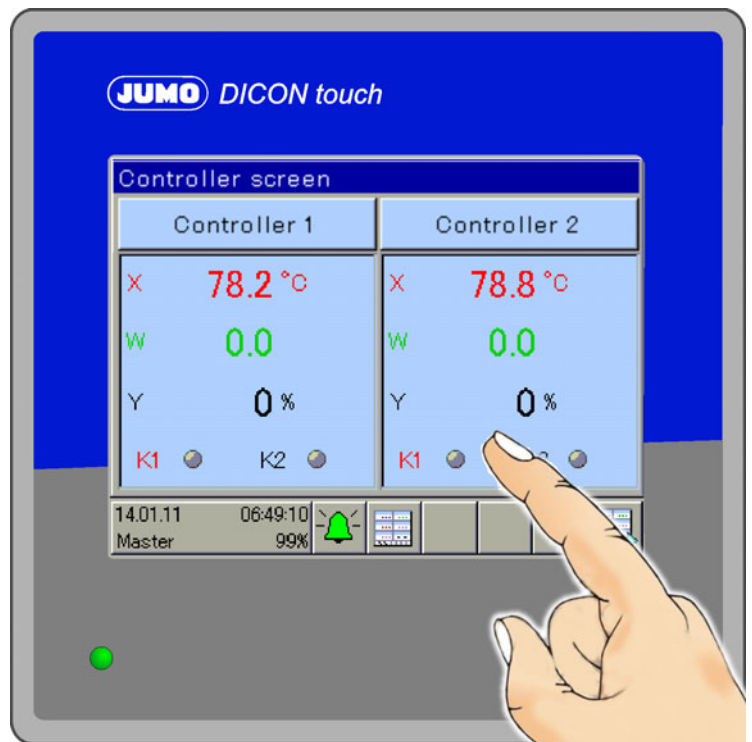


JUMO DICON touch

Two-channel process and program controller with paperless recorder and touchscreen 8.9 cm (3.5")



70357107T92Z001K000

V6.00/EN/00661200/2020-01-29

1	Safety information	5
1.1	Warning symbols	5
1.2	Note signs	5
2	Compatibility and system requirements	7
2.1	Certification	7
2.2	PROFINET IO and Ethernet standard services	7
2.3	Requirements for hardware, software, and cabling	7
3	Electrical connection	9
4	The GSDML file	11
4.1	General information	11
4.2	Module description	11
4.2.1	Introduction to the module concept	11
4.2.2	Modules	11
5	Noncyclical data exchange	19
5.1	Programming the noncyclical data exchange in the IO controller	19
5.1.1	Single-ID and Multi-ID	21
5.1.2	Programs for program controllers	26
5.2	Data tables of noncyclical data	36
5.2.1	Configuration data and parameters	36
5.2.2	Process values	42
6	Project planning	45
6.1	Project integration of JUMO IO devices	45
6.2	Configuring PROFINET modules	46
6.3	Configuring the JUMO IO device	50
6.3.1	Start-up parameters	50
6.3.2	Communication parameters	50
6.3.3	Device information	53
7	Error messages	57
7.1	Error messages for invalid values	57
7.2	Error messages for acyclic services	57

Contents

1.1 Warning symbols



DANGER!

This symbol indicates that **personal injury from electrocution** may occur if the appropriate precautionary measures are not taken.



WARNING!

This symbol in connection with the signal word indicates that **personal injury** may occur if the respective precautionary measures are not carried out.



CAUTION!

This symbol in connection with the signal word indicates that **material damage or data loss** will occur if the respective precautionary measures are not taken.



CAUTION!

This symbol indicates that **components could be destroyed** by electrostatic discharge (ESD = Electro Static Discharge) if the respective cautionary measures are not taken.

Only use the ESD packages intended for this purpose to return device inserts, assembly groups, or assembly components.



READ THE DOCUMENTATION!

This symbol, which is attached to the device, indicates that the associated **documentation for the device** must be **observed**. This is necessary to identify the nature of the potential hazard, and to take measures to prevent it.

1.2 Note signs



NOTE!

This symbol refers to **important information** about the product, its handling, or additional benefits.



REFERENCE!

This symbol refers to **additional information** in other sections, chapters, or other manuals.



FURTHER INFORMATION!

This symbol is used in tables and indicates that **further information** is provided after the table.



DISPOSAL!

At the end of its service life, the device and any batteries present do not belong in the trash! Please ensure that they are **disposed of** properly and in an **environmentally friendly** manner.

1 Safety information

2 Compatibility and system requirements

2.1 Certification

JUMO IO devices are certified by the PNO to Conformance Class B (abbreviated to CC-B) and net load class "Netload Class III".

The certificate for the JUMO DICON touch is available for download from the product page on the Internet.

The following QR code or web address will take you to the product page of the JUMO DICON touch:



qr-703571-en.jumo.info

2.2 PROFINET IO and Ethernet standard services

The PROFINET interface supports all Ethernet standard services. Over this interface, the device can also be accessed via the JUMO PC setup program and the JUMP PCC/PCA software. Additionally, the JUMO field device can send its e-mails over this interface. The Ethernet standard services are also available during the cyclical PROFINET communication.

The PROFINET IO communication occurs in a parameterizable time pattern (RT channel). This ensures that the IO data are transferred in real-time without being affected by Ethernet standard services. The remaining time between the RT phases (NRT channel) is used for the communication between the Ethernet standard services. Broadband bottlenecks impair the performance of the standard services. The real-time capability of the PROFINET IO communication is guaranteed by the reserved RT channel.

2.3 Requirements for hardware, software, and cabling

Network installation

Any switches used to network PROFINET IO devices must support the following standards and functions:

- 100 Mbit/s (transfer rate of the switchports)
- Auto negotiation (automatic setting of the switchport transfer rate)
- Cut through (direct relaying of the data between the switchports)

Full duplex support for the switchports

IEEE 802.1 Q (VLAN support for at least 4 priority classes)

The network installation must be executed compliant with the requirements of a 100Base TX Ethernet network with cabling to CLASS D or higher. The "PROFINET Cabling and Interconnection Technology" guideline also provides information on correct cabling.

JUMO DICON touch

To use the PROFINET IO with JUMO DICON touch, the device must meet the following minimum requirements:

- Device software version 206.03.xx or higher
- Hardware compatibility index 0x2 or higher

2 Compatibility and system requirements

The information can be displayed and reviewed with the aid of the device menu.

Display device software version:

Device menu > Device info > General > Version > SW version

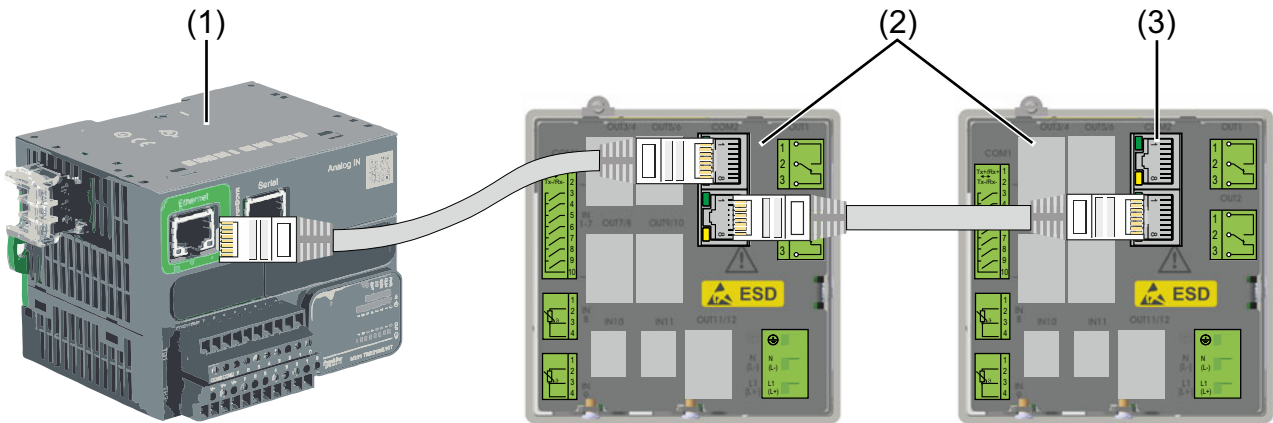
Display hardware compatibility index:

Device menu > Device info > General > Version > HW compatibility index

3 Electrical connection

Cabling

JUMO field devices with PROFINET IO interface have two Ethernet switchports. Additional field devices, IO controllers, an IO supervisor (programming device and/or PC for project planning) or other Ethernet components such as switches, routers etc. can be connected to each Ethernet switchport on the device.



- (1) PROFINET IO controller (PLC, control station PC or similar)
- (2) JUMO PROFINET IO devices
- (3) free Ethernet port (e.g. for additional PROFINET IO devices)



NOTE!

The cyclical data exchange with JUMO PROFINET IO devices is based on the RT protocol (Conformance Class B). PROFINET RT communication cannot be routed. It is thus necessary that the PROFINET IO controller and the IO devices be in a common broadcast domain (not connected via a router). Ethernet standard services (webserver, access via JUMO PC setup program/PCA/PCC and e-mail function) and NRT communication (e.g. project planning on the IO supervisor, system boot, noncyclical data, etc.) function via routers as before.

3 Electrical connection

4.1 General information

The GSDML file of a PROFINET device contains all the information required to provide your PLC engineering system with the PROFINET IO functionality of the JUMO DICON touch. It must be imported into the engineering system so that the JUMO DICON touch is available here for project planning. Once the GSDML file has been imported, the relevant device and its IO functions can be integrated into the programming system of your PLC projects.

4.2 Module description

4.2.1 Introduction to the module concept

The configuration of the IO data of JUMO field devices to be transferred via the PROFINET IO is defined by the PROFINET modules of the device software.

The "DeviceStatusBlock" module of JUMO field devices has a defined configuration and cannot be changed. It contains a range of cyclical IO data on the device status and the PROFINET start-up parameters for the JUMO field device concerned. In the project structure of the IO controller, this module is always located in Slot 1 of JUMO IO devices. It cannot be relocated or removed.

The JUMO PC setup program can be used to create additional modules in the module configuration of the device concerned. This involves device data being selected from the selectors in the JUMO field device and assigned to the corresponding IO items in the modules. The PC setup program can be used to print the finished module configuration as a basis for the project planning of the IO controller. In the project configuration of the IO controller, the modules to be transferred via the PROFINET-IO have to be assigned to the free slots of the JUMO IO device in the project structure of the IO controller. The IO items of the inserted modules are then available for the programming of the IO controller. The assignment of the JUMO field device data to the IO items of the individual modules is provided for the programmer in the printed module configuration list.

⇒ chapter 6 "Project planning", Page 45

4.2.2 Modules



CAUTION!

All temperature values of the JUMO IO device are transferred in the unit "°C".

Misinterpretation of measured values of a JUMO IO device in the IO controller can cause errors in the system control.

► Note the unit for the transferred temperature values!

"DeviceStatusBlock" module of the JUMO DICON touch

The "DeviceStatusBlock" module of JUMO IO devices has a defined configuration and cannot be changed. It contains a range of cyclical IO data on the device status and the PROFINET start-up parameters. The IO controller is only able to read the data of this module. The tables below list the data in the "DeviceStatusBlock" module of the JUMO DICON touch.

4 The GSDML file

Cyclical IO data, DeviceStatusBlock

IO item	Type	Bit item	Bit address	Explanation
Sub-module 1				
Status bits 1	USINT	Collective alarm	0	Internal signals of the JUMO DICON touch ⇒ See operating manual, chapter on digital selector configuration
		Collective alarm acknowledged	1	
		Memory alarm	2	
		Pre-alarm battery	3	
		Battery empty	4	
		Logon	5	
		USB inserted	6	
		Temperature in °F	7	
Status bits 2	USINT	–	0	Not used
		–	1	
		–	2	
		–	3	
		–	4	
		–	5	
		–	6	
		–	7	
Status bits 3	USINT	Manual mode, Controller 1	0	Internal signals of the JUMO DICON touch ⇒ See operating manual, chapter on digital selector configuration
		Tune controller 1	1	
		Manual mode, Controller 2	2	
		Tune controller 2	3	
		–	4	Not used
		–	5	
		–	6	
		–	7	
Status bits 4	USINT	–	0	
		–	1	
		–	2	
		–	3	
		–	4	
		–	5	
		–	6	
		–	7	

Cyclical IO data, DeviceStatusBlock

IO item	Type	Bit item	Bit address	Explanation
Encoder status	USINT	–	0	Internal signals of the JUMO DICON touch ⇒ See operating manual, chapter on digital selector configuration
		Programstopsignal	1	
		Basic status	2	
		Pre-programming	3	
		Manual operation mode	4	
		Programautosignal	5	
		Program end signal	6	
		Tolerancebandsignal	7	
Dummy	USINT	–	0	Fill byte (no function; used to satisfy requirements for certain data structures)
		–	1	
		–	2	
		–	3	
		–	4	
		–	5	
		–	6	
		–	7	
Operating contacts	UINT	Operating contact 1	0	Internal signals of the JUMO DICON touch ⇒ See operating manual, chapter on digital selector configuration
		Operating contact 2	1	
		Operating contact 3	2	
		Operating contact 4	3	
		Operating contact 5	4	
		Operating contact 6	5	
		Operating contact 7	6	
		Control contact 8	7	
		–	8	
		–	9	
		–	10	
		–	11	
		–	12	
		–	13	
		–	14	
		–	15	

4 The GSDML file

Cyclical IO data, DeviceStatusBlock

IO item	Type	Bit item	Bit address	Explanation
Current program number	UINT	–	–	Internal signals of the JUMO DICON touch ⇒ See operating manual, chapter on integer selector configuration
Current section number	UINT	–	–	
Current program runtime	UDINT	–	–	
Current remaining program runtime	UDINT	–	–	
Current section runtime	UDINT	–	–	
Current remaining section runtime	UDINT	–	–	
Current program setpoint value 1	REAL	–	–	Internal signals of the JUMO DICON touch ⇒ See operating manual, chapter on analog selector configuration
Current program setpoint value 2	REAL	–	–	
Controller 1 actual value	REAL	–	–	Internal signals of the JUMO DICON touch ⇒ See operating manual, chapter on analog selector configuration
Controller 1 setpoint value	REAL	–	–	
Controller 1 output level	REAL	–	–	
Controller 2 actual value	REAL	–	–	
Controller 2 setpoint value	REAL	–	–	
Controller 2 output level	REAL	–	–	
Analog input 1	REAL	–	–	
Analog input 2	REAL	–	–	
Analog input 3	REAL	–	–	
Analog input 4	REAL	–	–	
Analog output 1	REAL	–	–	
Analog output 2	REAL	–	–	
Analog output 3	REAL	–	–	
Analog output 4	REAL	–	–	
Analog output 5	REAL	–	–	

Cyclical IO data, DeviceStatusBlock

IO item	Type	Bit item	Bit address	Explanation
Binary values	UDINT	Digital input IN1	0	Internal signals of the JUMO DICON touch ⇒ See operating manual, chapter on digital selector configuration
		Digital input IN2	1	
		Digital input IN3	2	
		Digital input IN4	3	
		Digital input IN5	4	
		Digital input IN6	5	
		Digital input IN7	6	
		Digital output OUT1	7	
		Digital output OUT2	8	
		Digital output OUT3	9	
		Digital output OUT4	10	
		Digital output OUT5	11	
		Digital output OUT6	12	
		Digital output OUT7	13	
		Digital output OUT8	14	
		Digital output OUT9	15	
		Digital output OUT10	16	
		Digital output OUT11	17	
		Digital output OUT12	18	
		Limit value output 1	19	
		Limit value output 2	20	
		Limit value output 3	21	
		Limit value output 4	22	
		Limit value output 5	23	
Limit value output 6	24			
Limit value output 7	25			
Limit value output 8	26			
Limit value output 9	27			
Limit value output 10	28			
Limit value output 11	29			
Limit value output 12	30			
Limit value output 13	31			
Empty byte	USINT	–		Not used

4 The GSDML file

DeviceStatusBlock start-up parameter

Parameter	Type	Values	Explanation
Byte sequence for cyclical data	Bit	0 (Big Endian) 1 (Little Endian)	Selection of byte sequence for transferred cyclical data The byte sequence to be set is specified by the IO controller. Please refer to the documentation for your IO controller
Byte sequence for non-cyclical data	Bit	0 (Big Endian) 1 (Little Endian)	Selection of byte sequence for transferred noncyclical data The byte sequence to be set is specified by the IO controller. Please refer to the documentation for your IO controller
IOPS handling	Bit	0 (OFF) 1 (ON)	Handling of the IOPS status via the IO device 0 (OFF): The IOPS status of the relevant IO device slot is valid following successful system boot and no longer depends on the validity of the input data. The validity check of the input values must be implemented in the IO controller if necessary. 1 (ON): The IOPS status of the respective IO device slot communicates to the IO controller the validity status of the input values following successful system boot. If the IOPS status is valid, all values of the slot are valid. If at least 1 invalid input value occurs in the slot concerned, the IOPS status is likewise set to invalid. In case of invalid float values (REAL type), JUMO field devices map an error code in the transferred input values. As a result, the error number rather than the input value is contained. ⇒ chapter 7.1 "Error messages for invalid values", Page 57

Configurable modules

The configurable modules are intended exclusively for transferring cyclical IO data. The assignment of the modules can be compiled and printed via the JUMO PC setup program of the DICON touch. The printed list will help you maintain an overview of the configured modules of the JUMO IO device during the project planning and programming of the IO controller (PLC, host PC or similar).

⇒ chapter 6.2 "Configuring PROFINET modules", Page 46

The JUMO DICON touch supports up to 10 slots in total (9 configurable slots + DeviceStatusBlock) in the device and 10 slots in the IO controller project. The possible types of configurable modules are listed in the table below.

4 The GSDML file

Module types	IO items	Data type	Data direction IO controller
Analog input values	Analog values 1 to 16,	REAL	read
Digital input values	Digital values 1 to 32	BOOL	read
Integer input values	Integers 1 to 8	DINT	read
Analog output values	Analog values 1 to 8	REAL	write
Digital output values	Digital values 1 to 3	BOOL	write

4 The GSDML file

5 Noncyclical data exchange



CAUTION!

All temperature values of the JUMO IO device are transferred in the unit "°C".

Misinterpretation of measured values of a JUMO IO device in the IO controller can cause errors in the system control.

- ▶ Note the unit for the transferred temperature values!

5.1 Programming the noncyclical data exchange in the IO controller

In addition to the cyclical data exchange between IO controller and IO device in the RT channel, PROFINET IO also offers the option of event-controlled noncyclical data exchange. The noncyclical communication is controlled by the IO controller (similar to the master-slave principle) via write/read requests and has to be implemented by the user. Noncyclical data are provisioned by PROFINET IO users as "Record Data". The transfer occurs in the NRT channel.

For the programming of write/read requests, the engineering systems of the various manufacturers contain libraries with relevant function blocks such as "RDREC" (Read Record) and "WRREC" (Write Record).

With JUMO PROFINET IO devices, the write/read requests do not access the "Record Data" directly. Instead, they transfer data exchange packets between IO controller and IO device, which are further processed in the background by the field devices. For an IO controller to be able to exchange noncyclical data with a JUMO IO device, appropriate data structures (data blocks/data type objects) have to be created in the IO controller for the data exchange packet concerned. These data structures provide the memory for the outgoing and incoming data exchange packets. Data exchange packets are identified by an index, which has to be transferred to the write/read function blocks in the form of parameters. JUMO PROFINET IO devices have 3 indices for 3 different types of data exchange packets, which are transferred via the write/read commands. The table below lists the available indices of JUMO field devices.

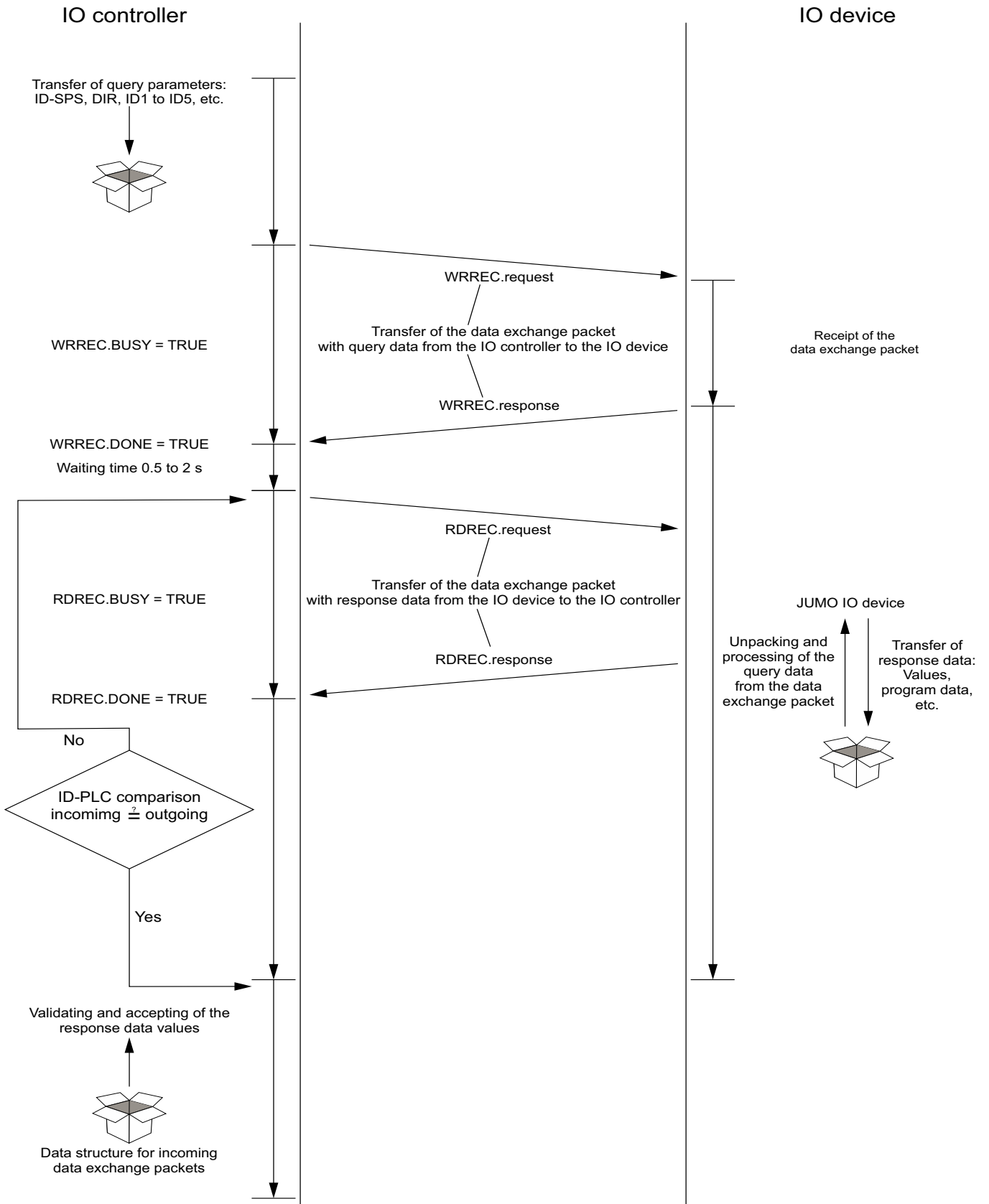
Index	Data exchange packet	Explanation
201	Single-ID	For the write and read transfer of single noncyclical data points with a length of up to 243 bytes within a data exchange sequence Length of the data exchange packet: 255 bytes
202	Multi-ID	For the write and read transfer of multiple noncyclical data points within a data exchange sequence, the single data points are allowed to be up to 4 bytes long. Length of the data exchange packet: 640 bytes
203	Program memory	For the write and read transfer of program memories (on program controllers) within a data exchange sequence Length of the data exchange packet: 1024 bytes

Within a data exchange sequence, the IO controller transfers a data exchange packet to the IO device. The IO device processes the data exchange packet and re-provisions it for pick-up by the IO controller (response with data or status messages). In order to control processing by the IO device (e.g. defining the data direction and selection of noncyclical data), the data exchange packet must be parameterized accordingly.

With JUMO IO devices, the noncyclical services are always assigned to Slot 1. In JUMO IO devices, Slot 1 is always pre-configured. Consequently, the noncyclical services are not dependent upon the module configuration or project planning in the IO controller, and are available as soon as the system has successfully booted.

5 Noncyclical data exchange

Sequence of a WRREC-RDREC cycle for transferring data exchange packets



5 Noncyclical data exchange

5.1.1 Single-ID and Multi-ID

To implement the exchange of single (Single-ID) or multiple (Multi-ID) noncyclical data points, you must take the following steps:

- Declare 1 data structure each for both the outgoing and the incoming direction. Select the "Single-ID" format to transfer 1 data point within a data exchange sequence. Select the "Multi-ID" format to transfer up to 4 data points within a data exchange sequence.
- Program a sequence control for the data exchange sequence.

The procedure is described below.

5 Noncyclical data exchange

Declaration of a data structure in the "Single-ID" format

In the IO controller, you need to declare 1 data structure each (user-defined "STRUCT variable") for incoming and outgoing data exchange packets as the target and source memory for data exchange packets. Data exchange packets in the "Single-ID" format have a fixed length of 65 bytes and must be created with the following structure:

Structure variable for data exchange packets in the "Single-ID" format

Data type	Name	Explanation
BYTE	ID-PLC	Sequential numbering for assigning the data exchange packets from queries and responses The "ID-PLC" value can be used by the IO controller for sequentially numbering the outgoing data exchange packets. JUMO IO devices enter the same number in the ID-PLC in the data exchange packet of the ensuing response, so that response and query can be assigned at the IO controller, or so that an error in the sequence of query and response can be intercepted by suitable control structures in the IO controller.
BYTE	Outgoing: DIR Incoming: ERROR	Data structure for outgoing data exchange packets: „DIR“ ist die Data direction for transferring noncyclical data. The data direction must be specified by the PLC programmer in the data exchange packet of the query from the IO controller and controls the processing of the data in the JUMO IO device accordingly. Coding: Value = 1: Write (from IO controller to IO device) Value = 2: Read (from IO device to IO controller) Data structure for incoming data exchange packets: In case of "ERROR", the JUMO IO device enters an integer value as error indicator. In the IO controller, "ERROR" can be evaluated via a corresponding control structure. List of error values: ⇒ chapter 7 "Error messages", Page 57
WORD	ID1	5-element ID of the data point from the noncyclical data table (see chapter 5.2 "Data tables of noncyclical data", Page 36)
WORD	ID2	
WORD	ID3	
WORD	ID4	
WORD	ID5	
ARRAY[53 bytes] (e.g. REAL, INT, BYTE)	VALUE	Data point to be queried or overwritten of the noncyclical data with a length of 53 bytes; however, only a maximum of 4 bytes are currently used. Value to be read/written of the data point; this variable can be declared as any data type with a length of 4 bytes.

5 Noncyclical data exchange

Declaration of a data structure in the "Multi-ID" format

In the IO controller, you need to declare 1 data structure each (user-defined "STRUCT variable") for incoming and outgoing data exchange packets as the target and source memory for incoming/outgoing data exchange packets. Data exchange packets in the "Multi-ID" format have a fixed length of 65 bytes and must be created with the following structure:

Structure variable for data exchange packets in the "Multi-ID" format

Data type	Name	Explanation
BYTE	ID-PLC	Sequential numbering for assigning the data exchange packets from queries and responses The "ID-PLC" value must be used by the IO controller for sequentially numbering the outgoing data exchange packets. JUMO IO devices enter the same number in the ID-PLC in the data exchange packet of the ensuing response, so that response and query can be assigned at the IO controller, or so that an error in the sequence of query and response can be intercepted by suitable control structures in the IO controller.

5 Noncyclical data exchange

Structure variable for data exchange packets in the "Multi-ID" format

Data type	Name	Explanation
BYTE	DIR_1 ^a	Query/re-response data for data point 1
BYTE	ERROR_1 ^a	
WORD	ID1_1 ^a	
WORD	ID2_1 ^a	
WORD	ID3_1 ^a	
WORD	ID4_1 ^a	
WORD	ID5_1 ^a	
Any basic data type with a length of 32 bits	VALUE_1 ^a	
.	.	Query/re-response data for data points 2 and 3 with identical structure
.	.	
.	.	
BYTE	DIR_32 ^a	Query/re-response data for data point 4
BYTE	ERROR_32 ^a	
WORD	ID1_32 ^a	
WORD	ID2_32 ^a	
WORD	ID3_32 ^a	
WORD	ID4_32 ^a	
WORD	ID5_32 ^a	
Any basic data type with a length of 32 bits	VALUE_32 ^a	

Data for transferring up to 4 data points of noncyclical data; the following variables are declared for each data point:

DIR: Data direction for transferring the respective data point;
The data direction of each data point must be specified by the PLC programmer in the data exchange packet of the query from the IO controller. The "DIR" variables control the processing of the single data points in the JUMO IO device accordingly.

Coding:
Value = 1: Write (from IO controller to IO device)
Value = 2: Read (from IO device to IO controller)

ERROR:
Data structure for incoming data exchange packets:

In case of "ERROR", the JUMO IO device enters an integer value as error indicator. In the IO controller, "ERROR" can be evaluated via a corresponding control structure.

List of error values:
⇒ chapter 7 "Error messages", Page 57

ID1 to ID5: 5-element ID of the respective data point from the noncyclical data table
⇒ chapter 5.2 "Data tables of noncyclical data", Page 36

VALUE: Value to be read/written of the respective data point; this variable must always be declared as a basic data type with a length of 32 bits. If the query aims at an IO device data type with 16 or 8 bits, only the corresponding lower-value bytes are relevant. The higher-value bytes are to be considered pure fill bits.

^a In the "Multi-ID" format, all variables are declared and also transferred for 4 data points.



CAUTION!

The data exchange packets have a predefined format.

If the data structures are not filled with fill bytes to the correct length, errors can occur during the data exchange.

- ▶ Be sure to respect the data structure defined above (data types, sequence of the variables and total length, including fill bytes) when declaring the data structures for data exchange packets.

5 Noncyclical data exchange

Data exchange sequence for data exchange packets in the "Single-ID" and "Multi-ID" formats

With JUMO IO devices, the noncyclical data are not directly accessed via the "WRREC" and "RDREC" PROFINET IO function blocks. Instead, data exchange packets that are processed by the IO device are transferred. The IO controller has to write a data exchange packet with query data to the JUMO IO device (WRREC). JUMO IO devices accept the data from the data exchange packet and then prepare this packet with response data for the IO controller. The IO controller must pick up this data exchange packet with response data again (RDREC). The data exchange packets in the "Single-ID" format can be used to transfer single data points within a data exchange sequence. The "Multi-ID" format can be used to transfer up to 4 data points within a data exchange sequence. This requires a suitable sequence control to be implemented in the IO controller. The sequence control of a data exchange sequence in the "Single-ID" and "Multi-ID" formats can be designed as follows:

Sequence control of a data exchange sequence for single-ID data exchange packets

1. **Parameterize data structure for outgoing data exchange packets with query data:** To be able to correctly control the processing of the query in the JUMO IO device, the "ID-PLC" count variable is incremented. It can be used to check the affiliation of query and response data. In the "DIR" variable, the transfer direction (write/read) for the single data points must be defined (see description of the data structures for the "Single-ID" and "Multi-ID" formats above). The 5-element IDs of the respective data points to be written to/read in the JUMO IO device are entered into fields ID1 to ID5 (see chapter 5.2 "Data tables of noncyclical data", Page 36). For write data exchange sequences (IO controller sends data points to IO device), the values to be transferred must be entered into the "VALUE" variables. For queries in "Multi-ID" format, the number of data points to be transferred must be entered into the "NUMBER" variable. The first of the indicated number of corresponding data points are read/written. The subsequent data points are not processed by the JUMO IO device.
2. **Transfer the data exchange packet to the IO device:** The data exchange packet must now be transferred to the IO device. This is done by calling up the write function (WRREC) in the IO controller. The parameters that have to be transferred at call-up appear in the table at the end of this section.
3. **Wait for the successfully completed transfer of the outgoing data exchange packet:** The JUMO IO device acknowledges the "Write Request" of the IO controller following successful transfer (Write Response). "WRREC.DONE", "WRREC.BUSY", "WRREC.ERROR" and "WRREC.STATUS" can be evaluated to be able to query the status of the transfer. The JUMO IO device reports back WRREC.DONE = TRUE to the IO controller to signal successful receipt of the data exchange packet. The JUMO IO device starts to process the transferred query data. At this point, the IO controller should respect a waiting period of 0.5 to 2 seconds before continuing with the next steps in the data exchange sequence.
4. **Pick-up of the data exchange packet with response data from the IO device:** The IO controller must pick-up data exchange packets from the device in cycles and check their validity as response data by means of "ID-PLC" (polling). If the values of outgoing and incoming "ID-PLC" match, valid response data have been read from the JUMO IO device. The IO controller can then end the polling process and accept the response data from the data structure of the incoming data exchange packets.
 - a) **Polling:** The IO controller has to pick-up, via cyclical read accesses (RDREC), data exchange packets from the JUMO IO device, place them in the data structure for incoming data exchange packets and compare the "ID-PLC" variables in the data structures for outgoing and incoming data exchange packets with one another (polling). Unless the outgoing and incoming values of "ID-PLC" are identical, processing in the JUMO IO device is still ongoing and the IO controller must continue with the polling process. If the values of outgoing and incoming "ID-PLC" match, the IO controller has received valid response data from the JUMO IO device and polling can be ended.

In order to query the status of the individual RDREC transfers within the poll cycles, "RDREC.VALID", "RDREC.BUSY", "RDREC.ERROR" and "RDREC.STATUS" can be evaluated. When RDREC.VALID = TRUE, the data exchange packet has been successfully received and transferred to the data structure for incoming data exchange packets. Following the successful transfer of a data exchange packet, outgoing and incoming "ID-PLC" must be compared and

5 Noncyclical data exchange

a decision made as to whether a further poll cycle is required ("ID-PLC" different) or valid response data were received ("ID-PLC" identical). The parameters that need to be transferred at cyclical call-ups of RDREC appear in the table at the end of this section.

- b) **Acceptance of response data:** As soon as a data exchange packet is received from the JUMO IO device and the values for "ID-PLC" in the data structures for outgoing and incoming data exchange packets are identical, the data structure for incoming data exchange packets contains the valid response data from the JUMO IO device. The data have to be copied from here to the target before they are overwritten by another read request.

Parameterizing function blocks WRREC/RDREC

Input parameters for WRREC/ RDREC	Transfer values	
	Single-ID	Multi-ID
LEN (Length of the data to be written for WRREC in bytes)	65	65
MLEN (Maximum length of the data to be read for RDREC in bytes)		
ID (Hardware detection of slots/subslots to be addressed on the IO device)	Hardware detection of Slot 1 on JUMO IO devices (DeviceStatusBlock)	
INDEX (Index of the target range for data exchange packets of JUMO IO devices)	201	202
RECORD (Pointer on data points to be read/written; with JUMO IO devices, data structures for incoming and outgoing data exchange packets are transferred.)	for WRREC: Pointer on the data structure for outgoing data exchange packets for RDREC: Pointer on the data structure for incoming data exchange packets	

5.1.2 Programs for program controllers

Program memories must be split within a data exchange sequence over multiple partial data records and transferred in succession. The individual partial data records are transferred packed in data exchange packets. For the outgoing and incoming data exchange packets, data structures in the "Program data transfer" format must be created in the IO controller as target and source memories. The data exchange packets with the partial data records are transferred in succession within a data exchange sequence in separate WRREC-RDREC cycles. To allow the "program data transfer" type data structures to be overwritten again during the WRREC-RDREC cycle, the content of these structures between the WRREC-RDREC cycle is synchronized with a further data structure, which holds ready for sending all data of the entire program memory to be transferred and/or accepts them at the point of receipt. This data structure must be declared in the IO controller for both the incoming and outgoing direction with the "Editing memory" format. For the complete transfer of a program memory, up to 3 partial data records have to be transferred:

- 1) 1 partial data record for program names
- 2) up to 2 partial data records for program segments, each with up to 25 program sections

5 Noncyclical data exchange

To implement the exchange of program memories, you must take the following steps:

- Declare 1 data structure in the "Program data transfer" format (partial data records of a program memory to be transferred) for both the outgoing and the incoming direction.
- Declare 1 data structure in the "Editing memory" format (total data of the program memory to be transferred) for both the outgoing and the incoming direction.
- Program a sequence control for the data exchange sequence.

The procedure is described below.

5 Noncyclical data exchange

Declaration of data structures for the transfer of program memories

In the IO controller, you need to declare 1 data structure (user-defined "STRUCT variable") in the "Program data transfer" format as both the target and the source memory for all data exchange packets for incoming and outgoing transfer direction, and one data structure for provisioning and one for accepting all data of a program memory during the data exchange sequence in the "editing memory" format.

Data structures in "Editing memory" format

The data structures in "Editing memory" format contain all the data for a program memory to be transferred in an incoming/outgoing direction. They must be created with the following structure:

5 Noncyclical data exchange

Structure variable for program data in "Editing memory" format

Data type	Name	Explanation	
REAL	Setpoint 1 [°C]	Section 1	<p>Program sections:</p> <p>Data records of individual program sections 1 to 50</p> <p>When transferring more than 25 program sections, the data of the program sections is split, due to the data volume, into 2 program sections each containing a maximum of 25 program sections, and transferred as separate partial data records.</p> <p>For more details about the individual data of a program section, see the operating manual for the JUMO DICON touch.</p> <p>The dummy variables are strictly fill bytes. They are used to satisfy requirements for certain data structures)</p>
REAL	Setpoint 2 [°C]		
REAL	Tolerance band min.		
REAL	Tolerance band max.		
UDINT	Section time		
REAL	Gradient		
WORD	Operating contacts		
USINT	Number of repetitions		
USINT	Start section		
USINT	Parameter block number		
USINT	Gradient programming		
UINT	Dummy		
.	.	Sections 2 to 49	
.	.		
.	.		
REAL	Setpoint 1 [°C]	Section 50	
REAL	Setpoint 2 [°C]		
REAL	Tolerance band min.		
REAL	Tolerance band max.		
UDINT	Section time		
REAL	Gradient		
WORD	Operating contacts		
USINT	Number of repetitions		
USINT	Start section		
USINT	Parameter block number		
USINT	Gradient programming		
UINT	Dummy		

5 Noncyclical data exchange

Structure variable for program data in "Editing memory" format

Data type	Name	Explanation
BYTE[73]	Program name	Partial data record for program names Program name: 24 characters containing 3 bytes each (array elements 0 to 71) for Unicode characters (UTF-8) The last byte (array element 72) contains the 1-byte long character for the end code (NULL). Icon: The "Icon" variable determines the selection of one of the icons in the JUMO IO device for visualizing the program profile on the JUMO IO device display. Dummy: Fill bytes are required in order to guarantee the length of 960 bytes (corresponds to the length of the data fields in "program data transfer" type structures).
UINT	Icon	
BYTE[885]	Dummy	

5 Noncyclical data exchange

Data structures in "Program data transfer" format

Data exchange packets in the "Transfer data" format have a fixed length of 1024 bytes and must be created with the following structure:

Structure variable for data exchange packets in the "Program data transfer" format

Data type	Name	Explanation
BYTE	ID-PLC	Sequential numbering for assigning the data exchange packets from queries and responses The "ID-PLC" value can be used by the IO controller for sequentially numbering the outgoing data exchange packets. JUMO IO devices enter the same number in the ID-PLC in the data exchange packet of the ensuing response, so that response and query can be assigned at the IO controller, or so that an error in the sequence of query and response can be intercepted by suitable control structures in the IO controller.
BYTE	DIR	Data direction for transferring noncyclical data The data direction must be specified by the PLC programmer in the data exchange packet of the query from the IO controller and controls the processing of the data in the JUMO IO device accordingly. Coding: Value = 1: Write (from IO controller to IO device) Value = 2: Read (from IO device to IO controller)
BYTE	EditFunct	This value is used for internal diagnostic purposes. It is irrelevant to the processing of a data exchange sequence and can be ignored.
BYTE	ProgNo	Number of the program to be transferred
BYTE	SecNumber	Writing: Total number of all sections of a program, which are to be written from the IO controller in the current data exchange sequence to the JUMO IO device. In the last WRREC-RDREC cycle of a data exchange sequence, this parameter must be assigned the number of sections of the program to be written in the JUMO IO device. In the JUMO IO device, it controls the number of program sections to be accepted. This parameter has no function for reading.
BYTE	PartSecNo	First program section number of the program segment in a data exchange packet to be transferred The program segments to be transferred (up to 2) must contain, without interruption, all sections of a program, starting from section number 1 up to the last section number of the program sections to be transferred.
BYTE	PartSecNumber	Number of program sections (max. 25) to be written/read in the program segment to be transferred in this data exchange packet
BYTE	isProgName	Indicator for program name: Indicates whether the data exchange packet transfers a partial data record for program names. Value= 0: The partial data record in this data exchange packet is not a program name (it is instead a program section). Value= 1: The partial data record in this data exchange packet is a program name.

5 Noncyclical data exchange

Structure variable for data exchange packets in the "Program data transfer" format

Data type	Name	Explanation
BYTE	ErrorCode	In case of an error, the JUMO IO device enters into this variable error messages, which can be evaluated by the IO controller. Value = 0: No error Value = 1: Program does not exist
BYTE	Dummy	Fill byte (no function; used to satisfy requirements for certain data structures)
UINT	Dummy	
BYTE[960]	Data	The partial data record for program name or program segment to be transferred; The data of the partial data record to be transferred are entered here. The data field must have a total length of 960 bytes. The data format of the data field corresponds to the formats of the partial data records for program names and program sections in the data structures of the "Editing memory" type.
BYTE[52]	Dummy	Fill byte array for filling the data exchange packet to the fixed size of 1024 bytes



CAUTION!

The data exchange packets have a predefined format.

If the data structures are not filled with fill bytes to the correct length, errors can occur during the data exchange.

- ▶ Be sure to respect the data structure defined above (data types, sequence of the variables and total length, including fill bytes) when declaring the data structures for data exchange packets.
-

5 Noncyclical data exchange

Data exchange sequence for data exchange packets in the "Program data transfer" formats

With JUMO IO devices, the noncyclical data are not directly accessed via the "WRREC" and "RDREC" PROFINET IO function blocks. Instead, data exchange packets that are processed by the IO device are transferred. The IO controller has to write a data exchange packet with query data to the JUMO IO device (WRREC). JUMO IO devices accept the data from the data exchange packet and then prepare this packet with response data for the IO controller. The IO controller must pick up this data exchange packet with response data again (RDREC). All data for a program profile (setpoint profile of program controllers) can be transferred in a data exchange sequence via data exchange packets in "Program memory" format. This requires a suitable sequence control to be implemented in the IO controller. In contrast to the "Single-ID" and "Multi-ID" formats, multiple data exchange packets within a data exchange sequence are transferred in succession with the "program data transfer" format. In order to completely transfer all the data of a program, up to 3 WRREC-RDREC cycles must occur within a data exchange sequence. 1 partial data record of the entire program is transferred in each cycle:

- 1) 1 partial data record for program names
- 2) up to 2 partial data records for program segments, each with up to 25 program sections

The partial data records can be transferred in any sequence. However, when writing programs to the JUMO IO device, it must be ensured that in the last WRREC-RDREC cycle, the number of sections of the program to be written is transferred in order to correctly control the acceptance of the program in the JUMO IO device at the end of the data exchange sequence (once all partial data records have been transferred). For this purpose, the total number of sections of the program to be written is assigned to the "SecNumber" variable in the data structure for outgoing data exchange packets ("Program data transfer" format) during the last WRREC-RDREC cycle.

The sequence control of the individual WRREC-RDREC cycles for data exchange packs in the "Program data transfer" format can be executed as follows:

Sequence control of the individual WRREC-RDREC cycles within a data exchange sequence for program memory data

1. **Parameterize data structure in the "Editing memory" format with query data:** The incoming and outgoing "Editing memories" are required for buffering the total data of program memories across the individual steps of a data exchange sequence. For write data exchange sequences, the program data to be transferred to the JUMO IO device must be entered into the "Editing memory" for outgoing program data.
The "Editing memory" for incoming data does not have to be filled. It accepts the transferred partial data records during the read and collects them. At the end of the complete data exchange sequence, it contains all data of the program read from the JUMO IO device.
2. **Parameterize data structure for outgoing data exchange packets:** To be able to correctly control the processing of the query in the JUMO IO device, the "ID-PLC" count variable is incremented. It can be used to check the affiliation of query and response data. In the "DIR" variable, the transfer direction for the program data must be defined (see description of the data structures for the "Program data transfer" format above). Parameters ProgNo., PartSecNo. PartSecNumber and isProgName serve to identify the partial data record to be transferred. These parameters must be indicated for each WRREC-RDREC cycle according to the partial data record to be transferred. For writing programs, in the last WRREC-RDREC cycle of the data exchange sequence, the "SecNumber" parameter must be assigned the number of sections of the program to be transferred. "SecNumber" has no function for reading programs. The meaning of the individual parameters is explained in the table of data structures in the "Program data transfer" format above.
For write data exchange sequences of program memories, program data have to be transferred in the data field of the outgoing data exchange packet. This requires the partial data record to be transferred during the upcoming WRREC-RDREC cycle to be copied from the "Editing memory" for outgoing program data to the data field of the "program data transfer" data structure for the outgoing data exchange packets.
3. **Transfer the data exchange packet to the IO device:** The data exchange packet must now be transferred to the IO device. This is done by calling up the write function (WRREC) in the IO controller. The parameters that have to be transferred at call-up appear in the table at the end of this section.

5 Noncyclical data exchange

4. **Wait for the successfully completed transfer of the outgoing data exchange packet:** The JUMO IO device acknowledges the "Write Request" of the IO controller following successful transfer of the data exchange packet (Write Response). "WRREC.DONE", "WRREC.BUSY", "WRREC.ERROR" and "WRREC.STATUS" can be evaluated to be able to query the status of the transfer. The JUMO IO device reports back WRREC.DONE = TRUE to the IO controller to signal successful receipt of the data exchange packet. The JUMO IO device starts to process the transferred query data. At this point, the IO controller should respect a waiting period of 0.5 to 2 seconds before continuing with the next steps in the data exchange sequence.
5. **Pick-up of the data exchange packet with response data from the IO device:** The IO controller must pick-up data exchange packets from the device in cycles and check their validity as response data by means of "ID-PLC" (polling). If the values of outgoing and incoming "ID-PLC" match, valid response data have been read from the JUMO IO device. The IO controller can then end the polling process and accept the response data from the data structure of the incoming data exchange packets.
 - a) **Polling:** The IO controller has to pick-up, via cyclical read accesses (RDREC), data exchange packets from the JUMO IO device, place them in the data structure for incoming data exchange packets and compare the "ID-PLC" variables in the data structures for outgoing and incoming data exchange packets with one another (polling). Unless the outgoing and incoming values of "ID-PLC" are identical, processing in the JUMO IO device is still ongoing and the IO controller must continue with the polling process. If the values of outgoing and incoming "ID-PLC" match, the IO controller has received valid response data from the JUMO IO device and polling can be ended.

In order to query the status of the individual RDREC transfers within the poll cycles, "RDREC.VALID", "RDREC.BUSY", "RDREC.ERROR" and "RDREC.STATUS" can be evaluated. When RDREC.VALID = TRUE, the data exchange packet has been successfully received and transferred to the data structure for incoming data exchange packets. Following the successful transfer of a data exchange packet, outgoing and incoming "ID-PLC" must be compared and a decision made as to whether a further poll cycle is required ("ID-PLC" different) or valid response data were received ("ID-PLC" identical). The parameters that need to be transferred at cyclical call-ups of RDREC appear in the table at the end of this section.
 - b) **Acceptance of response data:** As soon as a data exchange packet is received from the JUMO IO device and the values for "ID-PLC" in the data structures for outgoing and incoming data exchange packets are identical, the data structure for incoming data exchange packets contains the valid response data from the JUMO IO device. The data have to be copied from here to the target before they are overwritten by another read request.

For read data exchange sequences of program memories, the program data from the data field of the incoming data exchange packet must be accepted into the "Editing memory" for incoming program data and collected across the entire data exchange sequence. This requires the partial data record to be copied from the data field of the data structure in the "Program data transfer" format for incoming data exchange packets to the corresponding partial data record of the "Editing memory" for incoming program data.



NOTE!

The described sequence of a WRREC-RDREC cycle must be executed within a data exchange sequence for every partial data record. In contrast to the "Single-ID" and "Multi-ID" formats, a data exchange sequence for the "program data transfer" format therefore has a total of 3 WRREC-RDREC cycles.

5 Noncyclical data exchange

Parameterizing function blocks WRREC/RDREC

Input parameters for WRREC/ RDREC	Transfer values
LEN (Length of the data to be written for WRREC in bytes)	1024
MLEN (Maximum length of the data to be read for RDREC in bytes)	
ID (Hardware detection of slots/subslots to be addressed on the IO device)	Hardware detection of Slot 1 on JUMO IO devices (DeviceSta- tusBlock)
INDEX (Index of the target range for data ex- change packets of JUMO IO devices)	203
RECORD (Pointer on data points to be read/ written; with JUMO IO devices, data structures for incoming and outgoing data exchange packets are trans- ferred.)	for WRREC: Pointer on the data structure for outgoing data ex- change packets for RDREC: Pointer on the data structure for incoming data ex- change packets

5 Noncyclical data exchange

5.2 Data tables of noncyclical data



CAUTION!

All temperature values of the JUMO IO device are transferred in the unit "°C".

Misinterpretation of measured values of a JUMO IO device in the IO controller can cause errors in the system control.

- ▶ Note the unit for the transferred temperature values!

5.2.1 Configuration data and parameters



CAUTION!

Write operations of Read/Write parameters result in saving to the flash memory. This memory accommodates only a limited number of write cycles.

Writing to it with constantly changing values can therefore result in a memory error in the event of a power failure.

- ▶ Avoid rapid write accesses with constantly changing values!

Limit values for limit value monitoring 1 to 16

Index					Signal designation	Data type	Access
2	129	0	3	0	Limit value for limit value monitoring 1	REAL	Read/Write
2	129	1	3	0	Limit value for limit value monitoring 2	REAL	Read/Write
2	129	2	3	0	Limit value for limit value monitoring 3	REAL	Read/Write
2	129	3	3	0	Limit value for limit value monitoring 4	REAL	Read/Write
2	129	4	3	0	Limit value for limit value monitoring 5	REAL	Read/Write
2	129	5	3	0	Limit value for limit value monitoring 6	REAL	Read/Write
2	129	6	3	0	Limit value for limit value monitoring 7	REAL	Read/Write
2	129	7	3	0	Limit value for limit value monitoring 8	REAL	Read/Write
2	129	8	3	0	Limit value for limit value monitoring 9	REAL	Read/Write
2	129	9	3	0	Limit value for limit value monitoring 10	REAL	Read/Write
2	129	10	3	0	Limit value for limit value monitoring 11	REAL	Read/Write
2	129	11	3	0	Limit value for limit value monitoring 12	REAL	Read/Write
2	129	12	3	0	Limit value for limit value monitoring 13	REAL	Read/Write
2	129	13	3	0	Limit value for limit value monitoring 14	REAL	Read/Write
2	129	14	3	0	Limit value for limit value monitoring 15	REAL	Read/Write
2	129	15	3	0	Limit value for limit value monitoring 16	REAL	Read/Write

Ramp slope, controllers 1; 2

Index					Signal designation	Data type	Access
2	127	0	1	0	Ramp slope for positive gradient, controller 1	REAL	Read/Write
2	127	0	2	0	Ramp slope for negative gradient, controller 1	REAL	Read/Write
2	127	1	1	0	Ramp slope for positive gradient, controller 2	REAL	Read/Write
2	127	1	2	0	Ramp slope for negative gradient, controller 2	REAL	Read/Write

5 Noncyclical data exchange

Controller 1 parameter block 1

Index					Signal designation	Data type	Access
2	107	0	2	0	XP1 proportional band (P component)	REAL	Read/Write
2	107	0	3	0	XP2 proportional band (P component)	REAL	Read/Write
2	107	0	4	0	TV1 derivative time (D component)	REAL	Read/Write
2	107	0	5	0	TV2 derivative time (D component)	REAL	Read/Write
2	107	0	6	0	TN1 reset time (I component)	REAL	Read/Write
2	107	0	7	0	TN2 reset time (I component)	REAL	Read/Write
2	107	0	8	0	CY1 cycle time	REAL	Read/Write
2	107	0	9	0	CY2 cycle time	REAL	Read/Write
2	107	0	10	0	XSH switching hysteresis between 1st & 2nd digital output	REAL	Read/Write
2	107	0	11	0	XD1 switching differential of the 1st digital output	REAL	Read/Write
2	107	0	12	0	XD2 switching differential of the 2nd digital output	REAL	Read/Write
2	107	0	13	0	TT actuator time	INT	Read/Write
2	107	0	14	0	Y0 working point	INT	Read/Write
2	107	0	15	0	Y1 output value limit	INT	Read/Write
2	107	0	16	0	Y2 output value limit	INT	Read/Write
2	107	0	17	0	TK1 minimum start time of the 1st digital output	REAL	Read/Write
2	107	0	18	0	TK2 minimum start time of the 2nd digital output	REAL	Read/Write

Controller 1 parameter block 2

Index					Signal designation	Data type	Access
2	107	1	2	0	XP1 proportional band (P component)	REAL	Read/Write
2	107	1	3	0	XP2 proportional band (P component)	REAL	Read/Write
2	107	1	4	0	TV1 derivative time (D component)	REAL	Read/Write
2	107	1	5	0	TV2 derivative time (D component)	REAL	Read/Write
2	107	1	6	0	TN1 reset time (I component)	REAL	Read/Write
2	107	1	7	0	TN2 reset time (I component)	REAL	Read/Write
2	107	1	8	0	CY1 cycle time	REAL	Read/Write
2	107	1	9	0	CY2 cycle time	REAL	Read/Write
2	107	1	10	0	XSH switching hysteresis between 1st & 2nd digital output	REAL	Read/Write
2	107	1	11	0	XD1 switching differential of the 1st digital output	REAL	Read/Write
2	107	1	12	0	XD2 switching differential of the 2nd digital output	REAL	Read/Write
2	107	1	13	0	TT actuator time	INT	Read/Write
2	107	1	14	0	Y0 working point	INT	Read/Write
2	107	1	15	0	Y1 output value limit	INT	Read/Write
2	107	1	16	0	Y2 output value limit	INT	Read/Write
2	107	1	17	0	TK1 minimum start time of the 1st digital output	REAL	Read/Write
2	107	1	18	0	TK2 minimum start time of the 2nd digital output	REAL	Read/Write

5 Noncyclical data exchange

Controller 1 parameter block 3

Index					Signal designation	Data type	Access
2	107	2	2	0	XP1 proportional band (P component)	REAL	Read/Write
2	107	2	3	0	XP2 proportional band (P component)	REAL	Read/Write
2	107	2	4	0	TV1 derivative time (D component)	REAL	Read/Write
2	107	2	5	0	TV2 derivative time (D component)	REAL	Read/Write
2	107	2	6	0	TN1 reset time (I component)	REAL	Read/Write
2	107	2	7	0	TN2 reset time (I component)	REAL	Read/Write
2	107	2	8	0	CY1 cycle time	REAL	Read/Write
2	107	2	9	0	CY2 cycle time	REAL	Read/Write
2	107	2	10	0	XSH switching hysteresis between 1st & 2nd digital output	REAL	Read/Write
2	107	2	11	0	XD1 switching differential of the 1st digital output	REAL	Read/Write
2	107	2	12	0	XD2 switching differential of the 2nd digital output	REAL	Read/Write
2	107	2	13	0	TT actuator time	INT	Read/Write
2	107	2	14	0	Y0 working point	INT	Read/Write
2	107	2	15	0	Y1 output value limit	INT	Read/Write
2	107	2	16	0	Y2 output value limit	INT	Read/Write
2	107	2	17	0	TK1 minimum start time of the 1st digital output	REAL	Read/Write
2	107	2	18	0	TK2 minimum start time of the 2nd digital output	REAL	Read/Write

Controller 1 parameter block 4

Index					Signal designation	Data type	Access
2	107	3	2	0	XP1 proportional band (P component)	REAL	Read/Write
2	107	3	3	0	XP2 proportional band (P component)	REAL	Read/Write
2	107	3	4	0	TV1 derivative time (D component)	REAL	Read/Write
2	107	3	5	0	TV2 derivative time (D component)	REAL	Read/Write
2	107	3	6	0	TN1 reset time (I component)	REAL	Read/Write
2	107	3	7	0	TN2 reset time (I component)	REAL	Read/Write
2	107	3	8	0	CY1 cycle time	REAL	Read/Write
2	107	3	9	0	CY2 cycle time	REAL	Read/Write
2	107	3	10	0	XSH switching hysteresis between 1st & 2nd digital output	REAL	Read/Write
2	107	3	11	0	XD1 switching differential of the 1st digital output	REAL	Read/Write
2	107	3	12	0	XD2 switching differential of the 2nd digital output	REAL	Read/Write
2	107	3	13	0	TT actuator time	INT	Read/Write
2	107	3	14	0	Y0 working point	INT	Read/Write
2	107	3	15	0	Y1 output value limit	INT	Read/Write
2	107	3	16	0	Y2 output value limit	INT	Read/Write
2	107	3	17	0	TK1 minimum start time of the 1st digital output	REAL	Read/Write
2	107	3	18	0	TK2 minimum start time of the 2nd digital output	REAL	Read/Write

5 Noncyclical data exchange

Controller 2 parameter block 1

Index					Signal designation	Data type	Access
2	107	4	2	0	XP1 proportional band (P component)	REAL	Read/Write
2	107	4	3	0	XP2 proportional band (P component)	REAL	Read/Write
2	107	4	4	0	TV1 derivative time (D component)	REAL	Read/Write
2	107	4	5	0	TV2 derivative time (D component)	REAL	Read/Write
2	107	4	6	0	TN1 reset time (I component)	REAL	Read/Write
2	107	4	7	0	TN2 reset time (I component)	REAL	Read/Write
2	107	4	8	0	CY1 cycle time	REAL	Read/Write
2	107	4	9	0	CY2 cycle time	REAL	Read/Write
2	107	4	10	0	XSH switching hysteresis between 1st & 2nd digital output	REAL	Read/Write
2	107	4	11	0	XD1 switching differential of the 1st digital output	REAL	Read/Write
2	107	4	12	0	XD2 switching differential of the 2nd digital output	REAL	Read/Write
2	107	4	13	0	TT actuator time	INT	Read/Write
2	107	4	14	0	Y0 working point	INT	Read/Write
2	107	4	15	0	Y1 output value limit	INT	Read/Write
2	107	4	16	0	Y2 output value limit	INT	Read/Write
2	107	4	17	0	TK1 minimum start time of the 1st digital output	REAL	Read/Write
2	107	4	18	0	TK2 minimum start time of the 2nd digital output	REAL	Read/Write

Controller 2 parameter block 2

Index					Signal designation	Data type	Access
2	107	5	2	0	XP1 proportional band (P component)	REAL	Read/Write
2	107	5	3	0	XP2 proportional band (P component)	REAL	Read/Write
2	107	5	4	0	TV1 derivative time (D component)	REAL	Read/Write
2	107	5	5	0	TV2 derivative time (D component)	REAL	Read/Write
2	107	5	6	0	TN1 reset time (I component)	REAL	Read/Write
2	107	5	7	0	TN2 reset time (I component)	REAL	Read/Write
2	107	5	8	0	CY1 cycle time	REAL	Read/Write
2	107	5	9	0	CY2 cycle time	REAL	Read/Write
2	107	5	10	0	XSH switching hysteresis between 1st & 2nd digital output	REAL	Read/Write
2	107	5	11	0	XD1 switching differential of the 1st digital output	REAL	Read/Write
2	107	5	12	0	XD2 switching differential of the 2nd digital output	REAL	Read/Write
2	107	5	13	0	TT actuator time	INT	Read/Write
2	107	5	14	0	Y0 working point	INT	Read/Write
2	107	5	15	0	Y1 output value limit	INT	Read/Write
2	107	5	16	0	Y2 output value limit	INT	Read/Write
2	107	5	17	0	TK1 minimum start time of the 1st digital output	REAL	Read/Write
2	107	5	18	0	TK2 minimum start time of the 2nd digital output	REAL	Read/Write

5 Noncyclical data exchange

Controller 2 parameter block 3

Index					Signal designation	Data type	Access
2	107	6	2	0	XP1 proportional band (P component)	REAL	Read/Write
2	107	6	3	0	XP2 proportional band (P component)	REAL	Read/Write
2	107	6	4	0	TV1 derivative time (D component)	REAL	Read/Write
2	107	6	5	0	TV2 derivative time (D component)	REAL	Read/Write
2	107	6	6	0	TN1 reset time (I component)	REAL	Read/Write
2	107	6	7	0	TN2 reset time (I component)	REAL	Read/Write
2	107	6	8	0	CY1 cycle time	REAL	Read/Write
2	107	6	9	0	CY2 cycle time	REAL	Read/Write
2	107	6	10	0	XSH switching hysteresis between 1st & 2nd digital output	REAL	Read/Write
2	107	6	11	0	XD1 switching differential of the 1st digital output	REAL	Read/Write
2	107	6	12	0	XD2 switching differential of the 2nd digital output	REAL	Read/Write
2	107	6	13	0	TT actuator time	INT	Read/Write
2	107	6	14	0	Y0 working point	INT	Read/Write
2	107	6	15	0	Y1 output value limit	INT	Read/Write
2	107	6	16	0	Y2 output value limit	INT	Read/Write
2	107	6	17	0	TK1 minimum start time of the 1st digital output	REAL	Read/Write
2	107	6	18	0	TK2 minimum start time of the 2nd digital output	REAL	Read/Write

Controller 2 parameter block 4

Index					Signal designation	Data type	Access
2	107	7	2	0	XP1 proportional band (P component)	REAL	Read/Write
2	107	7	3	0	XP2 proportional band (P component)	REAL	Read/Write
2	107	7	4	0	TV1 derivative time (D component)	REAL	Read/Write
2	107	7	5	0	TV2 derivative time (D component)	REAL	Read/Write
2	107	7	6	0	TN1 reset time (I component)	REAL	Read/Write
2	107	7	7	0	TN2 reset time (I component)	REAL	Read/Write
2	107	7	8	0	CY1 cycle time	REAL	Read/Write
2	107	7	9	0	CY2 cycle time	REAL	Read/Write
2	107	7	10	0	XSH switching hysteresis between 1st & 2nd digital output	REAL	Read/Write
2	107	7	11	0	XD1 switching differential of the 1st digital output	REAL	Read/Write
2	107	7	12	0	XD2 switching differential of the 2nd digital output	REAL	Read/Write
2	107	7	13	0	TT actuator time	INT	Read/Write
2	107	7	14	0	Y0 working point	INT	Read/Write
2	107	7	15	0	Y1 output value limit	INT	Read/Write
2	107	7	16	0	Y2 output value limit	INT	Read/Write
2	107	7	17	0	TK1 minimum start time of the 1st digital output	REAL	Read/Write
2	107	7	18	0	TK2 minimum start time of the 2nd digital output	REAL	Read/Write

Setpoint values, controller 1

Index					Signal designation	Data type	Access
2	126	0	0	0	Setpoint W1 (controller 1)	REAL	Read/Write
2	126	0	0	1	Setpoint W2 (controller 1)	REAL	Read/Write

5 Noncyclical data exchange

Index					Signal designation	Data type	Access
2	126	0	0	2	Setpoint W3 (controller 1)	REAL	Read/Write
2	126	0	0	3	Setpoint W4 (controller 1)	REAL	Read/Write

Setpoint values, controller 2

Index					Signal designation	Data type	Access
2	126	1	0	0	Setpoint W1 (controller 2)	REAL	Read/Write
2	126	1	0	1	Setpoint W2 (controller 2)	REAL	Read/Write
2	126	1	0	2	Setpoint W3 (controller 2)	REAL	Read/Write
2	126	1	0	3	Setpoint value W4 (controller 2)	REAL	Read/Write

Filter time constant, offset

Index					Signal designation	Data type	Access
2	92	0	7	0	Filter time constant, analog input IN8	REAL	Read/Write
2	92	1	7	0	Filter time constant, analog input IN9	REAL	Read/Write
2	92	2	7	0	Filter time constant, analog input IN10	REAL	Read/Write
2	92	3	7	0	Filter time constant, analog input IN11	REAL	Read/Write
2	92	0	5	0	Offset, analog input IN8	REAL	Read/Write
2	92	1	5	0	Offset, analog input IN9	REAL	Read/Write
2	92	2	5	0	Offset, analog input IN10	REAL	Read/Write
2	92	3	5	0	Offset, analog input IN11	REAL	Read/Write

Analog and digital flags 1 to 8

Index					Signal designation	Data type	Access
2	142	0	0	0	Analog flag 1	REAL	Read/Write
2	142	1	0	0	Analog flag 2	REAL	Read/Write
2	142	2	0	0	Analog flag 3	REAL	Read/Write
2	142	3	0	0	Analog flag 4	REAL	Read/Write
2	142	4	0	0	Analog flag 5	REAL	Read/Write
2	142	5	0	0	Analog flag 6	REAL	Read/Write
2	142	6	0	0	Analog flag 7	REAL	Read/Write
2	142	7	0	0	Analog flag 8	REAL	Read/Write
2	143	0	0	0	Digital flag 1	BOOL	Read/Write
2	143	0	0	1	Digital flag 2	BOOL	Read/Write
2	143	0	0	2	Digital flag 3	BOOL	Read/Write
2	143	0	0	3	Digital flag 4	BOOL	Read/Write
2	143	0	0	4	Digital flag 5	BOOL	Read/Write
2	143	0	0	5	Digital flag 6	BOOL	Read/Write
2	143	0	0	6	Digital flag 7	BOOL	Read/Write
2	143	0	0	7	Digital flag 8	BOOL	Read/Write

5 Noncyclical data exchange

5.2.2 Process values

Setpoint values

Index					Signal designation	Data type	Access
2	294	0	15	0	Setpoint value, interface	REAL	Read
2	294	1	15	0	Setpoint value, interface	REAL	Read

Device operation

Index					Signal designation	Data type	Access
2	275	0	0	0	Operation, tune start, controller 1	BOOL	Read
2	275	0	1	0	Operation, tune stop, controller 1	BOOL	Read
2	275	1	0	0	Operation, tune start, controller 2	BOOL	Read
2	275	1	1	0	Operation, tune stop, controller 2	BOOL	Read
2	275	0	2	0	Operation, manual mode, controller 1	BOOL	Read
2	275	0	3	0	Operation, automatic, controller 1	BOOL	Read
2	275	1	2	0	Operation, manual mode, controller 2	BOOL	Read
2	275	1	3	0	Operation, automatic, controller 2	BOOL	Read
2	275	0	4	0	Operation, manual output level, controller 1	REAL	Read
2	275	1	4	0	Operation, manual output level, controller 2	REAL	Read
2	275	0	5	0	Operation, actuator drive on controller 1	BOOL	Read
2	275	0	6	0	Operation, actuator drive for controller 1	BOOL	Read
2	275	1	5	0	Operation, actuator drive on controller 2	BOOL	Read
2	275	1	6	0	Operation, actuator drive for controller 2	BOOL	Read
2	284	0	0	0	Operation, program start	BOOL	Read
2	284	0	1	0	Operation, program stop	BOOL	Read
2	284	0	2	0	Operation, program break	BOOL	Read
2	284	0	3	0	Operation, manual program	BOOL	Read
2	284	0	4	0	Operation, fast forward	BOOL	Read
2	284	0	5	0	Operation, next section	BOOL	Read
2	284	0	6	0	Operation, previous section	BOOL	Read
2	285	0	1	0	Operation, program start, program number	INT	Read
2	285	0	3	0	Operation, program start, start section	INT	Read
2	285	0	2	0	Operation, program start, lead time	DINT	Read
2	285	0	4	0	Operation, program start, section remaining time	DINT	Read
2	286	0	0	0	Operation, program start, start time	DINT	Read
2	286	0	1	0	Operation, program start, week day	DINT	Read
2	281	0	0	0	Manual operating mode, setpoint W1	REAL	Read
2	281	0	1	0	Manual operating mode, setpoint W2	REAL	Read
2	281	0	2	0	Manual operating mode, operating contacts	INT	Read
2	304	0	0	0	Operation, limit value monitoring acknowledgement	BOOL	Read
2	304	0	1	0	Operation, service acknowledgement	BOOL	Read
2	304	0	3	0	Operation, timer start T1	BOOL	Read
2	304	0	4	0	Operation, timer stop T1	BOOL	Read
2	304	0	5	0	Operation, timer hold T1	BOOL	Read
2	304	0	6	0	Operation, timer acknowledgement T1	BOOL	Read
2	304	0	7	0	Operation, timer restart T1	BOOL	Read

5 Noncyclical data exchange

Index					Signal designation	Data type	Access
2	304	0	3	1	Operation, timer start T2	BOOL	Read
2	304	0	4	1	Operation, timer stop T2	BOOL	Read
2	304	0	5	1	Operation, timer hold T2	BOOL	Read
2	304	0	6	1	Operation, timer acknowledgement T2	BOOL	Read
2	304	0	7	1	Operation, timer restart T2	BOOL	Read
2	304	0	10	0	Operation, hold, ramp 1	BOOL	Read
2	304	0	9	0	Operation, stop, ramp 1	BOOL	Read
2	304	0	8	0	Operation, restart, ramp 1	BOOL	Read
2	304	0	10	1	Operation, hold, ramp 2	BOOL	Read
2	304	0	9	1	Operation, stop, ramp 2	BOOL	Read
2	304	0	8	1	Operation, restart, ramp 2	BOOL	Read
2	304	0	11	0	Operation, function key 1	BOOL	Read

External analog and digital signals

Index					Signal designation	Data type	Access
2	171	0	0	0	External analog input1, (via interface)	REAL	Read
2	171	1	0	0	External analog input2, (via interface)	REAL	Read
2	171	2	0	0	External analog input3, (via interface)	REAL	Read
2	171	3	0	0	External analog input4, (via interface)	REAL	Read
2	171	4	0	0	External analog input5, (via interface)	REAL	Read
2	171	5	0	0	External analog input6, (via interface)	REAL	Read
2	171	6	0	0	External analog input7, (via interface)	REAL	Read
2	171	7	0	0	External analog input8, (via interface)	REAL	Read
2	172	0	0	0	External digital input 1	BOOL	Read
2	172	1	0	0	External digital input 2	BOOL	Read
2	172	2	0	0	External digital input 3	BOOL	Read
2	172	3	0	0	External digital input 4	BOOL	Read
2	172	4	0	0	External digital input 5	BOOL	Read
2	172	5	0	0	External digital input 6	BOOL	Read
2	172	6	0	0	External digital input 7	BOOL	Read
2	172	7	0	0	External digital input 8	BOOL	Read

5 Noncyclical data exchange

6.1 Project integration of JUMO IO devices

For JUMO field devices to be integrated into the project structure of the IO controller as IO device, the GSDML file of the respective device must be imported into the engineering system of your IO controller. The GSDML file describes all PROFINET-IO features of IO devices and delivers to the engineering system all the information required for the project planning phase. This file is located on the DVD from the supply scope of your JUMO field device, or can alternatively be downloaded from the JUMO website. Once the GSDML file has been imported into the engineering system, the corresponding field device is available as an IO device in the engineering system (e.g. "Hardware catalog" for SIMATIC®¹ or "Device repository" for CODESYS®²), and can be incorporated into the hardware structure of your projects. With JUMO IO devices, modules can then be added as slots. In the engineering system, modules are treated as modular devices for expanding IO devices. They are selected from the engineering system catalog and assigned to the slots in the JUMO IO device. The procedure for integrating devices into project structures is described in the documentation for the engineering system of your IO controller.

Procedure

1. Import the GSDML file for your JUMO IO device into the engineering system of your IO controller. Make sure the device software version matches the version data in the GSDML file.



NOTE!

The GSDML file for the current device version is located on the DVD from the scope of delivery of your JUMO field device. Alternatively, you can download the GSDML file from the JUMO website.

2. Configure the PROFINET modules for your JUMO field device using the JUMO PC setup program and print out the finished module configuration.
⇒ chapter 6.2 "Configuring PROFINET modules", Page 46
3. Integrate the desired JUMO field device into the project structure of your IO controller. Make sure the device software version matches the version data of the JUMO field device integrated into the project structure.



NOTE!

The JUMO IO device appears in the project structure with the "DeviceStatusBlock" module in Slot 1. The "DeviceStatusBlock" module is immovably placed in Slot 1.



NOTE!

The exact procedure for integrating PROFINET IO devices can be found in the description of your engineering system.

4. Assign a device name.



NOTE!

Engineering systems offer a function for **identifying** field devices. If JUMO field devices are addressed via the "**Identification function**" of the engineering system, this is signaled by the flashing of the front display.

5. Insert the configured modules into the project structure of the IO controller in the desired slot position below the 1st slot on the JUMO IO device.
 6. Set the communication parameters of the JUMO field device in the project structure of the IO controller.
-

¹ SIMATIC is a registered trademark of Siemens AG in 80333 Munich, Germany.

² CODESYS is a registered trademark of 3S-Smart Software Solutions GmbH in 87439 Kempten, Germany.

6 Project planning

⇒ chapter 6.3.2 "Communication parameters", Page 50

7. Set the start-up parameters in the project structure of the IO controller correctly. Pay attention in particular to the settings for the byte sequence for cyclical and noncyclical data. The byte sequence to be set is specified by the IO controller. Please refer to the documentation for your IO controller.
8. The IO data of the JUMO IO device are now available for the programming of the IO controller. To localize the IO data configured in the JUMO field device, refer to the printed module configuration list. During programming, the assignment of the device data to the IO items of the modules can be taken from this list.

6.2 Configuring PROFINET modules

For project configuration in the engineering system of your PROFINET IO controller, you can populate all slots with modules apart from Slot 1 on your JUMO IO device. Slot 1 is permanently assigned with the immovable "DeviceStatusBlock" module. You can create additional modules for various data types via the JUMO PC setup program for the field device concerned and configure their contents to suit your needs. During this process, the selectable IO data in a module are filtered by data type suitable for the module type. An overview of the module types can be found in the "Module description" chapter.

The placeholders in the respective module can be assigned data from the selectors of the JUMO field device and be provided with a comment. In other words, the content of the configurable modules can be determined by the user. In the project configuration of the IO controller, the modules have to be assigned to the free slots in the JUMO IO device.

Once the configuration has been created by means of the JUMO PC setup program and loaded into the field device, the PROFINET configuration can be printed. The printout contains an overview of the complete module configuration of the JUMO field device with all IO data, which the field devices provisions for the PROFINET IO communication. In the engineering system, the modules must then be assigned to the slots in the JUMO IO device, so that they can be queried by the IO controller during the cyclical communication.



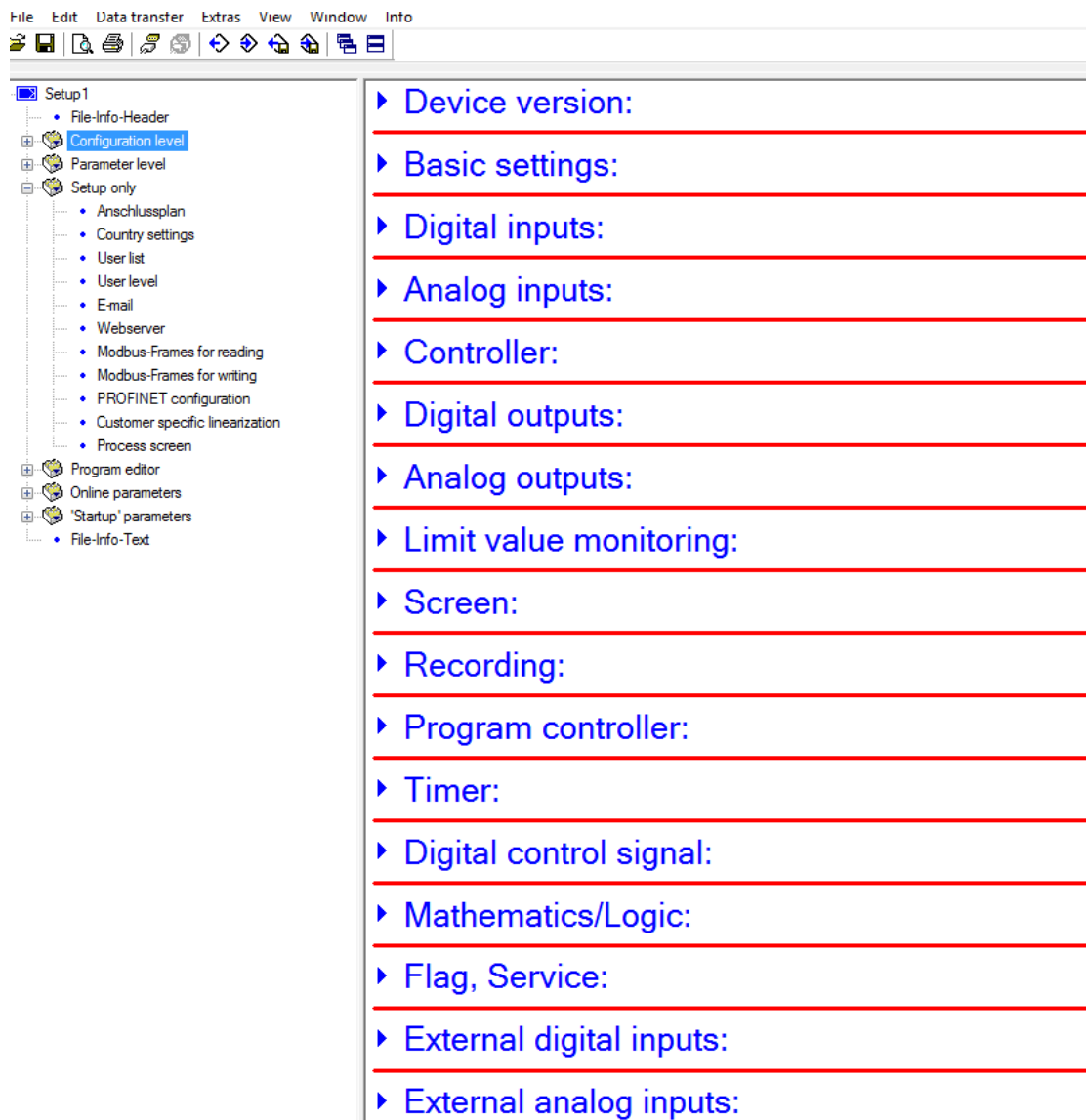
NOTE!

Make sure that your JUMO IO device has the software version required to use PROFINET IO (see chapter 4.2.2 "Modules", Page 11). The PROFINET configuration in the JUMO setup program of your device is available only for device software versions that are compatible with PROFINET IO.

Procedure for module configuration

1. Call up the module configuration dialog in the JUMO PC setup program:

Double-click on: Setup only > PROFINET configuration



The screenshot shows the JUMO PC setup program interface. The menu bar includes File, Edit, Data transfer, Extras, View, Window, and Info. The toolbar contains icons for file operations and navigation. The left pane shows a tree view of the configuration structure:

- Setup1
 - File-Info-Header
 - Configuration level
 - Parameter level
 - Setup only
 - Anschlussplan
 - Country settings
 - User list
 - User level
 - E-mail
 - Websserver
 - Modbus-Frames for reading
 - Modbus-Frames for writing
 - PROFINET configuration
 - Customer specific linearization
 - Process screen
 - Program editor
 - Online parameters
 - 'Startup' parameters
 - File-Info-Text

The right pane displays a list of configuration options, each with a blue arrow button and a red horizontal line below it:

- ▶ Device version:
- ▶ Basic settings:
- ▶ Digital inputs:
- ▶ Analog inputs:
- ▶ Controller:
- ▶ Digital outputs:
- ▶ Analog outputs:
- ▶ Limit value monitoring:
- ▶ Screen:
- ▶ Recording:
- ▶ Program controller:
- ▶ Timer:
- ▶ Digital control signal:
- ▶ Mathematics/Logic:
- ▶ Flag, Service:
- ▶ External digital inputs:
- ▶ External analog inputs:

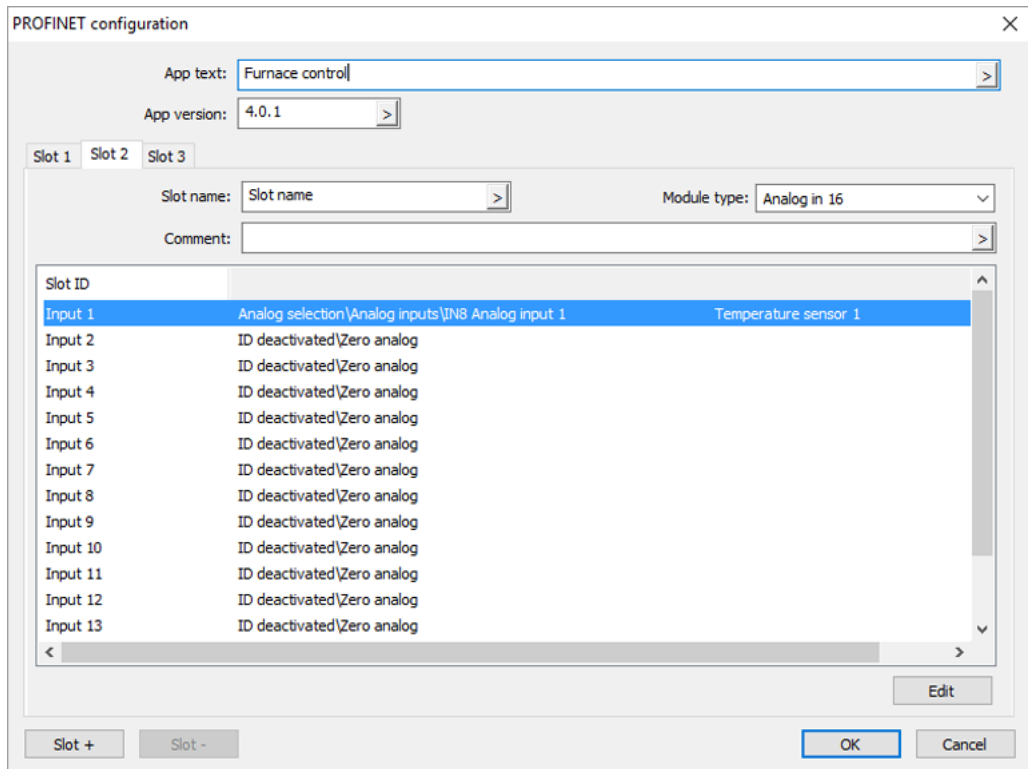
2. In the "App text" and "App Version" fields of the module configuration dialog, assign information about the application (e.g. system or system part), for which the configuration is intended. This information will appear later on the printout of the module configuration. In addition, the "App-Version" is displayed in the "Device info" of the JUMO field device.



NOTE!

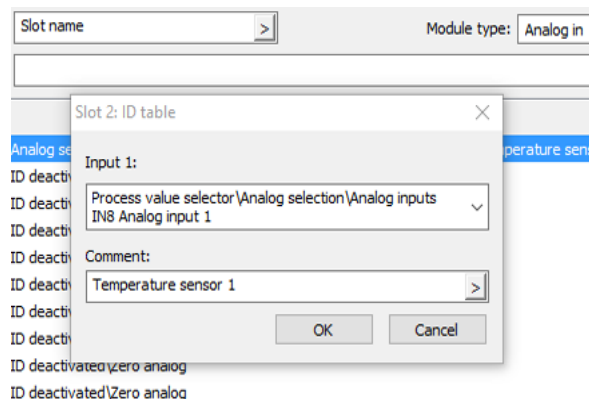
Entry fields with an arrow button are multilingual. By pressing the arrow button, you can open a dialog for entry and maintenance of the field contents for all languages installed on the device.

6 Project planning



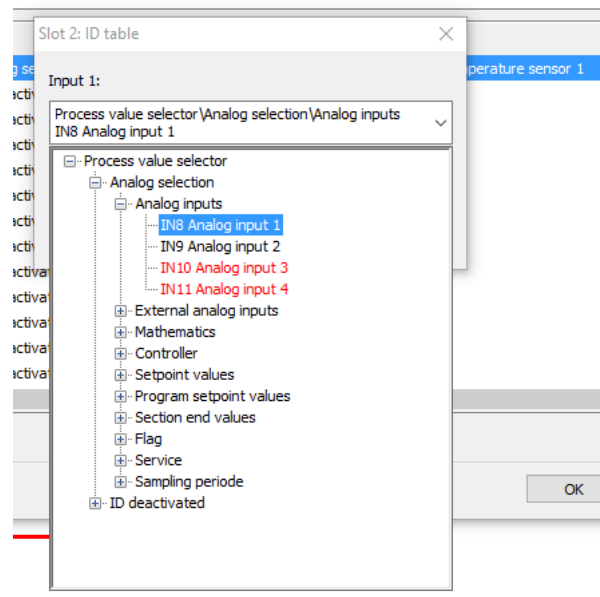
3. The module configuration contains 1 tab for each activated module. Define the number of modules you want to configure. To add modules, press the "Slot +" button. To reduce the number of modules, click on the next module tab to activate it and press the "Slot -" button.
4. Now determine the type you require for each module in the "Module type" field (see chapter 4.2.2 "Modules", Page 11)
5. In each module, select the IO data required for the cyclical communication.
 - a) Open the configuration dialog for each item in the module list by double-clicking on the required item or by selecting it and pressing the "Edit" button.

Configuration dialog for IO data



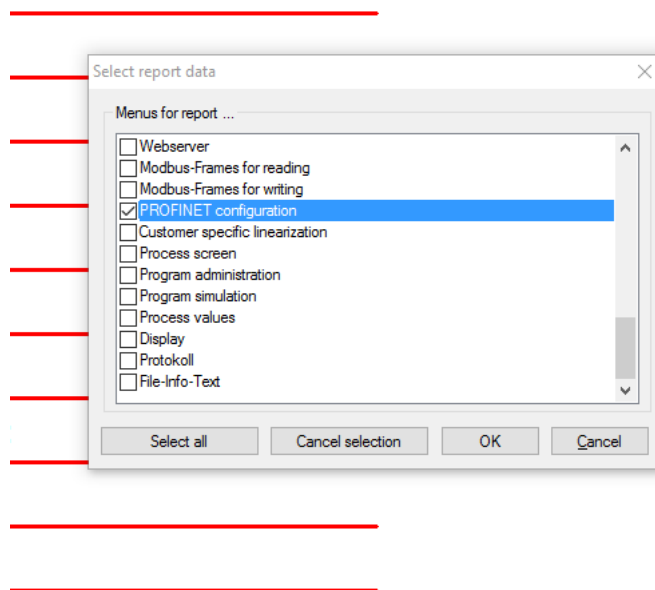
- b) Double-clicking on the selection field at the top opens the selector with selectable device data, which you can assign to the module list item. In the "Comment" field, you can enter a name for the data point (e.g. equipment designation). The content of the comment field also appears in the module configuration printout to make it easier to identify the IO data.

Configuration dialog for IO data with selector expanded



6. Once you have completed the configuration for all the required modules, click the "OK" button in the module configuration window to confirm the configuration.
7. Transfer the configuration to the JUMO field device and save the configuration file on the PC if required.
8. Print out the module configuration.
 - a) Do this by calling up the print command from the menu bar of the PC setup program. This opens "Select protocol data".

File> Print
 - b) In this selection field, activate the "PROFINET configuration" item and click on "OK".



The module list will be printed on your default printer. This is helpful for the configuration and programming of your IO controller.

6 Project planning

6.3 Configuring the JUMO IO device

6.3.1 Start-up parameters

When the system boots, the start-up parameters are transferred from the IO controller to the JUMO field device during the course of the IO device parameterization procedure. Prior to start-up, you must configure the settings for the start-up parameters with the engineering system in the project planning of your IO controller. The start-up parameters for each JUMO field device are located in the "DeviceStatusBlock" module (Slot 1.)

The list of start-up parameters to be configured can be found in the module description.

⇒ chapter 6.3.1 "Start-up parameters", Page 50

6.3.2 Communication parameters

Station name

To improve the overview in the IO controller project structure, each device should be assigned a self-explanatory and preferably unique station name. This permits a better overview for programming and project planning. The station name is entered in the engineering system of your IO controller at the PNIO identification data of the JUMO IO device.

IP configuration

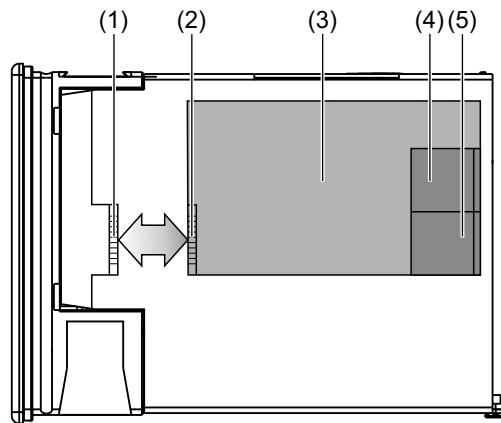
JUMO IO devices communicate with your PROFINET optional board via the internal Ethernet interface of the device. In order for the PROFINET optional board in a JUMO IO device to become operational, internal communication must be established between the optional board and the device. This requires valid IP configuration of the Ethernet interface of the JUMO IO device. Without valid IP configuration of the Ethernet interface, the device does not connect to the PROFINET IO interface and the PROFINET optional board is not initialized.



NOTE!

The IP addresses of the internal Ethernet interface and PROFINET interface are both visible from the outside and can be reached via both ports of the PROFINET optional board. The networks can be separately logically into different networks through use of different network IDs. The internal Ethernet interface support only standard Ethernet services and cannot be used for PROFINET IO. For PROFINET network segments, the PROFINET interface must be used.

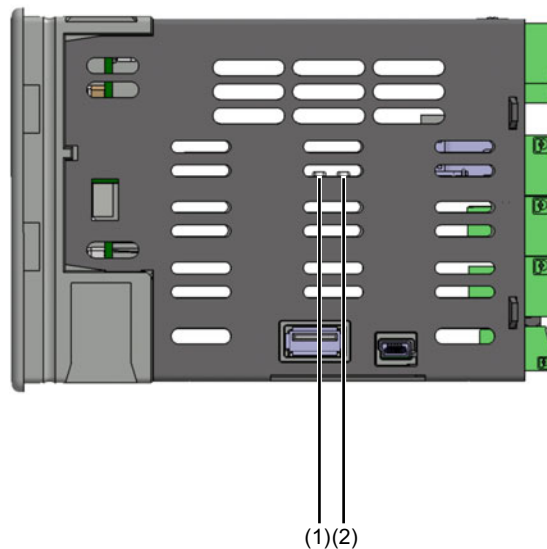
Block diagram of a PROFINET optional board with its interfaces



- (1) Ethernet slot in a JUMO IO device (COM2) with its own MAC address (cf. chapter 6.3.3 "Device information", Page 53)
- (2) Internal interface (SMK socket) of the PROFINET optional board for the slot in the JUMO IO device with its own MAC address (PROFINET MAC address of the JUMO IO device, cf. chapter 6.3.3 "Device information", Page 53)
- (3) PROFINET optional board
- (4) Switchport RJ-45 with its own MAC address (cf. chapter 6.3.3 "Device information", Page 53)
- (5) Switchport RJ-45 with its own MAC address (cf. chapter 6.3.3 "Device information", Page 53)

The operating status and IP configuration data of the PROFINET interface can be viewed on the device's display with the information in **Device menu > Device info > PROFINET** and **Device menu > Device info > Ethernet** as well as with the aid of 2 status LEDs located on the PROFINET optional board. The status LEDs can be viewed through a ventilation slot on the top of the device housing.

Status LED on PROFINET optional board



- (1) Front status LED on PROFINET optional board
- (2) Rear status LED on PROFINET optional board

6 Project planning

A **non-initialized** PROFINET optional board can be recognized from the following indicators:

- the MAC addresses of the interfaces on the PROFINET optional board **Device menu > Device info > PROFINET > Info** are not initialized (display: 00-00-00-00-00-00)
- the connection status of the PROFINET interface in **Device menu > Device info > PROFINET > Status > Status** shows "unknown"
- the front status LED on the PROFINET optional board is off
- the rear status LED on the PROFINET optional board flashes red

Cyclical transfer timing

Send clock and reduction ratio determine the frequency at which an IO device transfers cyclical data in a PROFINET IO network. The transfer cycle time of the JUMO IO device is calculated by dividing the send clock by the reduction ratio. On JUMO field devices, these parameters apply globally for all slots. The setting is configured in the engineering system of your IO controller at the PNIO parameters of the JUMO IO device.



NOTE!

In order to set meaningful transfer cycles, it is advisable to observe the processing cycle time of the respective JUMO field device.

JUMO DICON touch: 150 ms

Watchdog

If the watchdog function is activated, the cyclical communication is monitored in cycles. The watchdog cycle time can be set as a multiple of the transfer cycle time (maximum 1.92 s). If a "consumer" detects a communication failure, it dismantles the "Application Relation" and hence also the "Communication Relation" to the provider.



NOTE!

The watchdog cycle time can be set up to a maximum of 1.92 s. This is defined by the PROFINET IO standard for RT communication.

Starting up the PROFINET interface of a JUMO IO device

Proceed as follows to start up the PROFINET interface of a JUMO IO device:

1. Log on to the JUMO IO device as "Master" (see JUMO IO device operating manual).

Open user log-on: *Device menu > Log-on/Log-out*

2. The Ethernet interface of the JUMO IO device must be configured for the PROFINET optional board to connect internally to the JUMO IO device and be initialized. Either manually assign an IP configuration that is valid in your network or set the Ethernet interface to automatic configuration. If automatic IP configuration is set, the device must be physically connected to a network with a DHCP server. The DHCP server assigns a valid IP configuration to the Ethernet interface. If manual IP configuration is set, a physical connection is not yet absolutely essential at this point.

Open the Ethernet configuration: *Device Menu > Configuration > Ethernet*

3. After a valid IP configuration has been saved in the JUMO IO device, the PROFINET IO interface connects to the JUMO IO device and is initialized. To test, check the following indicators: The **front status LED on the PROFINET optional board must glow green** and the **MAC addresses of the PROFINET optional board** must now have valid content (not 00-00-00-00-00-00).

Check the MAC addresses: *Device menu > Device info > PROFINET > Info*

⇒ chapter 6.3.3 "Device information", Page 53

4. If you have not yet inserted a network connector into the switchports of the PROFINET interface, establish the connections to your network now. As soon as the physical connection to the network has been established, the PROFINET status changes to "Link Up".

Check the PROFINET status: *Device menu > Device info > PROFINET > Status*

⇒ chapter 6.3.3 "Device information", Page 53

5. Make sure that the correct communication settings have been defined in the project planning for your IO controller. Check for correct assignment of the IP configuration and unique station name for the JUMO IO device.

The subnet configuration for PROFINET communication must be saved in the IO controller. Then, as the system is booted, the IP configuration (IP address, subnet mask and default gateway) is transferred from the PROFINET IO controller to the IO device according to the IO controller configuration. The procedure for project planning can be found in the documentation of the engineering system you are using for the project planning of your system.

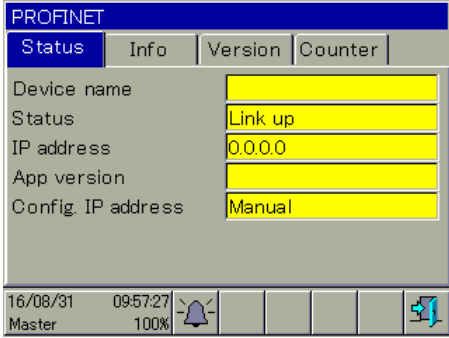
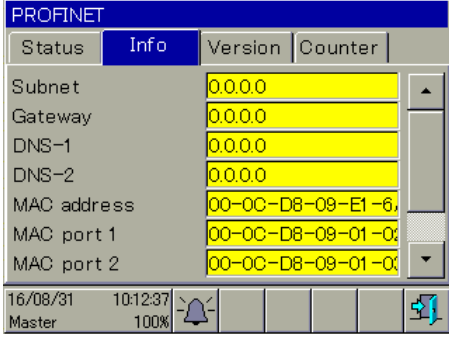
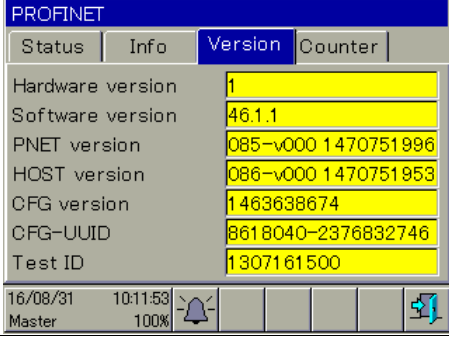
6. After the system has booted successfully with DCP assignment of the IP configuration to the JUMO IO device by the IO controller, the device is ready to communicate with the IO controller. The PROFINET status changes to "Network Up". Immediately thereafter, communication with the IO controller starts and the PROFINET status changes to "PLC Connection Up".

6.3.3 Device information

The device info for the JUMO IO device displays, for control and error diagnostics, information about network configuration, hardware and software component versions, as well as counters for evaluating data traffic. For more details about the displayed data, see the operating manual for your JUMO IO device.

Call-up: *Device menu> Device info> PROFINET*

6 Project planning

<p>Status: Data for connecting the JUMO IO device to the IO controller</p> <p>In addition to the essential IP configuration data for the JUMO IO device, additional important information is displayed. For example: The "App Version" from the module configuration of the device (see chapter 6.2 "Configuring PROFINET modules", Page 46) and in the "Status" field the connection status of the JUMO IO device. The last PROFINET diagnosis event is displayed as the current connection status:</p> <ul style="list-style-type: none"> • Unknown Connection status unknown (no previous diagnosis event) • Link Up/Down At least one RJ45 switchport of the PROFINET optional board has established or lost a physical connection. • Network Up/Down IP configuration status of the IO devices valid or invalid (e. g. address conflict). • PLC Connection Up/Down The IO device has established or lost a communication connection to the IO controller. 	
<p>Info: Network data of the IO device</p> <p>The network configuration data of the PROFINET interface such as the MAC addresses of the 3 interfaces of the PROFINET optional board are displayed here (cf. chapter 6.3.2 "Communication parameters", Page 50).</p> <p>The individual MAC addresses:</p> <ul style="list-style-type: none"> • MAC address MAC address of the internal interface (SMK socket) of the PROFINET optional board for connection to the internal Ethernet interface of the JUMO IO device (COM2 slot); this MAC address represents the JUMO IO device in your project planning for device detection and topology recognition • MAC port 1/2 MAC addresses of the 2 switchports (RJ 45) of the PROFINET optional board 	
<p>Version: Version numbers of hardware and software components</p>	

Counter: Packet statistics of the JUMO IO device for control and error diagnostics for evaluating data traffic

PROFINET	
Status	Info
P1 RxPacket	3
P1 RxBroadcast	3
P1 RxMulticast	0
P1 RxCRCError	0
P1 TooShortOrLong	0
P1 TxPacket	1

16/08/31 10:13:36
Master 100%

The status of the Ethernet interface is relevant for the PROFINET interface because the PROFINET optional board communicates internally with the JUMO IO device via the Ethernet interface (COM2 slot). It is essential for communication that the Ethernet interface of the JUMO IO device be configured with a valid IP address (cf. chapter 6.3.2 "Communication parameters", Page 50). For checking and for diagnostic purposes, the IP configuration of the Ethernet interface can be viewed in Device info.

Open: Device menu > Device info > Ethernet > Ethernet info

IP configuration data of the internal Ethernet interface of the JUMO IO device (cf. chapter 6.3.2 "Communication parameters", Page 50)

Ethernet	
Ethernet	Counter
MAC address	00-0C-D8-09-4E-CD
IP address	192.168.1.11
IP mask	255.255.255.0
Gate address	0.0.0.0
DNS address	10.10.0.175
DNS device name	
Transfer rate	00 MBit/s full duplex

16/03/14 02:24:58
Master 100%

6 Project planning

7.1 Error messages for invalid values

For measured values in float format, the error number itself is displayed as a value, i.e. instead of the measured value, the error number is returned.

Error code for float values	Error
1.0×10^{37}	Underrange
2.0×10^{37}	Overrange
3.0×10^{37}	Not a valid input value
4.0×10^{37}	Division by zero
5.0×10^{37}	Mathematical error
6.0×10^{37}	Invalid terminal temperature of thermocouple
7.0×10^{37}	Invalid floating-point value
8.0×10^{37}	Integrator or statistics destroyed

7.2 Error messages for acyclic services

The ERROR variable contains error messages from the IO device that can be evaluated by the IO controller. The following error IDs are used:

Error ID	Description
0	No error
3	Incorrect ID

7 Error messages



JUMO GmbH & Co. KG

Street address:
Moritz-Juchheim-Straße 1
36039 Fulda, Germany

Delivery address:
Mackenrodtstraße 14
36039 Fulda, Germany

Postal address:
36035 Fulda, Germany

Phone: +49 661 6003-0
Fax: +49 661 6003-607
Email: mail@jumo.net
Internet: www.jumo.net

JUMO Instrument Co. Ltd.

JUMO House
Temple Bank, Riverway
Harlow, Essex, CM20 2DY, UK

Phone: +44 1279 63 55 33
Fax: +44 1279 62 50 29
Email: sales@jumo.co.uk
Internet: www.jumo.co.uk

JUMO Process Control, Inc.

6733 Myers Road
East Syracuse, NY 13057, USA

Phone: +1 315 437 5866
Fax: +1 315 437 5860
Email: info.us@jumo.net
Internet: www.jumousa.com

