Use of IO-Link profile blocks

IO-Link / V1.0 / IO-Link Profile

https://support.industry.siemens.com/cs/ww/en/view/109766016

Siemens Industry Online Support

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. You yourself are responsible for the proper and safe operation of the products in accordance with applicable regulations and must also check the function of the respective application example and customize it for your system.
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. The application examples are not required to undergo the customary tests and quality inspections of a chargeable product; they may have functional and performance defects as well as errors. It is your responsibility to use them in such a manner that any malfunctions that may occur do not result in property damage or injury to persons.

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. 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://support.industry.siemens.com) shall also apply.

Security information

Siemens provides products and solutions with industrial security 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 security 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 security measures that may be implemented, please visit https://www.siemens.com/industrialsecurity.
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 Security RSS Feed at: https://www.siemens.com/industrialsecurity.

Table of contents

1 Library overview "LIoLinkProfile"

1.1 Overview

The blocks of the library "LIoLinkProfile" are described in this document.

This block library is compatible from TIA Portal SIMATIC STEP 7 Professional as of V14 SP1.

The core concern of the document is the description

- of all the blocks belonging to the library
- of the functionality implemented by these blocks

1.2 Operating principle of the IO-Link profiles

The IO-Link Community has defined standardized device profiles for IO-Link devices so that access to the IO-Link device via the controller is standardized in the user program.

One of these device profiles is the "Common Profile". It provides uniform information for identification and diagnosis of the IO-Link device. The "Common Profile" is generally valid for IO-Link devices.

In addition, there are the "Smart Sensor Profiles", which additionally differ in switching (adjustable switching sensors) and measuring (measurement data channel) profiles, i.e. switching points or measured values are transmitted with these sensor profiles. The "Smart Sensor Profile" devices are device groups that correspond to an implemented profile.

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 correspond to the same device profile, and the number of different function blocks from different 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 IO-Link homepage:

https://io-link.com/en/Download/Download.php?thisID=48

1.3 Components used

The blocks work in the current version of the library with the following IO-Link Master modules:

Table 1-1

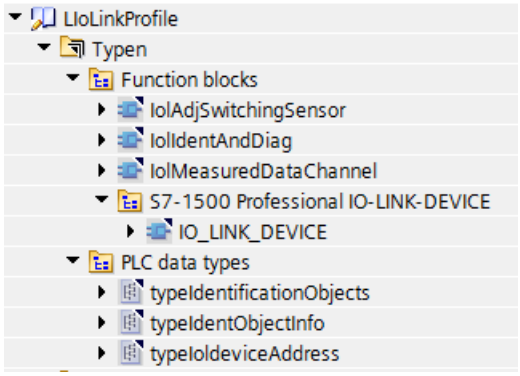
IO-Link master	Item number
ET 200SP CM 4xIO-Link	6ES7137-6BD00-0BA0
ET 200AL CM 4xIO-Link	6ES7147-5JD00-0BA0
ET 200pro EM 4 IO-LINK HF	6ES7147-4JD0-0AB0
ET 200ecoPN IO-Link master	6ES7148-6JD00-0AB0
S7-1200 SM 1278 4 IO-Link	6ES7278-4BD32-0XB0
ET 200MP CM 8xIO-Link	6ES7547-1JF00-0AB0

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.

1.4 Functional range of the library

The library "LIoLinkProfile" provides the following functions:

Figure 1-1 Functional range of the "LIoLinkProfile" library



The following IO-Link blocks are included in the library:

- IolIdentAndDiag
- IolAdjSwitchingSensor
- IolMeasuredDataChannel

You will also find the block "IO_LINK_DEVICE" in the folder structure. To use the blocks "IolIdentAndDiag" and "IolAdjSwitchingSensor" in the TIA Portal project, you have to also include the block "IO_LINK_DEVICE" in the project. The block "IO_LINK_DEVICE" is used for the internal communication to the IO-Link device.

You can find a link to the description of the block at:

https://support.industry.siemens.com/cs/de/en/view/82981502

You can also find the following PLC data types in the library:

- "typeIdentificationObjects"
- "typeIdentObjectInfo"
- "typeIoldeviceAddress"

1.5 Scope of the library

This description applies to the use of the block library "LIoLinkProfile" with the following configuration software and the corresponding SIMATIC controllers:

Table 1-2

Library	Engineering software	S7-1200	S7-1500
LIoLinkProfile	TIA Portal SIMATIC STEP 7 Professional from V14 SP1	X	X

The functionality and interfaces of the modules are discussed individually below.

2 Common profile "IoIIdentAndDiag" block

Overview

The "IoIIdentAndDiag" block reads and acyclically 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.

To use the block "IoIIdentAndDiag" in the TIA Portal project, you have to include the block "IO_LINK_DEVICE" in the project. The block "IO_LINK_DEVICE" is used for the internal communication to the IO-Link device on which the "IoIIdentAndDiag" is based.

The block "IO_LINK_DEVICE" can be found in another library at the following link:

https://support.industry.siemens.com/cs/de/en/view/82981502

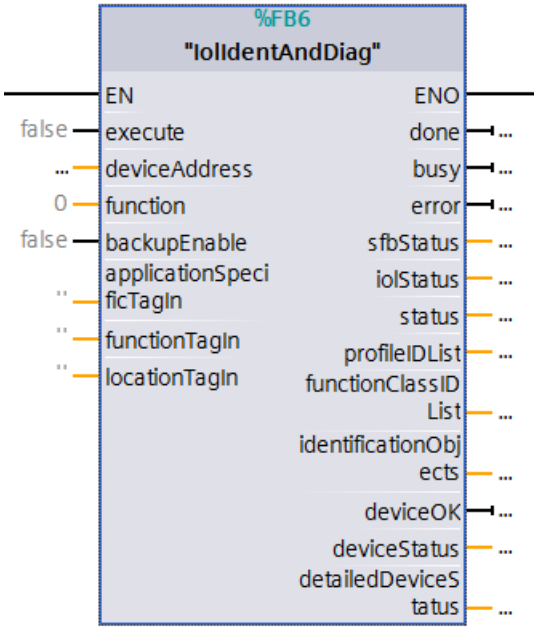
Note

Note that if the function block "IoIIdentAndDiag" is called several times at the same time for the same master (e.g. information retrieval for several ports simultaneously), only one block call is successfully terminated. A status conflict "iolStatus" = 16#7000 is returned to the other blocks (transmission and response data inconsistent).

2.1 Interface description

Block parameters

Figure 2-1 IO-Link block "IoIIdentAndDiag"



Input parameters

Table 2-1

Parameters	Data type	Description
EN	BOOL	Enable
execute	BOOL	Rising edge: Initiate data transfer
deviceAddress	typeIolDeviceAddress	PLC data type for assigning the address information of the IO-Link master to the IO-Link device/port (For an explanation see chapter 0)
function	INT	This input determines which function is to be executed: 0: no function; 1: rd_all The current identification and diagnostic data of the selected IO-Link device are read out. 2: rd_diag The current diagnostic data of the selected IO-Link device are read out. 3: wr_ident The "execute" writes the value at the input "applicationSpecificTagIn", "locationTagIn" and "functionTagIn" on the selected IO-Link device.
backupEnable	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 module.
applicationSpecificTagIn	STRING [32]	This input contains the info of the ApplicationSecificTag, which is written to the IO-Link device via wr_ident.
functionTagIn	STRING [32]	This input contains the info of the functionTag, which is written to the IO-Link device via wr_ident.
locationTagIn	STRING [32]	This input contains the info of the locationTag, which is written to the IO-Link device via wr_ident.

Output parameters

Table 2-2

Parameters	Data type	Description
ENO	BOOL	Enable
done	BOOL	TRUE: Execute job was completed FALSE: Execute job was not completed
busy	BOOL	TRUE: Job in progress FALSE: Job completed (valid or error)
error	BOOL	TRUE: Abort with errors FALSE: no error

Parameters	Data type	Description
sfbStatus	DWORD	Status output: error flag set = function error error flag reset = function status DW#16#000x0000 (x: Processing step 0 … 3)
iolStatus	DWORD	System error status: error flag set = specification of which system function the status belongs to error flag reset = DW#16#00000000
status	WORD	Outputs the status information of the function module (see Table 2-6 Error codes of the output "status" Table 2-6)
profileIdList	Array [32] of UNIT	Outputs a list of profile IDs supported by the device. (A list of the IDs can be found at **Fehler! Verweisquelle konnte nicht gefunden werden.**)
functionClassIdList	Array [32] of UNIT	Outputs a list of function classes supported by the device. (A list of the classes can be found at **Fehler! Verweisquelle konnte nicht gefunden werden.**)
identificationObjects	typeIdentObjectInfo	The information about the identification objects of the sensor are stored in this PLC data type (for an explanation, see chapter 0)
deviceOk	BOOL	Device information TRUE: no additional diagnostic information is available. FALSE: additional information is output at the "deviceStatus" and "detailedDeviceStatus" outputs
deviceStatus	BYTE	Displays the current status of the device (see 0 Explanation "Device status")
detailedDeviceStatus	Array [64] of DWORD	Provides additional information about the IO-Link device defined by the manufacturer (see 0 Explanation "detailedDeviceStatus")

Explanation "Device status"

Table 2-3 Value "deviceStatus"

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 peripherals. The device cannot perform its intended function.
16#05 … FF	Reserved

Explanation "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 displayed 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 status 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 EventCodes 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 basic settings - EventQualifier "0x00", EventCode "0x000000".

Table 2-4 Allocation "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.

2.2 How the block works

With the "IolIdentAndDiag" block, identification and diagnostic data can be read and written acyclically from an IO-Link device. The address information for the device must be specified at the input parameter "deviceAdress" so that the correct assignment to the IO-Link device can be made. The assignment can be made via the PLC data type "typeIolDeviceAdress".

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

1. rd_all ("function" = 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 5 are read on the device. If an optional parameter cannot be read, the default value is output.

2. rd_diag ("function" = 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.

3. wr_ident ("function" = 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.

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

An error is displayed if mandatory (M) parameters (see Table 5) cannot be read or written. In this case, the write or read job stops in the FB and a detailed status is output at the outputs "sfbStatus", "iolStatus" and "status".

Use of PLC data type "typeIolDeviceAddress"

With the PLC data type "typeIolDeviceAddress" you can create a simple address structure of the IO-Link master with assignment of the IO-Link devices at the connected port directly in a data block. When using several IO-Link masters and devices, you thus have an overview of how the ports on the IO-Link masters are occupied.

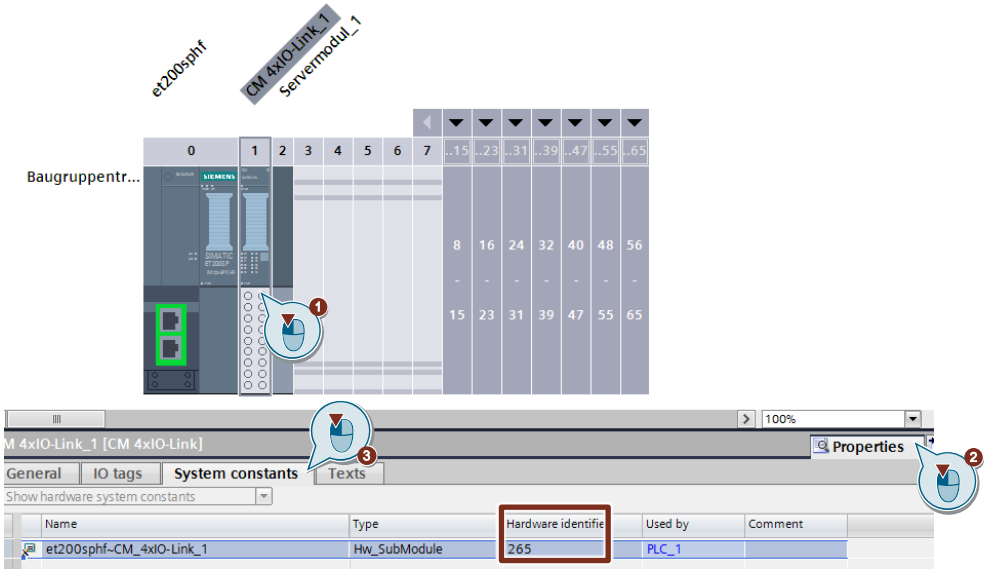
Figure 2-2 PLC data types "typeIolDeviceAddress"

AdressStructure

	Name	Data type	Start value
1	Static		
2	Device1	"typeIolDeviceAddress"	
3	id	HW_IO	265
4	cap	DInt	227
5	portNo	Int	1
6	Device2	"typeIolDeviceAddress"	
7	id	HW_IO	265
8	cap	DInt	227
9	portNo	Int	2

The parameters "id", "cap" and "portNo" must be assigned to the corresponding hardware. The hardware identification (HW-ID) of the IO-Link master used is integrated at the parameter "id". The HW-ID can be found in the properties of the IO-Link master under "System constants".

Figure 2-3 HW-ID selection



The access point to the IO-Link master is defined via the client access point (parameter "cap").

If no further information is given in the manual of the IO-Link Master, the value 227 is used (applies to all IO-Link communication modules of Siemens AG).

The assignment of the connected IO-Link device to the port used at the IO-Link master is made at the parameter "portNo".

Use of PLC data type "typeIdentObjectInfo"

The static tag "statIdentObjecInfo" is used to transfer the length of the transferred data as well as the index when calling the block "IO_LINK_DEVICE".

Figure 2-4 Data structure PLC data type "typeIdentObjectInfo"

typeIdentObjectInfo

	Name	Data type	Default value	Acc
1	index	Int	16#0000	
2	len	Int	12	
3	optional	Bool	false	

Use of 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.

Figure 2-5 Data structure PLC data type "typeIdentificationObjects"

typeIdentificationObjects

	Name	Data type	Default value
1	vendorID	Word	16#0
2	deviceID	DWord	16#0
3	vendorName	String[64]	''
4	vendorText	String[64]	''
5	productName	String[64]	''
6	productID	String[64]	''
7	productText	String[64]	''
8	serialNumber	String[16]	''
9	hardwareRevision	String[64]	''
10	firmwareRevision	String[64]	''
11	applicationSpecificTag	String[32]	''
12	locationTag	String[32]	''
13	functionTag	String[32]	''

IO-Link device parameters

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

There are parameters which must be output by the manufacturer (M) or which are optionally available (O).

Table 5 IO-Link device parameters

Parameters	Name	Type	Default value	mandatory (M) /optional (O)	Description
16#0000 bytes: 0x07, 0x08	Vendor ID:	WORD	-	M	Unique vendor identification assigned by the IO-Link community
16#0000 bytes: 0x09,	Device ID:	DWORD	-	M	Unique device ID that is assigned by a vendor.

Parameters	Name	Type	Default value	mandatory (M) /optional (O)	Description
0x0A, 0x0B					
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 (The assignment of the different profiles can be found at **Fehler! Verweisquelle konnte nicht gefunden werden.**)
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 of Table 2-3 Value "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.

2.3 Error handling

If an error occurs during the execution of the block, the output "error" is set to TRUE. There are three different status outputs that help to identify the error.

"sfbStatus" and "iolStatus" are forwarded directly from the outputs of the internally used function block "IO_LINK_DEVICE". These outputs are used to determine communication errors and errors with the IO-Link device.

The output "status" supplies the internal error codes of the function block "IolIdentAndDiag". The error codes are listed in the following table.

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

Output status	Meaning	Explanation
16#0000	no error	No error
16#1xxx	Readout error	The parameters cannot be read. xxx contains the index of the parameter
16#2xxx	Write error	The parameters cannot be written. xxx contains the index of the parameter
16#7003	Write warning	An optional parameter cannot be written.
16#8001	Incorrect operation of the function block	An error has occurred while executing the block. Check your input values.
16#82x3	Function is not defined	The value at the "function" input is outside the defined range.
16#8400	Faults when executing from outside	Optional parameters without default value

In the event of a communication error, the status is passed on from the blocks "RDREC" or "WRREC" to the output parameter "status".

In addition, the output parameter "iolStatus" indicates which S7 system function caused the error. In this case, the description of the status can be found in the online help of the corresponding S7 system function ("RDREC" or "WRREC").

If there is an error specific to IO-Link, it will be displayed on the output parameter "iolStatus" (the parameter "status" in this case has the value DW#16#00000000).

Error parameter "status"

The STATUS output parameter contains error information for the system functions used

- "RDREC" (SFB52)
- "WRREC" (SFB53)

Observe the notes on the "status" parameter and read the corresponding error information in the help for the corresponding system function.

Error parameter "iolStatus"

If the block "IO_LINK_DEVICE" runs without errors, the variable "iolStatus" is set to DW#16#00000000.

In the event of a block cancellation, error signaling is performed on the corresponding output variable "iolStatus".

The IO-Link error codes (device error codes) are passed directly into the output variable "iolStatus" of the data type "DWORD". You can find the coding in the respective IO-Link device documentation.

Error codes from the IO-Link master (IOL-M Error_Code) are also mapped into the "iolStatus" tag.

Table 2-7

"iolStatus" = DW#16#00000000			
IOL-M Error_Code		**Device error code (device-specific)**	
W#16#0000		W#16#0000	
B#16#00	B#16#00	B#16#00	B#16#00

Examples of "iolStatus"

Status information that is generated in the block "IO_LINK_DEVICE":

DW#16#80520000 = Error message from IO-Link master

DW#16#00008011 = Error message from device (sensor etc.)

Table 2-8 Coding IOL-M Error_Code within "iolStatus"

IO-Link master Error_Code	Meaning	Explanation
16#0000	No error	There is no error
16#0001	No call	Function ready for new job
16#0002	IO_LINK_CALL write	Function is in send state (SEND_REQUEST)
16#0003	IO_LINK_CALL read	Function is in poll state (WAIT_ON_RESPONSE)
16#0004 .. 06FF	-	Reserved
16 #7000	IO_LINK_CALL conflict	Send and response data inconsistent
16#7001	Wrong IO_LINK_CALL	Decoding error
16#7002	Port blocked	Port occupied by another job or not available
16#7003 .. 7FFF	-	Reserved
16#8000	Timeout	Timeout, job could not be carried out within the timeout time
16#8001	Wrong port address	Port address less than 0 or greater than 63
16#8002	Wrong index	Index less than 0 or greater than 32767
16#8003	Wrong subindex	Subindex less than 0 or greater than 255
16#8004	No device	No device connected (port but in IO-Link mode)
16#8005	Wrong LEN	Invalid length when writing, less than 1 or greater than 232
16#8006	Wrong LEN	Invalid length when reading, less than 0 or greater than 232
16#8007	DI/DO mode	Port in DI or DO mode
16#8008	No SPDU	Device in IO-Link mode does not support SPDUs
16#8009	-	An upload is not possible because the function is deactivated (data storage).
16#8010 .. 8051	-	Reserved
16#8052	RDREC fault	Error occurred when calling the "RDREC",

IO-Link master Error_Code	Meaning	Explanation
		see STATUS
16#8053	WRREC fault	Error occurred when calling the "WRREC", see STATUS
16#8054	Unexpected acknowledge	Internal error in IO-Link technology (unexpected status during an IO-Link request)
16#8055	Port function failed	Only relevant for port functions
16#8056 .. FFFF	-	Reserved

3 Block smart sensor profile "IolAdjSwitchingSensor"

Overview

This function block provides a uniform interface for accessing and parameterizing IO-Link devices that support the "Smart Sensor Profile". In particular, the device 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.

To use the block "IolAdjSwitchingSensor" in the TIA Portal project, you must also include the block "IO_LINK_DEVICE" in the project. The block "IO_LINK_DEVICE" is used for the internal communication to the IO-Link device, on which the "IolAdjSwitchingSensor" is based.

The block "IO_LINK_DEVICE" can be found in another library at the following link:

https://support.industry.siemens.com/cs/de/en/view/82981502

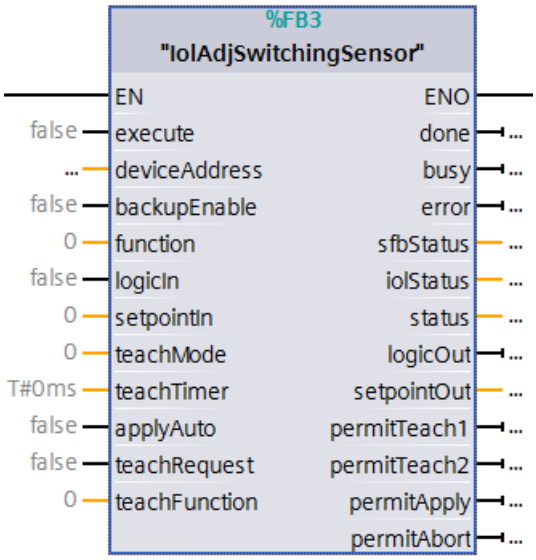
Note

Note that if the function block "IolAdjSwitchingSensor" is called several times at the same time for the same master (e.g. information retrieval for several ports simultaneously), only one block call is successfully terminated. A status conflict "iolStatus" = 16#7000 is returned to the other blocks (transmission and response data inconsistent).

3.1 Interface description

Block parameters

Figure 3-1 IO-Link block "IolAdjSwitchingSensor"



Input parameters

Table 3-1

Parameters	Data type	Description
EN	BOOL	Enable
execute	BOOL	The function is performed at the "function" input via the positive edge.
deviceAddress	typeIolDeviceAddress	PLC data type for assigning the address information of the IO-Link master to the IO-Link device/port
backupEnable	BOOL	This input is used to configure the behavior when a parameter in the device has been changed. TRUE: the backup mechanism is executed by the block, and the attached information is stored in the device FALSE: the backup mechanism is not executed by the module.
function	INT	This input is used to select the block functionality. 0 = no_func No function is performed. 1 = rd_all With this function, the current switching signals and parameter values are read by the sensor. The read values are available at the outputs "logicOut" and "setpointOut". 2 = wr_conf This function causes a previously created value for "logicIn" to be written into the sensor. 3 = wr_param This function causes a previously created value for "setpointIn" to be written into the sensor.

Parameters	Data type	Description
		4 = teach This function causes the FB to enter the teach event.
logicIn	BOOL	TRUE: The switching point currently present at the sensor is transferred to the sensor via the function "function" = 2 (wr_conf) FALSE: the readout of the current switching point at the sensor is inactive
setpointIn	INT	Defines the value for a new setpoint which is written to the sensor with the function "wr_param" when a request is made.
teachMode	INT	The possible teach operations are selected via this input: 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
teachTimer	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	BOOL	Defines the behavior during a two-value teach event. FALSE = automatic acceptance deactivated The transfer function must be triggered by the user program 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 user application program activity is required.
teachRequest	BOOL	A rising edge triggers a teach event, which is executed at the "teachFunction" input according to the selected function.
teachFunction	INT	The selected value defines the teach functionality that is to be executed with "teachRequest" = TRUE. 0 = no teach - no function selected 1 = teach 1 - start teach step 1 functionality 2 = teach 2 - start teach step 2 functionality 3 = Apply - accept two-value teach results 4 = Abort - aborting of the current teach sequence

Output parameters

Table 3-2

Parameters	Data type	Description
ENO	BOOL	Enable
done	BOOL	TRUE: Execute - job was completed FALSE: Execute - job was not completed
busy	BOOL	TRUE: Job in progress FALSE: Job completed (valid or error)
error	BOOL	TRUE: Abort with errors FALSE: no error
sfbStatus	DWORD	Status output: error flag set = function error error flag reset = function status DW#16#000x0000 (x: Processing step 0 … 3)
iolStatus	DWORD	System error status: error flag set = specification of which system function the status belongs to error flag reset = DW#16#00000000
status	WORD	Outputs the status information of the function module see Table 2-6 Error codes of the output "status"
logicOut	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 event, a write event or a request signal is completed.
setpointOut	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 event, a write event or a request signal is completed.
permitTeach1	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	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	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	BOOL	The signal is set if a trigger signal for the teach function "abort" is possible according to the pending status of the FB.

3.2 How the block works

The function block can be used to set or teach the setpoint and to change the switching point logic of adjustable switching sensors (AdSS) that support the IO-Link Smart Sensor Profile Type2. The address information must be specified at the "deviceAddress" input (for more information on "deviceAddress", see chapter 0).

The function block does not run in cyclic operation, but only on request of the "execute" input if, for example, a setpoint is to be set or taught-in.

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.

- Function: 0 = no_func

  No function is performed.

- Function: 1 = rd_all

  With this function, the current switching signals and parameter values are read by the sensor. The read values are available at the outputs "logicOut" and "setpointOut".

- Function: 2 = wr_conf

  This function causes a previously created value for "logicIn" to be written into the sensor.

- Function: 3 = wr_param

  This function causes a previously created value for "setpointIn" to be written into the sensor.

- Function: 4 = teach

  This function causes the block to switch over to the teach process. Four teach-in procedures are available on the block, and these can be selected via the "teachMode" input:

  - 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

  The teach process can be controlled via the inputs "teachTimer", "applyAuto", "teachRequest" and "teachFunction". The different teach-in mechanisms are explained in more detail in chapter 3.3 Explanation of the teach-in functions.

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

Use of PLC data type "typeIolDeviceAddress"

With the PLC data type "typeIolDeviceAddress" you can create a simple address structure of the IO-Link master with assignment of the IO-Link devices at the connected port directly in a data block. When using several IO-Link masters and devices, you thus have an overview of how the ports on the IO-Link masters are occupied.

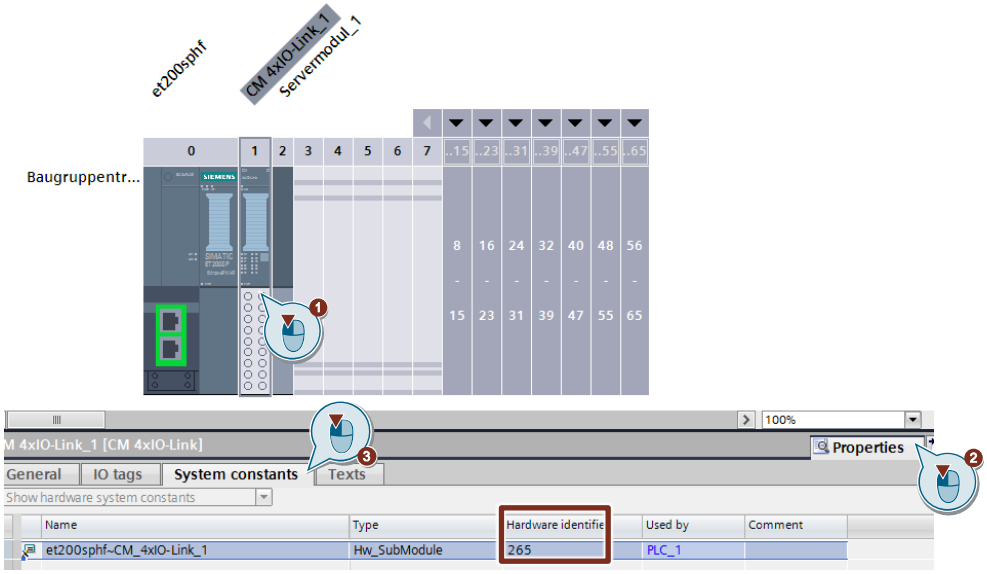
Figure 3-2 PLC data types "typeIolDeviceAddress"

AdressStructure

	Name	Data type	Start value
1	Static		
2	Device1	"typeIolDeviceAddress"	
3	id	HW_IO	265
4	cap	DInt	227
5	portNo	Int	1
6	Device2	"typeIolDeviceAddress"	
7	id	HW_IO	265
8	cap	DInt	227
9	portNo	Int	2

The parameters "id", "cap" and "portNo" must be assigned to the corresponding hardware. The hardware identification (HW-ID) of the IO-Link master used is integrated at the parameter "id". The HW-ID can be found in the properties of the IO-Link master under "System constants".

Figure 3-3 HW-ID selection



The access point to the IO-Link master is defined via the client access point (parameter "cap").

If no further information is given in the manual of the IO-Link Master, the value 227 can be used (applies to all IO-Link communication modules of Siemens AG).

The assignment of the connected IO-Link device to the port used is made at the parameter "portNo".

3.3 Explanation of the teach-in functions

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.

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

Table 3-3 Output state for a teach event

Input	Value
execute	FALSE
deviceAddress	Address information of the connected IO-Link device (For explanations see chapter 0)
backupEnable	TRUE or FALSE depending on whether the data storage mechanism is to be triggered.
function	4 (Teach)
logicIn	not relevant
setpointIn	not relevant
teachMode	0
teachRequest	FALSE
teachFunction	0

Single value teach-in

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

1. Set input "teachMode"=1
2. Set input "execute"=TRUE
3. Wait till the output "permitTeach1"=TRUE
4. Output "status" indicates 16#7021 (teach-in process successful/ready to start | Teach process is running, status single value; status see Table 3-4)
5. Set input "teachFunction"=1
6. Move the object to the desired position
7. Set input "teachRequest"=TRUE
8. Wait till the output "done"=TRUE
9. The teach process was successful if the output "status"=16#0000

Two-value teach-in

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

1. Set input "teachMode"=2
2. Set input "execute"=TRUE
3. Wait until the outputs "permitTeach1"=TRUE and "permitTeach2"=TRUE

4. Output "status" indicates 16#7022 (teach-in process successful/ready to start | Teach process is running, status two-value; status see Table 3-4)
5. Move the object to the desired position for teach point 1
6. Set input "teachFunction"=1
7. Set input "teachRequest"=TRUE
8. Wait till the output "done"=TRUE
9. Output "status" indicates 16#7122 (teach-in process is waiting for command | Teach process is running, status two-value; status see Table 3-4)
10. Move the object to the desired position for teach point 2
11. Set input "teachFunction"=2
12. Set input "teachRequest"=FALSE
13. Set input "teachRequest"=TRUE
14. If "applyAuto"=TRUE then "teachApply" is triggered automatically
15. If "applyAuto"=FALSE then "teachApply" must be triggered manually
    a. Wait until the outputs "done"=TRUE and "permitApply"=TRUE
    b. Output "status" indicates 16#7122 (teach-in process is waiting for command | Teach process is running, status two-value; status see Table 3-4)
    c. Set input "teachFunction"=3
    d. Set input "teachRequest"=FALSE
    e. Set input "teachRequest"=TRUE
16. Wait till the output "done"=TRUE
17. The teach process was successful if the output "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 performing a rising edge at the "teachRequest" input.

Dynamic teach-in

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

1. Set input "teachMode"=3 (dynamic teach-in)
2. Set input "execute"=TRUE
3. Wait till the output "permitTeach1"=TRUE
4. Output "status" indicates 16#7023 (teach-in process successful/ready to start | Teach process is running, status dynamic; status see Table 3-4)
5. Set input "teachFunction"=1
6. Set input "teachRequest"=TRUE
7. Wait till the output "done"=TRUE
8. Output "status" indicates 16#7123 (teach-in process is waiting for command | Teach process is running, status dynamic; status see Table 3-4)
9. The dynamic teach-in process has started to move the object within the desired range.

10. If "teachTimer"≠0s, then the teach point 2 is automatically triggered and the teach process ends when the time has elapsed. The timer starts immediately after the successful transmission of "Teach Starts" to the IO-Link device.
11. If "teachTimer"=0s, then teach point 2 is triggered manually.
    a. Wait until the outputs "done"=TRUE and "permitTeach2"=TRUE
    b. Set input "teachFunction"=2
    c. Set input "teachRequest"=FALSE
    d. Set input "teachRequest"=TRUE
12. Wait till the output "done"=TRUE
13. The teach process was successful if the output "status"=16#0000

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

3.4 Error handling

If an error occurs during the execution of the block, the output "error" is set to TRUE. There are three different status outputs that help to identify the error.

"sfbStatus" and "iolStatus" are forwarded directly from the outputs of the internally used function block "IO_LINK_DEVICE". These outputs are used to determine communication errors and errors with the IO-Link device. Additional information is available in the documentation on the block "IO_LINK_DEVICE":

https://support.industry.siemens.com/cs/de/en/view/82981502

The output "status" supplies the internal error codes of the function block "IolIdentAndDiag". The error codes are listed in the following table.

Table 3-4 Error codes of the output "status"

Output status	Explanation
Function block internal status	
16#XX00	Performed, successful
16#XX10	Occupied
16#XX11	Occupied reading data
16#XX12	Occupied writing data
16#XX20	Teach process is running
16#XX21	Teach process is running, status single value
16#XX22	Teach process is running, status two-value
16#XX23	Teach process is running, status dynamic
16#XX24	Teach process is running, perform action
16#XX25	Teach process is running, abort action
16#XX26	Occupied, backup
16#XX30	Performed, error (e.g. the written setpoint at input "setpointIn" is not supported or the sensor used does not support the Smart Sensor profile)
16#XX31	Error, request for teach function not permitted in this state
Additional, simultaneous teach states of the IO-Link device	

Output status	Explanation
16#00XX	Teach-in procedure successful/ready to start
16#71XX	Teach process waiting for further command
16#72XX	Teach process is running
16#83XX	Teach process error

4 Block smart sensor profile "IolMeasuredDataChannel"

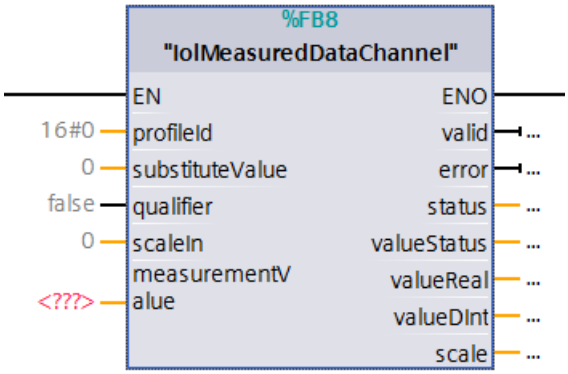
Overview

The "IolMeasuredDataChannel" function block provides a uniform interface for access by IO-Link devices that support the "Smart Sensor Profile". In particular, the device can be used by sensors that can be assigned to the measuring device profile type 3, i.e. IO-Link devices that support the smart sensor profile "Measurement Data Channel".

4.1 Interface description

Block parameters

Figure 4-1 IO-Link block "IolMeasuredDataChannel"



Input parameters

Table 4-1

Parameters	Data type	Description
EN	BOOL	Enable
profileId	WORD	Selected profile ID or process data structure 1 = SSP 3.1 2 = SSP 3.2 3 = SSP 3.3 4 = SSP 3.4 (SSP = Smart Sensor Profile)
substituteValue	DINT	The specified initial value is applied to the "valueReal" and "valueDINT" if "valueStatus" is not equal to 0.
qualifier	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.
scaleIn	SINT	The scaling information of the sensor from the process data is applied to this input. NOTE: The width of the process data input depends on the

Parameters	Data type	Description
		profile of the sensor (either INT16 or INT32).
measurementValue	Variant	The measured value information of the sensor from the process data is applied to this input. This input supports both 16 bit and 32 bit values.

Output parameters

Table 4-2

Parameters	Data type	Description
ENO	BOOL	Enable
valid	BOOL	If the value is TRUE, the specified values are valid and can be used for further calculations.
error	BOOL	If the value is TRUE, an internal error occurs and further information is provided at the function module via the output "status".
status	WORD	Provides internal error codes (see Table 4-8)
valueStatus	INT	Status of process data input 0 = ok 1 = Process data invalid 2 = No data 3 = Out of range (+) 4 = Out of range (-) 5 = not defined
valueReal	REAL	Process data in real format for evaluation within the PLC
valueDINT	DINT	Process data in double integer format
scale	INT	Process data scaling factor (depending on "scaleIn" input and connected sensor)

4.2 How the block works

With 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 via the scaling factor read in is already taken into account in the output measured value and does not have to be evaluated separately, i.e. both the physical unit of the sensor and the measured value are interpreted. The measurement data channel defines the process data structure, the functions and the representation of the measurement sensors.

The inputs "scaleIn" and "measurementValue" of the function block must be connected to the respective input addresses of the IO-Link device. The inputs can be read out either directly via the configuration in the TIA Portal or via the PCT tool. The scaling value acquired at input "scaleIn" is also available as information at output "scale".

The scaling address at input "scaleIn" is always two bytes higher than the address at input "measurementValue" for function class [0x800A] (16-bit measured values). With function class [0x800B] (32-bit measured values), the scaling address at input "scaleIn" is always four bytes higher than the measured value address.

The "profileId" input determines which type of Smart Sensor Profile (SSP) is used by the IO-Link device and whether the data is interpreted as a 16-bit or 32-bit value at "measurementValue".

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).

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

Table 4-3 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

If there is no error, the process input data are forwarded from "measurementValue" to the output "valueDINT". The value at output "valueReal" is calculated from: "measurementValue" * 10 ^ "scaleIn".

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

Table 4-4Different scenarios of signal states

Case	Setpoints at the inputs	Setpoints at the outputs
1	"qualifier" = FALSE	"valid" = FALSE "valueStatus" = 1 (Process data invalid) "valueReal" = initial value "valueDINT" = initial value "scale" = "scaleIn"
2	"qualifier" = TRUE "profilId" = 1, 2, 3 or 4 Low limit <= "measurementValue" <= high limit (see Table 4-5)	"valid" = TRUE "valueStatus" = 0 (ok) "valueReal" = "measurementValue" * 10 ^ "scaleIn" "valueDINT" = "measurementValue" "scale" = "scaleIn"
3	"qualifier" = TRUE "profilId" = 1, 2, 3 or 4 "measurementValue" = Out of range (+)	"valid" = FALSE "valueStatus" = 3 (Out of range (+)) "valueReal" = initial value

Case	Setpoints at the inputs	Setpoints at the outputs
	(see Table 4-6)	"valueDINT" = initial value "scale" = "scaleIn"
4	"qualifier" = TRUE "profilId" = 1, 2, 3 or 4 "measurementValue" = Out of range (-) (see Table 4-6)	"valid" = FALSE "valueStatus" = 4 (Out of range (-)) "valueReal" = initial value "valueDINT" = initial value "scale" = "scaleIn"
5	"qualifier" = TRUE "profilId" = 1, 2, 3 or 4 "measurementValue" = no measurement values (see Table 4-6)	"valid" = FALSE "valueStatus" = 2 (no data) "valueReal" = initial value "valueDINT" = initial value "scale" = "scaleIn"
6	"qualifier" = TRUE "profilId" = 1, 2, 3 or 4 "measurementValue" = measurement value does not correspond to any of the values of cases 2-5.	"valid" = FALSE "valueStatus" = 5 (not defined) "valueReal" = initial value "valueDINT" = initial value "scale" = "scaleIn"

Table 4-5 Limits of the measurement values

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

Table 4-6 Fixed special values (replacement values)

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

4.3 Error handling

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 4-7

Status "valueStatus"	Description
1	The process data are invalid
2	No data available.

Status "valueStatus"	Description
3	The process data are outside the upper limit.
4	The process data are outside the lower limit.
5	The status is not defined

In the event of an error, the output "error"=TRUE is set. The output "status" supplies the internal error codes of the function block. The following error codes are available:

Table 4-8 Error codes at the output "status"

Signal state at 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

5 Appendix

5.1 Service and support

Industry Online Support

Do you have any questions or need assistance?

Siemens Industry Online Support offers round the clock access to our entire service and support know-how and portfolio.

The Industry Online Support is the central address for information about our products, solutions and services.

Product information, manuals, downloads, FAQs, application examples and videos – all information is accessible with just a few mouse clicks:
https://support.industry.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:
www.siemens.com/industry/supportrequest

SITRAIN – Training for Industry

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:
www.siemens.com/sitrain

Service offer

Our range of services includes the following:

- Plant data services
- Spare parts services
- Repair services
- On-site and maintenance services
- Retrofitting and modernization services
- Service programs and contracts

You can find detailed information on our range of services in the service catalog web page:
https://support.industry.siemens.com/cs/sc

Industry Online Support app

You will receive optimum support wherever you are with the "Siemens Industry Online Support" app. The app is available for Apple iOS, Android and Windows Phone:
https://support.industry.siemens.com/cs/ww/en/sc/2067

5.2 Links and Literature

Table 5-1

No.	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to the entry page of the application example https://support.industry.siemens.com/cs/ww/en/view/Beitrags-ID
\3\	Link to the block "IO_LINK_DEVICE" https://support.industry.siemens.com/cs/de/en/view/82981502
\4\	Link to the document "Profile Identifier Overview" of the IO-Link Community https://io-link.com/share/Downloads/Profiles/IOL_ProfileIDOverview_V10_Mar2019.pdf

5.3 Change documentation

Table 5-2

Version	Date	Change
V1.0	03/2019	First version