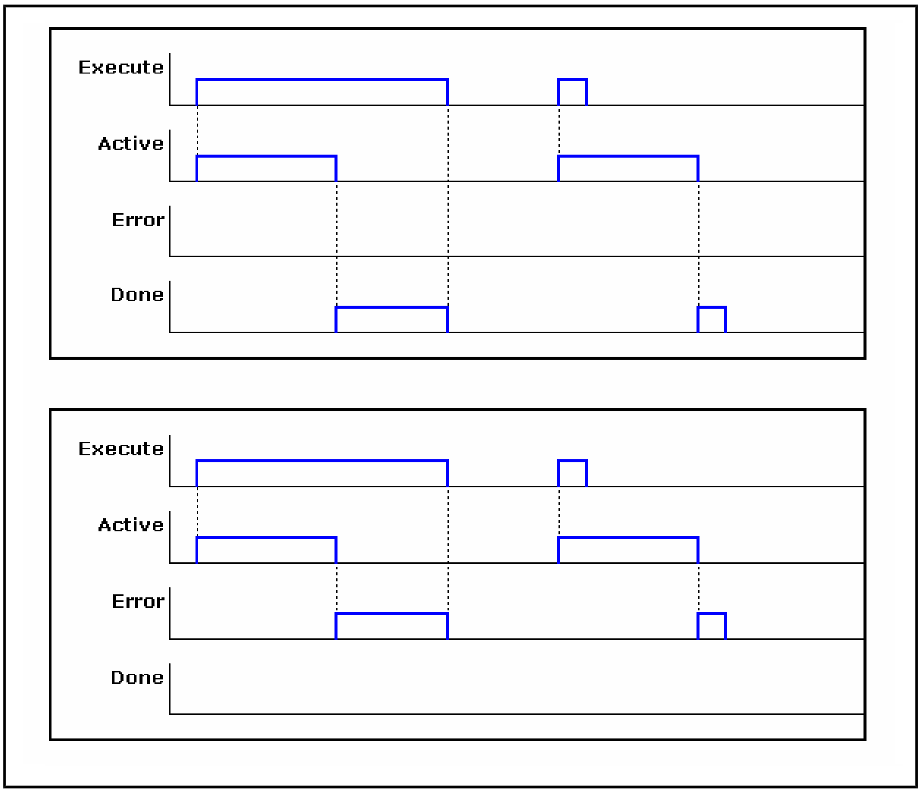


Fig.6-192: IL_DPSycFr signal time diagram

Functional Description

Example: To address all slaves of a group, the value of the ID is 16#007f0000.

The ID is formed from the various components by means of the "IL_DPIdent" function.

Outputs can only be synchronized if all slaves received the current output data before having received the sync command.

To achieve this, call the "IL_DPSycFr" function block from the same PLC task from which the output data of the slaves is written.

In a PLC task, the output data for the synchronized slaves is written first. Then, the SYNC command is started with "IL_DPSycF". As long as the Sync command is not completed yet (BUSY), the output data must not be modified.

If sync or freeze is used in the IndraWorks project explorer, an assignment of the groups at the master and the corresponding slaves must be set. For more information, please refer to the IndraWorks documentation or online help.

Possible control commands:

IL_DP_CMD_UNFREEZE	16#04	Freeze mode release
IL_DP_CMD_FREEZE	16#08	command Freeze
IL_DP_CMD_UNSYNC	16#10	Sync mode release
IL_DP_CMD_SYNC	16#20	command Sync

Fig.6-193: Possible IL_DPSycF control commands

6.11.11 IL_DPDeviceListGet

Brief Description

The "IL_DPDeviceListGet" function block determines a list of the currently available DP masters.

Library	Area
RIL_ProfibusDP.lib	RIL DPV1-Services

Fig.6-194: IL_DPDeviceListGet library assignment

Interface Description

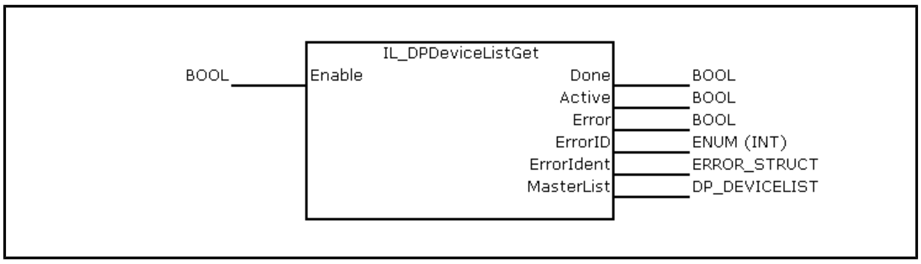


Fig.6-195: IL_DPDeviceListGet structure

	Name	Type	Description
VAR_INPUT	Enable	BOOL	Function release
VAR_OUTPUT	Done	BOOL	Done message (successful)
	Active	BOOL	Activity display
	Error	BOOL	Done message (unsuccessful)
	ErrorID	ERROR_CODE	See chapter "ErrorID" on page 171
	ErrorIdent	ERROR_STRUCT	Error structure from PB_DP_TABLE ErrorTable
	MasterList	DP_DEVICELIST	List of available DP masters

Fig.6-196: IL_DPDeviceListGet interface

Signal Time Diagram

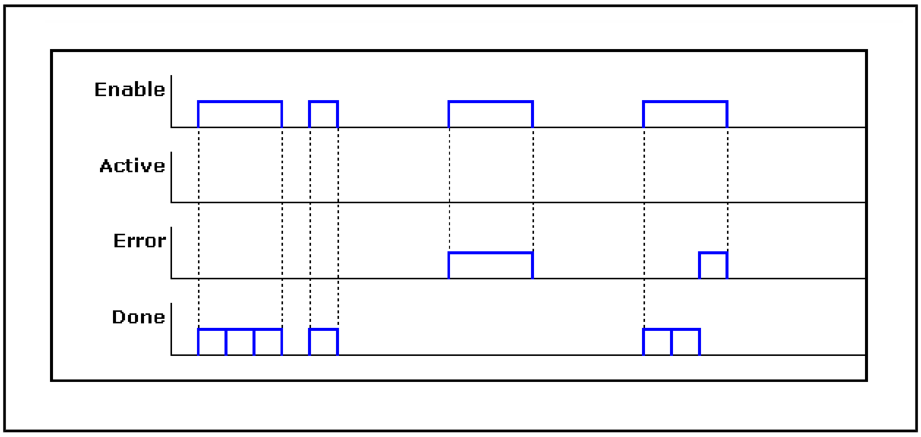


Fig.6-197: IL_DPDeviceListGet signal time diagram

Functional Description

This function is used to display the master instances active and addressable at the PLC. This also allows the user to monitor whether the configuration loaded is appropriate and whether it has been detected correctly.

6.11.12 IL_DPBaudrateGet

Brief Description

The "IL_DPBaudrateGet" function block determines the baud rate of the connected field bus. The baud rate is specified in bits per second.

Library	Area
RIL_ProfibusDP.lib	RIL DPV1-Services

Fig.6-198: Library assignment

Interface Description

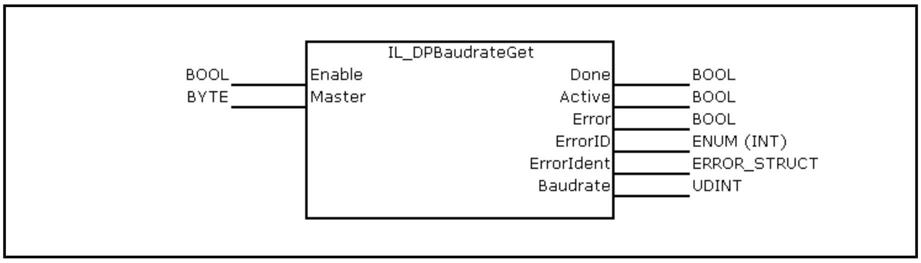


Fig.6-199: IL_DPBaudrateGet structure

	Name	Type	Description
VAR_INPUT	Enable	BOOL	Function release
	Master	BYTE	MasterID (see above)
VAR_OUTPUT	Done	BOOL	Done message (successful)
	Active	BOOL	Activity display
	Error	BOOL	Done message (unsuccessful)
	ErrorID	ERROR_CODE	See chapter "ErrorID" on page 171
	ErrorIdent	ERROR_STRUCT	Error structure from PB_DP_TABLE ErrorTable
	Baud rate	UDINT	Baud rate in 1/s

Fig.6-200: IL_DPBaudrateGet interface

Signal Time Diagram

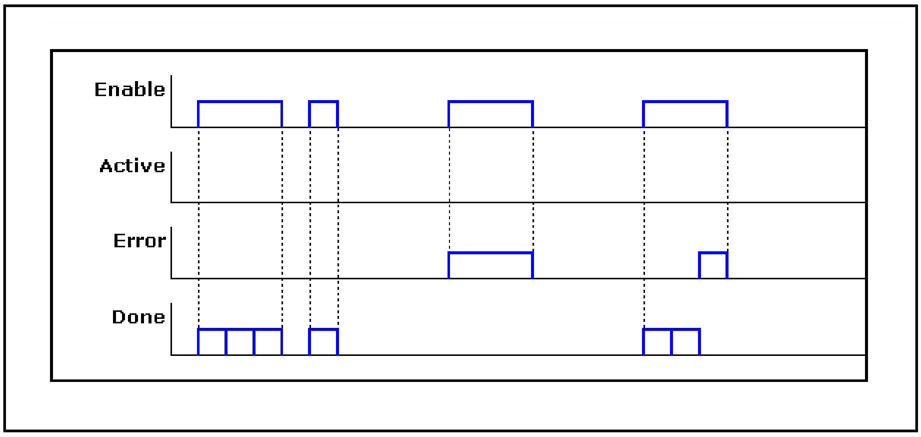


Fig.6-201: IL_DPBaudrateGet signal time diagram

Functional Description

This function is used to determine the operating state of the PB master that has been addressed. "BmState" and "BmError" must be encoded with 1.3.2 and 1.3.3.

6.11.13 IL_DPDevInfoGet

Brief Description

The "IL_DPDevInfoGet" is used to obtain information about the operating state of the particular PB device.

Library	Area
RIL_ProfibusDP.lib	RIL DPV1-Services

Fig.6-202: IL_DPDevInfoGet library assignment

Interface Description

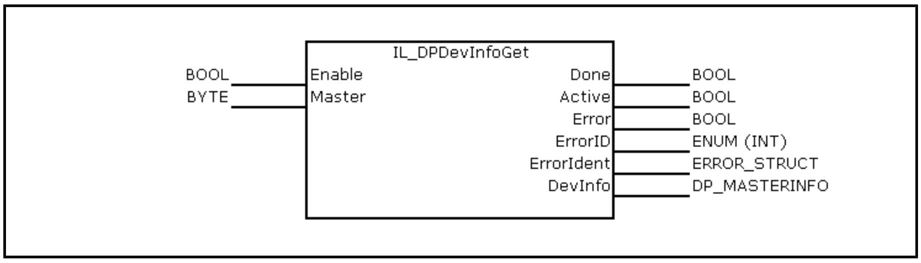


Fig.6-203: IL_DPDevInfoGet structure

	Name	Type	Description
VAR_INPUT	Enable	BOOL	Function release
	Master	BYTE	MasterID (see above)
VAR_OUTPUT	Done	BOOL	Done message (successful)
	Active	BOOL	Activity display
	Error	BOOL	Done message (unsuccessful)
	ErrorID	ERROR_CODE	See chapter "ErrorID" on page 171
	ErrorIdent	ERROR_STRUCT	Error structure from PB_DP_TABLE ErrorTable
	DevInfo	DP_MASTERINFO	DP_MASTERINFO

Fig.6-204: IL_DPDevInfoGet interface

Signal Time Diagram

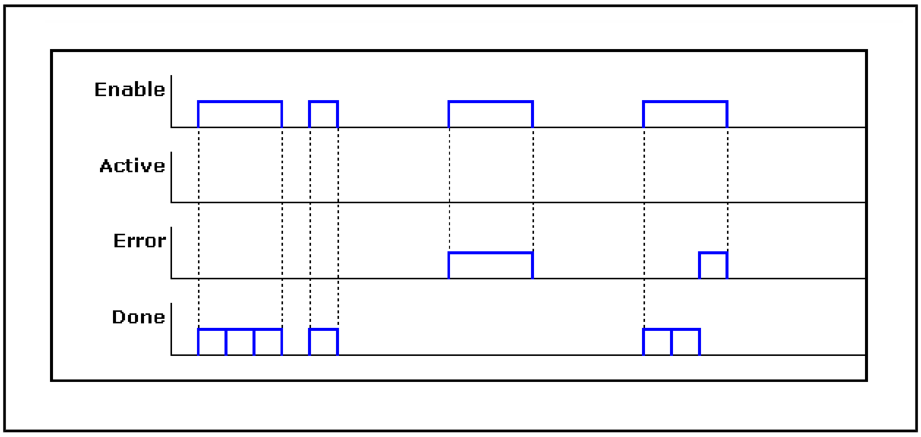


Fig.6-205: IL_DPDevInfoGet signal time diagram

6.11.14 IL_DPPrjSlaveGet

Brief Description

The "IL_DPPrjSlaveListGet" function block supplies the list of projected slaves. The list contains all slaves available in the master configuration file.

Library	Area
RIL_ProfibusDP.lib	RIL DPV1-Services

Fig.6-206: IL_DPDevInfoGet library assignment

Interface Description

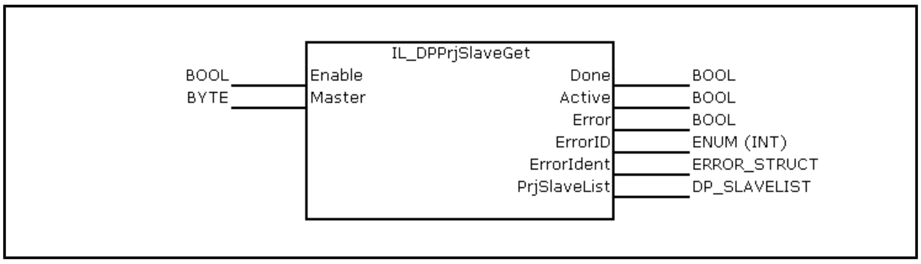


Fig.6-207: IL_DPPrjSlaveGet structure

	Name	Type	Description
VAR_INPUT	Enable	BOOL	Function release
	Master	BYTE	MasterID (see above)
VAR_OUTPUT	Done	BOOL	Done message (successful)
	Active	BOOL	Activity display
	Error	BOOL	Done message (unsuccessful)
	ErrorID	ERROR_CODE	See chapter "ErrorID" on page 171
	ErrorIdent	ERROR_STRUCT	Error structure from PB_DP_TABLE ErrorTable
	PrjSlaveList	DP_SLAVELIST	Bit list with set bit for every projected slave

Fig.6-208: IL_DPDevInfoGet interface

Signal Time Diagram

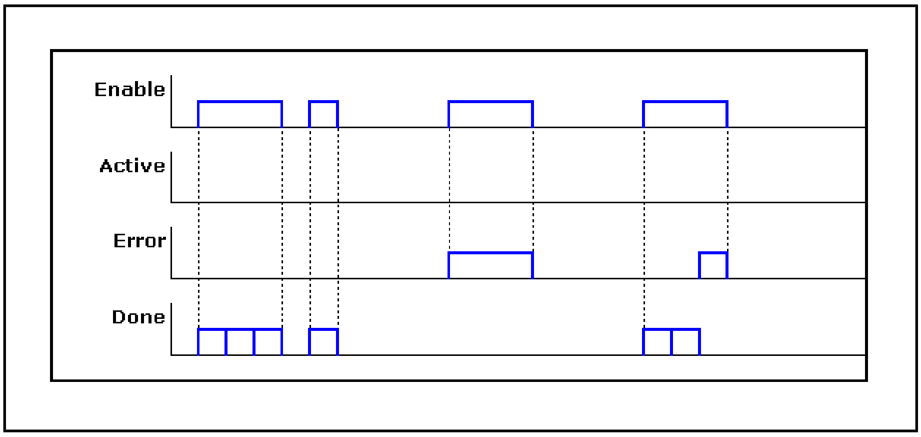


Fig.6-209: IL_DPPrjSlaveGet signal time diagram

Functional Description

The bit list is used to set a bit for each projected slave of a master. The function is based on the configuration data available.

6.11.15 IL_DPSlaveDiagListGet

Brief Description

The "IL_DPSlaveDiagListGet" supplies the list of slaves indicating diagnostics.

Library	Area
RIL_ProfibusDP.lib	RIL DPV1-Services

Fig.6-210: IL_DPSlaveDiagListGet library assignment

Interface Description

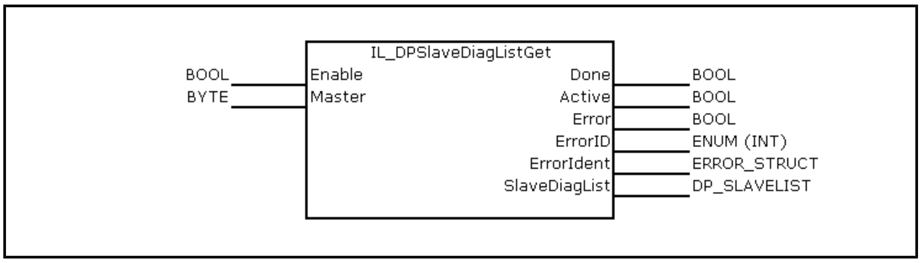


Fig.6-211: IL_DPSlaveDiagListGet structure

	Name	Type	Description
VAR_INPUT	Enable	BOOL	Function release
	Master	BYTE	MasterID (see above)
VAR_OUTPUT	Done	BOOL	Done message (successful)
	Active	BOOL	Activity display
	Error	BOOL	Done message (unsuccessful)
	ErrorID	ERROR_CODE	See chapter "ErrorID" on page 171
	ErrorIdent	ERROR_STRUCT	Error structure from PB_DP_TABLE ErrorTable
	SlaveDiagList	DP_SLAVELIST	Bit list with set bit for every slave indicating diagnostics

Fig.6-212: IL_DPSlaveDiagListGet interface

Signal Time Diagram

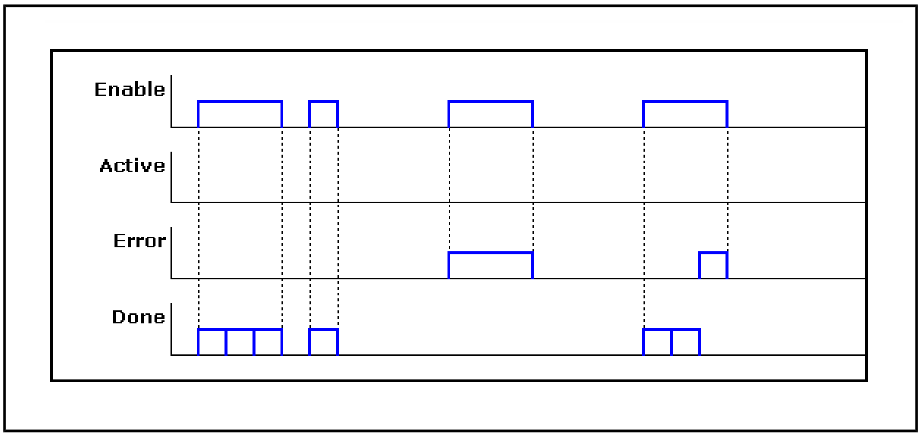


Fig.6-213: IL_DPSlaveDiagListGet signal time diagram

Functional Description

The "SlaveDiagList" is used to set a bit for each slave indicating diagnostics. This allows reading selected diagnostics. This function block does not trigger any DP services. It is used to evaluate data provided in the master.

6.11.16 Error Identifications

ErrorID

Enumerator	Code	Description
NONE_ERROR	16#0000	No error code available
INPUT_INVALID_ERROR	16#0001	Invlaid input assignment
COMMUNICATION_ERROR	16#0002	Communication error
RESOURCE_ERROR	16#0003	Source not available
ACCESS_ERROR	16#0004	Faulty or invalid access to data
STATE_MACHINE_ERROR	16#0005	Invalid state machine value
INPUT_RANGE_ERROR	16#0006	The value of one or more inputs is outside of the defined limits
CALCULATION_ERROR	16#0007	Calculation error
DEVICE_ERROR	16#0008	Drive error
OTHER_ERROR	16#7FFE	Undefined error (assignment to any of the other IDs not possible)
SYSTEM_ERROR	16#7FFF	System error

Fig.6-214: Possible ErrorIDs

ErrorIdent

Structure

ErrorIdent is a structure comprising three elements. Its default value is 0.

- ErrorTable : PB_DP_TABLE (16#0130) Profibus DP error
- ErrorAdditional1: see below
- ErrorAdditional1: see below

ErrorTable

Enumerator	Code	Description
NO_TABLE_USED	16#0000	No table assigned
SERCOS_TABLE	16#0010	Sercos error (error in communication via Sercos)
MLD_TABLE	16#0020	Drive error (error of PLC FBs of the drive PLC)
MLC_TABLE	16#0030	MLC error (error of PLC FBs of the MLC control)
MTX_TABLE	16#0040	MTX error (error of PLC FBs of the MTX control)
MLP_TABLE	16#0050	MLP error (error of PLC FBs of the PC-based control)
PLC_TABLE	16#0060	PLC error
INDRV_TABLE	16#0070	IndraDrive error (IndraDrive signals error via PLC FB)
DIAX_TABLE	16#0080	Diax error (Diax drive signals error via PLC FB)
ECO_TABLE	16#0090	EcoDrive error (EcoDrive signals error via PLC FB)
PB_DP_TABLE	16#0130	Profibus DP error (Profibus DP signals error via PLC FB)
DEVICENET_TABLE	16#0140	DeviceNet error (DeviceNet signals error via PLC FB)
ETHERNET_TABLE	16#0150	Ethernet error (Ethernet signals error via PLC FB)
ETHERNET_IP_TABLE	16#0151	EthernetIP error (EthernetIP signals error via PLC FB)
INTERBUS_TABLE	16#0160	Interbus error (Interbus signals error via PLC FB)
F_RELATED_TABLE	16#0170	System-crossing error messages (e.g. from technology FBs)
USER1_TABLE	16#1000	Can be used as desired
USER2_TABLE	16#1001	Can be used as desired
USER3_TABLE	16#1002	Can be used as desired
USER4_TABLE	16#1003	Can be used as desired
USER5_TABLE	16#1004	Can be used as desired
USER6_TABLE	16#1005	Can be used as desired
USER7_TABLE	16#1006	Can be used as desired
USER8_TABLE	16#1007	Can be used as desired

Enumerator	Code	Description
USER9_TABLE	16#1008	Can be used as desired
USER10_TABLE	16#1009	Can be used as desired

Fig.6-215: "ErrorTable" (ENUM)

ErrorAdditional1

Structure

ErrorAdditional1 is used for a superordinate distinction by error source. There are the following error sources:

16#0001: Onboard Profibus Device

16#0002: FunctionModule (FM) Profibus Device

16#0100: Function not supported

16#0101: Device(Master) not found

ErrorAdditional2

Structure

ErrorAdditional2 also comprises 4 bytes for "Onboard Profibus Device".

The meaning of the bytes is as follows:

Byte no.	Meaning	Description
Byte 3	Error_Source	Distinctionn by error origin: 16#00 Profibus (slave) 16#10 Masterstack 16#20 Profibus FDL layer
Byte 2	Error_Code_DP Error_Code_OB Error_Code_FDL	For Error_Source = 16#00 For Error_Source = 16#10 For Error_Source = 16#20
Byte 1	AddInfo_1	Reserved
Byte 0	AddInfo_2	Reserved

Fig.6-216: Onboard Profibus coding

ErrorAdditional2 also comprises 4 bytes for "FunctionModule (FM) Profibus Device".

The meaning of the bytes is as follows:

Byte no.	Meaning	Description
Byte 3	FM_Error_Code	FM master error code
Byte 2	Error_Code_DP	See tables below
Byte 1	Error_Code_1	DP-user-specific
Byte 0	AddInfo_1	Reserved

Fig.6-217: Function module coding

FM_Error_Code

The following error description corresponds to the error response definitions of the Hilscher Profibus DP master.

FM_Error_Code	Error
16#02	The slave does not provide any memory or buffer for this service.

16#03	The slave does not support any DPV1 services.
16#09	The slave did not transmit any data.
16#11	The slave did not respond/is not applied to the bus.
16#12	The DP master is not applied to the ProfiBus (check cabling)
16#19	The slave does not comply with DPV1.
16#36	The slave rejected the access. Evaluate Error_Code_DP!
16#81	DPV1 is not configured on the master.
16#82	The slave did not respond with plausible parameters.
16#83	Another service already in progress; parallel services not allowed.
16#84	Data capacity exceeds configured size.
16#85	Wrong parameter in request.
16#9a	Unknown command
16#F0	Invalid state

Fig.6-218: ErrorCode function module

Error_Code_OB

Error_Code_OB	Error
16#11	Invalid order parameters
16#23	RequestList full
16#25	SemTake error
16#31	Unallowed call
16#32	Invalid call parameters
16#33	Invalid data length
16#34	Faulty call state
16#35	Slave not configured
16#36	Slave configured but not in cyclic mode
16#61	DPV1 request to non-DPV! slave
16#62	The slave does not respond within timeout.
16#63	DPV1 telegram format error
16#64	Order was withdrawn.
16#65	Pertinent RQB not found.
16#66	Invalid parameter
16#67	Unknown AMPRO2 opcode

Fig.6-219: Onboard ErrorCode

Error_Code_FDL

Error_Code_FDL	Error
16#61	FE: format error in a request APDU
16#62	NI: service not implemented
16#63	AD: access denied
16#64	EA: area to large (up/download)
16#65	LE: data block length too large (up/download)
16#66	RE: format error in a request APDU
16#67	IP: invalid parameter
16#68	SC: sequence conflict
16#69	SE: sequence error
16#6A	NE: area non-existent
16#60	No slave found
16#6B	DI: data incomplete
16#6C	NC: master parameter set not compatible

Fig.6-220: Field bus data link layers (FDL) ErrorCode

Error_Code_DP

Structure

The meaning of "Error_Code_DP" corresponds to that of "Error_Code_1" described in the DPV1 standard. Bits 4..7 of the error byte constitute the "Error_Class", while Bits 0..3 constitute the "Error_Code".

7	6	5	4	3	2	1	0	Meaning
				————				Error Code
————								Error Class

Fig.6-221: ErrorCode DP

Error_Class	Meaning	Error_Code
0 to 9	reserved[3]	
10	Application	0 = read error 1 = write error 2 = module failure 3 to 7 = reserved[4] 8 = version conflict 9 = feature not supported 10 to 15 = user specific

[3] *reserved: values are intended to be passed on to the user as they are.*

[4] *reserved: values are intended to be passed on to the user as they are.*

Error_Class	Meaning	Error_Code
11	Access	0 = invalid index 1 = write length error 2 = invalid slot 3 = type conflict 4 = invalid area 5 = state conflict 6 = access denied 7 = invalid range 8 = invalid parameter 9 = invalid type 10 to 15 = user specific
12	Resource	0 = read constrain conflict 1 = write constrain conflict 2 = resource busy 3 = resource unavailable 4 to 7 = reserved[5)] 8 to 15 = user specific
13 to 15	User-specific	

Fig.6-222: ErrorCode DP

Additional Iínfo: user(slave)-specific

6.12 RIL_Utilities

6.12.1 Overview

Designation	Type	Description
Version_RIL_Utilities_01V*	FNC	Version management of the RIL_Utilities.lib.
IL_HighResTimeTick	FNC	Read the high-resolution time tick of the system.
IL_HighResTimeDiff	FNC	Calculation of the time difference of two high-resolution time ticks of the system in microseconds.
IL_Date	FNC	Read the current system date.
IL_TimeOfDay	FNC	Read the current system time.
IL_DateAndTime	FNC	Read the current system date and the current system time (format according to IEC 61131-3).
IL_SysTime64	FB	Read the current system date and the current system time (in microseconds since 1970-01- 01).
IL_SysTimeDate	FB	Read the current system date and the current system time (in system format).
IL_ExtSysTimeDate	FB	Read the current system date and the current system time (in extended system format).

[5)] *reserved: values are intended to be passed on to the user as they are.*

Designation	Type	Description
IL_SysTime64ToSysTimeDate	FB	Format conversion of system date and system time.
IL_SysTimeDateToSysTime64	FB	Format conversion of system date and system time.

Fig.6-223: Overview of the function blocks and functions contained in RIL_Utilities.lib

6.12.2 Data Types

The RIL_Utilities.lib library does not contain any separate data types.

6.12.3 Global Variables

The RIL_Utilities.lib library does not contain any separate global variables.

6.12.4 Version_RIL_Utilities_01V*

Brief Description

The Version_RIL_Utilities_01V*[6] is used for RIL_Utilities.lib version control.

Interface Description

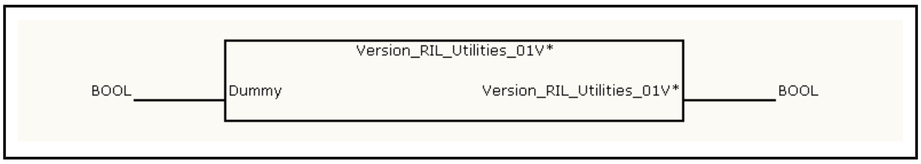


*Fig.6-224: Interface of: Version_RIL_Utilities_01V**

	Name	Type	Comment
VAR_INPUT	Dummy	BOOL	Dummy
Function value	Version_ RIL_Utilities_01V*	BOOL	Acknowledges that the library is valid

Fig.6-225: Version_RIL_Utilities_01V interface*

Specification

The "Version_RIL_Utilities_01V*" function restricts the use of "RIL_Utilities.lib" to the valid system as well as to a certain number of valid releases.

The "Version_RIL_Utilities_01V*" function also shows the current release of "RIL_Utilities.lib" and contains an overview of all modifications made up to that point.

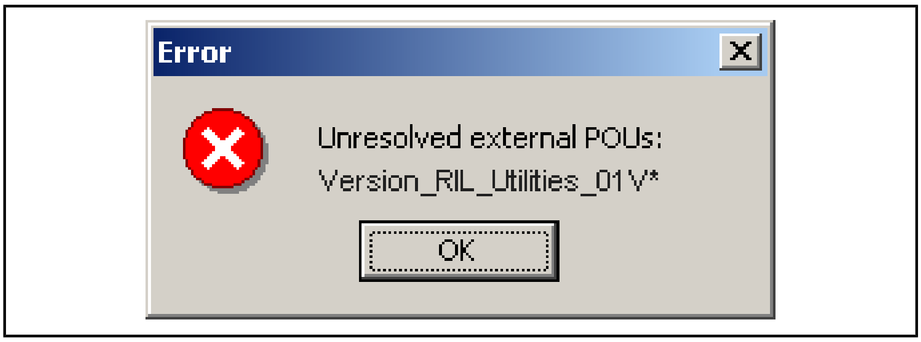
☞ "RIL_Utilities.lib" is available on several platforms (systems) in the "IndraLogic" programming system. Its functional range is adapted to the respective system.

Functional Description

If "RIL_Utilities.lib" is integrated in a project, the "Version_RIL_Utilities_01V*" function restricts the download of the whole project to the valid system as well as to a valid release, since the respective system function is not available on invalid systems or in invalid releases and can, therefore, not be addressed, see Message: Version check of the system failed.

☞ Verification of the system and the releases is active even if the "Version_RIL_Utilities_01V*" function is not used.

[6] *Release*



Error

Unresolved external POUs:
Version_RIL_Utilities_01V*

Fig.6-226: Message: Version check of the system failed

☞ In addition to the version function, further functions are indicated as not existing for which reason they cannot be integrated in the project.

If the "Version_RIL_Utilities_01V*" function can be addressed without errors and if it was used, it constantly returns TRUE as the return value.

☞ The "Version_RIL_Utilities_01V*" function has no functional meaning for parts of the library and does not need to be used in the project.

6.12.5 IL_HighResTimeTick

Brief Description The "IL_HighResTimeTick" function reads the high-resolution time tick of the system.

Interface Description

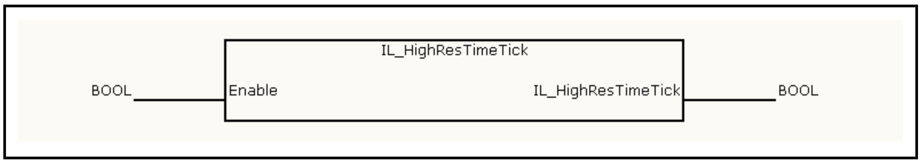


Fig.6-227: IL_HighResTimeTick

	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Enable function processing (cyclic, status-controlled)
Function value	IL_HighResTimeTick	UDINT	High-resolution time tick of the system

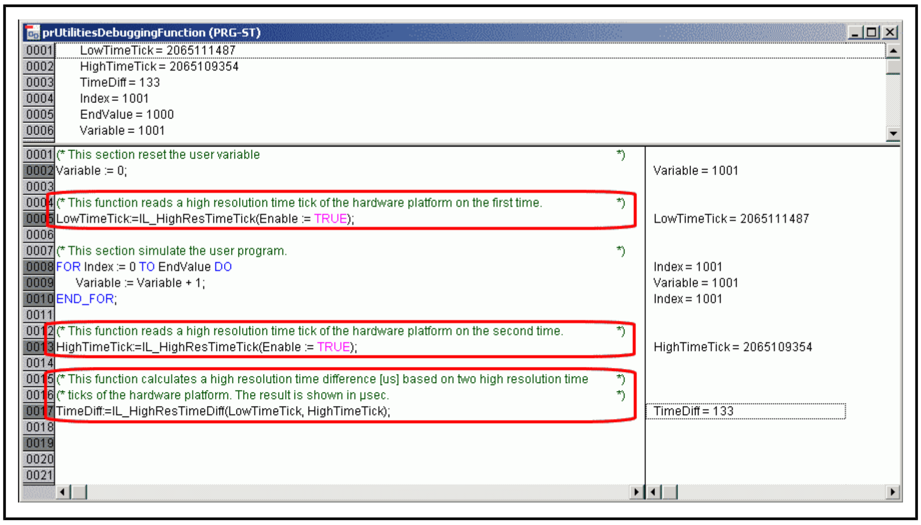
Fig.6-228: IL_HighResTimeTick interface

Specification The "IL_HighResTimeTick" and "IL_HighResTimeDiff" functions are used jointly to determine the runtime of a code segment, see Application example of the IL_HighResTimeTick function.

☞ The return value of the "IL_HighResTimeTick" function should not be used as time value due to the special system-specific time basis.

Functional Description After having been enabled for processing with "Enable", the "IL_HighResTimeTick" function cyclically reads the high-resolution time tick of the system.

Example



prUtilitiesDebuggingFunction (PRG-ST)

```
    LowTimeTick = 2065111487
    HighTimeTick = 2065109354
    TimeDiff = 133
    Index = 1001
    EndValue = 1000
    Variable = 1001
```

```
(* This section reset the user variable                                              *)
Variable := 0;                                                                          Variable = 1001

(* This function reads a high resolution time tick of the hardware platform on the first time.    *)
LowTimeTick:=IL_HighResTimeTick(Enable := TRUE);                                        LowTimeTick = 2065111487

(* This section simulate the user program.                                           *)
FOR Index := 0 TO EndValue DO                                                           Index = 1001
    Variable := Variable + 1;                                                           Variable = 1001
END_FOR;                                                                                Index = 1001

(* This function reads a high resolution time tick of the hardware platform on the second time.   *)
HighTimeTick:=IL_HighResTimeTick(Enable := TRUE);                                       HighTimeTick = 2065109354

(* This function calculates a high resolution time difference [us] based on two high resolution time  *)
(* ticks of the hardware platform. The result is shown in µsec.                      *)
TimeDiff:=IL_HighResTimeDiff(LowTimeTick, HighTimeTick);                                TimeDiff = 133
```

Fig.6-229: Application example of the IL_HighResTimeTick function

6.12.6 IL_HighResTimeDiff

Brief Description

The "IL_HighResTimeDiff" function is used to calculate the time difference between two high-resolution time ticks of the system in microseconds.

Interface Description

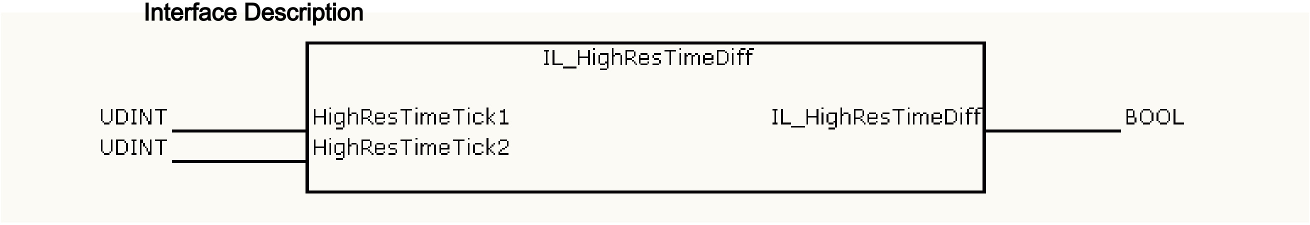


Fig.6-230: IL_HighResTimeDiff

	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Enable function processing (cyclic, status-controlled)
	HighResTimeTick1	UDINT	High-resolution time tick of the system before the code segment
	HighResTimeTick2	UDINT	High-resolution time tick of the system after the code segment
Function value	IL_HighResTimeDiff	UDINT	Time difference between two high-resolution time ticks of the system in [us]

Fig.6-231: IL_HighResTimeDiff interface

Specification

The "IL_HighResTimeDiff" function and the "IL_HighResTimeTick" function are used jointly to determine the runtime of a code segment, see IL_HighResTimeTick.

☞ Reading the high-resolution time tick requires approx. one to two microseconds depending on the system. If necessary, the user can add this time while determining the time difference.

Functional Description

After having been enabled for processing with "Enable", the "IL_HighResTimeDiff" function cyclically calculates the time difference between two high-resolution time ticks of the system in microseconds.

6.12.7 IL_Date

Brief Description The "IL_Date" function is used to the current system date.

Interface Description

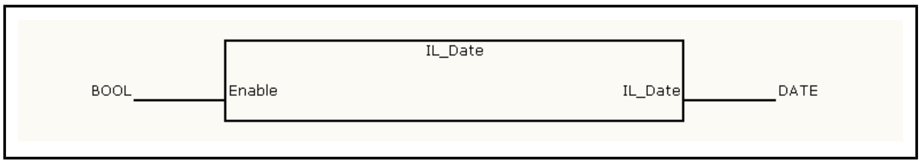


Fig.6-232: IL_Date

	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Enable function processing (cyclic, status-controlled)
RETURN	IL_Date	DATE	Current system date according to IEC 61131-3

Fig.6-233: IL_Date interface

Functional Description After having been enabled for processing with "Enable", the "IL_Date" function cyclically reads the system date which is formatted according to IEC 61131-3.

6.12.8 IL_TimeOfDay

Brief Description The "IL_TimeOfDay" function is used to read the current system time.

Interface Description

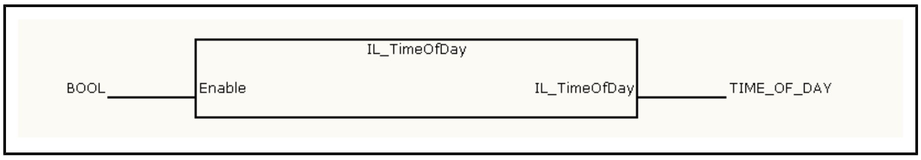


Fig.6-234: IL_TimeOfDay

	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Enable function processing (cyclic, status-controlled)
RETURN	IL_TimeOfDay	TOD	Current system time according to IEC 61131-3

Fig.6-235: IL_TimeOfDay interface

Functional Description After having been enabled for processing with "Enable", the "IL_TimeOfDay" function cyclically reads the system time which is formatted according to IEC 61131-3.

6.12.9 IL_DateAndTime

Brief Description The "IL_DateAndTime" function is used to read the current system date and time.

Interface Description

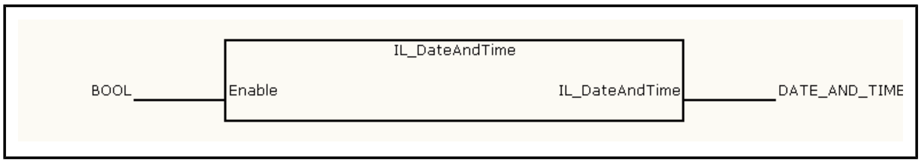


Fig.6-236: IL_DateAndTime

	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Enable function processing (cyclic, status-controlled)
RETURN	IL_DateAndTime	DT	Current system date and time according to IEC 61131-3

Fig.6-237: IL_DateAndTime interface

Functional Description After having been enabled for processing with "Enable", the "IL_DateAndTime" function cyclically reads the current system date as well as the current system time according to IEC 61131-3 format.

6.12.10 IL_SysTime64

Brief Description The "IL_SysTime64" function block is used to read the current system date and time.

Interface Description

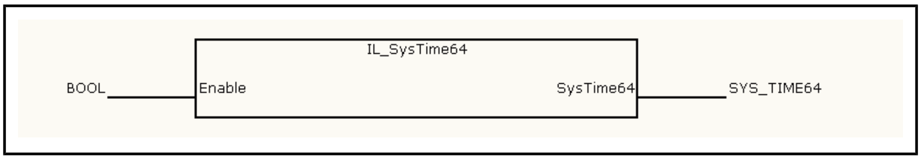


Fig.6-238: IL_SysTime64

	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Processing enable of the function block (cyclic, state-controlled)
VAR_OUTPUT	SysTime64	SYS_TIME64	Current system date and time in microseconds since 1970-01-01

Fig.6-239: IL_SysTime64 interface

Functional Description After having been enabled for processing with "Enable", the "IL_SysTime64" function block cyclically reads the current system date and time in microseconds since 1970-01-01.

6.12.11 IL_SysTimeDate

Brief Description The "IL_SysTimeDate" function block is used to read the current system date and time.

Interface Description

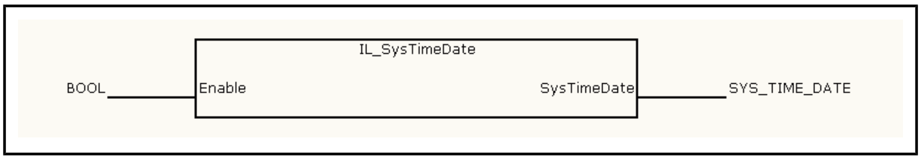


Fig.6-240: IL_SysTimeDate

	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Processing enable of the function block (cyclic, state-controlled)
VAR_OUTPUT	SysTimeDate	SYS_TIME_DATE	Current system date and time in system format

Fig.6-241: Interface variables of the IL_SysTimeDate function block

Functional Description After having been enabled for processing with "Enable", the "IL_SysTimeDate" function block cyclically reads the current system date and the current system time in system format.

6.12.12 IL_ExtSysTimeDate

Brief Description The "IL_ExtSysTimeDate" function block is used to read the current system date and time.

Interface Description

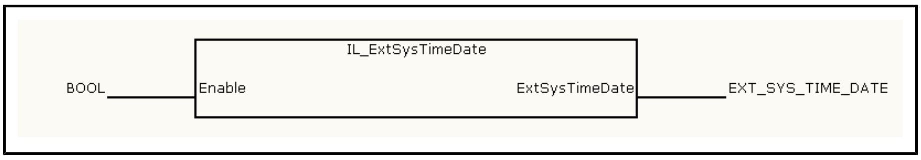


Fig.6-242: IL_ExtSysTimeDate

	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Processing enable of the function block (cyclic, state-controlled)
VAR_OUTPUT	ExtSysTimeDate	EXT_SYS_TIME_DATE	Current system date and time in extended system format

Fig.6-243: Interface variables of the IL_ExtSysTimeDate function block

Functional description

After having been enabled for processing with "Enable", the "IL_ExtSysTimeDate" function block cyclically reads the current system date and the current system time in extended system format.

☞ The extended system format consists of the "SYS_TIME64" as well as "SYS_TIME_DATE" system formats.

6.12.13 IL_SysTime64ToSysTimeDate

Brief Description

The "IL_SysTime64ToSysTimeDate" function block is used to convert the format of the system date and time.

Interface Description

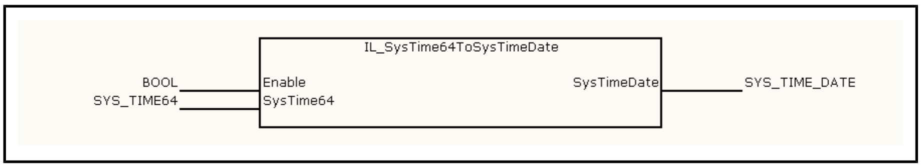


Fig.6-244: IL_SysTime64ToSysTimeDate

	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Processing enable of the function block (cyclic, state-controlled)
	SysTime64	SYS_TIME64	System date and time in microseconds since 1970-01-01
VAR_OUTPUT	SysTimeDate	SYS_TIME_DATE	System date and time in system format

Fig.6-245: Interface variables of the IL_SysTime64ToSysTimeDate function block

Functional Description

After having been enabled for processing with "Enable", the "IL_SysTime64ToSysTimeDate" function block cyclically converts the system data and time applied in microseconds to the "SysTime64" input (basis: 1970-01-01) to the system date and time in system format.

6.12.14 IL_SysTimeDateToSysTime64

Brief Description

The "IL_SysTimeDateToSysTime64" function block is used to convert the format of the system date and time.

Interface Description

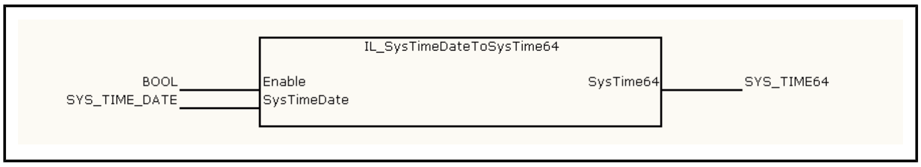


Fig.6-246: IL_SysTimeDateToSysTime64

	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Processing enable of the function block (cyclic, state-controlled)
	SysTimeDate	SYS_TIME_DATE	System date and time in system format
VAR_OUTPUT	SysTime64	SYS_TIME64	System date and time in microseconds since 1970-01-01

Fig.6-247: Interface variables of the IL_SysTimeDateToSysTime64 function block

Functional Description

After having been enabled for processing with "Enable", the "IL_SysTimeDateToSysTime64" function block cyclically converts the system date and time applied to the "SysTimeDate" input in system format to the system date and time in microseconds (basis: 1970-01-01).

6.13 RIL_VExUtil.lib

6.13.1 IL_VExKeys

Brief Description

With the IL_VExKeys function block the user can safely transmit the keystrokes on an HMI device to the control. The connection occurs via UDP. Each time when the function block is called, the last data is copied. If the connection is interrupted, all outputs are reset to 0. Thus, the user is able to program safe properties, whereby in the output data of the function block the safety mode is assigned to 0.

Interface Description

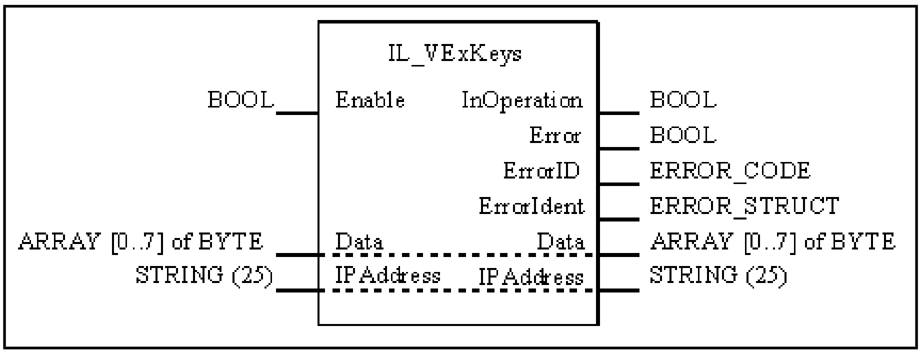


Fig.6-248: Function block IL_VExKeys

	Name	Type	Comment
VAR_IN_OUT	Data	ARRAY [0..7] OF BYTE	Contains data as byte array. The meaning of the particular bit is described in the documentation of the HMI.
	IPAddress	STRING(25)	Entered IP address "."§ notation, e.g. "10.104.73.193"
VAR_INPUT	Enable	BOOL	As long as the input is TRUE, the data is permanently updated. If it is inactive, the output data are all 0.
VAR_OUTPUT	InOperation	BOOL	If TRUE, the output data are valid and are updated.
	Error	BOOL	Indicates that an error has occurred within the function block. The output data are all set to 0.
	ErrorID	ERROR_CODE	Error recognition (see next table "Error codes").
	ErrorIdent	ERROR_STRUCT	Error structure with further division of the errors.

Fig.6-249: Function block IL_VExKeys

Error Messages

The function block generates the following error messages in Additional1/Additional2 for table "**MLC_ETHERNET**":

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR (16#0001)	10	0	Wrong IP syntax
COMMUNICATION_ERROR (16#0002)	11	0	Unknown IP address or no connection
DEVICE_ERROR (16#0008)	12	0	Invalid internal buffer size
STATE_MACHINE_ERROR (16#0005)	1	0	Invalid status of the state machine

Fig.6-250: Generated error numbers of the IL_VexKeys function block

7 Display and Operating Components

7.1 General Information

On its front, the IndraLogic L40 DP is provided with the following display and operating components: single-line display, operating keys, reset button, and Stop LED.

In addition, you can use the "IndraControl Service Tool" of the IndraLogic L40 DP for display and operating functions. The Ethernet interface allows access to the control via a web browser. On delivery, the IP address of the control is set to 192.168.1.10. Network settings can be managed in the IndraControl Service Tool (see chapter 7.5 "Web-based IndraControl Service Tool " on page 196).

7.2 Display and Operating Keys

Display

The display is an LC display comprising 8 digits (5 x 10 dot matrix).

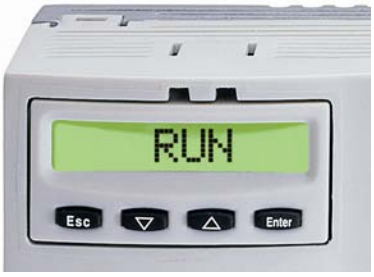


Fig.7-1: Display with four operating keys

Operating Keys

The four keys below the display have the following functions:

Key	Menu navigation	Input functions
<Esc>	One level up	Cancel input
<Down> (down arrow)	One menu entry down	Reduce parameter value
<Up> (up arrow)	One menu entry up	Increase parameter value
<Enter>	One level down	Confirm input

Fig.7-2: Operating key functions

7.3 Reset Button S1 and STOP LED

The reset button and a red light-emitting diode are arranged below the display.

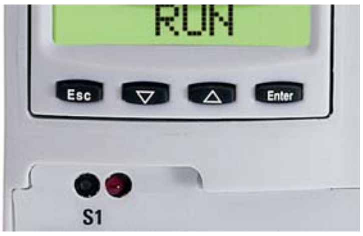


Fig.7-3: Stop LED and recessed reset button S1

Reset Button S1

The reset button resets the entire assembly and initiates a mandatory restart without the supply voltages having to be switched off.

The reset button can only be pressed using a tool, for instance with the tip of a pencil.

☞ A reset interrupts processing of a running program!

STOP LED

The STOP LED indicates basic PLC states.

LED OFF	Normal state (run, PLC program is running)
LED red	PLC stopped
LED red and flashing	Outputs are disabled

Fig.7-4: Meanings of the stop LED

7.4 Available Menu Levels

7.4.1 General Information

The display of the device allows reading of information while the operating keys can be used to make the necessary settings. The data is always subdivided in function-related menu levels so that a clear structure is provided.

7.4.2 Default and Status Displays

General Information

After the IndraLogic L40 DP has been switched on, the display shows the default setting.

Use the <up>, <down>, <Enter> and <Esc> keys to switch between the various menu items and menu levels.

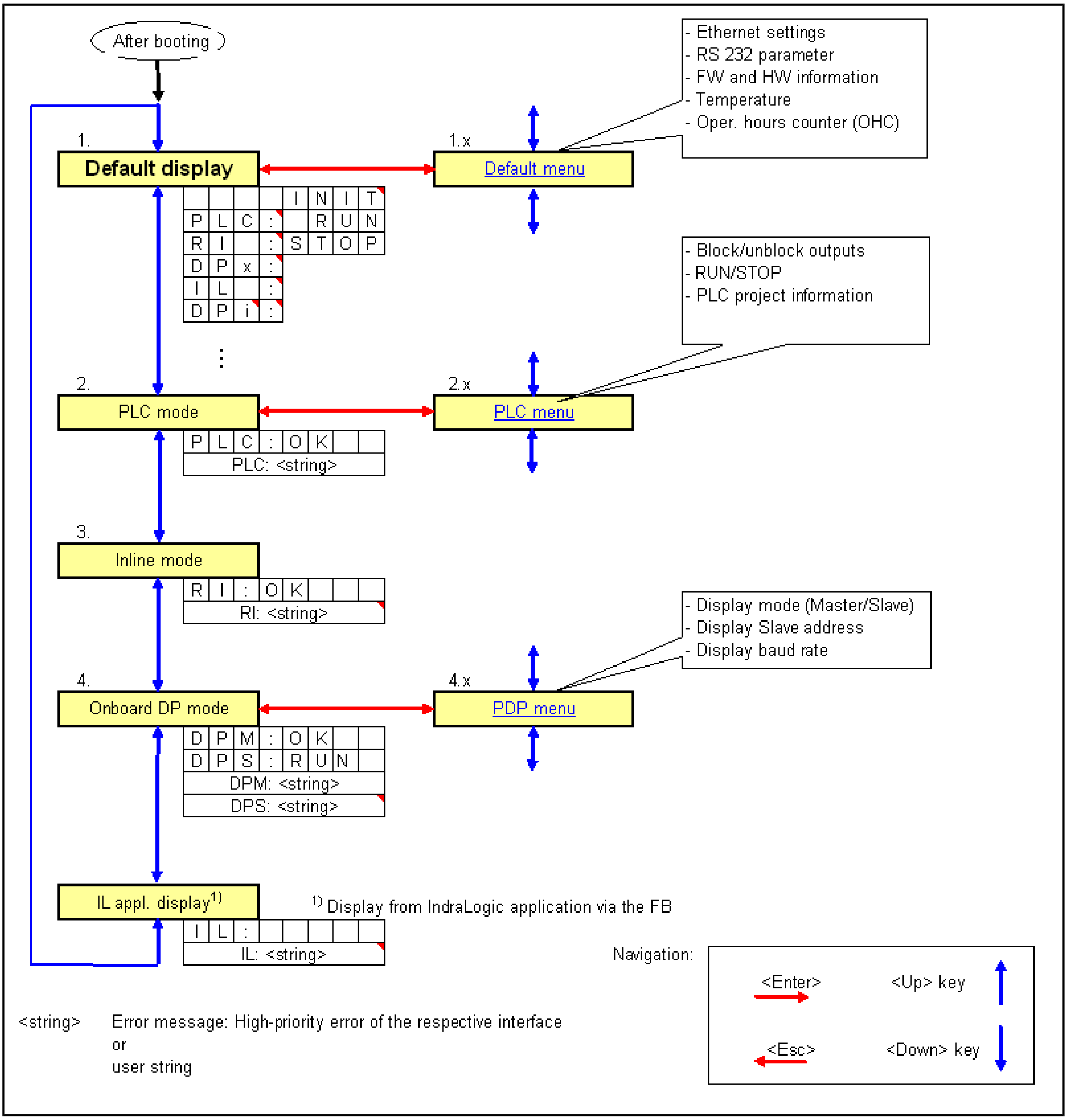


Fig.7-5: Overview: default and status displays

Default Display

"INIT": PLC boots.

"PLC: RUN": Normal mode. PLC is running.

"RI" or "DP" is flashing: An Inline (RI) or Profibus DP error (DPM: master; DPS: slave) is present. For details on the Inline error, refer to the "Inline mode" menu; for details on the Profibus DP error, refer to the "Onboard DP mode" menu. If several errors are present, only the error with the highest priority is displayed.

"IL:" flashing: a user output is present. This output can be displayed through the "IL app. display" menu.

Press the <Enter> key to go to the "Default menu".

PLC Mode

Indicates the current PLC state.

Press the <Enter> key to go to the "PLC menu".

Inline Mode

Indicates an active Rexroth Inline error.

Example: "RI: Rexroth Inline configuration error at module: xxx"

Onboard DP Mode

DPM:..": (only when the IndraLogic L40 DP is used as Profibus master with the onboard DP interface X7P) indicates the Profibus state of the master.

DPS:..": (only when the IndraLogic L40 DP is used as Profibus slave with the onboard DP interface X7P) indicates the Profibus state of the slave.

Press the <Enter> key to go to the "Profibus DP menu".

IL Appl. Display

Indicates a present IndraLogic user output, also see chapter 6.4.5 "IH_Set-Display " on page 99

.

7.4.3 Default Menu

General Information

After the default menu has been called from the default display with the <Enter> key, the "Ethernet" menu item is displayed.

Use the <up>, <down>, <Enter> and <Esc> keys to switch between the various menu items and menu levels.

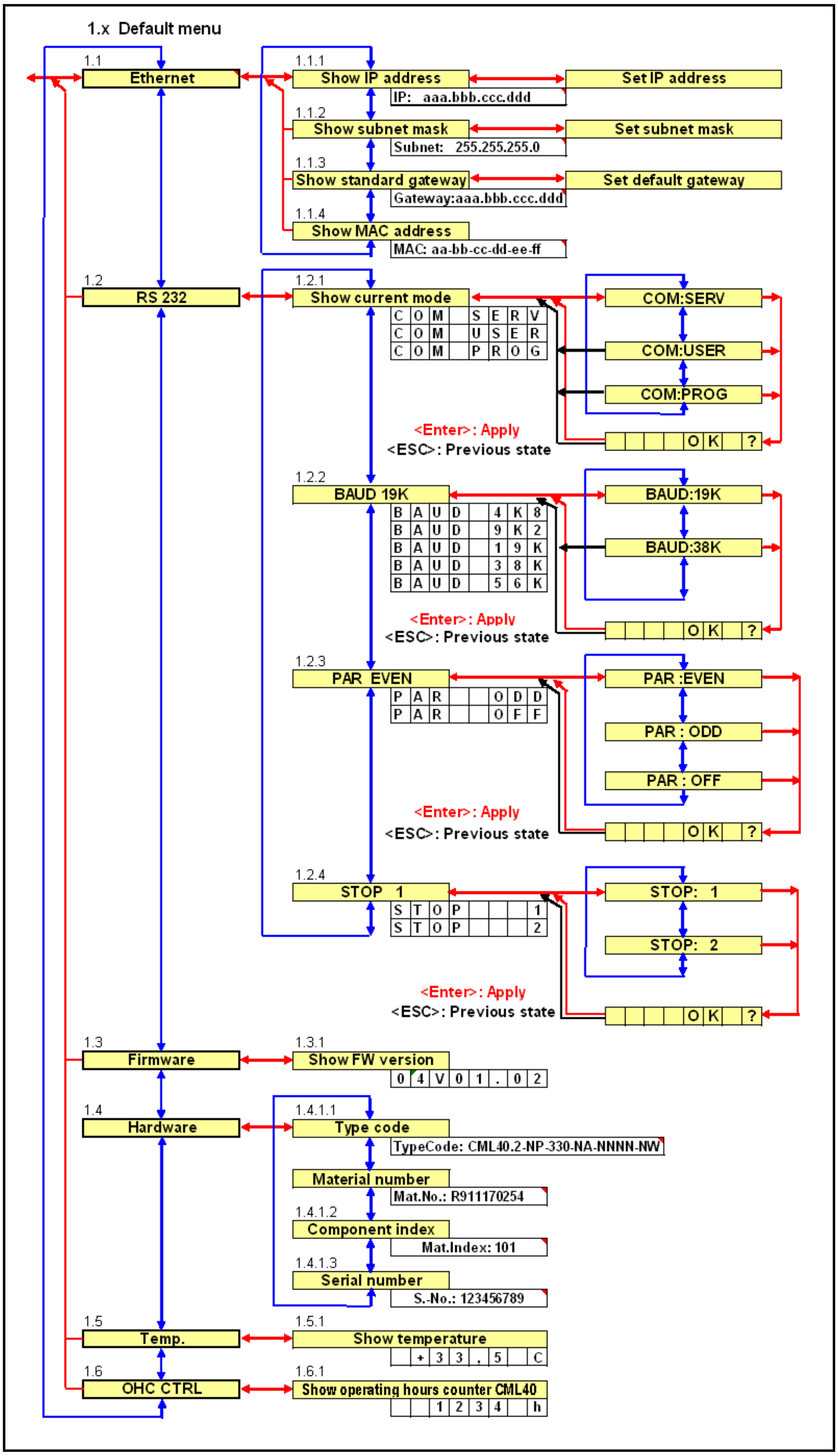


Fig.7-6: Overview: default menu

Ethernet

Allows displaying/setting of the IP address, subnet mask and default gateway. Moreover, you will find the MAC address here.

For more information about the setting, please refer to Setting the IP address, subnet mask and default gateway.

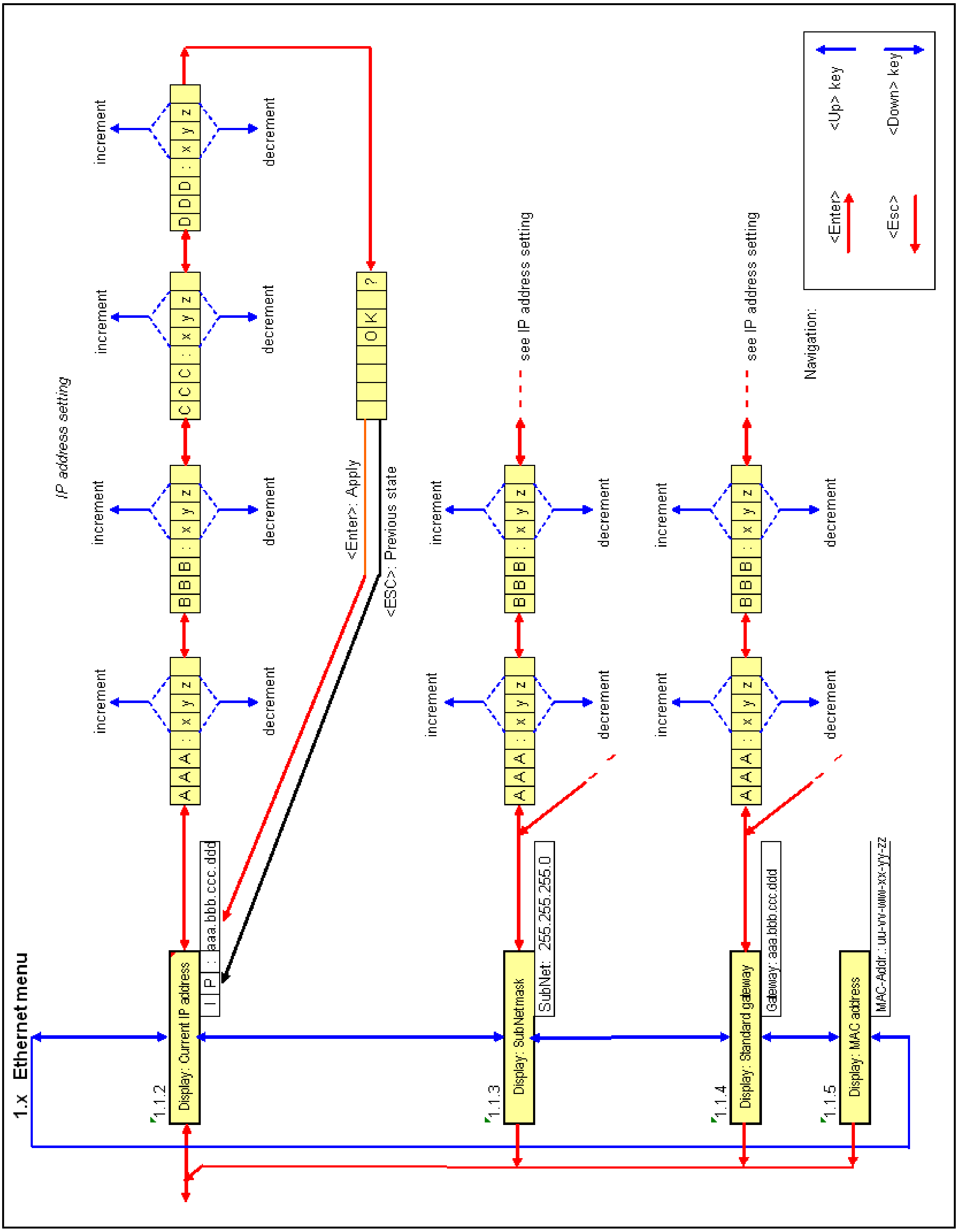


Fig.7-7: Setting the IP address, subnet mask and default gateway

1. To edit an address, press the <Enter> key while the current address is displayed.

   The display shows "AAA:" which identifies the first address byte, followed by its current decimal value.
2. Set the desired value with the <up> and <down> keys.
3. Acknowledge your setting with <Enter>.

   The display shows "BBB:" which identifies the second address byte, followed by its current decimal value.
4. Set the remaining address bytes (CCC, DDD) as described above.

   After you have acknowledged the last setting, the system displays "OK ?" asking you whether the new address value is to be applied.
5. Confirm with <Enter>, the new address value will be displayed, written to the die Compact Flash card and used **on the next start** of the control. If this is not desired, you can reject the change with <Esc>. Then, the previous address value remains active.

RS 232

The "RS 232" menu item defines how the onboard RS232 interface (X3C) will be used. The RS232 interface can only be used for one operating mode (Commserver, user interface or programming interface). Its use must be set on the display. A different setting will not become active before the control is restarted.

COM SERV	The "COM_USER" mode provides communication with the compact operator terminal. Configuration of the interface is assumed by the compact operator terminal.
COM USER	In the "COM_USER" mode, the serial interface can be used for data input and output from the PLC program (application). This can be achieved with the appropriate functions from the "SysLibCom.lib" library.
COM PROG	For communication with a programming device. In the "COM_PROG" mode, the serial interface can be used as alternative programming device interface. To achieve this, the RS232 interface must be selected from the communication settings (3S serial RS232 driver), see the chapter on "Communcation Settings".

☞ After having applied an edited setting, switch the IndraLogic L40 DP off and on again. Only thereafter will the settings become effective.

Firmware

The "Firmware" menu item displays the version number of the installed firmware.

Temp.

The "Temp." menu item displays the internal temperature of the IndraLogic L40 DP.

☞ If the internal temperature reaches 80 °C, the control automatically switches to stop mode, switches the outputs to the safe status and displays the "Temp !!!" warning. This mode can only be exited by switching the power supply off and on again.

Before restarting the control after overheating, you should find the cause of the problem. Ensure that the ambient temperature of the IndraLogic L40 DP is not higher than 45 °C.

☞ The IH_Temperature library function (see device-specific "RIL_Lxx_Util" library) allows programm-controlled reading of the internal temperature of the IndraLogic L40 DP. Thus, critical temperature rises can be detected at an early stage and measures can be taken to avoid an over-temperature.

OHC CTRL

The "OHC CTRL" menu item displays the value of the operating hours meter of the IndraLogic L40 DP.

7.4.4 PLC Menu

General Information

After the PLC menu has been called with the <Enter> key, the "PLC mode" menu displays the "OUTP ON" or "OUTP OFF" menu item.

Use the <up>, <down>, <Enter> and <Esc> keys to switch between the various menu items and menu levels.

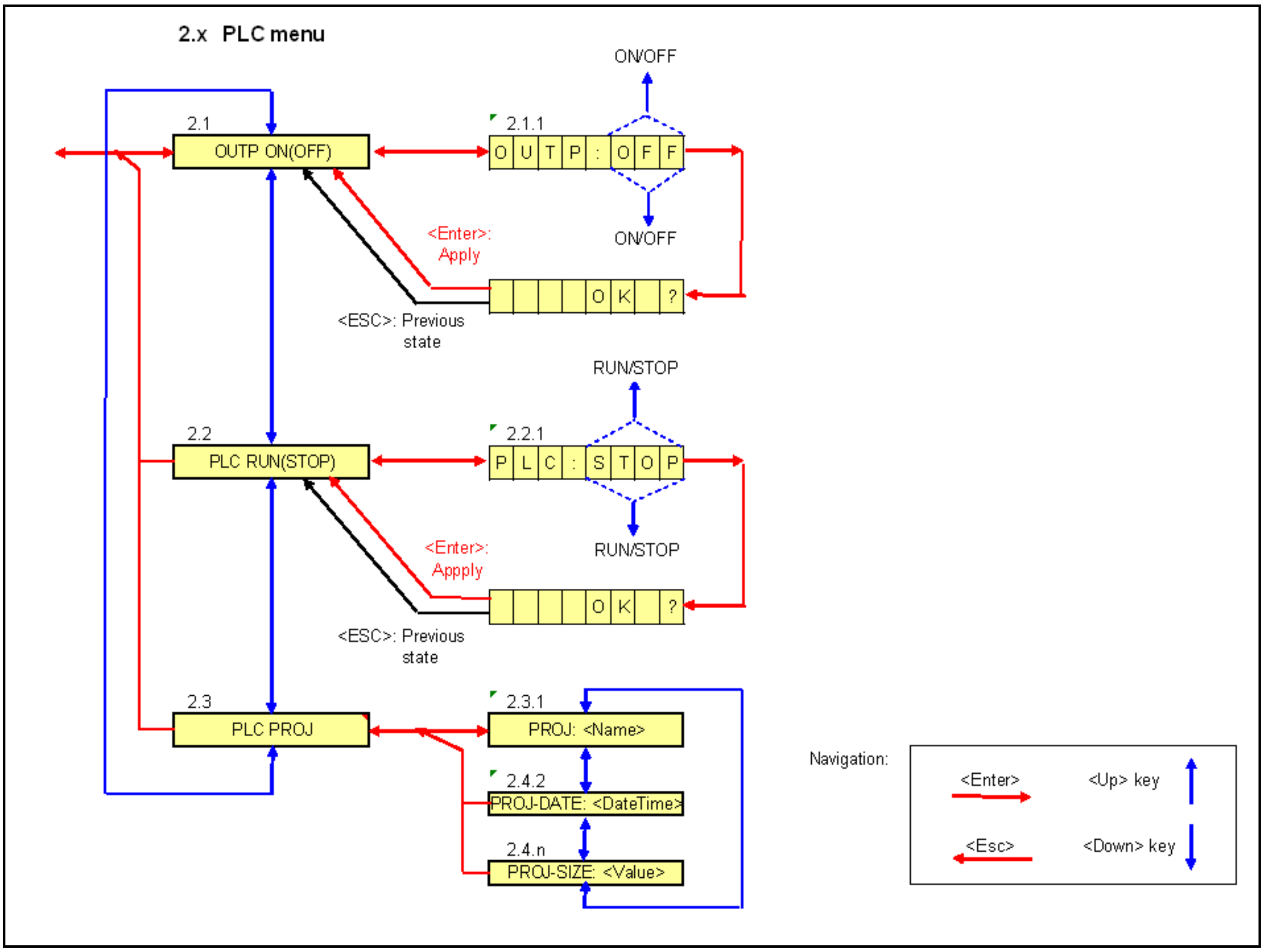


Fig.7-8: Overview: PLC menu

OUTP ON / OUTP OFF

The "OUTP ON" or "OUTP OFF" menu item involves all local digital outputs of the IndraLogic L40 DP, all outputs of the connected Rexroth Inline modules and all outputs activated via Profibus DP:

OUTP ON:	Enables outputs
OUTP OFF:	Resets outputs (safe status). The STOP LED flashes.

PLC RUN / PLC STOP

- The "PLC RUN" or "PLC STOP" menu item starts or stops the PLC program run.
- "PLC RUN" or "PLC STOP" involves all local digital outputs of the IndraLogic L40 DP, all outputs of the connected Rexroth Inline modules and all outputs activated via Profibus DP:

PLC RUN:	Enables outputs and starts the PLC program.
PLC STOP:	Resets the outputs to the safe status and stops the PLC program.

☞ The same functions can be activated by "Start" and "Stop" at a connected programming device. Please note that if, e.g., PLC STOP has been activated at the IndraLogic L40 DP, it can also be cancelled at the programming device.

PLC PROJ

Displays name, creation date and size of the currently loaded PLC project.

7.4.5 Profibus-DP Menu

General Information

After the Profibus DP menu has been called with the <Enter> key in the "Onboard DP mode" menu, detailed information on some Profibus DP settings can be displayed.

Use the <up>, <down>, <Enter> and <Esc> keys to switch between the various menu items and menu levels.

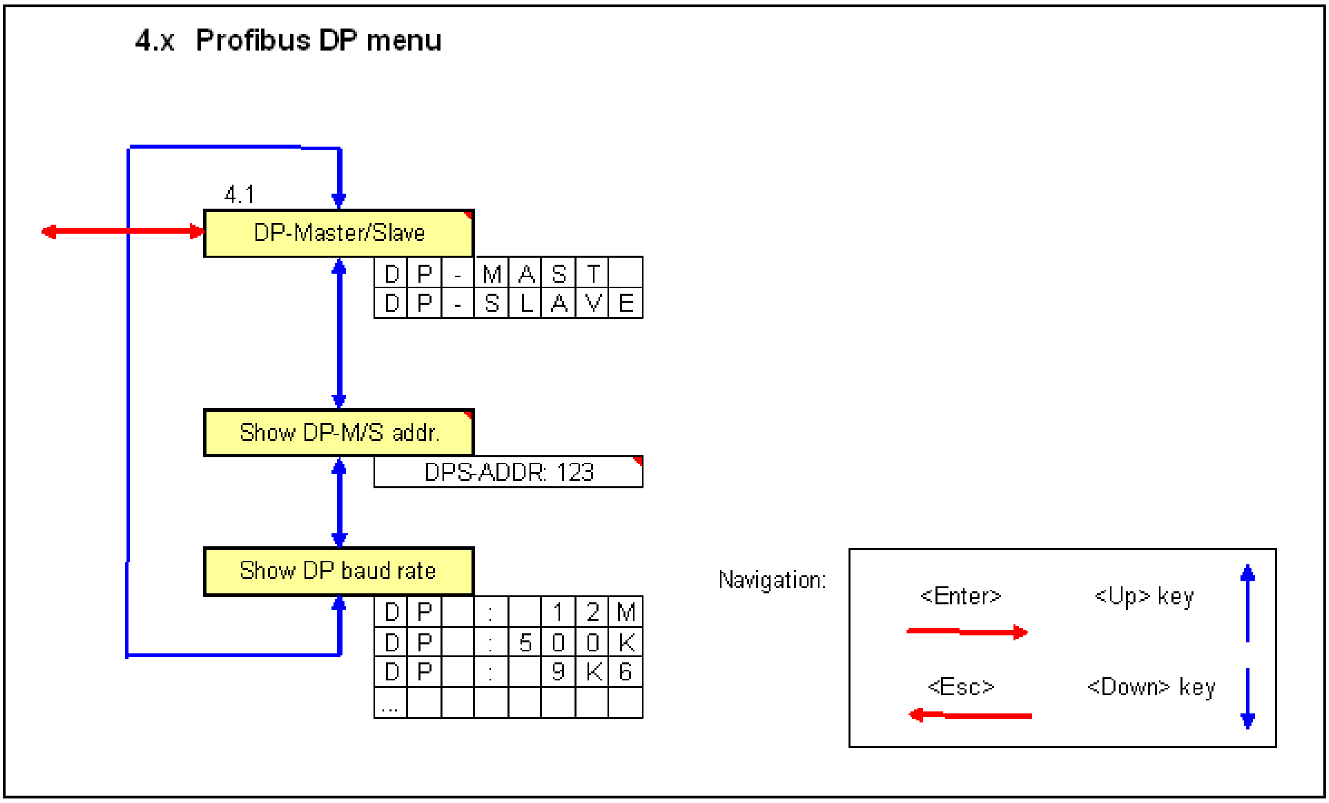


Fig.7-9: Overview: Profibus DP menu

DP Master/Slave

Displays whether the onboard DP interface IndraLogic L40 DP is parameterized as master ("DP-MAST") or as slave (DP-SLAVE).

Display DP-M/S Address

The "Display DP-M/S address" function can be used to display the active slave address of the IndraLogic L40 DP. This display is only available if the IndraLogic L40 DP has been parameterized as slave.

Display DP Baud Rate

Displays the active baud rate of the onboard DP interface (X7P).

Examples: "12M" = 12 Mbauds; "500K" = 500 kbauds; "9K6" = 9600 bauds

7.4.6 Boot Menu: Deleting the Boot Project

General Information

If you press <Esc> + <Enter> while the control is powering up, the IndraLogic L40 DP branches to the boot menu.

The boot menu provides the following function:

Bootproj

The "Bootproj" function deletes the boot project stored on the Compact Flash card (corresponds to the "Reset (origin)" IndraLogic function).

☞ The function of deleting the boot project should be used in exceptional cases only (e.g. if the control fails to start because of a faulty boot project).

Exiting the Boot Menu Press <Esc> to exit the boot menu without deleting the boot project

Deleting the Boot Project Press <Enter> twice to delete the boot project and then <Esc> to exit the boot menu.

This will open the default display of the IndraLogic L40 DP.

7.5 Web-based IndraControl Service Tool

7.5.1 General Information

The control can be accessed via an integrated, web-based interface. The application is called "IndraControl Service Tool". This application is for displaying, operating and configuring the control.

In this section, the required browser settings as well as basic pages are described. The basic pages include:

- Login
- Network configuration
- Hardware overview
- Diagnostics
- Variable editor
- User management

Depending on the rights (refer to user management), not all pages are displayed to every user. The writing rights of individual pages ("Network configuration" or "Diagnostics" for example) are restricted for some users.

7.5.2 Browser Settings

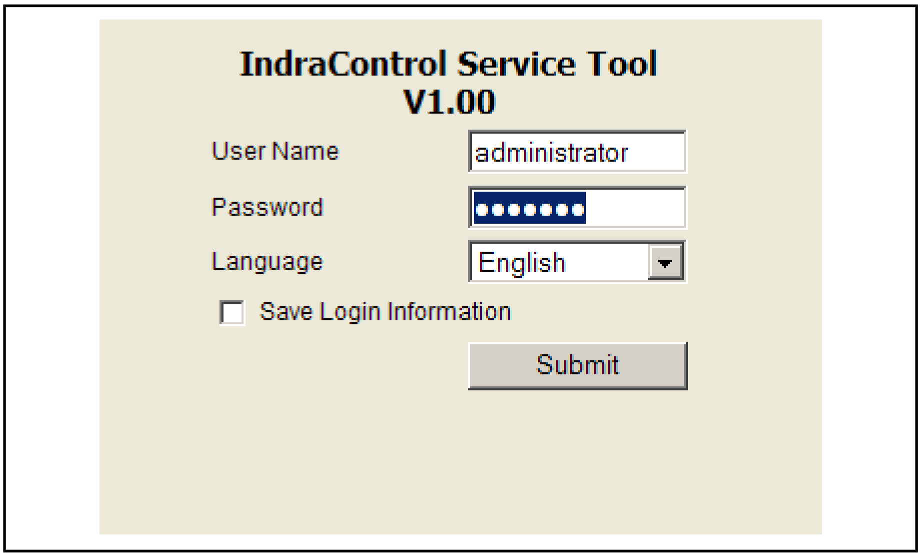
In order to display the IndraControl Service Tools correctly and to ensure that all functions are available, some security setting changes are required. As example, the following list shows the required settings (security settings for the web content zone) for the Internet Explorer 6:

Function	Setting
Execute ActiveX control elements secure for the scripting	Activate
Initialize and execute ActiveX control elements that are not secure	Prompt
Execute ActiveX control elements and plug-in	Activate
Automatic prompt for ActiveX control elements	Activate
Binary behavior and script behavior	Activate
Download of ActiveX control elements with sign	Prompt
Download of ActiveX control elements without sign	Activate
Active Scripting	Activate
Permit insert operations via a script	Activate

Fig.7-10: Security settings for the IndraControl Service Tool

This function uses popups.Therefore, they may not be blocked for the IP address of the control.

7.5.3 Login



IndraControl Service Tool
V1.00

User Name: administrator

Password: ●●●●●●●

Language: English ▾

□ Save Login Information

Fig.7-11: First page of the IndraControl Service Tools

Login The user name and a password are to be entered on the first page. The following user names and passwords are already assigned when delivered from the supplier:

User	Password
administrator	rexroth
service	service
user	user

Fig.7-12: User names and passwords when delivered from the supplier

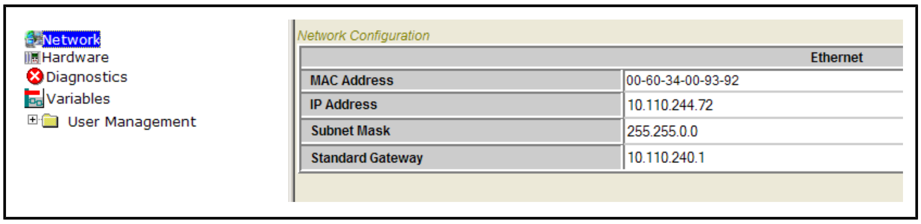
Language of the application The language of the application can be set via the selection field "Language selection". German and English are currently available.

Save login information With the option "Save login information" the login data is saved locally as "Cookie".

7.5.4 Symbols

Symbol	Description
🌳	Click on interface in order to hide the tree view of the IndraControl service tool.
🏃	The display device indicates that PLC is running.
🧍	The display device indicates that PLC is in stop state.
⊗	Click on interface in order to get diagnostic information on the PLC (also refer to chapter 7.5.7 " Diagnostics " on page 199).

7.5.5 Network Configuration



Network Configuration

	Ethernet
MAC Address	00-60-34-00-93-92
IP Address	10.110.244.72
Subnet Mask	255.255.0.0
Standard Gateway	10.110.240.1

Fig.7-13: Network configuration in the IndraControl Service Tool

The page "Network configuration " displays the following network parameters:

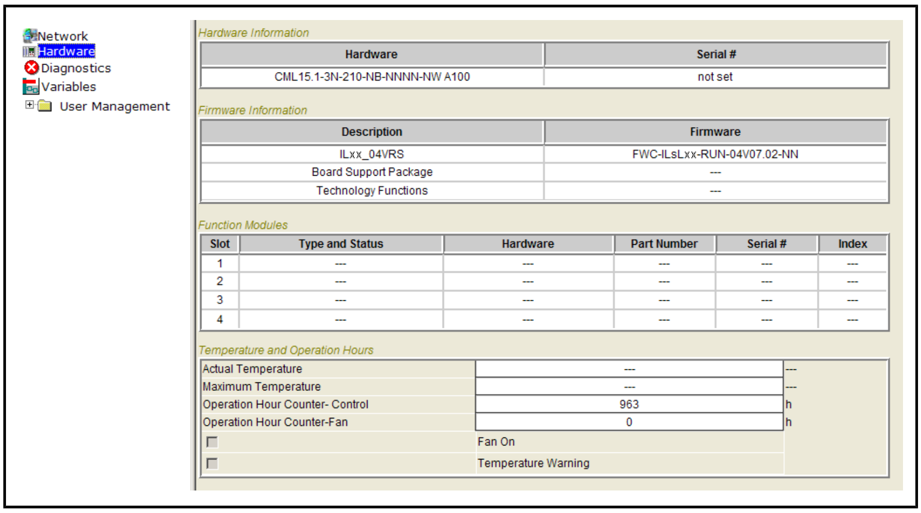
- MAC address
- IP address
- Subnet mask
- Default gateway

Apart from the MAC address, all values can be edited if the rights are given to the current user.

To change these parameters, set the curser in the requested field using the mouse. The highlighted presentation shows the modified values.

The modified values are applied after closing the network configuration and after a question for confirmation. To activate the modified values for the control, restart the control.

7.5.6 Hardware Overview



Hardware Information

Hardware	Serial #
CML15.1-3N-210-NB-NNNN-NW A100	not set

Firmware Information

Description	Firmware
ILxx_04VRS	FWC-ILsLxx-RUN-04V07.02-NN
Board Support Package	---
Technology Functions	---

Function Modules

Slot	Type and Status	Hardware	Part Number	Serial #	Index
1	---	---	---	---	---
2	---	---	---	---	---
3	---	---	---	---	---
4	---	---	---	---	---

Temperature and Operation Hours

Actual Temperature	---	---
Maximum Temperature	---	---
Operation Hour Counter- Control	963	h
Operation Hour Counter-Fan	0	h
□	Fan On	
□	Temperature Warning	

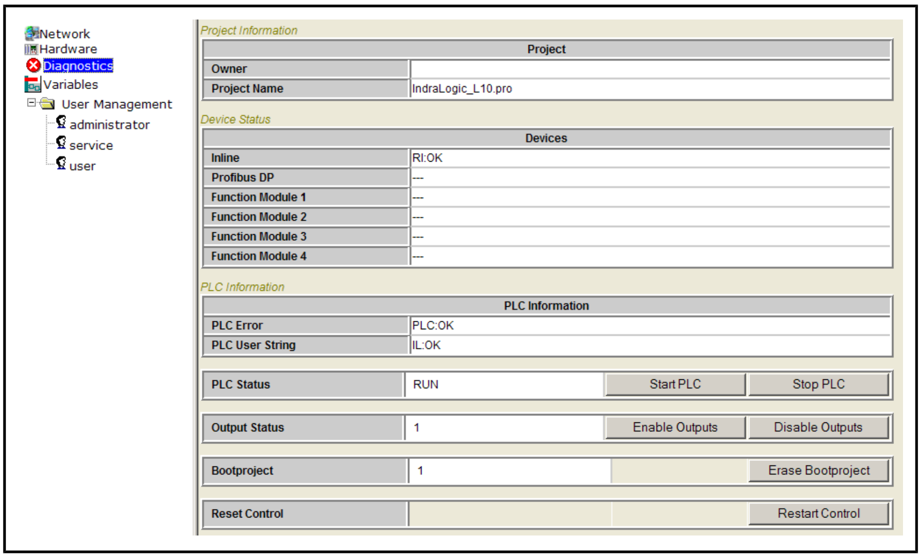
Fig.7-14: Hardware overview in the IndraControl Service Tool

The page "Hardware overview" provides the following information:

- Part short text of the hardware
- Change index
- Serial number
- Firmware version
- Board support package
- Technology function
- Type and status, hardware, part number, serial number and index of the function modules

- Temperature specifications (current and maximum temperature, fan status and temperature warning)
- Operating hours meter of the control as well as of the fan

7.5.7 Diagnostics



Project Information

Project	
Owner	
Project Name	IndraLogic_L10.pro

Device Status

Devices	
Inline	RI:OK
Profibus DP	---
Function Module 1	---
Function Module 2	---
Function Module 3	---
Function Module 4	---

PLC Information

PLC Information	
PLC Error	PLC:OK
PLC User String	IL:OK

PLC Status	RUN
Output Status	1
Bootproject	1
Reset Control	

Fig.7-15: Diagnostics overview in the IndraControl Service Tool

The page "Diagnostics" provides information on the PLC project as well as several PLC operating functions. The following information is shown.

- The user or the creator as well as the project name of the project currently loaded in the IndraControl.
- Inline status
- Profibus status
- Status of the equipped function modules
- Status of the PLC
- Display of the PLC user string

The following PLC operating functions are available if the rights were given to the current user:

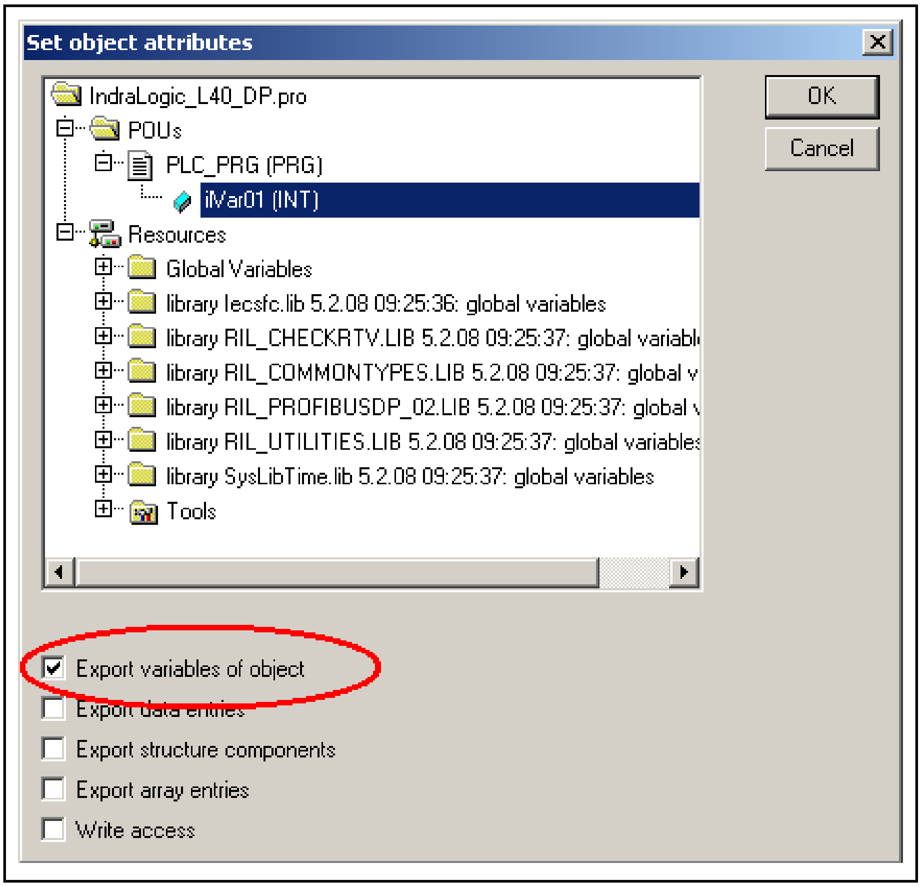
- Starting and stopping the PLC
- Blocking and unblocking of outputs
- Deleting the boot project
- Restarting the PLC

7.5.8 Variable Editor

Requirements

The following preconditions have to be fulfilled before using the variable editor.

- Double click on "PLC_PRG" in the project tree of your device. The IndraLogic programming platform opens.
- Select in the main menu **Project** ▸ **Options**. Select "Symbol configuration" in the options dialog. Tick "Create symbol entries".
- Click on "Configure symbol file...". The dialog "Set objects attributes" opens. Select the variables you want to edit in the variable editor. Subsequently tick "Display object variables" (refer to fig. 7-16 "Set dialog object attributes" on page 200).
- Confirm your entry by clicking on "OK".



Set object attributes

- IndraLogic_L40_DP.pro
  - POUs
    - PLC_PRG (PRG)
      - iVar01 (INT)
  - Resources
    - Global Variables
    - library Iecsfc.lib 5.2.08 09:25:36: global variables
    - library RIL_CHECKRTV.LIB 5.2.08 09:25:37: global variabl
    - library RIL_COMMONTYPES.LIB 5.2.08 09:25:37: global v
    - library RIL_PROFIBUSDP_02.LIB 5.2.08 09:25:37: global v
    - library RIL_UTILITIES.LIB 5.2.08 09:25:37: global variables
    - library SysLibTime.lib 5.2.08 09:25:37: global variables
    - Tools

☑ Export variables of object

□ Export data entries

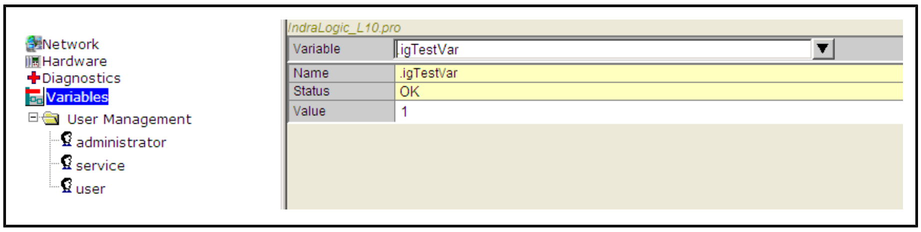
□ Export structure components

□ Export array entries

□ Write access

Fig.7-16: Set dialog object attributes

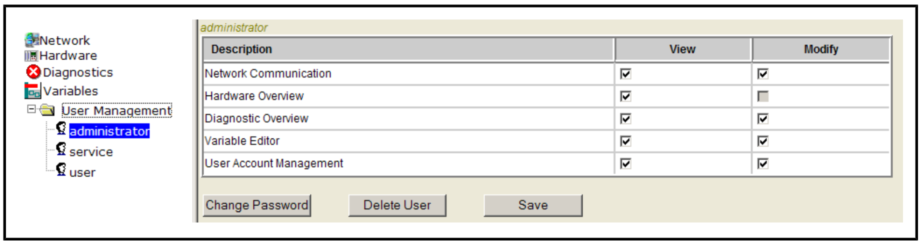
The variable editor provides the possibility to edit single PLC variables. The icon name of the variable is entered in the field "Variable" and confirmed using "Enter". If the input of the variable name is correct, the field "value" shows the current value. This value can be modified and transferred to the control using "Enter". The new value is immediately applied by the control.



Variable	.igTestVar
Name	.igTestVar
Status	OK
Value	1

Fig.7-17: Variable editor overview in the IndraControl Service Tool

7.5.9 User Management



Description	View	Modify
Network Communication	☑	☑
Hardware Overview	☑	□
Diagnostic Overview	☑	☑
Variable Editor	☑	☑
User Account Management	☑	☑

Fig.7-18: User management in the IndraControl Service Tool

The page "User Management" provides the opportunity to assign certain rights for the pages listed. Select the user on the left of the Explorer. Assign the rights "View" and "Change" on the right. Save the settings using the "Save" button.

Changing the passwords of single users and deleting existing users are part of the basic functions.

In order to add or delete a new user profile or change the password, right click on an empty spot in the Exploror in the user profiles.

Add User

New users can be defined via the Explorer contex menu using the comman "Add user". The entry dialog "Add User" opens.

Entry field	Description
User name	Enter your user name.
Password	Enter a safe password for the new user.
Confirm password	Confirm the password.

Confirm your entry by clicking on "OK". The new user is displayed in the Explorer.

Delete User

New users can be deleted via the Explorer context menu using the command "Delete user". A dialog asks you if you really want to delete the user. Confirm your entry by clicking on "OK".

Change Password

A new passoword can be entered via the Explorer context menu using the command "Change password". The entry dialog "Change password" opens.

Entry field	Description
Old password	Enter your old password.
New password	Enter your new password.
Confirm New Password	Confirm your new password.

Confirm your entry by clicking on "OK". The new password is activated immediately.

8 Technical Data

8.1 Equipment

IndraLogic L40 DP-DPM-04VRS	
Processor	National Geode LX800 with at least 500 MHz
Random access memory (RAM)	At least 128 Mbytes DRAM and at least 128 Kbytes NvRAM
Interfaces:	
Interface to functional modules	• Bosch Rexroth PC104^{Plus}
Interface to I/O modules	• Rexroth Inline interface
Communication interfaces	• 1 x Ethernet connection (RJ 45, 10/100 Base-T) • 1 x serial RS 232 interface • 1 x PROFIBUS DP master-slave interface
Inputs and outputs	• 8 electrically isolated digital inputs • 8 electrically isolated digital outputs

Fig.8-1: Equipment

8.2 Performance Data

IndraLogic L40 DP-DPM-04VRS	
User memory for program code	16 Mbyte
User memory for data	8 Mbyte
Remanent memory	128 kbytes
Number of tasks	16
Processing time, typical (1000 instructions in IL, bit and word commands mixed)	35 µs
Profibus DP master according to IEC 61158-3	

Fig.8-2: Performance data

9 Service and Support

9.1 Helpdesk

Our service helpdesk at our headquarters in Lohr, Germany, will assist you with all kinds of inquiries.

Contact us:

- By phone through the Service Call Entry Center

  Monday to Friday: 7:00 - 18:00 Central European Time

  +49 (0) 9352 40 50 60
- per Fax

  +49 (0) 9352 40 49 41
- By E-mail: service.svc@boschrexroth.de

9.2 Service Hotline

Out of helpdesk ours please contact our German service department directly:

+49 (0) 171 333 88 26

or

+49 (0) 172 660 04 06

Hotline numbers for other countries can be found in the addresses of each region on the Internet (see below).

9.3 Internet

Additional notes regarding service, maintenance and training, as well as the current addresses of our sales and service offices can be found on

http://www.boschrexroth.com

Outside Germany please contact our sales/service office in your area first.

9.4 Helpful Information

For quick and efficient help please have the following information ready:

- detailed description of the fault and the circumstances
- Information on the name plate of the affected products, especially type codes and serial numbers
- Your phone and fax numbers and E-mail address, so we can contact you in case of questions.

Index

Notes

Rexroth
Bosch Group

Bosch Rexroth AG
Electric Drives and Controls
P.O. Box 13 57
97803 Lohr, Germany
Bgm.-Dr.-Nebel-Str. 2
97816 Lohr, Germany
Phone +49 (0)93 52-40-50 60
Fax +49 (0)93 52-40-49 41
service.svc@boschrexroth.de
www.boschrexroth.com

R911323262

Printed in Germany
DOK-CONTRL-IC*L40**V04-AW01-EN-P