Siemens
Industry
Online
Support

APPLICATION EXAMPLE

IO-Link Library (LIOLink)

SIMATIC / TIA Portal / Block library

SIEMENS

Legal information

Use of application examples

Application examples illustrate the solution of automation tasks through an interaction of several components in the form of text, graphics and/or software modules. The application examples are a free service by Siemens AG and/or a subsidiary of Siemens AG ("Siemens"). They are non-binding and make no claim to completeness or functionality regarding configuration and equipment. The application examples merely offer help with typical tasks; they do not constitute customer-specific solutions. **The application examples are not subject to standard tests and quality inspections of a chargeable product and may contain functional and performance defects or other faults and security vulnerabilities. You are responsible for the proper and safe operation of the products in accordance with all applicable regulations, including checking and customizing the application example for your system, and ensuring that only trained personnel use it in a way that prevents property damage or injury to persons. You are solely responsible for any productive use.**

Siemens grants you the non-exclusive, non-sublicensable and non-transferable right to have the application examples used by technically trained personnel. Any change to the application examples is your responsibility. Sharing the application examples with third parties or copying the application examples or excerpts thereof is permitted only in combination with your own products. Any further use of the application examples is explicitly not permitted and further rights are not granted. You are not allowed to use application examples in any other way, including, without limitation, for any direct or indirect training or enhancements of AI models.

Disclaimer of liability

Siemens shall not assume any liability, for any legal reason whatsoever, including, without limitation, liability for the usability, availability, completeness and freedom from defects of the application examples as well as for related information, configuration and performance data and any damage caused thereby. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence, or culpable loss of life, bodily injury or damage to health, non-compliance with a guarantee, fraudulent non-disclosure of a defect, or culpable breach of material contractual obligations. Claims for damages arising from a breach of material contractual obligations shall however be limited to the foreseeable damage typical of the type of agreement, unless liability arises from intent or gross negligence or is based on loss of life, bodily injury or damage to health. The foregoing provisions do not imply any change in the burden of proof to your detriment. You shall indemnify Siemens against existing or future claims of third parties in this connection except where Siemens is mandatorily liable.

By using the application examples you acknowledge that Siemens cannot be held liable for any damage beyond the liability provisions described.

Other information

Siemens reserves the right to make changes to the application examples at any time without notice and to terminate your use of the application examples at any time. In case of discrepancies between the suggestions in the application examples and other Siemens publications such as catalogs, the content of the other documentation shall have precedence.

The Siemens terms of use (https://www.siemens.com/global/en/general/terms-of-use.html) shall also apply.

Cybersecurity information

Siemens provides products and solutions with industrial cybersecurity functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial cybersecurity concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial cybersecurity measures that may be implemented, please visit www.siemens.com/cybersecurity-industry.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Cybersecurity RSS Feed under https://www.siemens.com/cert.

Table of Contents

1. Introduction

Overview

Systematic diagnostics concepts and the efficient handling of parameter data are demanded at all levels of automation technology. Therefore, it is essential that sensors and switching devices are integrated into the automation network.

The IO-Link communication standard allows for new possibilities thanks to the intelligent linking of sensors and switching devices to the controller level. Core aspects are switching, protection and monitoring on the field level.

The IO-Link system consists of an IO-Link Master and one or more IO-Link devices (sensors or actuators). The IO-Link Master serves as an interface to the higher-level controller; it controls communication with the connected IO-Link devices.

This library provides blocks and PLC data types to facilitate communication between the SIMATIC Controller and the IO-Link Master or IO-Link device.

Range of functions

The blocks in this library are subdivided into three groups:

- General-purpose **base blocks** for communication with SIMATIC IO-Link Masters and any IO-Link devices and for reading out their IO-Link diagnostics.
- **Master-specific blocks** that are used for backup and restore of SIMATIC IO-Link Masters and for executing port functions.
- **Device-specific blocks** each simplify communication with a specific IO-Link device respectively, thanks to a custom-tailored interface and predefined PLC data types. These blocks are based on the base blocks.
- **Profile blocks** simplify communication with a device via a specific IO-Link profile. These blocks are based on the base blocks.

Validity

This library is available for TIA Portal V18 and higher.

All blocks in the library are available for SIMATIC S7-1200/1500 controllers.

2. Components of the Library

Below you will find a listing of all blocks in this library with the current version.

Base Blocks

Table 2-1: Base Blocks

Name	Version	Description
LIOLink_Device	V 5.1.1	Facilitates reading and writing of acyclic data of an IO-Link device
LIOLink_Diagnose	V 2.0.0	Lets you export diagnostics from all ports of the IO-Link Master.

Master-Specific Blocks

Table 2-2: Master-Specific Blocks

Name	Version	Description
LIOLink_Master	V 4.0.0	Facilitates backup and restore of device parameters and device settings of an IO-Link Master via the S7 program (master swap without Engineering Tool)
LIOLink_Toolchanger	V1.0.0	Allows you to suspend and resume port operation or diagnostics of ports on the IO-Link Master

Device-Specific Blocks

Table 2-3: Device-Specific Blocks

Name	Version	Description
LIOLink_3RA	V 4.1.1	Used for simple communication with SIRIUS 3RA devices
LIOLink_3RB24	V 4.1.1	Used for simple communication with SIRIUS 3RB24 devices
LIOLink_3RR24	V 4.1.1	Used for simple communication with SIRIUS 3RR24 devices
LIOLink_3RS1	V 4.1.1	Used for simple communication with SIRIUS 3RS1 devices
LIOLink_3RS2	V 2.1.1	Used for simple communication with SIRIUS 3RS2 devices
LIOLink_3SU1_ElecModule	V 3.1.1	Used for simple communication with SIRIUS 3SU1 electronics modules
LIOLink_3SU1_IDKeySwitch	V 3.1.1	Used for simple communication with SIRIUS 3SU1 ID key-operated switches
LIOLink_8WD46	V 1.1.1	Used for simple communication with SIRIUS 8WD46 indicator lights
LIOLink_3UG481x	V 4.1.1	Used for simple communication with SIRIUS 3UG481 devices
LIOLink_3UG4822	V 4.1.1	Used for simple communication with SIRIUS 3UG4822 devices
LIOLink_3UG4825	V 4.1.1	Used for simple communication with SIRIUS 3UG4825 devices
LIOLink_3UG4832	V 4.1.1	Used for simple communication with SIRIUS 3UG4832 devices
LIOLink_3UG4841	V 4.1.1	Used for simple communication with SIRIUS 3UG4841 devices
LIOLink_3UG4851	V 4.1.1	Used for simple communication with SIRIUS 3UG4851 devices
LIOLink_RF200	V 4.2.0	Used for simple communication with SIMATIC RF200 readers
LIOLink_RF200_ReadTag	V 3.0.1	Used specifically for reading a transponder via a SIMATIC RF200 reader
LIOLink_RF200_SwitchAntenna	V 3.0.0	Used specifically to switch on and off the antenna field of a SIMATIC RF200 reader
LIOLink_RF200_WriteTag	V 3.0.1	Used specifically for reading a transponder via a SIMATIC RF200 reader

Profile Blocks

Table 2-4: Profile Blocks

Name	Version	Description
LIOLink_AdjSwitchingSensor	V 2.2.0	Used for setting or teaching the setpoint value and for modifying the switching point logic of adjustable switching sensors (AdSS).
LIOLink_IdentAndDiag	V 2.2.0	Acyclically reads and writes identification and diagnostic data, and outputs the status of the connected IO-Link device. This function block supports the "Identification and Diagnosis" profile
LIOLink_MeasuredDataChannel	V 2.0.0	With this function block, measured values of the sensors can be acquired cyclically. The measured raw values from the sensors are processed and are available as Real or DInt measured values at the respective outputs. This function block supports Smart Sensor Profile type 3 ("Digital Measuring Sensors") and type 4 ("Digital Measuring Switching Sensors").
LIOLink_MultiAdjSwitchingSensor	V 2.1.0	Offers a unified interface for access and parameter assignment of IO Link devices that support the Smart Sensor Profile. In particular, the block can be used by sensors that can be assigned to the measuring device profile type 2, i.e. IO-Link devices that support the Smart Sensor Profile "Multiple Adjustable Switching Sensors".

Master Copies

Table 2-5: Master Copies

Name	Version	Description
LIOLink_MeasuredData		Used for the automatic acquisition and processing of process data from IO-Link sensors, regardless of the sensor type.

PLC data types

Table 2-6: PLC data types

Name	Version	Description
LIOLink_3RA_typeAll	V 3.0.1	Structure for all supported datasets for SIRIUS 3RA devices
LIOLink_3RA_typeConfig	V 3.0.1	Describes the datasets "actual configuration" (105) and "target configuration" (130)
LIOLink_3RA_typeConfigStarterLong	V 1.0.1	Describes the configurations for starters with type "long"
LIOLink_3RA_typeConfigStarterShort	V 1.0.1	Describes the configurations for starters with type "short"
LIOLink_3RA_typeDiag	V 3.0.1	Describes the dataset "Diagnostics" (92)
LIOLink_3RA_typeDiagStarterLong	V 1.0.1	Describes the diagnostics for starters with type "long"
LIOLink_3RA_typeDiagStarterShort	V 1.0.1	Describes the diagnostics for starters with type "short"
LIOLink_3RA_typeParameterPage1	V 3.0.1	Describes the dataset "Parameter page 1"
LIOLink_3RA_typePII	V 1.0.0	Describes the process image of the inputs
LIOLink_3RA_typePIQ	V 1.0.0	Describes the process image of the outputs
LIOLink_3RB24_typeAll	V 3.0.1	Structure for all supported datasets for SIRIUS 3RB24 devices
LIOLink_3RB24_typeDiag	V 3.0.1	Describes the dataset "Diagnostics" (92)
LIOLink_3RB24_typeMeasure	V 3.0.1	Describes the dataset "Measured values" (94)
LIOLink_3RB24_typePII	V 1.0.0	Describes the process image of the inputs
LIOLink_3RB24_typePIQ	V 1.0.0	Describes the process image of the outputs
LIOLink_3RB24_typePresetConfig	V 3.0.1	Describes the dataset "target configuration" (130)
LIOLink_3RB24_typeTechFunctions	V 3.0.1	Describes the dataset "technology functions" (131)
LIOLink_3RR24_typeAll	V 3.0.1	Structure for all supported datasets for SIRIUS 3RR24 devices
LIOLink_3RR24_typeDiag	V 3.0.1	Describes the dataset "Diagnostics" (92)
LIOLink_3RR24_typeMeasure	V 3.0.1	Describes the dataset "Measured values" (94)
LIOLink_3RR24_typeParam	V 3.0.1	Describes the dataset "Parameter" (131)
LIOLink_3RR24_typePII	V 1.0.0	Describes the process image of the inputs
LIOLink_3RR24_typePIQ	V 1.0.0	Describes the process image of the outputs

Name	Version	Description
LIOLink_3RS1_typeAll	V 3.0.1	Structure for all supported datasets for SIRIUS 3RS1 devices
LIOLink_3RS1_typeDiag	V 3.0.1	Describes the dataset "Diagnostics" (92)
LIOLink_3RS1_typeMeasure	V 3.0.1	Describes the dataset "Measured values" (94)
LIOLink_3RS1_typeParam	V 3.0.1	Describes the dataset "Parameter" (131)
LIOLink_3RS1_typePII	V 1.0.0	Describes the process image of the inputs
LIOLink_3RS1_typePIQ	V 1.0.0	Describes the process image of the outputs
LIOLink_3RS1_typeSensor	V 3.0.1	Describes the state of a temperature sensor
LIOLink_3RS2_typeAll	V 1.0.1	Structure for all supported datasets for SIRIUS 3RS2 devices
LIOLink_3RS2_typeCurrentInput	V 1.0.1	Describes the 4..20mA current input
LIOLink_3RS2_typeDiag	V 1.0.1	Describes the dataset "Diagnostics" (92)
LIOLink_3RS2_typeMeasure	V 1.0.1	Describes the dataset "Measured values" (94)
LIOLink_3RS2_typeMeasureFloat	V 1.0.1	Describes the dataset "Measured values" (94) in floating-point format
LIOLink_3RS2_typeParam	V 1.0.1	Describes the dataset "Parameter" (131)
LIOLink_3RS2_typePII	V 1.0.0	Describes the process image of the inputs
LIOLink_3RS2_typePIQ	V 1.0.0	Describes the process image of the outputs
LIOLink_3RS2_typeSensor	V 1.0.1	Describes the state of a temperature sensor
LIOLink_3SU1_ElecModule_typeAll	V 2.0.1	Structure for all supported datasets for SIRIUS ACT 3SU1 electronics modules
LIOLink_3SU1_ElecModule_typeDiag92	V 2.0.1	Describes the dataset "Diagnostics" (92)
LIOLink_3SU1_ElecModule_typeDiag94	V 2.0.1	Describes the dataset "Diagnostics" (94)
LIOLink_3SU1_ElecModule_typeParam	V 2.0.1	Describes the dataset "Parameter" (131)
LIOLink_3SU1_ElecModule_typePII	V 1.0.1	Describes the process image of the inputs
LIOLink_3SU1_ElecModule_typePIQ	V 1.0.1	Describes the process image of the outputs
LIOLink_3SU1_IDKeySwitch_typeAll	V 2.0.1	Structure for all supported datasets for SIRIUS ACT 3SU1 ID key-operated switches
LIOLink_3SU1_IDKeySwitch_typeDiag92	V 2.0.1	Describes the dataset "Diagnostics" (92)
LIOLink_3SU1_IDKeySwitch_typeDiag94	V 2.0.1	Describes the dataset "Diagnostics" (94)
LIOLink_3SU1_IDKeySwitch_typeKeyList1	V 2.0.1	Describes the dataset for "Individually codable ID key (1-30)" (81)
LIOLink_3SU1_IDKeySwitch_typeKeyList2	V 2.0.1	Describes the dataset for "Individually codable ID key (31-50)" (82)
LIOLink_3SU1_IDKeySwitch_typeKeyState	V 2.0.1	Describes the state of an ID key
LIOLink_3SU1_IDKeySwitch_typeParam	V 2.0.1	Describes the dataset "Parameter" (131)
LIOLink_3SU1_IDKeySwitch_typePII	V 1.0.0	Describes the process image of the inputs
LIOLink_3UG481_typeAll	V 3.0.1	Structure for all supported datasets for SIRIUS 3UG481 devices
LIOLink_3UG481_typeDiag	V 3.0.1	Describes the dataset "Diagnostics" (92)
LIOLink_3UG481_typeMeasure	V 3.0.1	Describes the dataset "Measured values" (94)
LIOLink_3UG481_typeParam	V 3.0.1	Describes the dataset "Parameter" (131)
LIOLink_3UG4822_typeAll	V 3.0.1	Structure for all supported datasets for SIRIUS 3UG4822 devices
LIOLink_3UG4822_typeDiag	V 3.0.1	Describes the dataset "Diagnostics" (92)
LIOLink_3UG4822_typeMeasure	V 3.0.1	Describes the dataset "Measured values" (94)
LIOLink_3UG4822_typeParam	V 3.0.1	Describes the dataset "Parameter" (131)
LIOLink_3UG4825_typeAll	V 3.0.1	Structure for all supported datasets for SIRIUS 3UG4825 devices
LIOLink_3UG4825_typeDiag	V 3.0.1	Describes the dataset "Diagnostics" (92)
LIOLink_3UG4825_typeMeasure	V 3.0.1	Describes the dataset "Measured values" (94)
LIOLink_3UG4825_typeParam	V 3.0.1	Describes the dataset "Parameter" (131)
LIOLink_3UG4832_typeAll	V 3.0.1	Structure for all supported datasets for SIRIUS 3UG4832 devices
LIOLink_3UG4832_typeDiag	V 3.0.1	Describes the dataset "Diagnostics" (92)
LIOLink_3UG4832_typeMeasure	V 3.0.1	Describes the dataset "Measured values" (94)
LIOLink_3UG4832_typeParam	V 3.0.1	Describes the dataset "Parameter" (131)
LIOLink_3UG4841_typeAll	V 3.0.1	Structure for all supported datasets for SIRIUS 3UG4841 devices
LIOLink_3UG4841_typeDiag	V 3.0.1	Describes the dataset "Diagnostics" (92)
LIOLink_3UG4841_typeMeasure	V 3.0.1	Describes the dataset "Measured values" (94)
LIOLink_3UG4841_typeParam	V 3.0.1	Describes the dataset "Parameter" (131)

Name	Version	Description
LIOLink_3UG4851_typeAll	V 3.0.1	Structure for all supported datasets for SIRIUS 3UG4851 devices
LIOLink_3UG4851_typeDiag	V 3.0.1	Describes the dataset "Diagnostics" (92)
LIOLink_3UG4851_typeMeasure	V 3.0.1	Describes the dataset "Measured values" (94)
LIOLink_3UG4851_typeParam	V 3.0.1	Describes the dataset "Parameter" (131)
LIOLink_8WD46_typeAll	V 1.0.0	Structure that summarizes all possible parameters of 8WD46
LIOLink_8WD46_typeLightEffects	V 1.0.0	Structure for parameterizing lighting effects
LIOLink_8WD46_typeSoundMode	V 1.0.0	Structure for parameterizing sound effects
LIOLink_RF200_typeAll	V 3.0.0	Structure for all supported datasets for SIMATIC RF200 readers
LIOLink_RF200_typeEventHistory	V 3.0.1	Describes the dataset "Event history" (0x4A)
LIOLink_RF200_typeParameters	V 1.0.0	Describes the dataset "Reader parameters" (0x40)
LIOLink_RF200_typePII	V 1.0.0	Describes the process image of the inputs
LIOLink_RF200_typePIQ	V 1.0.0	Describes the process image of the outputs
LIOLink_RF200_typeReaderStatus	V 3.0.0	Describes the dataset "Reader status" (0x5A)
LIOLink_RF200_typeTagStatus	V 3.0.0	Describes the dataset "Transponder status" (0x5B)
LIOLink_RF200_typeUIDHistory	V 3.0.0	Describes the dataset "UID history" (0x5C)
LIOLink_typeDiagnostics	V 1.0.0	Provides a diagnostic structure for various blocks in the library in order to give detailed information in the event of an error.
LIOLink_typeEvents	V 1.0.0	Describes the events of the individual ports of an IO-Link Master
LIOLink_typePortEventCodes	V 1.0.0	Structure for event information: Event codes and EventQualifier
LIOLink_typePortEventQualifier	V 1.0.0	Structure for EventQualifier: Instance, source, type, mode
LIOLink_typePortEvents	V 1.0.0	Structure for displaying the last 5 events of a port
LIOLink_typeIdentificationObjects	V 1.0.1	Describes all profile-relevant device parameters according to Common profile
LIOLink_typeParameterPage0	V 3.0.1	Describes parameter page 0, which all IO-Link devices support by default.
LIOLink_typeConfigMultiAdjSwSensor	V 1.0.0	Defines the values for the configuration settings to be read/written on a request using the rd_all/wr_conf function
LIOLink_typeParamMultiAdjSwSensor	V 1.0.0	Defines the values for the setpoint parameters to be read/written on a request with the function rd_all/wr_conf
LIOLink_typeSensorInfo		Information about the sensor. This information is required to detect the sensor and process the data coming from the sensor

NOTE Further information about the individual datasets can be found in the associated device manuals.

3. Base Blocks

3.1. LIOLink_Device

3.1.1. Description

By reading and writing acyclic data, you can write device parameters to an IO-Link device, or read parameters, measured values and diagnostic data from an IO-Link device.

The function block can only be used on a PLC of the S7-1200 family from firmware version 4.6.

The function block supports you in the following tasks:

- (Re)parameterization of an IO-Link device
- Diagnosing an IO-Link device
- Executing IO-Link port functions
- Backing up/restoring IO-Link device parameters

The data on the IO-Link device are uniquely addressed with index and subindex. Additionally, the function block can be used to execute port functions.

NOTE The structure of the data objects and port functions can be found in the respective documentation on the IO-Link device and IO-Link Master.

The function block is based on a standardized protocol (PROFIBUS DP/PROFINET IO), which ensures access to data from an IO-Link device via the IO-Link Master. Essentially, a sequence of acyclic read and write accesses is used, which are represented by the SIMATIC system functions "RDREC" and "WRREC".

The block is an asynchronously functioning function block, i.e. processing extends over multiple PLC cycles.

NOTE If a DPV1 slave is configured via GSD file (GSD rev. 3 or later) and the DP interface of the DP master is set to "S7 compatible", then the block will not function correctly.

Remedy: Set the interface for the DP master to "DPV1".

NOTE This block replaces the previously available blocks "IO_LINK_DEVICE" and "IO_LINK_CALL".

3.1.2. Parameter

Figure 3-1: LIOLink_Device

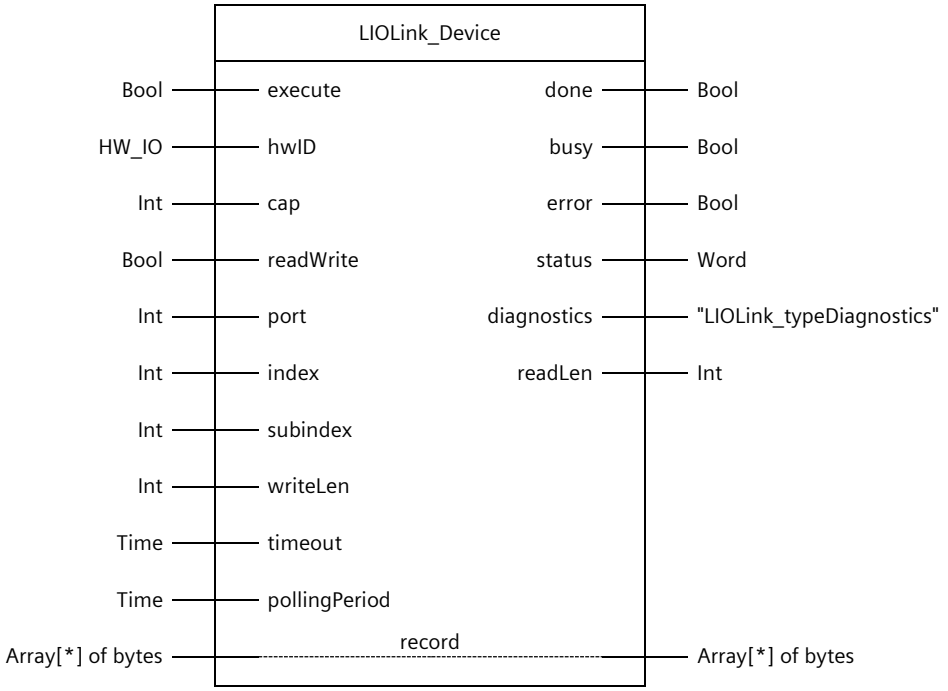


Table 3-1: Parameters of LIOLink_Device

Name	P type	Data type	Comment
execute	IN	Bool	Request to execute the function
hwID	IN	HW_IO	Hardware identifier of the IO-Link Master or of the first submodule.
cap (optional)	IN	Int	Access point (Client Access Point): When using Siemens IO-Link Masters, is automatically detected, and is therefore optional. If the parameter is changed, automatic detection is not active.
readWrite	IN	Bool	Mode FALSE: read TRUE: write
port	IN	Int	Port number at which the IO-Link device is operated Possible values: 0..63
index	IN	Int	Parameter index Possible values: 0..32767 65535 (0xFFFF): IOL-D - port functions
subindex	IN	Int	Parameter subindex 0: entire record 1..255: Parameter from record
writeLen	IN	Int	Length of the data in bytes+ (net data) to be written Possible values for writing: 1..232 Not relevant when reading
timeout	IN	Time	Time after which a command is canceled
pollingPeriod (optional)	IN	Time	Variably adjustable time that the block waits until the dataset is transmitted. Default value: 100 ms
done	OUT	Bool	TRUE: Command successfully executed
busy	OUT	Bool	TRUE: Command is currently being processed
error	OUT	Bool	TRUE: An error has occurred while processing the FB
status	OUT	Word	16#0000–16#7FFF: Status of the FB 16#8000–16#FFFF: Error codes (see Section 3.1.4).

Name	P type	Data type	Comment
diagnostics	OUT	"LIOLink_ typeDiagnostics"	Detailed diagnostic information of the FB (see Section 3.1.4)
readLen	OUT	Int	Length of data read in bytes (net data)
record	IN_OUT	Array[*] of bytes	Source/destination range for the data to be read/written. Possible range of values: 0..231

3.1.3. Principle of Operation

Addressing

The desired dataset is uniquely addressed with the parameters "index" and "subindex".

When writing data, the data quantity specified at the "writeLen" parameter is transmitted to the IO-Link device. The parameter is irrelevant when reading.

The access point to the IO-Link Master is defined using the "cap" parameter (Client Access Point): When using IO-Link Masters from Siemens AG, detection is automatic. However, if the parameter is changed, automatic detection is not active. Normally, the access point is 0xB400 or 0x00E3.

Chronological sequence of a transmission

Data transmission is started with a positive edge at the input "execute".

The outputs "done", "busy", "error" and "status" show the status of the command.

After successful execution, the parameter "len" shows the length of the received or written data.

As long as the input "execute" is set, the output parameters retain their value. If the "execute" input is reset before the processing of the FB is completed, the values of the output parameters are output for one cycle after the command is processed.

If the operation exceeds the time at the parameter "timeout", the processing will be aborted and an error will be output.

NOTE

The data transmission takes place in the form of raw data (ARRAY of bytes), i.e. the data cannot be interpreted in this form.

It is left to the user to format the data in accordance with the specifications of the device manufacturer (copy to a data structure or data type).

Confirming parameter changes

After all desired changes of the IO-Link device parameters have been written with the block, the data must be verified by a system command so that they are ready in the master as a backup.

You can execute this system command with the following operation:

Table 3-2: Confirming parameter changes

Parameter	Value
mode	1
index	2
subindex	0
writeLen	1
record[0]	0x05

Figure 3-2: Excerpt from the IO-Link specification

Table B.9 – Coding of SystemCommand (ISDU)

Command (hex)	Command (dec)	Command name	M/O	Definition
0x00	0	Reserved		
0x01	1	ParamUploadStart	O	Start parameter upload
0x02	2	ParamUploadEnd	O	Stop parameter upload
0x03	3	ParamDownloadStart	O	Start parameter download
0x04	4	ParamDownloadEnd	O	Stop parameter download
0x05	5	ParamDownloadStore	O	Finalize parameterization and start Data Storage

3.1.4. Error Handling

The "status" output outputs the current status and any errors, while the "diagnostics" output provides a diagnostic structure with detailed information in the event of an error.

status

Table 3-3: Output "status" from LIOLink_Device

Status	Meaning
16#0000	Operation completed, no warning and no further details
16#7000	No operation in progress (initial value)
16#7001	First call after input of a new command (rising edge on "execute")
16#7002	Subsequent call
16#8201	Unsupported port
16#8202	Unsupported index
16#8203	Unsupported subindex
16#8205	The length at the "writeLen" parameter does not match the dataset that will be written
16#8401	The IO-Link Master has reported an error code, see "diagnostics"
16#8402	Received dataset does not match operation
16#8403	Operation could not be completed in the specified time
16#8600	Internal state machine has reached an undefined state
16#8601	System function WRREC reports an error, see "diagnostics"
16#8602	System function RDREC reports an error, see "diagnostics"

diagnostics

In the event of an error, the "diagnostics" output gives detailed information about the pending error.

Table 3-4: "diagnostics" output of LIOLink_Device

Tag	Description
status	Last status code of the interface parameter "status" of the FB.
subfunctionStatus	Status of system functions RDREC/WRREC or error code from IO-Link Master (%W1: Error code from IO-Link Master, %W0: ISDU error code). For detailed information, refer to the Online Help for the system function in question, or the device manual of the IO-Link Master/device.
stateNumber	State of the FB's state machine in which the error occurred.

3.2. LIOLink_Diagnose

3.2.1. Description

Using this block, it is possible to export detailed diagnostic data from a Siemens IO-Link Master and the devices connected to it.

NOTE This function block provides diagnostics for IO-Link Masters with up to 8 ports. If there are more than 8 ports, the block only provides diagnostic data for the first 8 ports.

Validity

The block is approved for the following IO-Link Masters:

- ET 200SP, CM 4xIO-Link (6ES7137-6BD00-0BA0) firmware V2.2.0 or higher
- ET 200pro, EM 4 IO-LINK HF (6ES7147-4JD00-0AB0) firmware V1.0.0 or higher
- ET 200AL, CM 4xIO-Link (6ES7147-5JD00-0BA0) firmware V1.1.0 or higher
- ET 200AL, CM 4xIO-Link (6ES7 147-5JD01-0BA0) firmware V2.0.0 or higher
- ET 200AL, CM 8xIO-Link + DIQ 6x24VDC (6ES7147-5JJ00-0BA0) firmware V2.0.0 or higher
- ET 200eco PN, IO-Link Master (6ES7148-6JD00-0AB0) firmware V1.0.0 or higher
- ET 200eco PN, CM 8x IO-Link, M12-L (6ES7148-6JG00-0BB0) firmware V1.0.0 or higher
- ET 200eco PN, CM 4x IO-Link + DIQ 12x 24VDC (6ES7148-6JE00-0BB0) firmware V1.0.0 or higher
- ET 200eco PN, CM 8x IO-Link + DIQ 8x 24VDC (6ES7148-6JJ00-0BB0) firmware V1.0.0 or higher
- ET 200clean, CM 8x IO-Link + DIQ 4x 24VDC (6ES7148-7JH00-0BB0) firmware V1.1.0 or higher
- S7-1200, SM 1278 4 IO-Link (6ES7278-4BD32-0XB0) firmware V2.0.0 or higher
- S7-1500, CM 8xIO-Link (6ES7547-1JF00-0AB0) firmware V1.0.0 or higher

3.2.2. Parameter

Figure 3-4: LIOLink_Diagnose

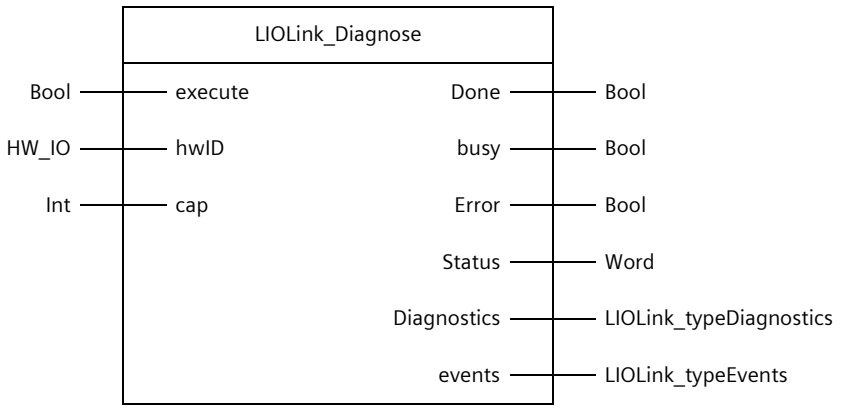


Table 3-9: Parameters of LIOLink_Diagnose

Name	P type	Data type	Comment
execute	IN	Bool	Request to execute the function
hwID	IN	HW_IO	Hardware identifier of the IO-Link Master or of the first submodule.

Name	P type	Data type	Comment
cap (optional)	IN	Int	Access point (Client Access Point): When using Siemens IO-Link Masters, is automatically detected, and is therefore optional. If the parameter is changed, automatic detection is not active. Normally, the access point is 0xB400 (dec. -19456) or 0x00E3 (dec. 227), and is specified in the respective manual.
done	OUT	Bool	TRUE: Command successfully executed
busy	OUT	Bool	TRUE: Command is currently being processed
error	OUT	Bool	TRUE: An error has occurred while processing the FB
status	OUT	Word	16#0000–16#7FFF: Status of the FB 16#8000–16#FFFF: Error codes (see Section 3.2.4).
diagnostics	OUT	"LIOLink_ typeDiagnostics"	Detailed diagnostic information of the FB (see Section 3.2.4)
events	OUT	"LIOLink_ typeEvents"	The events of the individual ports of an IO-Link Master (see Section 3.2.5)

3.2.3. Principle of Operation

Data transmission is started with a positive edge at the input "execute".

The outputs "done", "busy", "error" and "status" show the status of the command.

Once successfully executed, the "events" parameter shows the events/diagnostics for all IO-Link Master ports and the connected devices.

As long as the input "execute" is set, the output parameters retain their value. If the "execute" input is reset before the processing of the FB is completed, the values of the output parameters "done", "busy", "error" and "status" are output for one cycle after the job is processed. The "events" output parameter is updated every time the "execute" input is triggered.

3.2.4. Error Handling

Error handling is accomplished with the two outputs "status" and "diagnostics", and is explained in more detail below.

status

The "status" output outputs the current status of the block along with error codes for the executed system function RDREC.

Table 3-5: Output "status" from LIOLink_Diagnose

Code / Value	Identifier/Description
16#0000	Operation completed, no warning and no further details
16#7000	No operation in progress (initial value)
16#7001	First call after input of a new command (rising edge on "execute")
16#7002	Subsequent call
16#8600	Internal state machine has reached an undefined state
16#8602	System function RDREC reports an error, see "diagnostics"
16#8603	Error: the manually entered CAP is not supported; only CAP=227 or CAP=0xB400 (-19456) are supported

diagnostics

In the event of an error, the "diagnostics" output gives detailed information about the pending error.

Table 3-6: "diagnostics" output of LIOLink_Diagnose

Tag	Description
status	Last status code of the interface parameter "status" of the FB.
subfunctionStatus	Status of system functions RDREC/WRREC or error code from IO-Link Master (%W1: Error code from IO-Link Master, %W0: ISDU error code). For detailed information, refer to the Online Help for the system function in question, or the device manual of the IO-Link Master/device.
stateNumber	State of the FB's state machine in which the error occurred.

3.2.5. Events

The output parameter "events" provides the last 5 events for each port of the IO-Link Master. These events are based on the IO-Link standard; they can either be events of the port or of the device. This distinction is made with the "source" bit in Table 3-10: "LIOLink_typePortEventQualifier".

The "events" parameter is principally divided into various UDTs, which are listed in hierarchical order in the Tables below. The UDT "LIOLink_typeEvents" is first assigned to the output parameter. It has an array containing the events of all 8 ports (see Table 3-7: "LIOLink_typeEvents").

Table 3-7: "LIOLink_typeEvents"

Tag	Data type	Description
port	Array[1..8] of LIOLink_typePortEvents	Contains the events of all 8 ports.

The UDT "LIOLink_typePortEvents" in Table 3-8 in turn contains an array with 5 fields for storing the last 5 events.

Table 3-8: "LIOLink_typePortEvents"

Tag	Data type	Description
event	Array[0..4] of LIOLink_typePortEventCodes	Describes the latest 5 events that occurred

Here, each event is described with an "eventCode" and an "eventQualifier" (see Table 3-9). This is defined in the IO-Link specification.
Possible event codes are listed in Table 3-11 and in Table 3-12.

Table 3-9: "LIOLink_typePortEventCodes"

Tag	Data type	Description
eventCode	Word	IO-Link EventCode
eventQualifier	LIOLink_typePortEventQualifier	IO-Link EventQualifier

EventQualifier consists of the tags "instance", "source", "type" and "mode". Here, "instance" indicates the specific source (instance) of an event, thus making it easier for the recipient to evaluate. In addition, the "source" tag indicates the source of the event, while "type" indicates the event category and "mode" the event mode. These EventQualifiers are defined in the IO-Link specification.

Table 3-10: "LIOLink_typePortEventQualifier"

Tag	Data type	Description
instance	Byte	0: Unknown / 1-3: reserved / 4: Application / 5-7: reserved
source	Bool	0: Device (remote) / 1: Master/Port
type	Byte	0: reserved / 1: Notification / 2: Warning / 3: Error
mode	Byte	0: reserved / 1: Event single shot / 2: Event disappears / 3: Event appears

NOTE Depending on the "events.Port[x].event[y].eventQualifier.source" bit, there are two different Tables that define event codes for IO-Link devices and the event codes for the ports of the master.

3.2.6. Event Codes for IO-Link Devices

Table 3-11 lists all event codes that are defined for the IO-Link devices. Accordingly, the "source" bit (see Table 3-10) is not set. The list is taken from the IO-Link specification V1.1.3. Vendor-specific event codes can be found in the IODD of the respective devices under the event codes: 0x8CA0 - 0x8DFF.

Table 3-11: Event codes for IO-Link devices

EventCode	Definition and recommended maintenance action	Event category
0x0000	No malfunction	Notification
0x1000	General malfunction – unknown error	Error
0x1001 - 0x17FF	Reserved	
0x1800 - 0x18FF	Vendor specific	
0x1900 - 0x3FF	Reserved	
0x4000	Temperature fault – Overload	Error
0x4001 - 0x420F	Reserved	
0x4210	Device temperature overrun – Clear source of heat	Warning
0x4211 - 0x421F	Reserved	
0x4220	Device temperature underrun – Insulate Device	Warning
0x4221 - 0x4FFF	Reserved	
0x5000	Device Hardware fault – Device exchange	Error
0x5001 - 0x500F	Reserved	
0x5010	Component malfunction – Repair of exchange	Error
0x5011	Non volatile memory loss – Check batteries	Error
0x5012	Batteries low – Exchange batteries	Warning
0x5013 - 0x50FF	Reserved	
0x5100	General power supply fault – Check availability	Error
0x5101	Fuse blown/open – Exchange fuse	Error
0x5102 - 0x510F	Reserved	
0x5110	Primary supply voltage overrun – Check tolerance	Warning
0x5111	Primary supply voltage underrun – Check tolerance	Warning
0x5112	Secondary supply voltage fault (Port Class B) – Check tolerance	Warning
0x5113 - 0x5FFF	Reserved	
0x6000	Device software fault – Check firmware version	Error
0x6001 - 0x631F	Reserved	
0x6320	Parameter error – Check data sheet and values	Error
0x6321	Parameter missing – Check data sheet	Error
0x6322 - 0x634F	Reserved	
0x6350	Reserved	
0x6351 - 0x76FF	Reserved	
0x7700	Wire break of a subordinate device – Check installation	Error
0x7701 - 0x770F	Wire break of subordinate device 1 ...device 15 – Check installation	Error
0x7710	Short circuit – Check installation	Error
0x7711	Ground fault – Check installation	Error
0x7712 - 0x8BFF	Reserved	
0x8C00	Technology specific application fault – Reset Device	Error
0x8C01	Simulation active – Check operational mode	Warning
0x8C01 - 0x8C0F	Reserved	
0x8C10	Process variable range overrun – Process Data uncertain	Warning
0x8C11 - 0x8C1F	Reserved	

EventCode	Definition and recommended maintenance action	Event category
0x8C20	Measurement range exceeded – Check application	Error
0x8C21 - 0x8C2F	Reserved	
0x8C30	Process variable range underrun – Process Data uncertain	Warning
0x8C31 - 0x8C3F	Reserved	
0x8C40	Maintenance required – Cleaning	Warning
0x8C41	Maintenance required – Refill	Warning
0x8C42	Maintenance required – Exchange wear and tear parts	Warning
0x8C43 - 0x8C9F	Reserved	
0x8CA0 - 0x8DFF	Vendor specific	
0x8E00 - 0xAFFF	Reserved	
0xB000 - 0xB0FF	Reserved for Safety extensions	
0xB100 - 0xBFFF	Reserved for profiles	
0xC000 - 0xFF90	Reserved	
0xFF91	Data Storage upload request ("DS_UPLOAD_REQ") – internal, not visible to user	Notification (single shot)
0xFF92 - 0xFFAF	Reserved	
0xFFB0 - 0xFFB7	Reserved for Wireless extensions	
0xFFB8 - 0xFFFF	Reserved	

3.2.7. Event Codes for the IO-Link Master Ports

Table 3-12 lists all event codes that are defined for the ports of the IO-Link Master. Accordingly, the "source" bit (see Table 3-10) is set. The list is taken from the IO-Link specification V1.1.3.

Table 3-12: Event codes for IO-Link Master ports

EventCode	Definition and recommended maintenance action	Event category
0x0000 to 0x17FF	Reserved	
0x1800	No device (communication)	Error
0x1801	Startup parametrization error – check parameter	Error
0x1802	Incorrect VendorID – Inspection Level mismatch	Error
0x1803	Incorrect DeviceID – Inspection Level mismatch	Error
0x1804	Short circuit at C/Q – check wire connection	Error
0x1805	PHY overtemperature – check Master temperature and load	Error
0x1806	Short circuit at L+ - check wire connection	Error
0x1807	Overcurrent at L+ - check power supply (e.g. L1+)	Error
0x1808	Device Event overflow	Error
0x1809	Backup inconsistency – memory out of range (2048 octets)	Error
0x180A	Backup inconsistency – identity fault	Error
0x180B	Backup inconsistency – Data storage unspecific error	Error
0x180C	Backup inconsistency – upload fault	Error
0x180D	Parameter inconsistency - download fault	Error
0x180E	P24 (Class B) missing or undervoltage	Error
0x180F	Short circuit at P24 (Class B) – check wire connection (e.g. L2+)	Error
0x1810	Short circuit at I/Q – check wiring	Error
0x1811	Short circuit at C/Q (if digital output) – check wiring	Error
0x1812	Overcurrent at I/Q – check load	Error
0x1813	Overcurrent at C/Q (if digital output) – check load	Error
0x1814 to 0x1EFF	Reserved	
0x1F00	Vendor specific	
0x2000 to 0x2FFF	Safety extensions	
0x3000 to 0x3FFF	Wireless extensions	
0x4000 to 0x5FFF	Reserved	

EventCode	Definition and recommended maintenance action	Event category
0x6000	Invalid cycle time	Error
0x6001	Revision fault – incompatible protocol version	Error
0x6002	ISDU batch failed – parameter inconsistency?	Error
0x6003 to 0xFF20	Reserved	
0xFF21	DL: Device plugged in ("NEW SLAVE") – PD stop	Notification
0xFF22	Device communication lost	Notification
0xFF23	Data Storage identification mismatch	Notification
0xFF24	Data Storage buffer overflow	Notification
0xFF25	Data Storage parameter access denied	Notification
0xFF26	Port status changed	Notification
0xFF27	Data Storage upload completed and new data object available	Notification
0xFF28 to 0xFF30	Reserved	
0xFF31	DL: Incorrect Event signaling	Notification
0xFF32 to 0xFFFF	Reserved	

4. Master-Specific Blocks

4.1. LIOLink_Master

4.1.1. Description

Using this block, you can back up the device parameters and device settings of an IO-Link Master via the S7 program (Backup), or restore (Restore) (master swap without Engineering Tool).

NOTE This block replaces the previously available block "IO_LINK_MASTER_4" and "IO_LINK_MASTER_8".

Validity

The block is approved for the following IO-Link Masters:

- ET 200SP, CM 4xIO-Link (6ES7137-6BD00-0BA0) firmware V2.2.0 or higher
- ET 200pro, EM 4 IO-LINK HF (6ES7147-4JD00-0AB0) firmware V1.0.0 or higher
- ET 200AL, CM 4xIO-Link (6ES7147-5JD00-0BA0) firmware V1.1.0 or higher
- ET 200AL, CM 4xIO-Link (6ES7 147-5JD01-0BA0) firmware V2.0.0 or higher
- ET 200AL, CM 8xIO-Link + DIQ 6x24VDC (6ES7147-5JJ00-0BA0) firmware V2.0.0 or higher
- ET 200eco PN, IO-Link Master (6ES7148-6JD00-0AB0) firmware V1.0.0 or higher
- ET 200eco PN, CM 8x IO-Link, M12-L (6ES7148-6JG00-0BB0) firmware V1.0.0 or higher
- ET 200eco PN, CM 4x IO-Link + DIQ 12x 24VDC (6ES7148-6JE00-0BB0) firmware V1.0.0 or higher
- ET 200eco PN, CM 8x IO-Link + DIQ 8x 24VDC (6ES7148-6JJ00-0BB0) firmware V1.0.0 or higher
- ET 200clean, CM 8x IO-Link + DIQ 4x 24VDC (6ES7148-7JH00-0BB0) firmware V1.1.0 or higher
- S7-1200, SM 1278 4 IO-Link (6ES7278-4BD32-0XB0) firmware V2.0.0 or higher
- S7-1500, CM 8xIO-Link (6ES7547-1JF00-0AB0) firmware V1.0.0 or higher

NOTE The function "Master Backup" is only available for IO-Link devices that are specified for the IO-Link standard V1.1 or higher.

IO-Link Masters from third parties are not supported by this block.

4.1.2. Parameter

Figure 4-1: LIOLink_Master

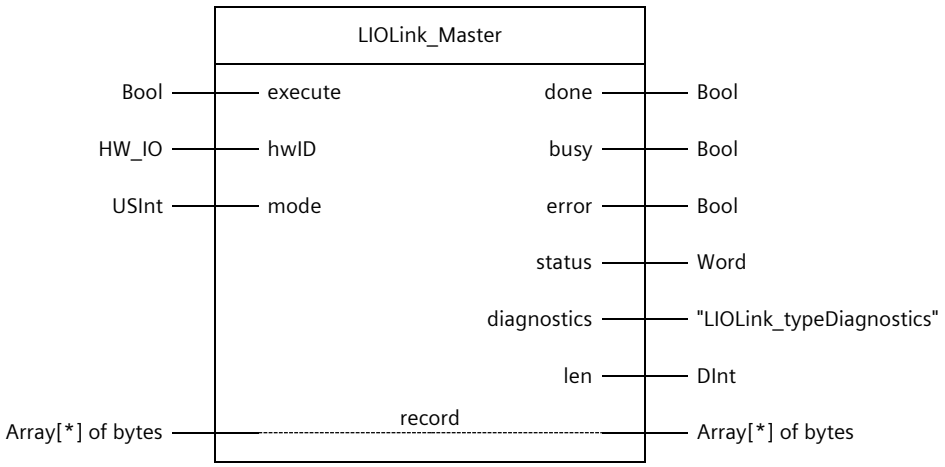


Table 4-1: LIOLink_Master parameters

Name	P type	Data type	Comment
execute	IN	Bool	Request to execute the function
hwID	IN	HW_IO	Hardware identifier of the IO-Link Master or of the first submodule.
mode	IN	USInt	Mode 0: Backup 1: Restore
done	OUT	Bool	TRUE: Command successfully executed
busy	OUT	Bool	TRUE: Command is currently being processed
error	OUT	Bool	TRUE: An error has occurred while processing the FB
status	OUT	Word	16#0000–16#7FFF: Status of the FB 16#8000–16#FFFF: Error codes (see Section 4.1.4).
diagnostics	OUT	"LIOLink_ typeDiagnostics"	Detailed diagnostic information of the FB (see Section 4.1.4)
len	OUT	DInt	Length in bytes of the read/written data (net data)
record	IN_OUT	Array[*] of bytes	Source/destination range for the data to be read/written. Valid limits: • 4-port master: 0..10239 • 8-port master: 0..17549

4.1.3. Principle of Operation

Data transmission is started with a positive edge at the input "execute".

The block functions asynchronously, i.e. processing extends over multiple PLC cycles. The block essentially uses a sequence of acyclic read/write accesses, which are handled by the system functions "RDREC" and "WRREC".

The outputs "done", "busy", "error" and "status" show the status of the command.

After successful execution, the parameter "len" shows the length of the received or written data.

NOTE In order to use as few resources as possible, the block works directly in the memory area at the "record" parameter. The data in the memory area are not allowed to change during a write operation, and may only be read once a read action is complete. For this purpose, evaluate the output parameter "done" or "status".

As long as the input "execute" is set, the output parameters retain their value. If the "execute" input is reset before the processing of the FB is completed, the values of the output parameters are output for one cycle after the command is processed.

NOTE Data is transmitted in the form of raw data (array of bytes), i.e. the data cannot be interpreted and must not be changed or manipulated!

Before a dataset can be written, it must be backed up first.

NOTE If a DPV1 slave is configured via GSD file (GSD rev. 3 or later) and the DP interface of the DP master is set to "S7 compatible", then the block "LIOLink_Master" will not function correctly.

Set the interface of the DP master to "DPV1" instead.

ATTENTION **Data loss**

The combination of a SIMATIC S7-1200 CPU with an 8-port IO-Link Master exceeds the level of retentive data available in the CPU.

If this combination is used, data loss will occur because the data cannot be completely retentively stored in the CPU.

4.1.4. Error Handling

The "status" output outputs the current status and any errors, while the "diagnostics" output provides a diagnostic structure with detailed information in the event of an error.

status

Table 4-2: Output "status" from LIOLink_Master

Status	Meaning
16#0000	Operation completed, no warning and no further details
16#7000	No operation in progress (initial value)
16#7001	First call after input of a new command (rising edge on "execute")
16#7002	Subsequent call during active processing without further details
16#71xx	Subsequent call during backup, xx = current sequence number
16#72xx	Subsequent call during restore, xx = current sequence number
16#80B0	Unknown module type
16#8200	Unsupported value at "mode" parameter
16#8201	Array at the "record" parameter does not match the expected limits
16#8401	IO-Link Master has reported back a sequence number that indexes an error, see "diagnostics"
16#8600	Internal state machine has reached an undefined state
16#8601	System function WRREC reports an error during a reset, see "diagnostics"
16#8602	System function RDREC reports an error during backup, see "diagnostics"
16#8603	System function WRREC reports an error during restore, see "diagnostics"
16#8604	System function RDREC reports an error during verification of the restore, see "diagnostics"

diagnostics

In the event of an error, the "diagnostics" output gives detailed information about the pending error.

Table 4-3: "diagnostics" output from LIOLink_Master

Tag	Description
status	Last status code of the interface parameter "status" of the FB.
subfunctionStatus	Status of system functions RDREC/WRREC or error code from IO-Link Master. For detailed information, refer to the Online Help for the system function in question, or the device manual of the IO-Link Master.
stateNumber	State of the FB's state machine in which the error occurred.

Table 4-4: Output "subfunctionStatus" Error code from LIOLink_Master

subfunctionStatus	Meaning
16#FFFF_FF01	Restore in progress
16#FFFF_FF02	Activation in progress
16#FFFF_FF03	Activation complete
16#FFFF_FF04	CRC check failed
16#FFFF_FF05	Blob data invalid

4.2. LIOLink_Toolchanger

4.2.1. Description

Port functions can be used to suspend and resume port diagnostics or error messages depending on the use case.

The function block supports you in the following tasks:

- "Suspend port operation"

- "Resume port operation"

The concept behind "suspend port operation" is to suppress all fault messages to the system, as it is an intentional action. Basically, all error messages of the port and device in question are deleted after the suspension. In this state, it is assumed that the user program continues to process data and Port Qualifier Information (PQI) and that these are therefore still active and updated. However, a device can be removed from a port in a suspended state without triggering specific events or diagnostics. System diagnostics will not report an error.

The concept behind "Resume Port Operation" is to resume the diagnostic mechanism. The current diagnostics are restored and updated.

A possible use case can be a machine that undocks a specific tool, such as a gripper, in a magazine and docks another. This docking and undocking can trigger events and diagnostics in the system or the user program, as communication and the power supply to the device or tool are interrupted. Suppressing the diagnosis before undocking the "previous" tool and resuming the diagnosis after docking the "new" tool eliminates this problem during operation.

NOTE For more information on Tool Changer use case and port functions, see IO-Link Integration – Profile for PROFINET

.

Validity

The block is approved for the following IO-Link Masters:

- ET 200AL, CM 4xIO-Link (6ES7 147-5JD01-0BA0) firmware V2.0.0 or higher
- ET 200AL, CM 8xIO-Link + DIQ 6x24VDC (6ES7147-5JJ00-0BA0) firmware V2.0.0 or higher
- ET 200eco PN, CM 8x IO-Link, M12-L (6ES7148-6JG00-0BB0) firmware V5.2.0 or higher
- ET 200eco PN, CM 4x IO-Link + DIQ 12x 24VDC (6ES7148-6JE00-0BB0) firmware V5.1.0 or higher
- ET 200eco PN, CM 8x IO-Link + DIQ 8x 24VDC (6ES7148-6JJ00-0BB0) firmware V5.1.0 or higher
- ET 200clean, CM 8x IO-Link + DIQ 4x 24VDC (6ES7148-7JH00-0BB0) firmware V1.1.0 or higher

4.2.2. Parameter

Figure 4-2: LIOLink_Toolchanger

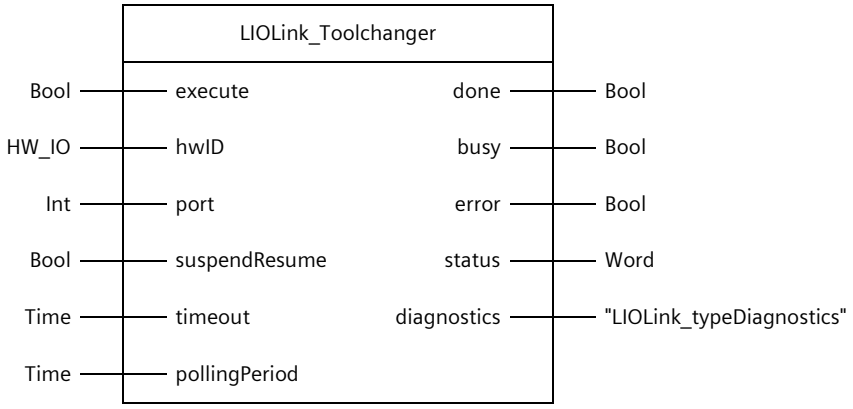


Table 4-5: Parameters of LIOLink_Toolchanger

Name	P type	Data type	Comment
execute	IN	Bool	Request to execute the function
hwID	IN	HW_IO	Hardware identifier of the IO-Link Master or of the first submodule.
port	IN	Int	Port number at which the IO-Link device is operated
suspendResume	IN	Bool	Port function; FALSE: Suspend port operation, TRUE: Resume port operations
timeout	IN	Time	Time after which a command is canceled Default value: 20s
pollingPeriod (optional)	IN	Time	Variably adjustable time that the block waits until the dataset is transmitted. Default value: 100 ms
done	OUT	Bool	TRUE: Command successfully executed
busy	OUT	Bool	TRUE: Command is currently being processed
error	OUT	Bool	TRUE: An error has occurred while processing the FB
status	OUT	Word	16#0000–16#7FFF: Status of the FB 16#8000–16#FFFF: Error codes (see Section 3.1.4).
diagnostics	OUT	"LIOLink_ typeDiagnostics"	Detailed diagnostic information of the FB (see Section 3.1.4)

NOTE

The "suspendResume" input interrupts or resumes the operation of a specific port (input "port"). Please note that the process data and the PQI are still active and updated even after a port has been suspended. If a device is removed, the process data is empty or 0 after the port is suspended and PQI indicates accordingly that the IO-Link data is invalid.

It is left to the user to handle the process data accordingly in the event of an interruption/resumption. The FB LIOLink_MeasuredData is an example of how this can be done.

4.2.3. Status and Error Handling

The "status" output outputs the current status and any errors, while the "diagnostics" output provides a diagnostic structure with detailed information in the event of an error.

status

Table 4-6: Output "status" of LIOLink_Toolchanger

Status	Meaning
16#0000	Job completed, no warning and no further detailing in the "diagnostics" output
16#7000	No operation in progress (initial value)
16#7001	First call after input of a new command (rising edge on "execute")
16#7002	Subsequent call
16#7003	Suspension was performed successfully
16#7004	Resumption was performed successfully
16#8201	Unsupported port
16#8401	The IO-Link Master has reported an error code, see "diagnostics"
16#8402	Received dataset does not match operation
16#8403	Operation could not be completed in the specified time
16#8600	Internal state machine has reached an undefined state
16#8601	System function WRREC reports an error, see "diagnostics"
16#8602	System function RDREC reports an error, see "diagnostics"

diagnostics

In the event of an error, the "diagnostics" output gives detailed information about the pending error.

Table 4-7: Output "diagnostics" of LIOLink_Toolchanger

Tag	Description
status	Last status code of the interface parameter "status" of the FB.
subfunctionStatus	Status of system functions RDREC/WRREC or error code from IO-Link Master (%W1: Error code from IO-Link Master, %W0: ISDU error code). For detailed information, refer to the Online Help for the system function in question, or the device manual of the IO-Link Master/device.
stateNumber	State of the FB's state machine in which the error occurred.

5. Device-Specific Blocks

The device-specific blocks in this library each simplify communication with a specific IO-Link device respectively, thanks to a custom-tailored interface and predefined PLC data types.

In this context, the user does not have to know the necessary parameters in order to read or write a specific dataset. Instead, the user chooses it via the interface.

For every supported dataset, PLC data types are provided which facilitate simple reading/writing of the dataset.

5.1. SIMATIC RF200

Using the library blocks described here, you can drive the SIMATIC RF200 IO-Link reader (V1.0/V1.1) via a simple user interface.

The FB "LIOLink_RF200" makes it possible to read and write various datasets, while the other FBs offer functions for reading a transponder, writing a transponder and turning the reader's antenna field on and off.

5.1.1. LIOLink_RF200

Description

The FB "LIOLink_RF200" facilitates reading and writing of various datasets via a simple user interface.

Parameter

Figure 5-1: LIOLink_RF200

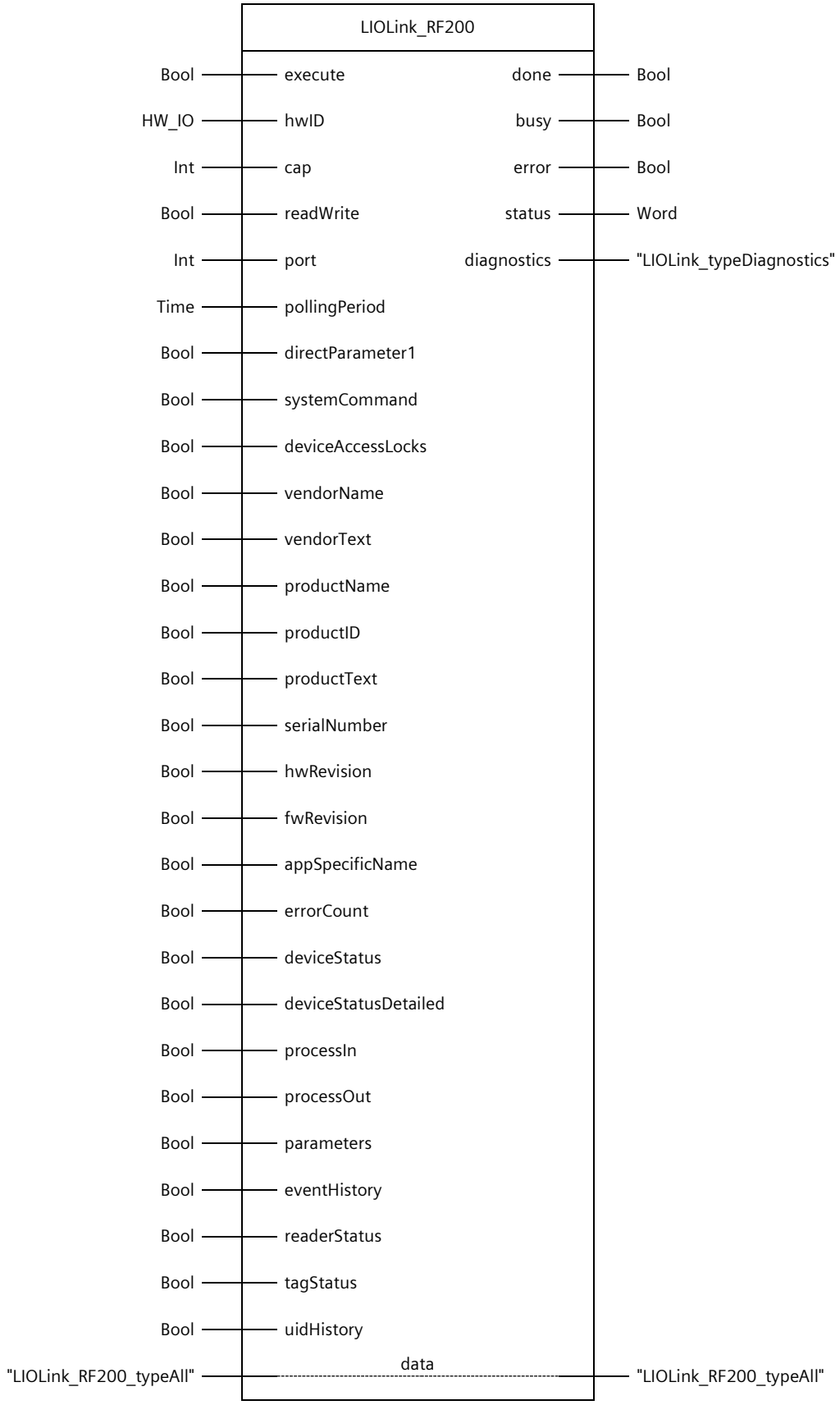


Table 5-1: Parameter von LIOLink_RF200

Name	P type	Data type	Comment
execute	IN	Bool	Request to execute the function
hwID	IN	HW_IO	Hardware identifier of the IO-Link Master or of the first submodule.
cap (optional)	IN	Int	Access point (Client Access Point): When using Siemens IO-Link Masters, is automatically detected, and is therefore optional. If the parameter is changed, automatic detection is not active.
readWrite	IN	Bool	Mode FALSE: read TRUE: write
port	IN	Int	Port number at which the IO-Link device is operated Possible values: 0..63
pollingPeriod	IN	Time	Variably adjustable time that the block waits until the dataset is transmitted. Default value: 100 ms
directParameter1	IN	Bool	Read "Direct parameter page 1"
systemCommand	IN	Bool	Write system command: Device- Reset: 0x80: Reset to factory settings: 0x82
deviceAccessLocks	IN	Bool	Read/write locking functions for device access
vendorName	IN	Bool	Read vendor name
vendorText	IN	Bool	Read vendor text
productName	IN	Bool	Read product name
productID	IN	Bool	Read product ID
productText	IN	Bool	Read product text
serialNumber	IN	Bool	Read serial number
hwRevision	IN	Bool	Read hardware version
fwRevision	IN	Bool	Read firmware version
appSpecificName	IN	Bool	Read/write user-specific marking
errorCount	IN	Bool	Read error counter
deviceStatus	IN	Bool	Read device status
deviceStatusDetailed	IN	Bool	Read detailed device status
processIn	IN	Bool	Read last process image of the inputs
processOut	IN	Bool	Read last process image of the outputs
parameters	IN	Bool	Read parameters
eventHistory	IN	Bool	Read event history
readerStatus	IN	Bool	Read reader status
tagStatus	IN	Bool	Read transponder status
uidHistory	IN	Bool	Read UID history
done	OUT	Bool	TRUE: Command successfully executed
busy	OUT	Bool	TRUE: Command is currently being processed.
error	OUT	Bool	TRUE: An error has occurred while processing the FB
status	OUT	Word	Status of LIOLink_Device (see Section 3.1.3)
diagnostics	OUT	"LIOLink_typeDiagnostics"	Diagnostic information of LIOLink_Device (see Section 3.1.3)
data	IN_OUT	"LIOLink_RF200_typeAll"	Device-specific target/source range

NOTE For each operation, only one dataset can be read or written. The FB reads/writes the dataset associated with the first parameter that is set.

Error Handling

The block outputs the status and diagnostic information from the internally-called FB "LIOLink_Device" at the outputs "status" and "diagDeviceFB" (see Section 3.1.3).

5.1.2. LIOLink_RF200_ReadTag

Description

The FB "LIOLink_RF200_Read" reads a data block from the transponder.

Parameter

Figure 5-2: LIOLink_RF200_ReadTag

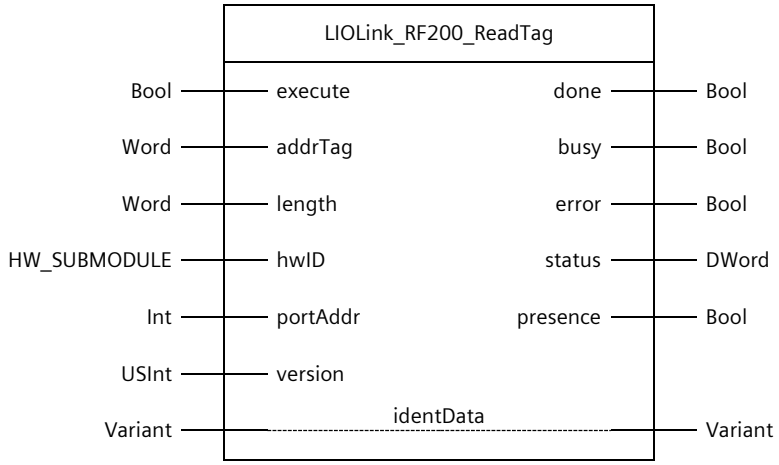
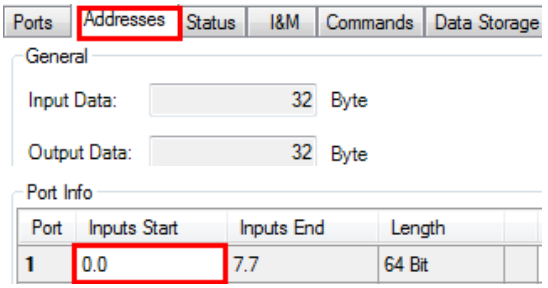


Table 5-2: Parameters of LIOLink_RF200_ReadTag

Name	P type	Data type	Comment
execute	IN	Bool	Activates the read operation at a positive edge.
addrTag	IN	Word	Start address of the data to be read on the transponder
length	IN	Word	Length of the data to be read from the transponder
hwID	IN	HW_SUBMODULE	Hardware identifier of the IO-Link Master
portAddr	IN	Int	Start address of the connected reader (PCT Tool).  Example: Start address 0.0, the value "0" must be entered for "portAddr".
version	IN	USInt	IO-Link version of the reader 11: IO-Link version 1.1 10: IO-Link version 1.0
done	OUT	Bool	TRUE: Command successfully executed
busy	OUT	Bool	TRUE: Command is currently being processed
error	OUT	Bool	TRUE: An error has occurred while processing the FB
status	OUT	DWord	Error code in the event of an error, see Section 5.1.6.
presence	OUT	Bool	Presence bit This bit is only set if a transponder is in the reader's field.
identData	IN_OUT	Variant	Data range to which the read data will be written.

5.1.3. LIOLink_RF200_SwitchAntenna

Description

The FB "LIOLink_RF200_Antenna" switches the antenna of a SIMATIC RF200 IO-Link reader on or off.

NOTE In normal operation, this command is not needed, as the antenna is always switched on after the reader is switched on.

Parameter

Figure 5-3: LIOLink_RF200_SwitchAntenna

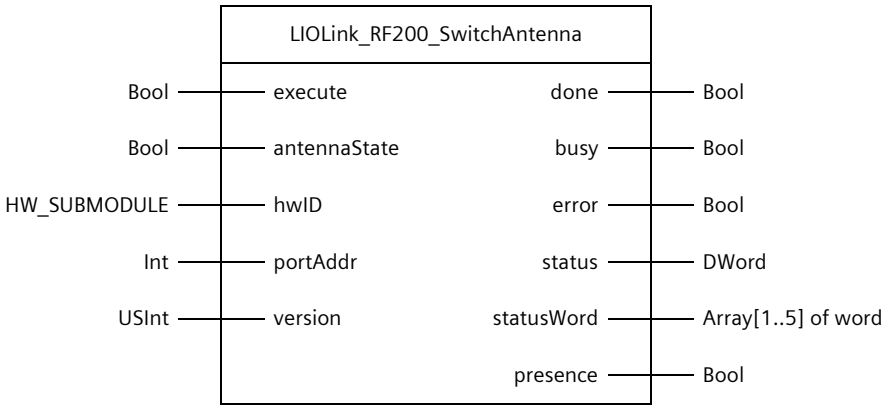
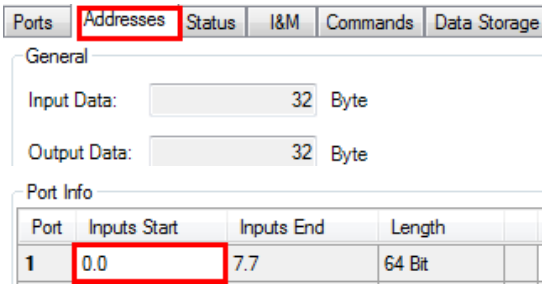


Table 5-3: Parameters of LIOLink_RF200_SwitchAntenna

Name	P type	Data type	Comment
execute	IN	Bool	Request to execute the function
antennaState	IN	Bool	Antenna state TRUE: Activate antenna FALSE: Deactivate antenna
hwID	IN	HW_SUBMODULE	Hardware identifier of the IO-Link Master
portAddr	IN	Int	Start address of the connected reader (PCT Tool).  Example: Start address 0.0, the value "0" must be entered for "portAddr".
version	IN	USInt	IO-Link version of the reader 11: IO-Link version 1.1 10: IO-Link version 1.0
done	OUT	Bool	TRUE: Command successfully executed
busy	OUT	Bool	TRUE: Command is currently being processed
error	OUT	Bool	TRUE: An error has occurred while processing the FB
status	OUT	DWord	Error code in the event of an error, see Section 5.1.6.
presence	OUT	Bool	Presence bit This bit is only set if a transponder is in the reader's field.

5.1.4. LIOLink_RF200_WriteTag

Description

The FB "LIOLink_RF200_Write" writes a data block from the user program to a transponder.

Parameter

Figure 5-4: LIOLink_RF200_WriteTag

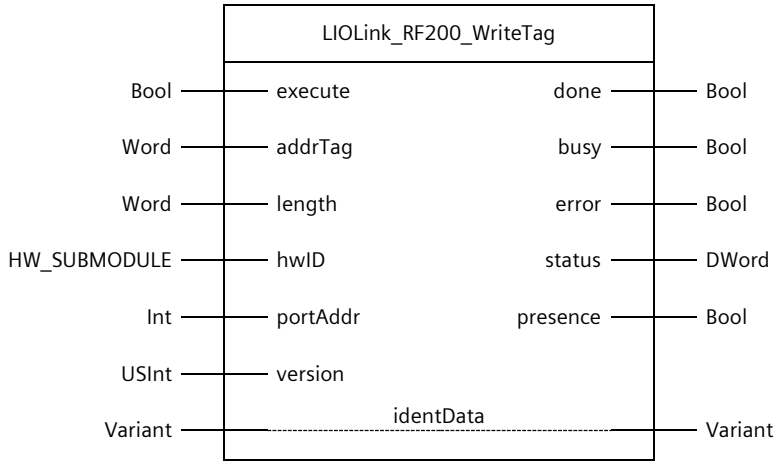
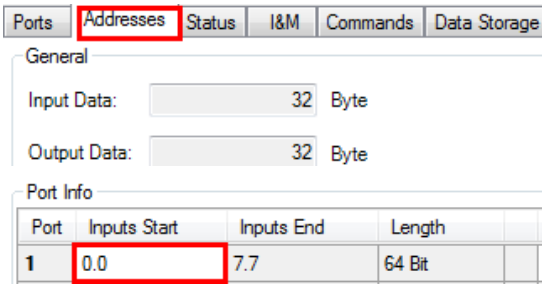


Table 5-4: Parameters of LIOLink_RF200_WriteTag

Name	P type	Data type	Comment
execute	IN	Bool	Activates the write operation if there is a positive edge.
addrTag	IN	Word	Start address of the data to be written to the transponder.
length	IN	Word	Length of the data to be written to the transponder. **Note:** The reader writes at least 4 (V1.0) or 28 bytes (V1.1) to the transponder. Therefore, the length must be selected as longer than 4 (V1.0) or 28 bytes (V1.1).
hwID	IN	HW_SUBMODULE	Hardware identifier of the IO-Link Master
portAddr	IN	Int	Start address of the connected reader (PCT Tool).  Example: Start address 0.0, the value "0" must be entered for "portAddr".
version	IN	USInt	IO-Link version of the reader 11: IO-Link version 1.1 10: IO-Link version 1.0
done	OUT	Bool	TRUE: Command successfully executed
busy	OUT	Bool	TRUE: Command is currently being processed
error	OUT	Bool	TRUE: An error has occurred while processing the FB
status	OUT	DWord	Error code in the event of an error, see Section 5.1.6.
presence	OUT	Bool	Presence bit This bit is only set if a transponder is in the reader's field.
identData	IN_OUT	Variant	Data range from which the data to be written are read

5.1.5. Integration in the User Project

Siemens Industry Online Support features a detailed application example on integrating the blocks into your user project:

https://support.industry.siemens.com/cs/ww/en/view/73565887

5.1.6. Error Handling

NOTE The following information only applies for the FBs "LIOLink_RF200_ReadTag", "LIOLink_RF200_SwitchAntenna" and "LIOLink_RF200_WriteTag".

Error messages of the FBs

Table 5-5: Error messages of the FBs

Status	Description	Remedy
16#00018101	The transponder has left the field before the read/write process finished.	Restart the process
16#00018102	The previous job is not yet complete. The operation will be terminated as soon as possible.	Restart the process
16#00018103	No transponder was recognized in the reader's field within the specified time period.	Restart the process
16#00018104	The specified length is shorter than 4 (IO-Link V1.0) or 28 (IO-Link V1.1).	Specify a length longer than 4 bytes (IO-Link V1.0) or 28 bytes (IO-Link V1.1).
16#000180xy 16#000187xy 16#000185xy 16#00018xyy	Error messages from the extended commands "BLKMOV" "DPRD_DAT" and "DPWR_DAT".	See TIA Portal information system
16#001100xx	Error messages from the connected reader.	See device manual

Error messages from the connected RF200 IO-Link reader

You can determine the error of the connected reader in the following ways:

- Directly on the reader by counting off the blink pattern of the red error LED
- Via the "status" parameter (16#001100xx)

A detailed overview of these errors can be found in the manual "SIMATIC Ident RFID systems SIMATIC RF200":
https://support.industry.siemens.com/cs/ww/en/view/109766065

5.2. SIRIUS

5.2.1. Description

Using the library blocks described here, you can drive various IO-Link-capable SIRIUS devices via a simple user interface.

This includes the following devices:

- SIRIUS 3RA compact starters
- SIRIUS 3RB24 solid-state overload relay
- SIRIUS 3RR24 monitoring relay
- SIRIUS 3RS1 temperature monitoring relay
- SIRIUS 3RS2 temperature monitoring relay
- SIRIUS ACT 3SU1 electronic module
- SIRIUS ACT 3SU1 ID key switch
- SIRIUS 3UG48 monitoring relay
- SIRIUS 8WD46 signal towers

5.2.2. Parameter

The parameters of the blocks for the different device types are almost identical. Therefore, the parameters are explained using the example of the FB "LIOLink_3SU1_ElecModule".

Figure 5-5: LIOLink_3SU1_ElecModule

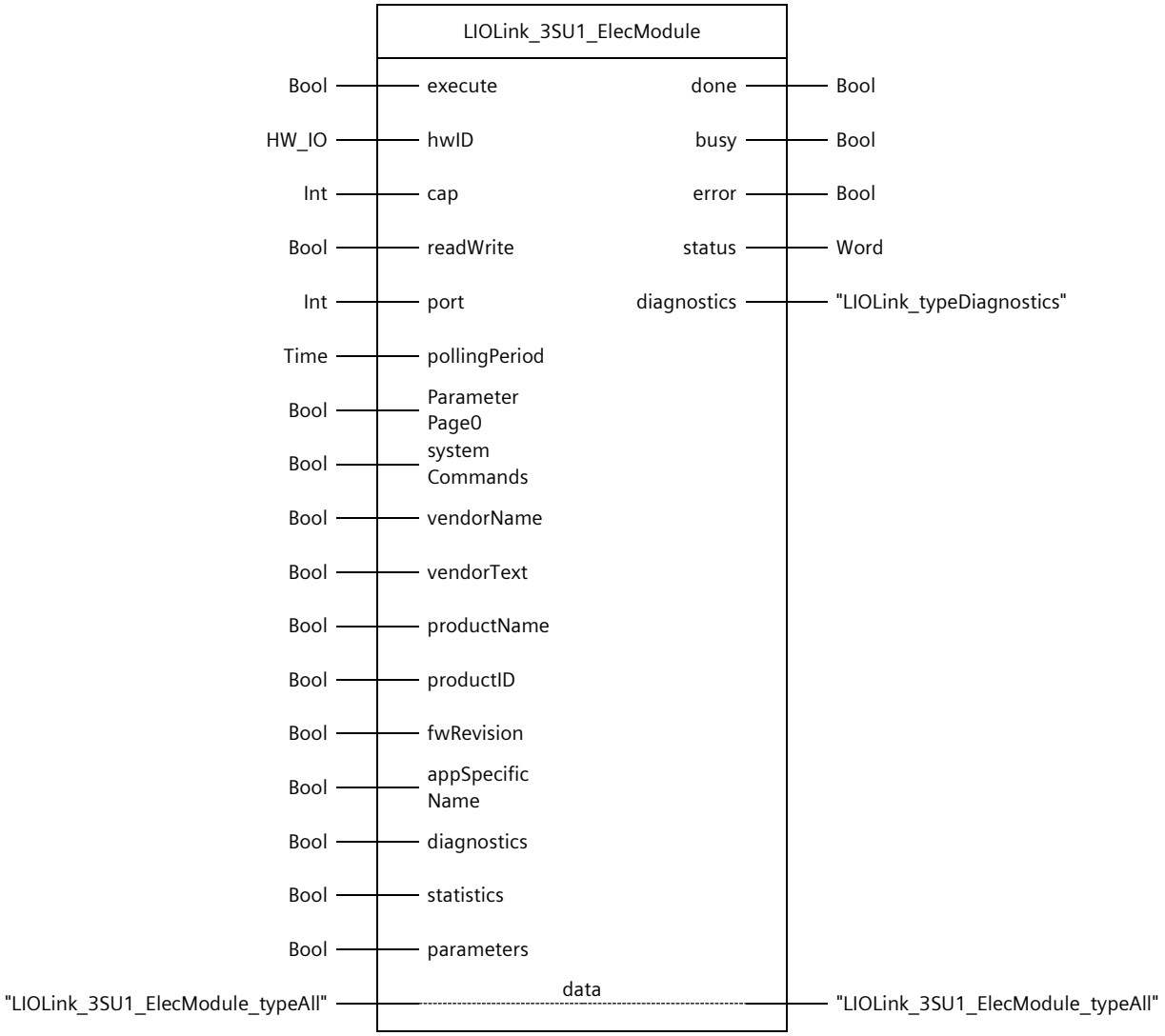


Table 5-6: Parameters of LIOLink_3SU1_ElecModule

Name	P type	Data type	Comment
execute	IN	Bool	Request to execute the function
hwID	IN	HW_IO	Hardware identifier of the IO-Link Master or of the first submodule.
cap (optional)	IN	Int	Access point (Client Access Point): When using Siemens IO-Link Masters, is automatically detected, and is therefore optional. If the parameter is changed, automatic detection is not active.
readWrite	IN	Bool	Mode FALSE: read TRUE: write
port	IN	Int	Port number at which the IO-Link device is operated Possible values: 0..63
pollingPeriod	IN	Time	Variably adjustable time that the block waits until the dataset is transmitted. Default value: 100 ms
parameterPage0	IN	Bool	Read parameter page 0
systemCommands	IN	Bool	Write system commands
vendorName	IN	Bool	Read vendor name

Name	P type	Data type	Comment
vendorText	IN	Bool	Read vendor text
productName	IN	Bool	Read product name
productID	IN	Bool	Read product ID
fwRevision	IN	Bool	Read firmware version
appSpecificName	IN	Bool	Read/write user-specific marking
diagnostics	IN	Bool	Read diagnostics (dataset 92)
statistics	IN	Bool	Read diagnostics (dataset 94)
parameters	IN	Bool	Read/write parameter (dataset 131)
done	OUT	Bool	TRUE: Command successfully executed
busy	OUT	Bool	TRUE: Command is currently being processed.
error	OUT	Bool	TRUE: An error has occurred while processing the FB
status	OUT	Word	Status of LIOLink_Device (see Section 3.1.3)
diagnostics	OUT	"LIOLink_ typeDiagnostics"	Diagnostic information of LIOLink_Device (see Section 3.1.3)
data	IN_OUT	"LIOLink_3SU1_ ElecModule_typeAll"	Device-specific target/source range

NOTE For each operation, only one dataset can be read or written. The FB reads/writes the dataset associated with the first parameter that is set.

5.2.3. Integration in the User Project

Siemens Industry Online Support features a detailed application example on integrating the blocks into your user project:

https://support.industry.siemens.com/cs/ww/en/view/90529409

5.2.4. Error Handling

The block outputs the status and diagnostic information from the internally called FB "LIOLink_Device" at the outputs "status" and "diagDeviceFB" (see Section 3.1.3).

6. Profile Blocks

The IO-Link community defines standardized device profiles for the IO-Link devices in order to unify access to IO-Link devices from a controller's user program.

The device profiles for IO-Link are based on uniform data structures, data contents and basic functionalities for the IO-Link devices. This means that a uniform interface can be created in the program for a large number of different IO-Link devices that match the same device profile, and the number of different function blocks from multiple manufacturers can be reduced to a minimum.

The prerequisite for using the blocks is that the data structure of the IO-Link device used supports the corresponding IO-Link profile.

The specification for the individual device profiles can be found on the web page of the IO-Link community: https://IO-Link.com/

The profile blocks in the library facilitate standardized access via the Common and Smart Sensor profiles.

Common profile

The Common profile provides uniform information for identification and diagnosis of the IO-Link device. It is generally valid for IO-Link devices.

Smart Sensor profile

The Smart Sensor profile is further divided into switching profiles (adjustable switching sensors) and measuring profiles (measurement data channel). This means that switching points of measured values are transmitted with these sensor profiles. The "Smart Sensor" devices are device groups that correspond to an implemented profile.

NOTE Note that the device profiles are only available for IO-Link devices that are specified for the IO-Link standard V1.1 or higher.

6.1. LIOLink_AdjSwitchingSensor

6.1.1. Description

The FB "LIOLink_AdjSwitchingSensor" offers a unified interface for access to and parameterization of IO-Link devices that support the Smart Sensor profile.

In particular, the block can be used by sensors that can be assigned to the measuring device profile type 2, i.e. IO-Link devices that support the smart sensor profile "Adjustable Switching Sensors".

Adjustable switching sensors (AdSS) within the Smart Sensor profile are devices that provide exactly one binary output signal (switching signal). The setpoint of this switching output can be defined by the application either by entering your own setpoint during configuration or by using a teach-in procedure.

In addition, various teach-in methods are possible, such as single value teach-in, two-value teach-in or dynamic teach-in, which facilitates the commissioning of the application. Depending on the sensor type, individual combinations of these teach-in methods are possible.

The switching point logic (high-active / low-active) can be defined by the application.

NOTE Note that if the FB "LIOLink_AdjSwitchingSensor" is called multiple times simultaneously for the same master (e.g. information retrieval for several ports simultaneously), only one block call will be completed successfully. A status conflict "diagnostics.status" = 16#8402 of the child FB "LIOLink_Device" will be output at the other blocks (send and response data inconsistent).

6.1.2. Parameter

Figure 6-1: LIOLink_AdjSwitchingSensor

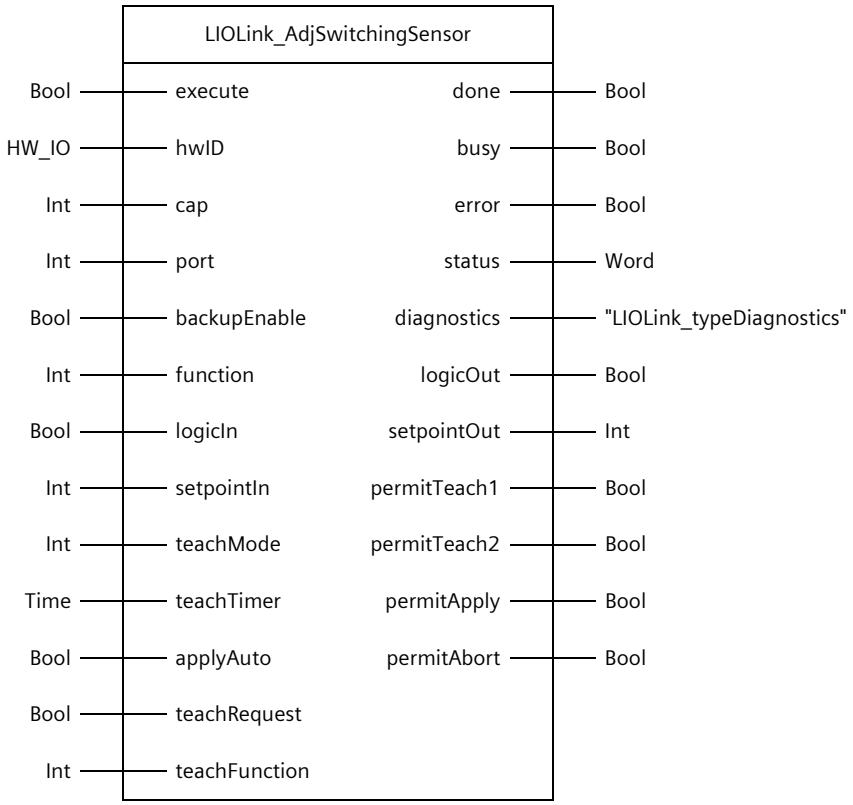


Table 6-1: Parameters of LIOLink_AdjSwitchingSensor

Name	P type	Data type	Comment
execute	IN	Bool	Request to execute the function
hwID	IN	HW_IO	Hardware identifier of the IO-Link Master or of the first submodule.

Name	P type	Data type	Comment
cap (optional)	IN	Int	Access point (Client Access Point): When using Siemens IO-Link Masters, is automatically detected, and is therefore optional. If the parameter is changed, automatic detection is not active. Normally, the access point is 0xB400 (dec. -19456) or 0x00E3 (dec. 227) and is specified in the respective manual.
port	IN	Int	Port number at which the IO-Link device is operated Possible values: 0..63
backupEnable	IN	Bool	TRUE: via the command "wr_ident" the information at the inputs "applicationSpecificTagIn", "locationTagIn" and "functionTagIn" are stored in the device FALSE: the backup mechanism is not executed by the block.
function	IN	Int	Function selection: 0: no_func 1: rd_all 2: wr_conf 3: wr_param 4: teach See Section 6.1.3.1
logicIn	IN	Bool	TRUE: The switching point currently present at the sensor is written to the sensor via the function "wr_conf". FALSE: the readout of the current switching point at the sensor is inactive
setpointIn	IN	Int	Defines the value for a new setpoint which will be written to the sensor at a request with the function "wr_param"
teachMode	IN	Int	Selecting the teach process: 0: no_teach: No teach-in action 1: single_value: Single value teach-in 2: two_value: Two-value teach-in 3: dynamic: Dynamic teach-in See Section 6.1.3.2
teachTimer	IN	Time	Defines the duration of the dynamic teach time. A value of "0" deactivates the activation of the automatic stop command. The teach function "teach_Stop" can always be used to trigger the dynamic teach stop and thus overwrites the teach timer.
applyAuto	IN	Bool	Defines the behavior during a two-value teach process. FALSE = automatic acceptance deactivated The acceptance function must be triggered by the user in order to evaluate the collected teach points and activate the new setpoint. TRUE = automatic acceptance activated If two teach points have been successfully taught in, automatic acceptance is triggered. No activity from the user is required.
teachRequest	IN	Bool	A rising edge triggers a teach step in accordance with the selected teach function at the "teachFunction" input.
teachFunction	IN	Int	Teach function which will be executed with a teach request at the input "teachRequest": 0: no teach function 1: Start teach step 1 2: Start teach step 2 3: Accept two-value teach results 4: Abort current teach sequence See Section 6.1.3.2
done	OUT	Bool	TRUE: Command successfully executed
busy	OUT	Bool	TRUE: Command is currently being processed
error	OUT	Bool	TRUE: An error has occurred while processing the FB

Name	P type	Data type	Comment
status	OUT	Word	16#0000–16#7FFF: Status of the FB 16#8000–16#FFFF: Error codes (see Section 6.1.4)
diagnostics	OUT	"LIOLink_type Diagnostics"	Detailed diagnostic information of the FB (see Section 6.1.4)
logicOut	OUT	Bool	This output represents the current value of the "Logic" parameter of the sensor. The tag is updated with the function "rd_all" each time a teach process, a write event or a request signal is completed.
setpointOut	OUT	Int	This output represents the current value of the "Setpoint" parameter of the sensor. The tag is updated with the function "rd_all" each time a teach process, a write event or a request signal is completed.
permitTeach1	OUT	Bool	The signal is set if a trigger signal for the teach function "teach_1" is possible according to the pending status of the FB.
permitTeach2	OUT	Bool	The signal is set if a trigger signal for the teach function "teach_2" is possible according to the pending status of the FB.
permitApply	OUT	Bool	The signal is set if a trigger signal for the teach function "apply" is possible according to the pending status of the FB.
permitAbort	OUT	Bool	The signal is set if a trigger signal for the teach function "abort" is possible according to the pending status of the FB.

6.1.3. Operation

6.1.3.1. Functions

The block offers five functions. A function can be selected by specifying the corresponding number at the "function" input. A rising edge at the "execute" input triggers the selected function.

- **no_func (0)**
  No function is performed.
- **rd_all (1)**
  With this function, the current switch signals and parameter values are read by the sensor. The read values are available at the outputs "logicOut" and "setpointOut".
- **wr_conf (2)**
  This function causes a previously created value for "logicIn" to be written into the sensor.
- **wr_param (3)**
  This function causes a previously created value for "setpointIn" to be written into the sensor.
- **teach (4)**
  This function causes the block to switch over to the teach process (see Section 6.1.3.2).

If the input "backupEnable" is set, the IO-Link system command "ParameterDownloadStore" is called. This starts the data storage mechanism (DataStorage) and saves the new parameterization in the IO-Link Master.

6.1.3.2. Teach-in

Overview

This section explains the various teach functions. The teach-in functions are used to teach in setpoints for switching the switching output at the sensor. Various teach-in methods are possible, for instance single value teach-in, two value teach-in or dynamic teach-in, which facilitates the commissioning of the application. Depending on the sensor type, individual combinations of these teach-in methods are possible.

Four teach-in procedures are available, and these can be selected via the "teachMode" input:

- **no_teach (0):** No teach-in action
- **single_value (1):** Single value teach-in
- **two_value (2):** Two-value teach-in
- **dynamic (3):** Dynamic teach-in

During a teach process, the individual teach steps are controlled via the inputs "teachRequest" and "teachFunction".

The different teach-in mechanisms are explained in more detail in the next section.

The "status" output provides information on the currently active teach-in step. If a "teachFunction" is requested but not permitted at this time, the function block stops the teach process and returns an error.

Single value teach-in

The steps for a single value teach-in are listed below:

1. Set inputs "function" = 4 and "teachMode" = 1.
2. Set input "execute" = TRUE.
3. Wait until outputs "permitTeach1" = TRUE and "status" = 16#7110 (teach process waits for another command).
4. Move the object to the desired position.
5. Set input "teachFunction" = 1.
6. Set input "teachRequest" = TRUE.

The teach process was successful if the outputs "done" = TRUE and "status" = 16#0000.

Two-value teach-in

The steps for a two-value teach-in are listed below:

1. Set inputs "function" = 4 and "teachMode" = 2.
2. Set input "execute" = TRUE.
3. Wait until outputs "permitTeach1" = TRUE, "permitTeach2" = TRUE and "status" = 16#7120 (teach process waits for another command).
4. Move the object to the desired position for teach point 1.
5. Set input "teachFunction" = 1.
6. Set input "teachRequest" = TRUE.
7. Wait until outputs "permitTeach1" = TRUE, "permitTeach2" = TRUE and "status" = 16#7120 (teach process waits for another command).
8. Move the object to the desired position for teach point 2.
9. Set input "teachFunction" = 2.
10. Set input "teachRequest" = FALSE.
11. Set input "teachRequest" = TRUE.

12. If "applyAuto" = FALSE, "teachApply" must be triggered manually:
    a. Wait until outputs "permitApply" = TRUE and "status" = 16#7140 (teach process waits for command to accept the taught values).
    b. Set input "teachFunction" = 3.
    c. Set input "teachRequest" = FALSE.
    d. Set input "teachRequest" = TRUE.

    If "applyAuto" = TRUE, then "teachApply" is triggered automatically.

The teach process was successful if the outputs "done" = TRUE and "status" = 16#0000.

Alternatively, "teachPoint2" can also be taught in first.

A teach point can also be taught in several times during the two-value teach-in process.

An abort of the teach function is available with "teachFunction" = 4 after teach-in of the first teach point. The teach process can be aborted by setting "teachFunction" = 4 and making a rising edge at the "teachRequest" input.

Dynamic teach-in

The steps for a dynamic teach-in are listed below:

1. Set inputs "function" = 4 and "teachMode" = 3 (dynamic teach-in).
2. Set input "execute" = TRUE.
3. Wait until outputs "permitTeach1" = TRUE and "status" = 16#7130 (teach process waits for another command).
4. Set input "teachFunction" = 1.
5. Set input "teachRequest" = TRUE.
6. Wait until output "status" = 16#7131 (teach process is active).
   The dynamic teach-in process has started.
7. Move object inside of the desired area.
8. If "teachTimer" = 0 s, teach point 2 has to be manually initiated.
   a. Wait until output "permitTeach2" = TRUE.
   b. Set input "teachFunction" = 2.
   c. Set input "teachRequest" = FALSE.
   d. Set input "teachRequest" = TRUE.

   If "teachTimer" ≠ 0 s, then teach point 2 is automatically triggered and the teach process ends when the time has elapsed. The timer starts after the teach request has been successfully sent to the IO-Link device.

The teach process was successful if the outputs "done" = TRUE and "status" = 16#0000.

An abort of the teach function is available after the start of the teach process. This can be aborted with "teachFunction" = 4 and a rising edge at the input "teachRequest".

6.1.4. Error Handling

The "status" output outputs the current status and any errors, while the "diagnostics" output provides a diagnostic structure with detailed information in the event of an error.

status

Table 6-2: Error codes of the "status" output

Output status	Explanation
Function block internal status	
16#0000	Operation successfully executed
16#7000	No operation in progress (initial value)
16#7011	Busy reading data
16#7012	Busy writing data
16#7026	Busy with backup
16#7101	Teach process started
16#7102	Teach process, subsequent call
16#7110	Teach process, single-value teach process waiting for another command
16#7111	Teach process, single-value teach process is busy
16#7120	Teach process, two-value teach process waiting for another command
16#7121	Teach process, two-value teach process is busy
16#7130	Teach process, dynamic teach process waiting for another command
16#7131	Teach process, dynamic teach process is busy
16#7140	Teach process is waiting on command to apply taught-in values
16#7141	Teach process, taught-in values are applied
16#7151	Teach process is aborted
16#8030	The value at the "function" input is outside the defined range
16#8330	An error occurred during the teach process
16#8331	Error, request for teach function not permitted in this state
16#8332	The value at the "teachMode" input is outside the defined range
16#8600	Error in child FB "LIOLink_Device", see "diagnostics"

diagnostics

In the event of an error, the "diagnostics" output gives detailed information about the pending error.

Table 6-3: "diagnostics" output

Tag	Description
status	Last status code of the interface parameter "status" of the FB.
subfunctionStatus	Status of LIOLink_Device, see Section 3.1.4.
stateNumber	State of the FB's state machine in which the error occurred.

6.2. LIOLink_IdentAndDiag

6.2.1. Description

The FB "LIOLink_IdentAndDiag" acyclically reads and writes identification and diagnostic data and outputs the status of the connected IO-Link device. The function block supports the "Common Profile" from the IO-Link specification.

This profile ID contains the function classes "DeviceIdentification", "DeviceDiagnosis", "ProcessDataVariable" and "ExtendedIdentification". They are combined to form a device profile.

NOTE Note that if the FB "LIOLink_IdentAndDiag" is called several times at the same time for the same master (e.g. information retrieval for several ports simultaneously), only one block call will be completed successfully. A status conflict "diagnostics.status" = 16#8402 of the child FB "LIOLink_Device" will be output at the other blocks (send and response data inconsistent).

6.2.2. Parameter

Figure 6-2: LIOLink_IdentAndDiag

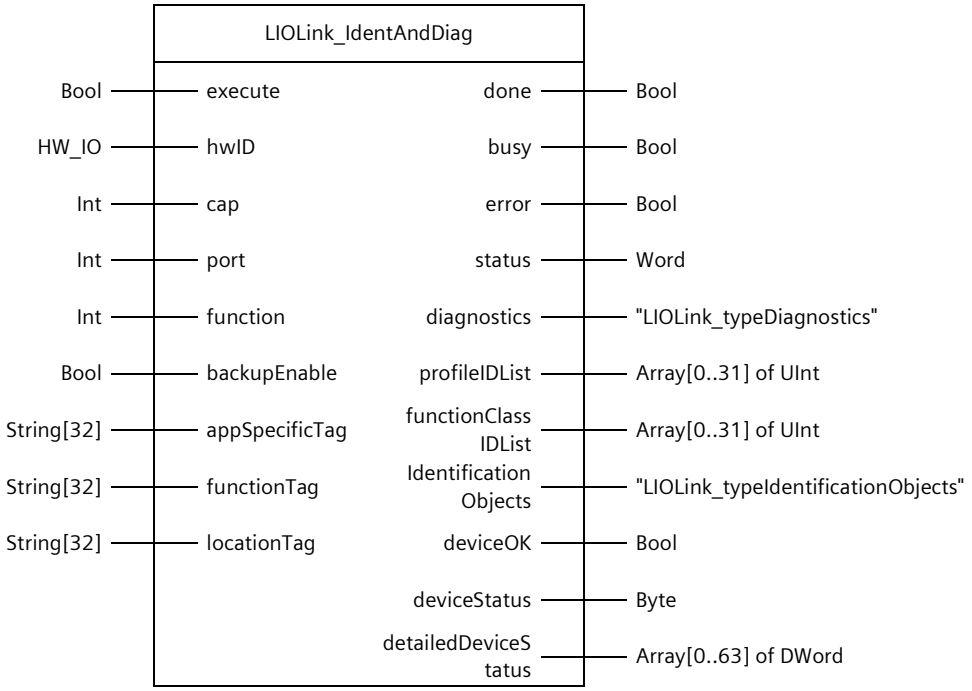


Table 6-4: Parameters of LIOLink_IdentAndDiag

Name	P type	Data type	Comment
execute	IN	Bool	Request to execute the function
hwID	IN	HW_IO	Hardware identifier of the IO-Link Master or of the first submodule.
cap (optional)	IN	Int	Access point (Client Access Point): When using Siemens IO-Link Masters, is automatically detected, and is therefore optional. If the parameter is changed, automatic detection is not active. Normally, the access point is 0xB400 (dec. -19456) or 0x00E3 (dec. 227) and is specified in the respective manual.
port	IN	Int	Port number at which the IO-Link device is operated Possible values: 0..63

Name	P type	Data type	Comment
function	IN	Int	Function selection 0: no_func No function 1: rd_all Identification and diagnostic data of the selected IO-Link device are read out. 2: rd_diag Diagnostic data of the selected IO-Link device are read out. 3: wr_ident Values at the inputs "appSpecificTag", "locationTag" and "functionTag" are written to the selected IO-Link device. You can find a detailed description of the functions in Section 6.2.3.
backupEnable	IN	Bool	TRUE: pending information in the device at the inputs "appSpecificTag", "locationTag" and "functionTag" is saved via the "wr_ident" command. FALSE: the backup mechanism is not executed by the block.
appSpecificTag	IN	String[32]	User-specific identifier which is written to the IO-Link device via wr_ident.
functionTag	IN	String[32]	Function identifier which is written to the IO-Link device via wr_ident.
locationTag	IN	String[32]	Location identifier which is written to the IO-Link device via wr_ident.
done	OUT	Bool	TRUE: Command successfully executed
busy	OUT	Bool	TRUE: Command is currently being processed
error	OUT	Bool	TRUE: An error has occurred while processing the FB
status	OUT	Word	16#0000–16#7FFF: Status of the FB 16#8000–16#FFFF: Error codes (see Section 6.2.4).
diagnostics	OUT	"LIOLink_type Diagnostics"	Detailed diagnostic information of the FB (see Section 6.2.4)
profileIDList	OUT	Array[0..31] of UInt	Outputs a list of profile IDs supported by the device
functionClassIDList	OUT	Array[0..31] of UInt	Outputs a list of function classes supported by the device
identificationObjects	OUT	"LIOLink_type IdentificationObjects"	The information about the identification objects of the sensor are stored in this PLC data type (see Table 6-5)
deviceOK	OUT	Bool	Device information TRUE: no additional diagnostic information is available. FALSE: additional information is output at the "deviceStatus" and "detailedDeviceStatus" outputs
deviceStatus	OUT	Byte	Outputs the current status of the device (see Section 6.2.4)
detailedDeviceStatus	OUT	Array[0..63] of DWord	Provides additional information about the IO-Link device defined by the manufacturer (see Section 6.2.4)

6.2.3. Principle of Operation

Function selection

Different functions can be controlled at the function block via the input parameter "function". The programmed function is executed with a rising edge at the "execute" input:

- **no_func (0)**
  No function is performed.
- **rd_all (1)**
  All current identification and diagnostic data of the IO-Link device are read out and displayed at the corresponding outputs. All parameters stored in Table 6-6 are read on the device. If an optional parameter cannot be read, the default value is issued.
- **rd_diag (2)**
  Read back of the current diagnostic parameter values from the device. In contrast to "function" = 1, only "deviceStatus" and "detailedDeviceStatus" are read back. If these parameters cannot be read, the default values are provided.
- **wr_ident (3)**
  The values provided at the inputs "applicationSpecificTagIn", "locationTagIn" and "functionTagIn" are written on the IO-Link device. These inputs are also written directly to their respective outputs. If "locationTagIn" or "functionTagIn" cannot be written, the default values are written to the outputs and the output "status" = 16#4000.

Backup

If the input "backupEnable" is set to TRUE, the IO-Link system command "ParameterDownloadStore" is called. This starts the data storage mechanism (DataStorage) and saves the new parameterization in the IO-Link Master.

An error is displayed if mandatory (M) parameters (see Table 6-6) cannot be read or written. In this case, the write or read operation stops in the FB and a detailed status is output at the "status" and "diagDeviceFB" outputs (see Section 6.2.4).

PLC data type "typeIdentificationObjects"

The PLC data type "typeIdentificationObjects" is connected to the output parameter "identificationObjects". The identification structure of the IO-Link device is stored in this data type.

All information on the various profile sensors can be stored in one data block.

Table 6-5: Parameters of LIOLink_typeIdentificationObjects

Name	Type	Comment
vendorID	Word	Unique manufacturer identification assigned by the IO-Link community
deviceID	DWord	Unique device ID that is assigned by the manufacturer
vendorName	String[64]	Manufacturer name
vendorText	String[64]	Additional information about the manufacturer
productName	String[64]	Product name
productID	String[64]	Product ID
productText	String[64]	Product text
serialNumber	String[16]	Serial number
hwRevision	String[64]	Hardware version
fwRevision	String[64]	Firmware version
appSpecificTag	String[32]	User-specific identifier
locationTag	String[32]	Advanced identification parameter that can be used for general device localization
functionTag	String[32]	Advanced identification parameter that can be used for general device identification

IO-Link device parameters

The following table shows which parameters can be read by an IO-Link device with the integrated common profile.

There are parameters that must be provided by the manufacturer (M) or that are optional (O).

Table 6-6: IO-Link device parameters

Parameter	Name	Type	Default value	Mandatory (M)/ optional (O)	Description
16#0000 Byte 0x07, 0x08	Vendor ID	WORD	-	M	Unique vendor identification assigned by the IO-Link community
16#0000 Byte 0x09, 0x0A, 0x0B	Device ID	DWORD	-	M	Unique device ID that is assigned by a vendor.
16#000D	Profile characteristic	Array [32] of UINT	0	M	The profiles are based on the definition of FunctionClasses. These FunctionClasses can be used as standalone properties or combined with the ProfileIdentifier, for example • DeviceProfileIDs for specific classes of devices, or • CommonApplicationProfileIDs for general use in all devices
16#0010	Vendor name	String [64]	-	M	Vendor name to which the VendorID is assigned.
16#0011	Vendor string	String [64]	"na"	O	Additional information about the vendor
16#0012	Product name	String [64]	-	M	Product name to differentiate between variants
16#0013	Product ID	String [64]	-	M	Manufacturer-specific product or type designation
16#0014	Product string	String [64]	"na"	O	Additional product information, e.g. product category
16#0015	Serial number	String [16]	"na"	M	Unique manufacturer-specific code for each individual device
16#0016	Hardware revision	String [64]	"na"	M	Manufacturer-specific coding for the hardware revision of the device
16#0017	Firmware revision	String [64]	-	M	Manufacturer-specific coding for the firmware revision of the device
16#0018	Application-specific tag	String [32]	-	M	Read/write data object for user application to identify specific device
16#0019	Location Tag	String [32]	"na"	M	Advanced identification parameter that can be used for general device localization. The content is not predefined, and every visible string can be written according to its own naming rules.
16#001A	Function tag	String[32]	"na"	M	Advanced identification parameter that can be used for general device identification. The content is not predefined, and every visible string can be written according to its own naming rules.
16#0024	Device status	BYTE	0	M	The information that is output corresponds to the values in Table 6-9: "deviceStatus.
16#0025	Detailed device status	Array[64] of DWORD	0	M	This array displays additional detailed device information defined by the manufacturer of the IO-Link device.

6.2.4. Error Handling

The "status" output outputs the current status and any errors, while the "diagnostics" output provides a diagnostic structure with detailed information in the event of an error.

status

Table 6-7: Error codes of the "status" output

Status	Meaning
16#0000	Operation completed, no warning and no further details
16#0003	Operation completed, at least one optional parameter could not be written
16#7000	No operation in progress (initial value)
16#7001	First call after input of a new command (rising edge on "execute")
16#7002	Subsequent call
16#8001	An error has occurred while executing the block. Check your input values.
16#8203	The value at the "function" input is outside the defined range
16#8400	External error during processing. Optional parameter without default value.
16#8600	Error in child FB "LIOLink_Device", see "diagnostics"
16#87xx	A parameter could not be read. xx contains the index (hex) of the parameter.
16#88xx	A parameter could not be written. xx contains the index (hex) of the parameter.

diagnostics

In the event of an error, the "diagnostics" output gives detailed information about the pending error.

Table 6-8: "diagnostics" output

Tag	Description
status	Last status code of the interface parameter "status" of the FB.
subfunctionStatus	Status of LIOLink_Device, see Section 3.1.4.
stateNumber	State of the FB's state machine in which the error occurred.

deviceStatus

Table 6-9: "deviceStatus" values

Value	Description
16#00	The device is working properly.
16#01	"Maintenance required" Although the process data is valid, the internal diagnosis indicates that the device is about to lose its ability to function correctly. e.g.: Optical lenses are getting dusty, deposits are forming, lubricant level low
16#02	"Out of specification" Although the process data are valid, the internal diagnosis indicates that the instrument is operating outside the specified measurement range or environmental conditions. e.g.: Power supply, auxiliary power, temperature, air pressure, magnetic disturbances, vibrations, acceleration, stray light, bubble formation in liquids
16#03	"Function test" Process data are temporarily invalid due to intentional manipulation of the device. e.g.: Calibrations, teach-in, position adjustments, simulation
16#04	"Error" Process data are invalid due to malfunctions in the device or its IO. The device cannot perform its intended function.
16#05..FF	Reserved

detailedDeviceStatus

The parameter provides information about currently pending events to the IO-Link device. Events of type "Error" or "Warning" and mode "Event appears" are output in the detailed device status list with EventQualifier and EventCode. When an event occurs with "Event disappears" mode, the corresponding Detailed Device Status entry is set to EventQualifier "0x00" and EventCode "0x000000". In this way, this parameter always provides the current diagnostic state of the device. The parameter is a read-only data object. A maximum of 64 array elements (event entries) can be displayed, but the number of array elements of this parameter is device-specific. The Event Codes supplied are also device-specific and are defined by the manufacturer. When the IO-Link device is switched off or reset, the contents of all array elements are set to the default settings – EventQualifier "0x00", EventCode "0x000000".

Table 6-10: Breakdown of "detailedDeviceStatus"

Entry	Event	Data type	Comment
1	Error/Warning_1	DWORD	All entries: 16#0000, there is no error or no warning 1st BYTE: EventQualifier 2nd, 3rd BYTE: EventCode 4th BYTE: includes no extra information
2	Error/Warning_2	DWORD	
[3]	Error/Warning_3	DWORD	
...	-	DWORD	
n (max.64)	Error/Warning_n	DWORD	

IO-Link device manufacturers can choose to implement a static list, i.e. a fixed array position for each event with a specific event code, or a dynamic list, i.e. each event entry is stored in the next free array position. Access to the subindex is not permitted for a dynamic list.

6.3. LIOLink_MeasuredDataChannel

6.3.1. Description

The FB "LIOLink_MeasuredDataChannel" provides a unified interface for access to IO-Link devices that support the Smart Sensor profile.

In particular, the block can be used by sensors that support the Smart Sensor profile type 3 ("Digital Measuring Sensors") and type 4 ("Digital Measuring Switching Sensors"). However, only 4.1 and 4.2 are supported for type 4 in the current version.

6.3.2. Parameter

Figure 6-3: LIOLink_MeasuredDataChannel

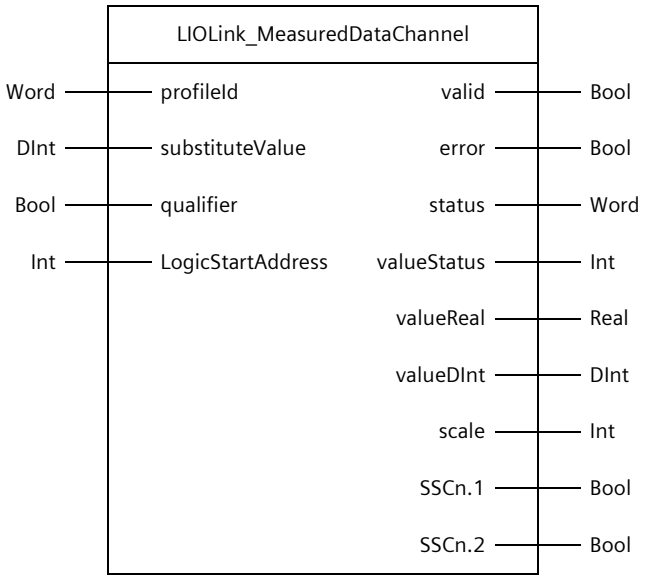


Table 6-11: Parameters of LIOLink_MeasuredDataChannel

Name	P type	Data type	Comment
profileId	IN	Word	Selected profile ID or process data structure 1 = SSP 3.1 2 = SSP 3.2 3 = SSP 3.3 4 = SSP 3.4 5 = SSP 4.1.x 6 = SSP 4.2.x (SSP = Smart Sensor Profile)
substituteValue	IN	DInt	The specified replacement value is applied to the "valueReal" and "valueDINT" if "valueStatus" is not equal to 0.
qualifier	IN	Bool	This signal corresponds to the port qualifier information of the sensor. FALSE = Process data are invalid TRUE = Process data are valid The port qualifier bit can be activated in the PCT tool. One bit is reserved for each IO-Link port.
LogicStartAddress	IN	Int	Logical address of the first byte of the process data image of the inputs.
valid	OUT	Bool	If the value is TRUE, the specified values are valid and can be used for further calculations.
error	OUT	Bool	If the value is TRUE, an internal error occurs, and further information is provided at the function block via the "status" output.
status	OUT	Word	Provides internal error codes (see Table 6-16)
valueStatus	OUT	Int	Status of process data input

Name	P type	Data type	Comment
			0 = ok 1 = Process data invalid 2 = No data 3 = Out of range (+) 4 = Out of range (-) 5 = not defined
valueReal	OUT	Real	Process data in real format for evaluation within the PLC
valueDInt	OUT	DInt	Process data in double integer format
scale	OUT	Int	Process data scaling factor (depending on the "scaleIn" input and connected sensor)
SSCn.1	OUT	Bool	Switching information channel 1
SSCn.2	OUT	Bool	Switching information channel 2

6.3.3. Principle of Operation

Using the function block, measured values of the sensors can be recorded cyclically. The measured raw values of the sensors are processed and output again as Real or DInt measured values at the respective outputs. The scaling of the sensor is already accounted for in the measured value that is output (thanks to the scaling value from the process data input structure); it does not have to be interpreted again, i.e. both the physical unit of the sensor and the measured value are interpreted.

The 'LogicStartAddress' input is the logical address of the first byte in the process data image of the inputs; it is used to read the measured values, the 'scale' outputs, and 'SSCn.1'/'SSCn.2' directly from the process data input structure. The outputs 'SSCn.1' and 'SSCn.2' are only relevant for SSP 4.1.x and 4.2.x. Otherwise, 'SSCn.1' and 'SSCn.2' are set to 0.

The 'LogicStartAddress' input can be read either directly via the configuration in the TIA Portal or with the PCT Tool.

The "profileId" input determines which type of Smart Sensor Profile (SSP) is used by the IO-Link device and whether the length of the measured value that is read from the process image is a 16-bit value or a 32-bit value.

SSP 3.1 and SSP 3.3 use the process data structure "PDI32.INT16_INT8" (16 bit measured value), and there is no difference between "profileId" = 1 and "profileId" = 3. The same applies to "profileId" = 2 (SSP 3.2) and "profileId" = 4 (SSP 3.4), since both process the data structure "PDI48.INT32_INT8" (32-bit measured value). "profileId" = 5 (SSP 4.1.x) uses a 16-bit measured value (structure MSDC48) and "profileId" = 6 (SSP 4.2.x) uses a 32-bit measured value (MSDC48).

The following Table provides an overview of profile types 3 and 4 with the corresponding process data structure.

Table 6-12: Measuring device profile types 3

Profile type	Profile ID	Name of the profile characteristic	Function class		Process data structure
			Measurement	Converter deactivation	
SSP 3.1	0x000A	Measuring sensor	0x800A	-	PDI32.INT16_INT8
SSP 3.2	0x000B	Measuring sensor, high resolution	0x800B		PDI48.INT32_INT8
SSP 3.3	0x000C	Measuring sensor, blocking function	0x800A	0x800C	PDI32.INT16_INT8 PDO8.BOOL1
SSP 3.4	0x000D	Measuring sensor, high resolution, blocking function	0x800B		PDI48.INT32_INT8 PDO8.BOOL1
SSP 4.1.1	0x0010	Measuring and switching sensor, Channel 1	0x800D 0x8010	0x800A	PDI32.MSDC32_1
SSP 4.1.2	0x0011	Measuring and switching sensor, Channel 2			PDI64.MSDC32_2
SSP 4.1.3	0x0012	Measuring and switching sensor, Channel 3			PDI96.MSDC32_3
SSP 4.1.4	0x0013	Measuring and switching sensor, Channel 4			PDI128.MSDC32_4
SSP 4.2.1	0x0014	Measuring and switching sensor, high-resolution, Channel 1		0x800B	PDI48.MSDC48_1
SSP 4.2.2	0x0015	Measuring and switching sensor, high-resolution, Channel 2			PDI96.MSDC48_2
SSP 4.2.3	0x0016	Measuring and switching sensor, high-resolution, Channel 3			PDI144.MSDC48_3
SSP 4.2.4	0x0017	Measuring and switching sensor, high-resolution, Channel 4			PDI192.MSDC48_4

If there is no error pending, the measured value that is read is forwarded directly to the "valueDInt" output. The value at the "valueReal" output is calculated from: "measured value" * 10 ^ "scale".

There are several signal states that can occur during the process:

Table 6-13: Different signal state scenarios

Case	Setpoints at the inputs	Setpoints at the outputs
1	"qualifier" = FALSE	"valid" = FALSE "valueStatus" = 1 (Process data invalid) "valueReal" = substitute value "valueDInt" = substitute value "scale" = "scaleIn"
2	"qualifier" = TRUE "profilId" = 1, 2, 3, 4, 5 or 6 lower limit < = "measurementValue" < = upper limit (see Table 6-14)	"valid" = TRUE "valueStatus" = 0 (ok) "valueReal" = "measurementValue" * 10 ^ "scaleIn" "valueDInt" = "measurementValue" "scale" = "scaleIn"
3	"qualifier" = TRUE "profilId" = 1, 2, 3, 4, 5 or 6 "measurementValue" = Out of range (+) (see Table 6-15)	"valid" = FALSE "valueStatus" = 3 (Out of range (+)) "valueReal" = substitute value "valueDInt" = substitute value "scale" = "scaleIn"
4	"qualifier" = TRUE "profilId" = 1, 2, 3, 4, 5 or 6 "measurementValue" = Out of range (-) (see Table 6-15)	"valid" = FALSE "valueStatus" = 4 (Out of range (-)) "valueReal" = substitute value "valueDInt" = substitute value "scale" = "scaleIn"
5	"qualifier" = TRUE "profilId" = 1, 2, 3, 4, 5 or 6 "measurementValue" = no measured values (see Table 6-15)	"valid" = FALSE "valueStatus" = 2 (no data) "valueReal" = substitute value "valueDInt" = substitute value "scale" = "scaleIn"
6	"qualifier" = TRUE "profilId" = 1, 2, 3, 4, 5 or 6 "measurementValue" = measured value does not correspond to any of the values of cases 2-5.	"valid" = FALSE "valueStatus" = 5 (not defined) "valueReal" = substitute value "valueDInt" = substitute value "scale" = "scaleIn"

Table 6-14: Limits of measured values

Limit	16 bits	32 bits
low limit	-32000	-2147482880
	16#8300	16#80000300
high limit	32000	2147482880
	16#7D00	16#7FFFFD00

Table 6-15: Fixed special values (substitute values)

Limit	16 bits	32 bits
Out of range (-)	-32760	-2147483640
	16#8008	16#80000008
Out of range (+)	32760	2147483640
	16#7FF8	16#7FFFFFF8
No measured value	32764	2147483644
	16#7FFC	16#7FFFFC

6.3.4. Error Handling

In the event of an error, the "error" output is set and the "status" output gives the internal error code of the FB.

Table 6-16: Error codes at the "status" output

status	Meaning	Explanation
16#0000	No error	There is no error.
16#8001	Unknown profile ID or wrong data type	Check the profile ID of the sensor or the data type used. An unknown profile ID was used or the data type does not match the profile ID. Profile ID1: WORD Profile ID2: DWORD Profile ID3: WORD Profile ID4: DWORD

The output "valueStatus" provides information about the quality of the process data. If there is no error, "valueStatus" = 0.

If there is an error, the output "valueStatus" may have the following states:

Table 6-17: Meanings of the "valueStatus" output

valueStatus	Description
1	The process data are invalid
2	No data available.
3	The process data are outside the upper limit.
4	The process data are outside the lower limit.
5	The status is not defined.

6.4. LIOLink_MultiAdjSwitchingSensor

6.4.1. Description

The FB "LIOLink_MultiAdjSwitchingSensor" offers a unified interface for access to and parameterization of IO-Link devices that support the Smart Sensor Profile.

In particular, the block can be used by sensors that can be assigned to the following measuring device profiles:

- Type 2.1 to 2.6: Adjustable switching sensors with one channel
- Type 2.7: Multi-channel adjustable switching sensors
- Type 4.x: Multi-channel adjustable switching sensors

The function block (FB) can be used to set or teach-in setpoints and to change the switching point logic of an adjustable switching sensor. This means that the block is compatible with IO-Link devices that support the smart sensor profile 'Multiple Adjustable Switching Sensors'.

The function block provides the state machines for accessing the profile-specific parameters and the processes for the three teach modes (single value teach-in, dual value teach-in and dynamic teach-in). The signals shown provide access to functions for various use cases and operating modes:

- Selecting the teach channel
- Reading switching signal channel parameters
- Writing parameters of the switching signal channel
- Single-value teaching
- Dual-value teaching
- Dynamic teach-in

NOTE Note that if the FB "LIOLink_MultiAdjSwitchingSensor" is called several times at the same time for the same master (e.g. information retrieval for several ports simultaneously), only one block call will be completed successfully. A status conflict "diagnostics.status" = 16#8402 of the child FB "LIOLink_Device" will be output at the other blocks (send and response data inconsistent).

6.4.2. Parameter

Figure 6-4: LIOLink_MultiAdjSwitchingSensor

Table 6-18: Parameters of LIOLink_MultiAdjSwitchingSensor

Name	P type	Data type	Comment
execute	IN	Bool	Request to execute the function
hwID	IN	HW_IO	Hardware identifier of the IO-Link Master (of the submodule for ET 200eco PN)
cap (optional)	IN	Int	Access point (Client Access Point): When using Siemens IO-Link Masters, is automatically detected, and is therefore optional. If the parameter is changed, automatic detection is not active. Normally, the access point is 0xB400 (dec. -19456) or 0x00E3 (dec. 227), and is specified in the respective manual.
port	IN	Int	Port number at which the IO-Link device is operated Possible values: 0..63
backupEnable	IN	Bool	TRUE: via the command "wr_ident" the information at the inputs "applicationSpecificTagIn", "locationTagIn" and "functionTagIn" are stored in the device FALSE: the backup mechanism is not executed by the block.
function	IN	Int	Function selection: 0: no_func 1: rd_all 2: wr_conf 3: wr_param 4: teach See Section 6.4.3.3
configIn	IN	LIOLink_type ConfigMultiAdjSwSensor	Defines the values for the configuration settings that are written to the sensor when a request is made using the 'wr_conf function. See Section 6.4.3.1

Name	P type	Data type	Comment
paramIn	IN	LIOLink_type ParamMultiAdjSwSensor	Defines the values for the new setpoints that are written to the sensor when a request is made using the 'wr_param' function. See Section 6.4.3.1
teachChannel	IN	Int	Defines the type of SSP used; SSP types 2.1 to 2.6, SSC = -1; SSC1.1=1; SSC1.2=2; SSC2.1=11; SSC2.2=12; SSC3.1=21; SSC3.2=22; SSC4.1=31; SSC4.2=32 See Section 6.4.3.4
teachMode	IN	Int	Selecting the teach process: 0: no_teach: No teach-in action 1: single_value: Single value teach-in SP1 2: two_value: Dual-value teach-in SP1 3: dynamic: Dynamic teach-in SP1 For SSP types 2.7 and 4.x, the following also applies: 11: single_value: Single value teach-in SP2 12: two_value: Dual-value teach-in SP2 13: dynamic: Dynamic teach-in SP2 See Section 0
teachTimer	IN	Time	Defines the duration of the dynamic teach time. A value of "0" deactivates the activation of the automatic stop command. The teach function "teach_Stop" can always be used to trigger the dynamic teach stop and thus overwrites the teach timer.
applyAuto	IN	Bool	Defines the behavior during a two-value teach process. FALSE = automatic acceptance deactivated The acceptance function must be triggered by the user in order to evaluate the collected teach points and activate the new setpoint. TRUE = automatic acceptance activated If two teach points have been successfully taught in, automatic acceptance is triggered. No activity from the user is required.
teachRequest	IN	Bool	A rising edge triggers a teach step in accordance with the selected teach function at the "teachFunction" input.
teachFunction	IN	Int	Teach function which will be executed with a teach request at the input "teachRequest": 0: no teach function 1: Start teach step 1 2: Start teach step 2 3: Accept two-value teach results 4: Abort current teach sequence See Section 6.1.3.2
done	OUT	Bool	TRUE: Command successfully executed
busy	OUT	Bool	TRUE: Command is currently being processed
error	OUT	Bool	TRUE: An error has occurred while processing the FB
status	OUT	Word	16#0000–16#7FFF: Status of the FB 16#8000–16#FFFF: Error codes (see Section 6.1.4).
diagnostics	OUT	"LIOLink_type Diagnostics"	Detailed diagnostic information of the FB (see Section 6.1.4)

Name	P type	Data type	Comment
isRealDint	OUT	Int	Displays the data type used for configuration and setpoint parameters (Hyst, SP1, SP2); depending on the sensor, these values can have the format Real or Dint; 0: Feature class not supported, 1: Real 2 Dint
configOut	OUT	LIOLink_type ConfigMultiAdjSwSensor	This output represents the current value of the sensor's configuration parameters. The tags are updated each time a teach process, a write process or a request signal with the "rd_all" function is completed. See Section 6.4.3.1
paramOut	OUT	LIOLink_type ParamMultiAdjSwSensor	This output represents the current values of the sensor's setpoints. The tags are updated with the rd_all function each time a teach process, write, or request signal is completed. See Section 6.4.3.1
permitTeach1	OUT	Bool	The signal is set if a trigger signal for the teach function "teach_1" is possible according to the pending status of the FB.
permitTeach2	OUT	Bool	The signal is set if a trigger signal for the teach function "teach_2" is possible according to the pending status of the FB.
permitApply	OUT	Bool	The signal is set if a trigger signal for the teach function "apply" is possible according to the pending status of the FB.
permitAbort	OUT	Bool	The signal is set if a trigger signal for the teach function "abort" is possible according to the pending status of the FB.

6.4.3. Operation

6.4.3.1. Configuration Settings and Target Values

The input tag "configIn" and the output tag "configOut" have the data type struct "LIOLink_typeConfigMultiAdjSwSensor". This data type consists of all possible configuration parameters for a channel.

Table 6-19 "LIOLink_typeConfigMultiAdjSwSensor" [1]

LIOLink_typeConfigMultiAdjSwSensor		
Tag	**Data type**	**Description**
logic	Int	Switching point logic; 0: high active; 1: low active;
mode	Int	Switching point mode; 0: disabled (1), single point, 2: windows, 3: dual-point;
hystDint	DInt	Switching point hysteresis (Dint)
hystReal	Real	Switching point hysteresis (Real)

The input tag "paramIn" and the output tag "paramOut" have the data type Struct "LIOLink_typeParamConfigMultiAdjSwSensor". This data type consists of all possible setpoints for a channel.

Table 6-20

LIOLink_typeParamMultiAdjSwSensor		
Tag	**Data type**	**Description**
sp1Dint	DInt	Setpoint 1 (DInt)
sp1Real	Real	Setpoint 1 (Real)
sp2Dint	DInt	Setpoint 2 (DInt)
sp2Real	Real	Setpoint 2 (Real)

More information about the functions of each configuration parameter and setpoint can be found under the "Profiles" section of the IO Link Community web page.

[1] Hyst is not relevant if the switching point mode is equal to a dual point.

The configuration parameters "hyst" as well as the setpoints "sp1" and "sp2" can be in either Dint or Real format, depending on the supported functional class of the device.

The FB recognizes which functional class is supported by the device and reads/writes this information in the correct format. As a helpful tip, the output "isRealDint" indicates in which format the device supports the values.

It is recommended that you first run a rd_all function to read out all configurations and setpoints. This updates the isRealDint output, and then you can adjust the appropriate tags to match the supported format.

6.4.3.2. Switching Point Modes

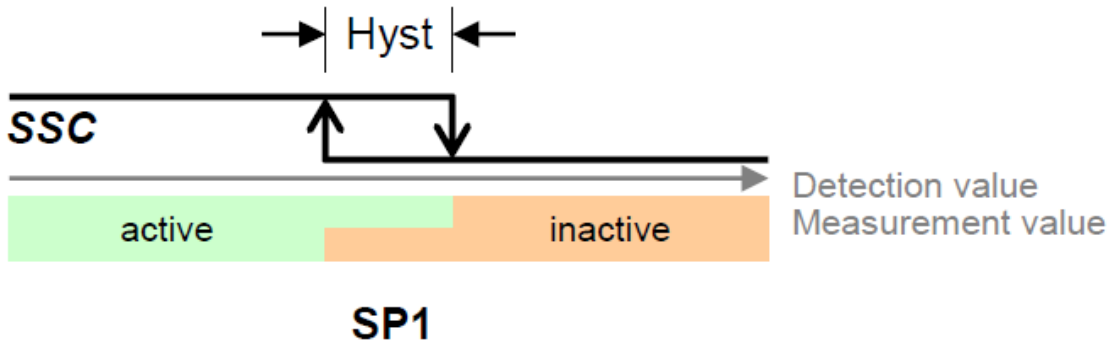
The "mode" parameter of the configuration parameter "LIOLink_typeConfigMultiAdjSwSensor" defines the behavior of the switching signal depending on setpoint parameters and the current detection or measured value. There are four different switching point modes:

- Deactivation mode:
- Single-point mode
- Window mode
- Dual-point mode

Single-point mode

In single-point mode, the switching state changes when the current value reaches the SP1 setpoint. This change occurs when values rise or fall. The SP2 setpoint is not relevant in this mode. This mode is typically used for object detection or quantity detection.

Figure 6-5 Example of a single-point mode for object detection[2]

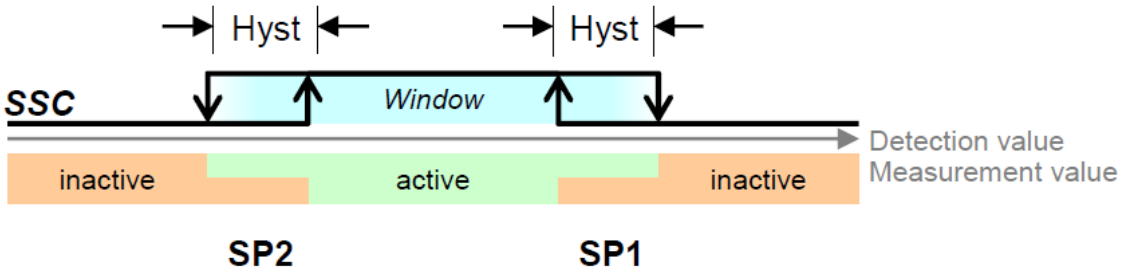


Window mode

In window mode, the switching state changes when the current measured value reaches either the setpoint SP1 or the setpoint SP2. This change occurs when values rise or fall.

[2] Source: IO-Link Smart Sensor 2nd Edition, https://io-link.com/de/Download/Download.php

Figure 6-6 Window Mode Example[3]

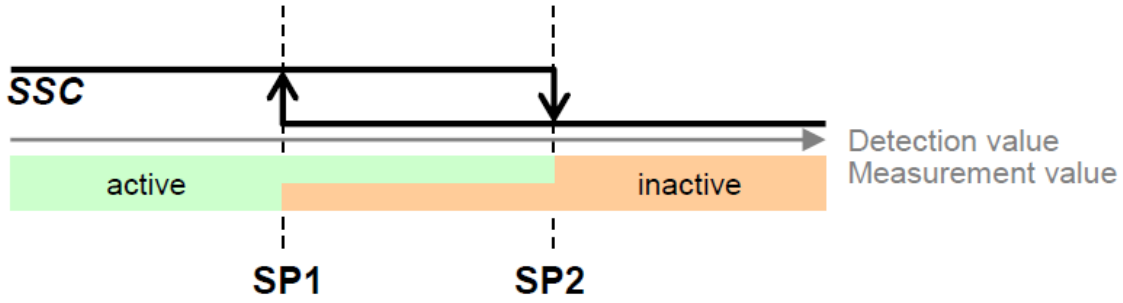


Dual-point mode (without hysteresis)

In dual-point mode, the switching state changes when the current measured value reaches the setpoint SP1. This change only occurs when measured values increase. The switching status also changes when the power value reaches the setpoint SP2. This change only takes place when measured values fall. Hysteresis is ignored in this case.

If the measured value is between SP1 and SP2 when the Smart Sensor is switched on, the behavior depends on the manufacturer/provider-specific design of the device.

Figure 6-7 Example of two-point mode for object detection[4]



Deactivation mode:

The switching state in deactivation mode must be "inactive".

6.4.3.3. Functions

The block offers five functions. A function can be selected by specifying the corresponding number at the "function" input. A rising edge at the "execute" input triggers the selected function.

- **no_func (0)**
  No function is performed.
- **rd_all (1)**
  With this function, the current switch signals and parameter values are read by the sensor. The read values are available at the "configOut" and "paramOut" outputs.
- **wr_conf (2)**
  This function causes the previously created values within the "configIn" to be written to the sensor.
- **wr_param (3)**
  This function causes the previously created values within the "paramIn" to be written to the sensor.
- **teach (4)**
  This function causes the block to switch over to the teach process (see Section Teach-In).

[3] Source: IO-Link Smart Sensor 2nd Edition, https://io-link.com/de/Download/Download.php

[4] Source: IO-Link Smart Sensor 2nd Edition, https://io-link.com/de/Download/Download.php

If the input "backupEnable" is set, the IO-Link system command "ParameterDownloadStore" is called. This starts the data storage mechanism (DataStorage) and saves the new parameterization in the IO-Link Master.

For each function, the "teachChannel" is used to first select the channel and then edit the function. The channel is selected and the function is processed in an execution (in a rising edge in the "execute" input) of the block. More information about "teachChannel" can be found here: 6.4.3.4.

6.4.3.4. Teach Channel

The block allows you to select the channel from which setpoints, configuration parameters and teach points are to be read/written to/from the sensor, depending on the selected function. A channel can be selected by entering the corresponding number at the "teachChannel" input.

The channel depends on the SSP type (Smart Sensor Profile) supported by the sensor.

- Switching Signal Channel (SSC) = -1: only applies to SSP types 2.1 to 2.6
- SSC1.1=1;
- SSC1.2=2;
- SSC2.1=11;
- SSC2.2=12;
- SSC3.1=21;
- SSC3.2=22;
- SSC4.1=31;
- SSC4.2=32

More information about SSP types and their properties can be found on the "Profiles" section of the IOLink web page.

6.4.3.5. Teach-In

Overview

This section explains the various teach functions. The teach-in functions are used to teach in setpoints for switching the switching output at the sensor. Various teach-in methods are possible, for instance single value teach-in, two value teach-in or dynamic teach-in, which facilitates the commissioning of the application. Depending on the sensor type, individual combinations of these teach-in methods are possible.

There are seven teach processes available, which can be selected via the "teachMode" input:

- **no_teach (0):** No teach-in action
- **single_value (1):** Single value teach-in
- **two_value (2):** Two-value teach-in
- **dynamic (3):** Dynamic teach-in

- For SSP types 2.7 and 4.x, the following also applies:
- **single_value (11):** Single value teach-in SP2
- **two_value (12):** Dual-value teach-in SP2
- **dynamic (13):** Dynamic teach-in SP2

During a teach process, the individual teach steps are controlled via the inputs "teachRequest" and "teachFunction".

The different teach-in mechanisms are explained in more detail in the next section.

The "status" output provides information on the currently active teach-in step. If a "teachFunction" is requested but not permitted at this time, the function block stops the teach process and returns an error.

Before a teach-in function is started, the switching point mode for the channel must first be configured. Depending on the switching point mode, some teach procedures are not relevant/allowed, e.g. the single value teach for SP2 is not relevant in single-point mode.

The teach channel must always be selected when a teach-in process is carried out. The teach-in process is performed on the selected channel.

Single value teach-in

The steps for a single value teach-in on the SSC2.1 are listed below:

1. Set inputs "teachChannel" = 11 (SSC2.1), "function" = 4 and "teachMode" = 1.
2. Set input "execute" = TRUE.
3. Wait until outputs "permitTeach1" = TRUE and "status" = 16#7110 (teach process waits for another command).
4. Move the object to the desired position.
5. Set input "teachFunction" = 1.
6. Set input "teachRequest" = TRUE.

The teach process was successful if the outputs "done" = TRUE and "status" = 16#0000.

Two-value teach-in

The steps for a two-value teach-in on the SSC1.2 are listed below:

1. Set inputs "teachChannel" = 2 (SSC1.2), "function" = 4 and "teachMode" = 2.
2. Set input "execute" = TRUE.
3. Wait until outputs "permitTeach1" = TRUE, "permitTeach2" = TRUE and "status" = 16#7120 (teach process waits for another command).
4. Move the object to the desired position for teach point 1.
5. Set input "teachFunction" = 1.
6. Set input "teachRequest" = TRUE.
7. Wait until outputs "permitTeach1" = TRUE, "permitTeach2" = TRUE and "status" = 16#7120 (teach process waits for another command).
8. Move the object to the desired position for teach point 2.
9. Set input "teachFunction" = 2.
10. Set input "teachRequest" = FALSE.
11. Set input "teachRequest" = TRUE.
12. If "applyAuto" = FALSE, "teachApply" must be triggered manually:
    a. Wait until outputs "permitApply" = TRUE and "status" = 16#7140 (teach process waits for command to accept the taught values).
    b. Set input "teachFunction" = 3.
    c. Set input "teachRequest" = FALSE.
    d. Set input "teachRequest" = TRUE.

    If "applyAuto" = TRUE, then "teachApply" is triggered automatically.

The teach process was successful if the outputs "done" = TRUE and "status" = 16#0000.

Alternatively, "teachPoint2" can also be taught in first.

A teach point can also be taught in several times during the two-value teach-in process.

An abort of the teach function is available with "teachFunction" = 4 after teach-in of the first teach point. The teach process can be aborted by setting "teachFunction" = 4 and making a rising edge at the "teachRequest" input.

Dynamic teach-in

The steps for a dynamic teach-in on the SSC2.2 are listed below:

1. Set inputs "teachChannel" = 12 (SSC2.2), "function" = 4 and "teachMode" = 3 (dynamic teach-in).
2. Set input "execute" = TRUE.
3. Wait until outputs "permitTeach1" = TRUE and "status" = 16#7130 (teach process waits for another command).
4. Set input "teachFunction" = 1.
5. Set input "teachRequest" = TRUE.
6. Wait until output "status" = 16#7131 (teach process is active).
   The dynamic teach-in process has started.
7. Move object inside of the desired area.
8. If "teachTimer" = 0 s, teach point 2 has to be manually initiated.
   a. Wait until output "permitTeach2" = TRUE.
   b. Set input "teachFunction" = 2.
   c. Set input "teachRequest" = FALSE.
   d. Set input "teachRequest" = TRUE.

   If "teachTimer" ≠ 0 s, then teach point 2 is automatically triggered and the teach process ends when the time has elapsed. The timer starts after the teach request has been successfully sent to the IO-Link device.

The teach process was successful if the outputs "done" = TRUE and "status" = 16#0000.

An abort of the teach function is available after the start of the teach process. This can be aborted with "teachFunction" = 4 and a rising edge at the input "teachRequest".

7. Master Copies

7.1. LIOLink_MeasuredData

7.1.1. Description

This FB is used to read and process the process data, regardless of which device is connected to the IO-Link Master. The FB first identifies the connected device and then reads and prepares its process data accordingly.

Supported devices for this FB can be found in a "database" in the form of a static array of the "Struct" data type. Some information is required from each device or sensor so that the FB works correctly and supports the device, e.g. Vendor ID, Device ID, Offset, Scale.

In the event of a device error or a specific Operation, such as a tool changer (see Section 4.1), the user can specify a substitute value which is then used as the process value for the specific sensor.

This FB serves as a Master template and can be customized by the user on the fundamental of their use case and their sensors and devices.

This FB can be helpful when replacing sensors, especially if different sensors from the same or different manufacturers need to be replaced. If a different sensor type is used, it must be re-parameterized, but the FB would automatically detect the new sensor type and read out its process value as long as the new sensor type is supported by the FB. This would make it possible to replace the sensor with a different sensor type without making any changes to the user program or TIA project.

Requirements

The ports of the IO-Link Master that are used must be activated in the device configuration of the IO-Link Master (Port Enable).

Due to the evaluation of the PQI, it is essential to set the process data length of the port to IO-Link 32 I/ 32 O + PQI (maximum). This means that different process data lengths from different sensors do not cause any problems when they are exchanged.

7.1.2. Parameter

Figure 7-1: LIOLink_MeasuredData

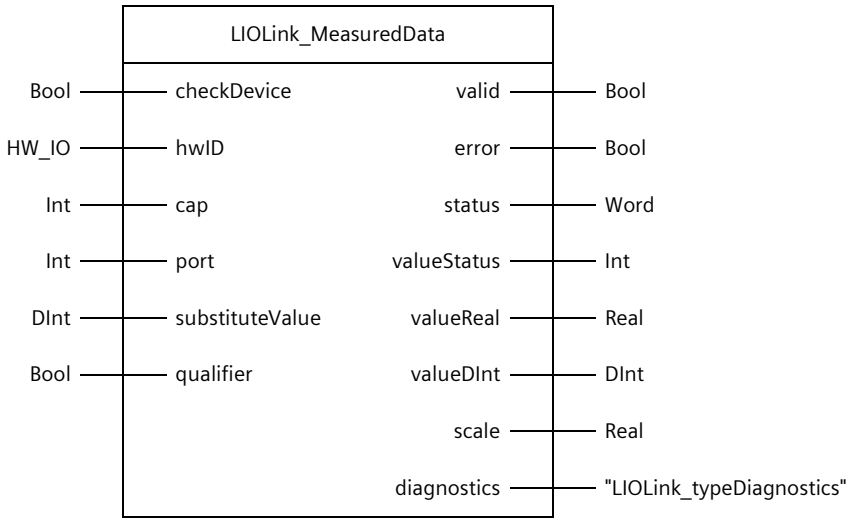


Table 7-1: Parameters of LIOLink_MeasuredData

Name	P type	Data type	Comment
checkDevice	IN	Bool	Request a check of the connected device (if a new device is connected)
hwID	IN	HW_IO	Hardware identifier of the port of the IO-Link Master
cap (optional)	IN	Int	Access point (Client Access Point): When using Siemens IO-Link Masters, is automatically detected, and is therefore optional. If the parameter is changed, automatic detection is not active. Normally, it has the value 0xB400 (dec. -19456) or 0x00E3 (dec. 227) and is specified in the respective manual.
port	IN	Int	Port number at which the IO-Link device is operated
substituteValue	IN	DInt	This value is used at the ValueReal and ValueDint outputs if ValueStatus is not equal to 0 ValueReal := DINT_TO_REAL(substituteValue)
qualifier	IN	Bool	Qualifier of the process data FALSE: ProcessData are invalid TRUE: ProcessData are invalid (see Section 7.1.3)
valid	OUT	Bool	TRUE: Command successfully executed
error	OUT	Bool	TRUE: Command is currently being processed
status	OUT	Word	16#0000–16#7FFF: Status of the FB 16#8000–16#FFFF: Error codes (see Section 3.1.4).
valueStatus	OUT	Int	Status of process data input 0 = OK; 1 = PD invalid; 2 = No data; 3 = Out of range (+) 4= Out of range (-); 5 = Not defined;
valueReal	OUT	Real	Process value in Real format for evaluation in the PLC
valueDInt	OUT	DInt	Process value in Double-Integer format for evaluation in the PLC
scale	OUT	Real	Scaling factor of the process data
diagnostics	OUT	"LIOLink_ typeDiagnostics"	Detailed diagnostic information of the FB (see Section 3.1.4)

7.1.3. Principle of Operation

Qualifier

The "qualifier" input indicates the validity of the data, with FALSE signaling that the process data is invalid, and the substitute value ("substituteValue" input) is used instead, while TRUE means that the process data is valid. For example, the "qualifier" input must be set to FALSE if port operation is to be interrupted (see Section 4.2.1) so that the substitute value is used instead of the process data. This also applies if an error occurs or if maintenance work needs to be carried out; in such cases, the "qualifier" input is set to FALSE to ensure that the invalid process data is replaced by the substitute value.

Device Detection

The FB first tries to find or recognize the device that is connected to the given port ("port" input). Device detection is triggered during the first CPU cycle or when the "checkDevice" input is triggered.

The FB uses the FB LIOLink_Device to read the device ID and vendor ID of the device. In this way, it recognizes the connected device. An attempt is then made to find this information, Device ID and Vendor ID, in the "sensor database" (static array "statSensors").

The static tag in the form of an array "statSensors" is an array with different types of sensors. Each element of this array represents a sensor type. Each sensor type has the same information that is required for its identification and for processing the process data. This information is structured in the "LIOLink_typeSensorInfo" data type.

Table 7-2 "LIOLink_typeSensorInfo"

LIOLink_typeSensorInfo		
Tag	**Data type**	**Description**
vendorID	DInt	Manufacturer ID of the sensor
deviceID	DInt	Device ID of the sensor
offset	Int	Offset for the value from the sensor
scale	Real	Scaling of the value from the sensor Default value: 1.0 (= no scaling)
fbState	DInt	FBState, in which the pre-processing of the IO data of this specific sensor is carried out. FBState must be unique for each sensor.

After the device is found in the statSensors array by comparing the readout device ID and vendor ID with the corresponding information in the array, the element that contains exactly the same information is assigned to the statCurrentSensor static tag. This tag then represents the currently connected sensor and is used to process the process data.

As the device detection process is an acyclical process and can take several cycles, the substitute value (input "substituteValue") is assigned to the process values or outputs. This scenario remains valid until the acyclical calls have been completed.

Table 7-3: Output states during a device detection procedure

Output tag	**Value**
valid	FALSE
error	FALSE
status	16#7003
valueStatus	1 (PD Invalid)
valueReal	DINT_TO_REAL(substituteValue)
valueDInt	substituteValue
scale	1.0

Reading and processing process data

The FB uses the GETIO command internally and reads the entire process image of the inputs of a port via the "hwID" input, i.e. 33 bytes (32 byte input + PQI). The entire process image is then stored in a static tag "statIOData".

After reading the process image, the PQI is evaluated to check whether the IO-Link data is valid. If the data is not valid, the FB goes into the error state and the replacement value is used as the process value.

NOTE For the "hwID" input, use the hardware identifier of the port of the IO-Link Master to which the sensor is connected. Each port of the IO-Link Master has a hardware identifier.

If the IO-Link data is valid, the FB switches to a state machine in which each state corresponds to a specific sensor type. In each of these states, the sensor data is extracted from the entire process image (previously read out with GETIO) and processed accordingly depending on the sensor type.

As different sensors can supply process data at different positions in the process image and with different accuracies, offsets and scaling, a separate status is used for each sensor type. This enables a high degree of flexibility, as the system is able to seamlessly integrate different sensor types into the same application.

The state in which the machine is operating is determined directly by the detected sensor, and the active state is stored in the static structure "statCurrentSensor".

Integration of new sensors in LIOLink_MeasuredData

Two sensors are integrated in the FB as an example to show how it should look. These two sensors are of different types and from different suppliers. Their process value is located at different positions in the process image and they have different scaling.

This shows how to integrate a new sensor into the LIOLink_MeasuredData FB:

1. Increase the constant "NUMBER_OF_SENSORS" by 1. This constant is used to determine the range of the "statSensors" array mentioned above. This adds a new sensor to the "statSensors" sensor database. This sensor is inserted at the end of the array, i.e. it is the last element of the array.
2. Add a new constant FB_SENSOR_STATE_XYZ, which is used as the new state in the state machine for the new sensor. As mentioned above, each sensor must have a unique state number, so the value of FB_SENSOR_STATE_XYZ must be unique compared to other states.

   Figure 7-2: Before/after carrying out steps 1 & 2

   Before carrying out steps 1 & 2:

49		Constant								
50		FB_SENSOR_STATE_3RS2	DInt	1						State for sensor 3RS2 (statSensor[1])
51		FB_SENSOR_STATE_TCC511	DInt	2						State for sensor TCC511 (statSensor[2])
52		NUMBER_OF_SENSORS	Int	2						Number of all possible sensors

   After carrying out steps 1 & 2:

49		Constant								
50		FB_SENSOR_STATE_3RS2	DInt	1						State for sensor 3RS2 (statSensor[1])
51		FB_SENSOR_STATE_TCC511	DInt	2						State for sensor TCC511 (statSensor[2])
52		FB_SENSOR_STATE_XYZ	DInt	3						State for sensor XYZ
53		NUMBER_OF_SENSORS	Int	3						Number of all possible sensors

3. The new sensor that was added to "statSensors" in step 1 has default values. We have to change these values with those of the new sensor. Here vendorID, deviceID, offset, scale are required and can be found on the IODDfinder website. In addition, fbState is required and corresponds to the new state that was added in step 2.
4. Add the new state created in step 2 to the state machine in the "Prepare Measured Data" region. First extract the process value from statIOData, bearing in mind that the position of the value may vary depending on the sensor. Convert the process value according to its accuracy and data type. You can use the tempWordMeasurementValue and tempIntMeasurementValue temporary tags to support data extraction and conversion.

   Once the value is processed, assign it to the statValueDInt and statValueReal static output tags. The statValid flag should be set to true and statValueStatus should be updated to show that the values are valid. However, the validity (statValid) and status (statValueStatus) of the process value should be determined on the basis of certain criteria, e.g. whether the value is within the expected range. The handling of scenarios, such as values that lie outside the expected range, is up to the user to decide how best to deal with such situations.

   Options for these outputs can be seen in the comment area (see Section 7.1.2).

7.1.4. Error Handling

The block outputs the status and diagnostic information from the internally called FB "LIOLink_Device" at the outputs "status" and "diagDeviceFB" (see Section 3.1.3).

Status

In addition to the status information read from "LIOLink_Device", the FB has the following additional status codes.

Table 7-4: Output "status" of LIOLink_MeasuredData

Status	Meaning
16#0000	Operation completed, no warning and no further details
16#7000	No operation in progress (initial value)
16#7001	First call after receipt of a new job (rising edge 'checkDevice')
16#7002	Subsequent call during active processing without additional information.
16#7003	Subordinate FB (LIOLink_Device) is busy, machine status is FB_STATE_ENABLING
16#7004	Reading the process image is active and successful
16#8600	Error due to an undefined state in the state machine
16#8601	Error due to an undefined state in the state machine; sensor state was not found in FB_STATE_ENABLING;
16#8603	IO Link Master has returned an error code, see "diagnostics"; Invalid IO-Link data
16#8604	IO Link Master has reported an error code, see "diagnostics"
16#8605	Qualifier is incorrect

8. Useful Information

8.1. IO-Link Fundamentals

Introduction

The point-to-point interface IO-Link, which can be used to transmit process data and diagnostic data as well as parameters, has been specifically developed for connecting all types of sensors and actuators to a control system. In this context, IO-Link does not use traditional bus wiring but instead keeps the parallel wiring that is typical with sensors and actuators.

The IO-Link communication standard is localized below the fieldbus level. It allows central fault diagnosis and tracking down to the sensor/actuator level and simplifies commissioning and maintenance by allowing the parameter data to be modified dynamically, directly from the application.

As an open interface, the IO-Link can be integrated into all common fieldbus and automation systems. Consistent interoperability ensures maximum protection of investment. This also applies in the context of existing machine concepts for continued use of sensors that have no IO-Link interface.

8.1.1. What is IO-Link?

Overview

"IO-Link is the first globally standardized IO technology (IEC 61131-9) for communicating with sensors as well as actuators. Its powerful point-to-point communication is based on the well-established 3-lead sensor and actuator connector with no additional requirements for the cabling material. IO-Link is therefore not a fieldbus but rather an evolutionary outgrowth of the existing, proven connection technology for sensors and actuators."[5]

Manufacturer

IO-Link is supported in a technical committee by many famous manufacturers, among them Siemens.

Connection

IO-Link communication can be connected via existing lines for digital inputs and outputs using the 3-lead technology. This makes for a homogeneous and much reduced deployment of wiring.

[5] https://io-link.com/

8.1.2. Configuring the IO-Link Master

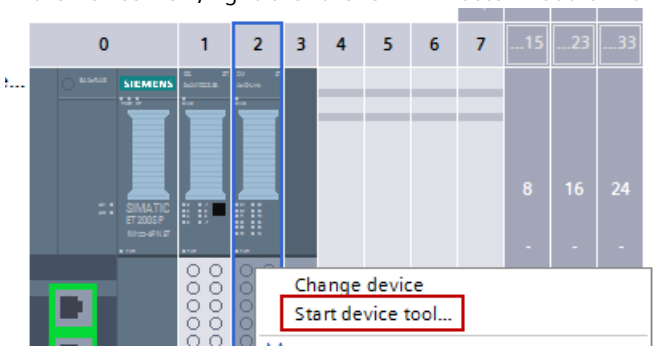
The configuration of the IO-Link Master and the devices connected to it is done in the Port Configuration Tool, or S7-PCT for short:

https://support.industry.siemens.com/cs/ww/en/view/32469496

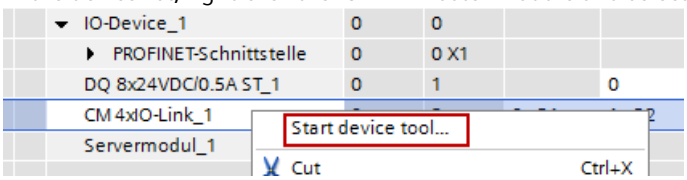
The S7-PCT can either driven standalone or launched from an existing TIA Portal project.

To open the S7-PCT from STEP 7 V16, you have the following options in the Device view of the IO-Link Master:

- In the Device view, right-click the IO-Link Master module in the editor area and select "Start device tool...".



- In the device list, right-click the IO-Link Master module and select "Start device tool...".



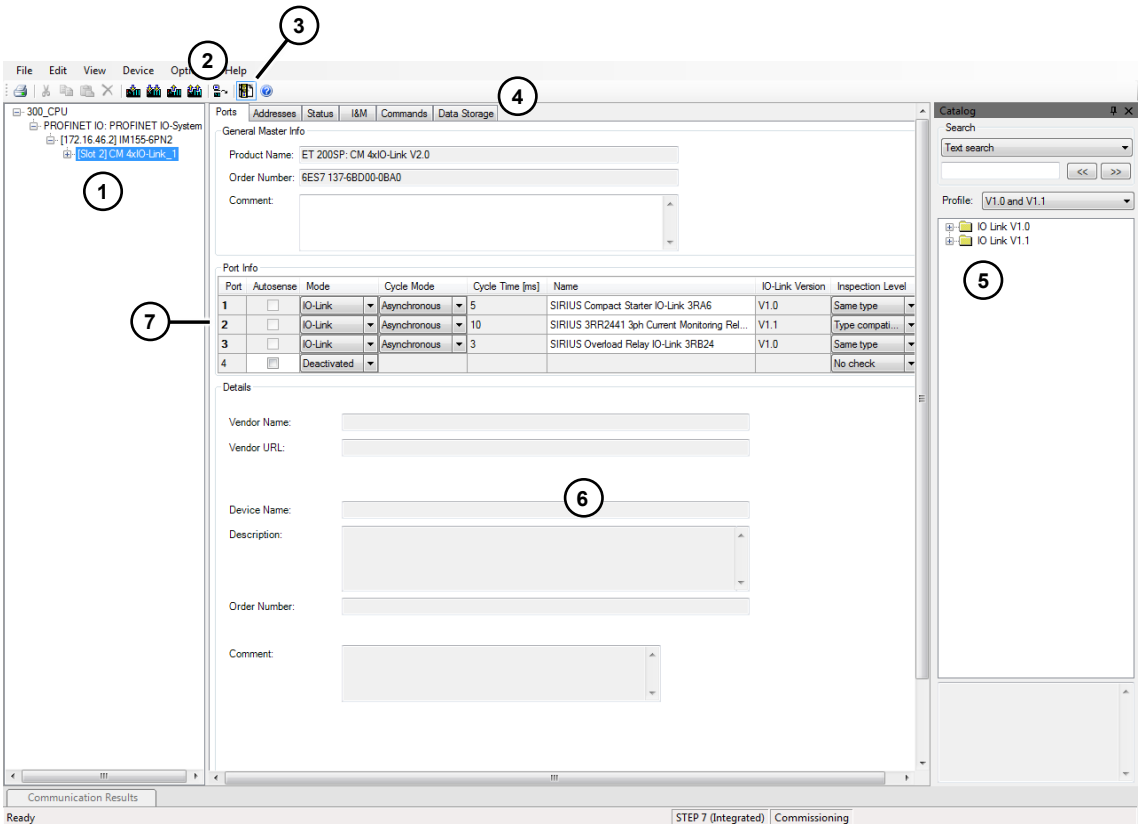
Overview and description of the S7-PCT user interface

The S7-PCT carries out the functions of:

- Assigning the respective sensors and actuators (e.g. IO-Link devices) to the ports of the IO-Link Master.
- Assigning the periphery addresses of the IO-Link devices to the S7 CPU.
- Defining the parameterization of the devices.

The Figure below shows an already configured IO-Link Master module:

Figure 8-1



Legend:

1. Project directory
2. Buttons for uploading and downloading the current IO-Link configuration
3. Button to show/hide the product catalog
4. Tabs
5. Product catalog for IO-Link devices
6. Work area
7. Port numbers with parameterized IO-Link devices

8.2. Libraries in the TIA Portal

Most of the blocks are stored as types in the library. Thus the blocks are versioned and can use the following benefits:

- Central update function for library elements
- Versioning of library elements

NOTE For information on library use in general, see the Guide to Library Use:
https://support.industry.siemens.com/cs/ww/en/view/109747503

NOTE All blocks in the library were created in accordance with the programming style guide:
https://support.industry.siemens.com/cs/ww/en/view/81318674

For more information on libraries in the TIA Portal:

- How do you open, edit and upgrade global libraries in the TIA Portal?
  https://support.industry.siemens.com/cs/ww/en/view/37364723
- TIA Portal in under 10 minutes: Time Savers – Global libraries
  https://support.industry.siemens.com/cs/ww/en/view/78529894
- Which elements of STEP 7 (TIA Portal) and WinCC (TIA Portal) can you store in a library as Type or as Master Copy?
  https://support.industry.siemens.com/cs/ww/en/view/109476862
- When starting TIA Portal V13 and higher, how do you get a global library to open automatically and use it as corporate library, for example?
  https://support.industry.siemens.com/cs/ww/en/view/100451450

9. Appendix

9.1. Service and support

SiePortal

The integrated platform for product selection, purchasing and support - and connection of Industry Mall and Online support. The SiePortal home page replaces the previous home pages of the Industry Mall and the Online Support Portal (SIOS) and combines them.

- Products & Services
  In Products & Services, you can find all our offerings as previously available in Mall Catalog.
- Support
  In Support, you can find all information helpful for resolving technical issues with our products.
- mySieportal
  mySiePortal collects all your personal data and processes, from your account to current orders, service requests and more. You can only see the full range of functions here after you have logged in.

You can access SiePortal via this address: sieportal.siemens.com

Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers – ranging from basic support to individual support contracts.
Please send queries to Technical Support via Web form: support.industry.siemens.com/cs/my/src

SITRAIN – Digital Industry Academy

We support you with our globally available training courses for industry with practical experience, innovative learning methods and a concept that's tailored to the customer's specific needs.
For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page: siemens.com/sitrain

Industry Online Support app

You will receive optimum support wherever you are with the "Industry Online Support" app. The app is available for iOS and Android:

9.2. Links and literature

Table 9-1: Links and Literature

No.	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to the article page of the application example https://support.industry.siemens.com/cs/ww/en/view/82981502
\3\	IO-Link at Siemens https://new.siemens.com/global/en/products/automation/industrial-communication/io-link.html
\4\	Web page of the IO-Link community: https://io-link.com/
\5\	Multi-manufacturer database for description files (IODDs) https://ioddfinder.IO-Link.com/
\6\	Port Configuration Tool https://support.industry.siemens.com/cs/ww/en/view/32469496
\7\	Website of the IO-Link Community – Files https://io-link.com/de/Download/Download.php
\8\	IO-Link Integration – Profile for PROFINET https://io-link.com/de/Download/Download.php

9.3. Change documentation

Table 9-2: Change Documentation

Version	Date	Change
V1.0	11/04/2013	First edition
V2.0	03/30/2015	• Complete revision and updated for STEP 7 V13 SP1 Copy protection removed • Added IO_LINK_MASTER
V2.1	08/21/2015	Changed limits for the port number with IO_LINK_DEVICE
V3.0	10/27/2016	FB "IO_LINK_MASTER" expanded to include the function "Master Backup" for the ET 200eco PN, IO-Link Master (6ES7148-6JD00-0AB0)
V3.1	03/16/2017	Library expanded with STEP 7 V14 Basic variant
V5.0	11/16/2017	Integrated new Master function for new IO-Link Master for STEP 7 V14 SP1
V5.1	01/10/2018	Addition to the library for STEP 7 V5.5 SP4 Hotfix 11
V5.2	12/21/2018	Functional expansion for S7-1500, CM 8xIO-Link with the function block IO_LINK_MASTER_8
V6.0	09/22/2020	• Completely revised blocks in accordance with programming style guide for S7-1200/1500 and PLCopen • Merged FBs for STEP 7 Basic and STEP 7 Professional • Merged FBs "IO_LINK_MASTER_4" and "IO_LINK_MASTER_8" • Integrated device-specific blocks from article 90529409 • Integrated profile blocks from article 109766016 • Added FB for SIRIUS 3RS2 devices
V7.0	04/01/2023	• Implemented automatic CAP detection and optional polling period. • Added "LIOLink_Diagnose" diagnostic block. • Added profile block "LIOLink_MultiAdjSwitchingSensor". • Overhauled profile block "LIOLink_MeasuredDataChannel", profile 4 is integrated. • Optimization and standardization of device-specific blocks. • Added device-specific block "LIOLink_8WD46".

Version	Date	Change
V7.1	01/27/2025	• Profile module "LIOLink_MultiAdjSwitchingSensor" updated and supports all SSP profiles. Previously, it only implemented the functions of the "LIOLink_AdjSwitchingSensor" profile block • Bug fix "LIOLink_Diagnosis"; it supports CAP input, and improved event mapping • New basic module "LIOLink_Toolchanger" added • New basic module added as Master Template "LIOLink_MeasuredData" • Improvement: "LIOLink_Device"; robust • Update of all "LIOLink_Device" dependent FBs with the new version V5.1
V7.2	01/30/2025	• Adjusted error status and error bit from SIMATIC RF200 blocks "LIOLink_RF200_ReadTag" and "LIOLink_RF200_WriteTag"
V7.3	08/25/2025	• The library is now only offered for S7-1200/1500. Functions for S7-300/400 have been archived • Bugfix: Some functions could not be used on an S7-1200 due to incompatible conversion. The conversion has been adjusted. • Library upgraded to TIA Portal V18 • Integration to TIA Package Manager