IO-Link specification in brief

1 Table of contents

2 IO-Link system overview

An IO-Link system consists of IO-Link *devices* - often sensors, actuators or combinations thereof - a standard 3-wire sensor/actuator cable and an IO-Link *master*. The master may be a device with any design and any degree of protection. The system architecture may be as follows:

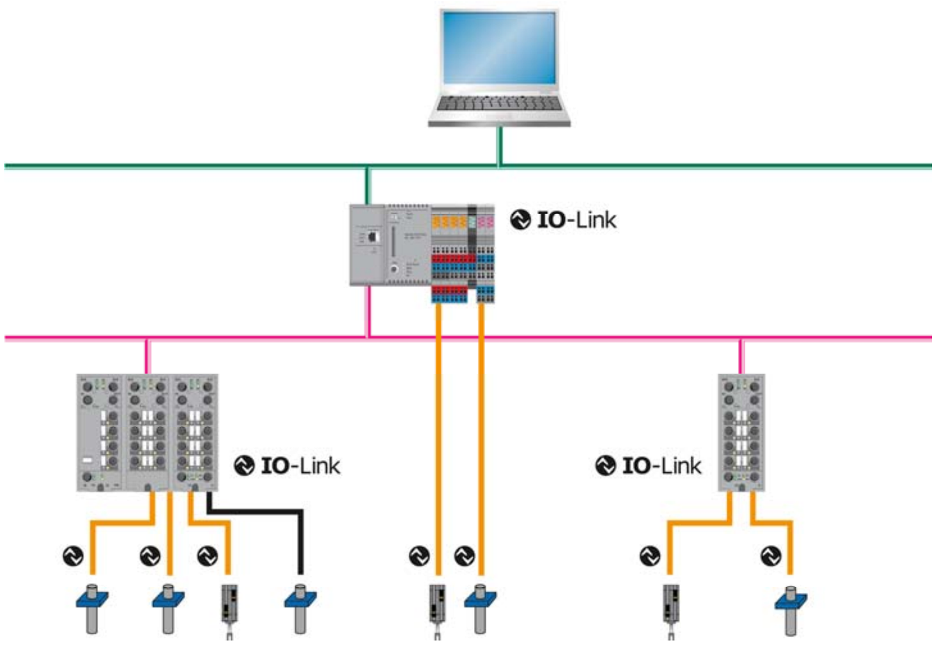

Figure 1: System architecture example

An IO-Link master can have one or several ports. Only one IO-Link device can be connected to each port. Thus IO-Link is point-to-point communication and not a fieldbus.

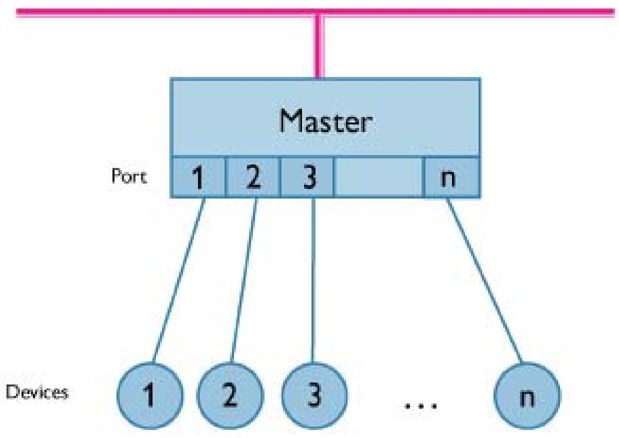

Figure 2: IO-Link point-to-point connection

2.1 After power up

At the beginning the device is always in the *SIO mode* (standard I/O mode). The ports of the master may have different configurations. If a port is set to *SIO mode*, the master acts on this port like a normal digital input. If the port is set to *communication mode*, the master tries to find the connected IO-Link device. This process is called *wake-up*.

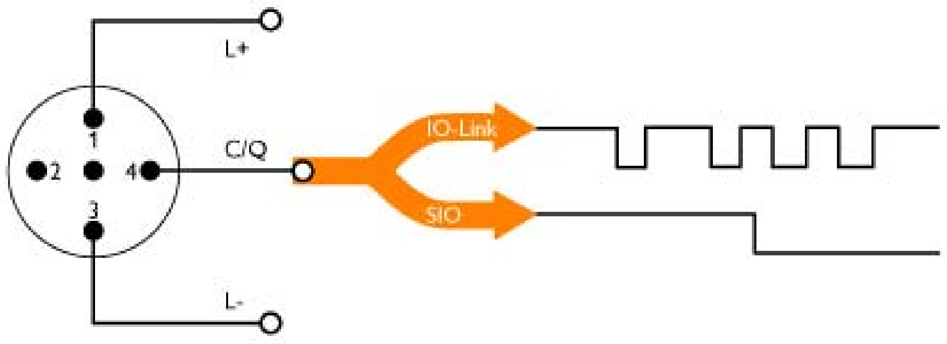

Figure 3: IO-Link SIO and COM mode

During the *wake-up* the master sends a defined signal and waits for the response of the device. The master tries to do this with the highest defined baud rate. If this is unsuccessful the master tries the next lower baud rate. With every baud rate, the master tries to address the device three times. If the master receives a response (i.e., the device was woken up) both will start communication. At first, they exchange the communication parameters then they start to exchange cyclic process data.

If the device is removed during operation, the master detects the communication abort, reports it, like it is specific to fieldbus, to the control system and tries cyclically to wake-up the device. After another successful wake-up the communication parameters will be read out again and validated, if necessary, and cyclic data exchange will be started.

If the master terminates communication both, the master and the device, return to the original mode that is the SIO mode.

This is called *fall back*.

3 IO-Link protocol

Three types of data may basically be exchanged:

- Cyclic data (also process data)
- Acyclic data or service data
- Events

The IO-Link device sends data only after a request of the IO-Link master. Acyclic data and events are requested explicitly by the master and cyclic data is sent after the IDLE telegram of the master.

3.1 Process data (PD)

The process data of the devices is transmitted cyclically in a data frame, provided that the process data width of 2 bytes is not exceeded. If the process data width is exceeded the process data is split up and transmitted in several cycles.

There will be a diagnostic message when the process data is invalid.

3.2 Service data (SD)

Service data is always exchanged acyclically and always upon request of the IO-Link master. First, the IO-Link master sends a request to the device and the device responds. This is true for writing data to the device as well as for reading data from the device. Service data can be used to read out parameter values or device states. It can also be used to write parameter values or to send commands.

SD and PD can be transmitted in one telegram or in separate telegrams. A typical data exchange may have the following structure.

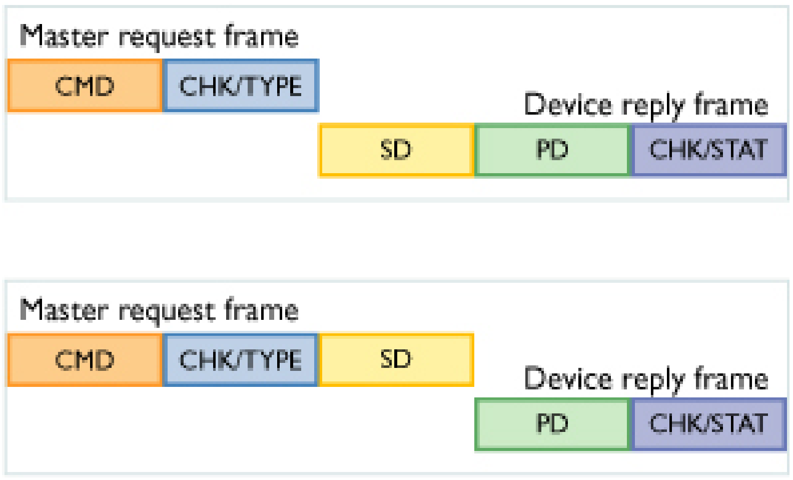

Figure 4: IO-Link telegram structure

For a detailed explanation of the telegram structure, please refer to Section “Telegram types and their structure”.

3.3 Events

When an event occurs the device sets the “Event Flag”, which is transmitted in bit 7 in the CHECK/STAT byte of the process data telegram. The master detects the set bit and reads out the reported event. No service data can be exchanged while an event is read out. This means that events or device states such as contamination, overheating, short circuit, etc. can be transmitted via the IO-Link master to the PLC or the visualization software.

The IO-Link master can generate its own events and states and transmit them over the respective fieldbus. Such event may be, for example, open circuits, communication abort, or overload, etc.

3.4 Transmission quality, retries, QoS

IO-Link is a very robust transmission system. It operates with a 24 V level. If one of the *frames* fails, the master’s request is repeated twice. Only when the second attempt to send data fails again, will the master detect a communication abort and report it to the higher-level control system. The master measures the quality of the transmission (QoS = Quality of Service) with the number of telegram repetitions (retries).

3.5 Transmission speed and synchronism

The IO-Link specification defines at least the transmission rates (*baud rates*) 4.8 and 38.4 kbaud. Normally, an IO-Link device supports one of the defined baud rates. The IO-Link master must support both baud rates.

The cycle time consists of the telegram length and the delay times in the master and the device. For a baud rate of 38.4 kbaud, the cycle time is typically 2 ms.

The total time results from the minimum cycle time specified by the device and the agreed or parameterized actual cycle time specified by the master afterwards.

A different time response can be parameterized for each port of the master. The device application can be synchronized with the master cycle. It is also possible to synchronize device applications at different ports of the same master.

3.6 Telegram types and their structure

The IO-Link specification defines different telegram types that differ in the size of the process input data and the process output data.

To establish communication, the master must determine the communication parameters of the device. One relevant piece of information is the process data length. On the basis of this information, the IO-Link master decides which telegram type is used for the cyclic data exchange. In the communication establishment phase the master uses telegram type 0. The following telegram types are defined:

Frame type	Length of process input data	Length of process output data
0	0	0
1	See below	See below
2.1	1	0
2.2	2	0
2.3	0	1
2.4	0	2
2.5	1	1

Table 1: Telegram types

Telegram type 1 is always used when the sum of the process input and output data of a device exceeds two bytes. Then the telegram structure consists of several IO-Link cycles.

The telegrams shown in Section “Service data” are telegrams of type 2.1. The device sends one byte of process data. In the top image the device sends one byte of service data in addition to the byte of process data. In the bottom image, the master sends one byte of service data to the device.

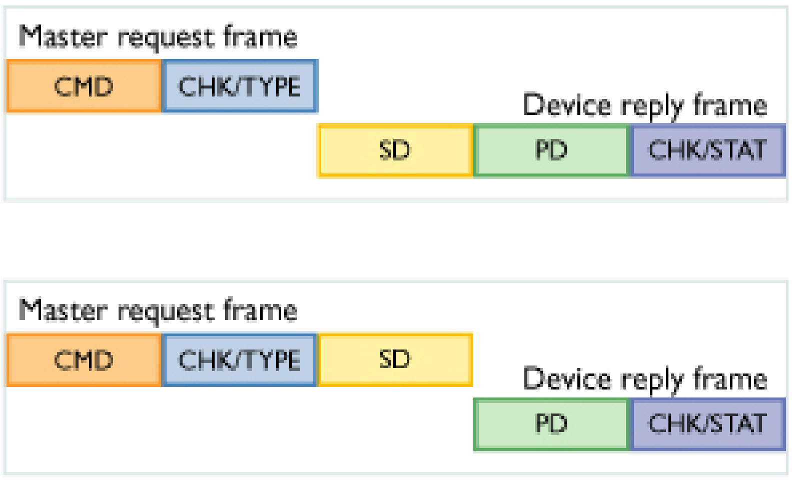

Figure 5: Frame type 2.1

The meanings of the various bits of the telegram are shown in the following tables.

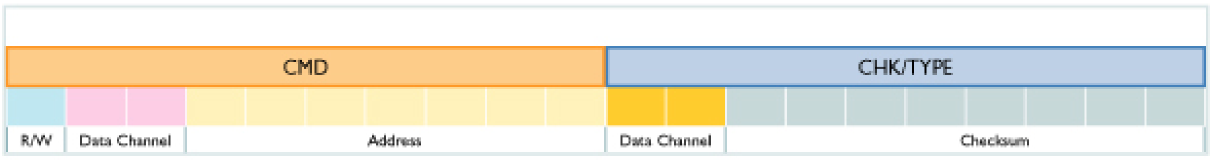

Figure 6: Command/check sequence of the master

Value	Meaning
0	Write access
1	Read access

Table 2: R/W values

Value	Meaning
0	Process data
1	Parameter data
2	Diagnosis data
3	Service PDU

Table 3: Data channel values

Value	Meaning
0	Type 0
1	Type 1
2	Type 2 (Note)
3	reserved

Table 4: Frame type values

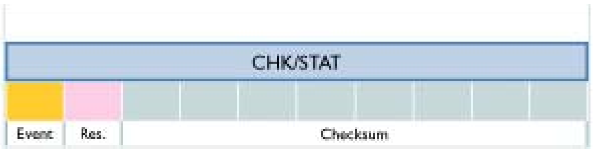

Figure 7: Check/status sequence of the device

Value	Meaning
0	No Event
1	Event

Table 5: Event bit values

For transmission over the IO-Link physics, each individual byte is packed in a UART frame and transmitted between master and device in half duplex mode.

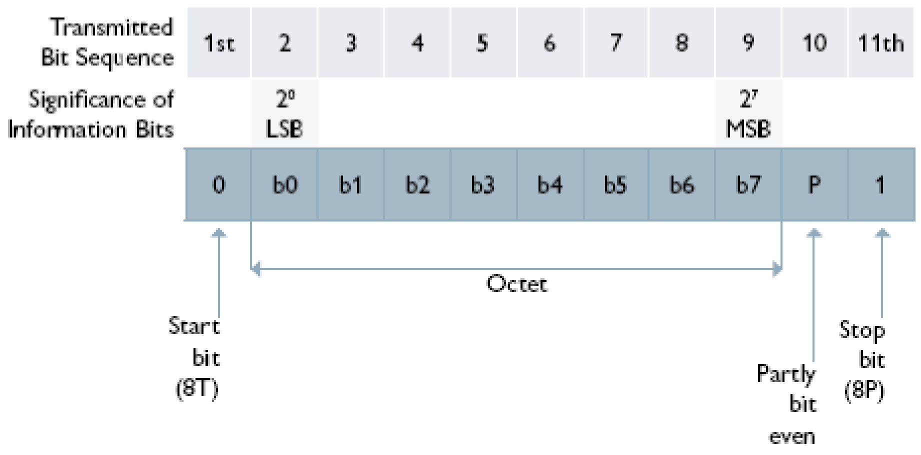

Figure 8: IO-Link UART frame

4 Parameter data exchange

To exchange data between an IO-Link device and a PLC, the IO-Link master maps the IO-Link data to the fieldbus used. This is known as an IO-Link map to the fieldbus. If the IO-Link master is directly connected with a PLC over a proprietary backplane bus (see Figure 1), the IO-Link data is mapped to this bus and transmitted to the PLC or from the PLC to the IO-Link master and further to the IO-Link device. IO-Link maps have already been specified for PROFIBUS, for Profinet, INTERBUS, AS-i and EtherCAT .

Process data from or to the IO-Link device is transferred in the cyclic data traffic over the fieldbus or the backplane bus. Service data must be explicitly requested by the PLC or identified as such. This is why the SPDU (Service Protocol Data Unit) was defined in the IO-Link specification.

With the help of function blocks (FB) – each PLC manufacturer provides several customized FBs for his system – the IO-Link master programs acyclic communication with the IO-Link device in the control program. The FB defines with which IO-Link master (that is with which fieldbus device) and over which of its ports data is to be exchanged. A request is also sent to the IO-Link device at the same time.

Parameter values and states can be requested with indices and sub-indexes in an IO-Link device. In the IO-Link master, the requests (Read Write Services) are coded to an IO-Link specific Service Protocol Data Unit (*SPDU*) and transferred via the IO-Link interface to the device.

The SPDU specifies whether data is read or written. The parameters whose values are read or written to are defined using indexes. The SPDU is structured as follows:

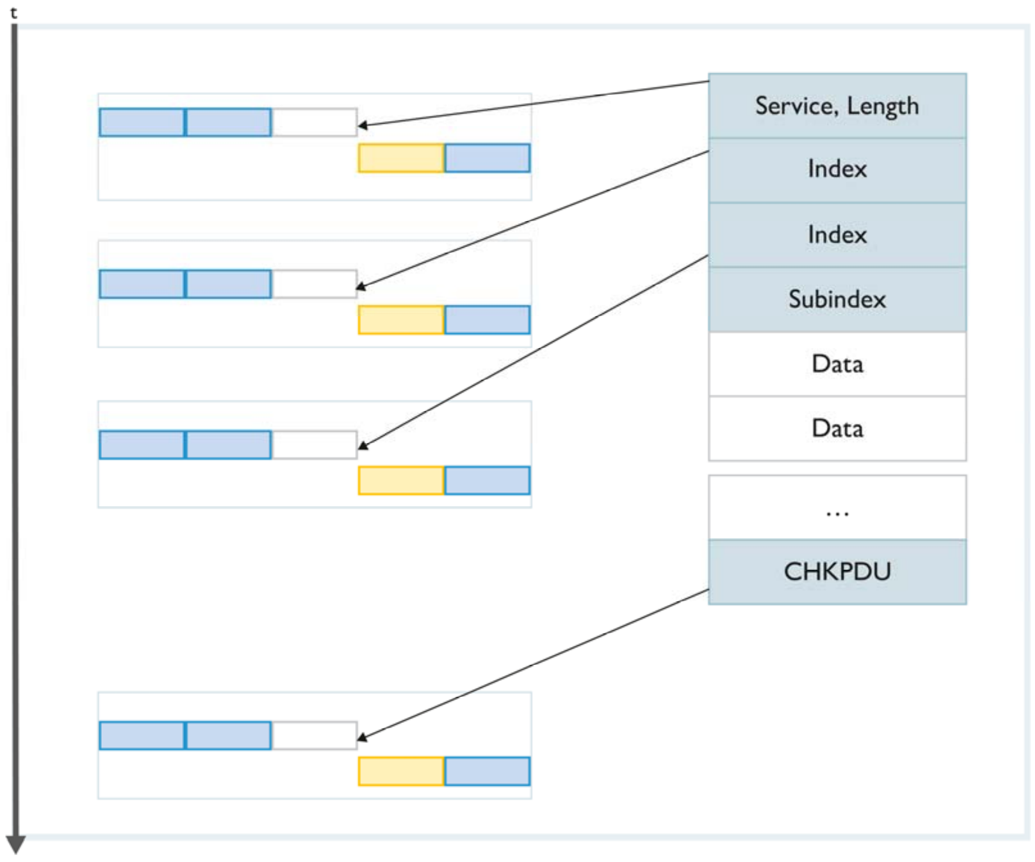

Figure 9: Structure of an SPDU

Up to 32768 indexes with a size of up to 232 bytes can be addressed over the IO-Link.

The IO-Link specification specifies several services such as

- D10 Vendor Name

- D12 Product Name

IO-Link devices can be uniquely identified with these services.

On the fieldbus, the IO-Link master is represented like a normal fieldbus device and is linked to the respective network configurator via the corresponding device description (e.g., GSD, FDCML, GSDML, etc.). These files describe the communication and further properties of the IO-Link master, such as the number of ports. However, which IO-Link devices are connected cannot be read here. The IO-Link Device Description (IODD) was defined to show the system architecture transparently and completely down to the IO-Link device. With the help of the IODD and the ODD Interpreter tool, the user can configure at which port of an IO-Link masters which IO-Link device is connected.

5 IO-Link system structure

5.1 The IO-Link device

Using the IO-Link protocol, the IO-Link device offers access to process data and variables of device functions. Certain variables have been defined, for instance, for identification purposes. The manufacturer must create the device variables in defined index areas. All this information is described in the *IODD* .

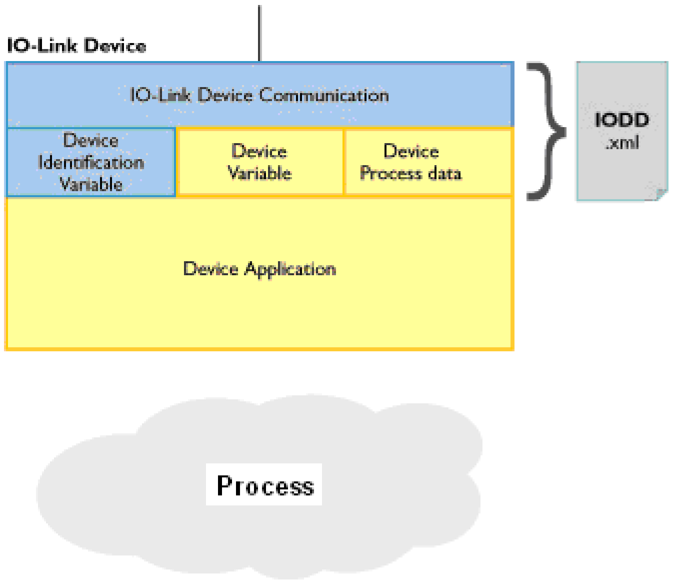

Figure 10: IO-Link device structure

5.2 IODD and the interpreter tool

The IODD contains information on communication properties, device parameters, identification, process and diagnostic data. It also includes an image of the device and the manufacturer's logo. The IODD structure is the same for all devices of all manufacturers and is always represented in the same way by the IODD interpreter tools. Therefore, the same handling of all IO-Link devices is guaranteed independent of the manufacturer.

The IODD is delivered as a package and consists of one or several *xml* files describing the device and image files in the *png* format. The "IODD-StandardDefinitions1.0.xml" file describes all universal and mandatory properties of the device. This file must be stored once in every supported language in the IODD directory. Further *xml* files describe the manufacturer-specific properties of a device.

An interpreter tool can read in an IODD and show the described device in a graphical form (to a limited extent only). It can be used to parameterize and to diagnose IO-Link devices of all manufacturers. At the same time an interpreter tool permits a transparent representation of the system architecture down to the field level.

The *IODD checker* was developed to validate IODDs. All IODDs must be tested with the IODD checker. The checker enters a checksum in the IODD. Interpreter tools that read in an IODD also create a checksum which must be identical with the checksum that has been entered in the IODD.

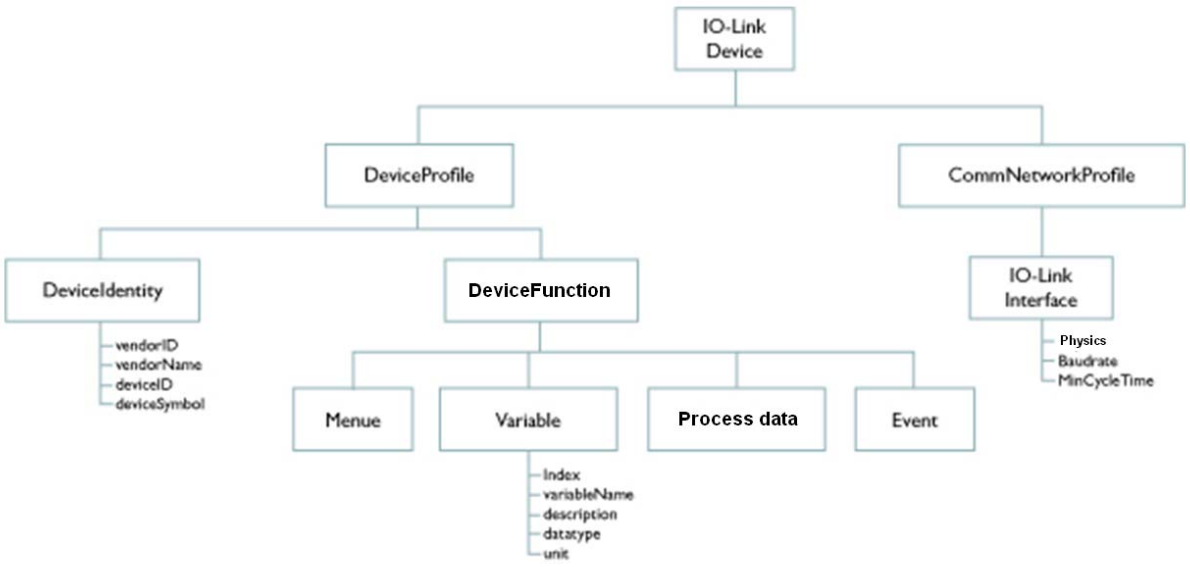

Figure 11: IODD structure

5.3 IO-Link master

The IO-Link master can be linked to the PLC in different ways (see Figure 1) and may have one or several ports.

The IO-Link specification distinguishes between two types of ports. Port *type A* where the function of pin 2 is not described in detail and can be freely defined by the manufacturer, and port *type B* for devices requiring a special power supply.

Port type A

For this port type, pin 4 can be configured as a digital input (DI) or an IO-Link. The manufacturers can also design pin 4 as a digital output (DO) with a limited output current. Pin 2 can also be used as desired. The manufacturers can design the pin, for example, as a DI or DO.

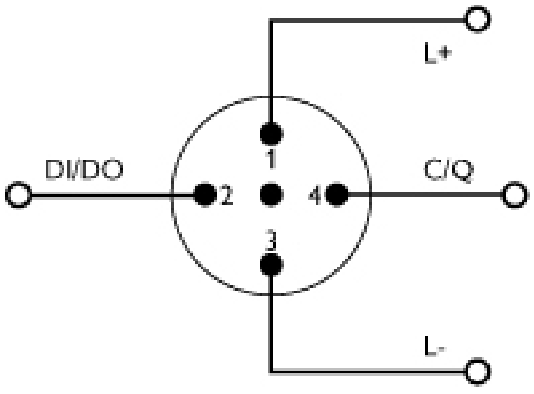

Figure 12: IO-Link master, port type A

Port type B

Port type B was designed, for instance, for sensors or actuators with electrically isolated power supply. Here, pins 2 and 5 are provided for additional power supply.

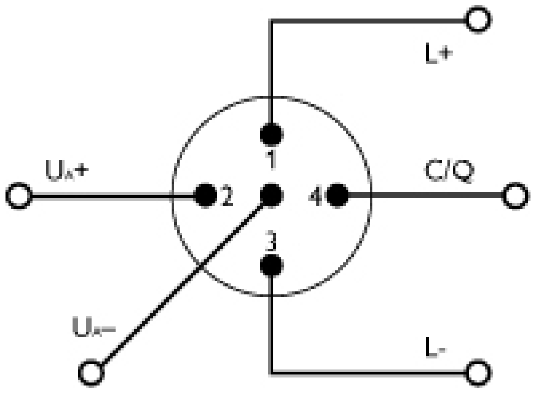

Figure 13: IO-Link master, port type B

6 List of figures

7 List of tables

8 Index