IO-Link
Smart Sensor Profile

Specification

Version 1.0
October 2011

Order No: 10.042

File name: **IOL-Smart-Sensor-Profile-Spec_10042_V10_Oct11.doc**

This specification has been prepared by the IO-Link Smart Sensor profile group and released by the Steering Committee of the IO-Link consortium.

Any comments, proposals, requests on this document are appreciated. Please use www.io-link-projects.com for your entries and provide name and email address.
Login: *IOL-SM-Profile*
Password: *Report*

Important notes:

NOTE 1 The IO-Link Consortium Rules shall be observed prior to the development and marketing of IO-Link products. The document can be downloaded from the www.io-link.com portal.

NOTE 2 Any IO-Link device shall provide an associated IODD file. Easy access to the file and potential updates shall be possible. It is the responsibility of the IO-Link device manufacturer to test the IODD file with the help of the IODD-Checker tool available per download from www.io-link.com.

NOTE 3 Any IO-Link devices shall provide an associated manufacturer declaration on the conformity of the device with this specification, its related IODD, and test documents, available per download from www.io-link.com.

Conventions:

In this specification the following key words (in **bold** text) will be used:
may: indicates flexibility of choice with no implied preference.
should: indicates flexibility of choice with a strongly preferred implementation.
shall: indicates a mandatory requirement. Designers **shall** implement such mandatory requirements to ensure interoperability and to claim conformity with this specification.

Publisher:
IO-Link Consortium
Haid-und-Neu-Str. 7
76131 Karlsruhe
Germany
Phone: +49 721 / 96 58 590
Fax: +49 721 / 96 58 589
E-mail: info@io-link.com
Web site: www.io-link.com

CONTENTS

0 Introduction

0.1 General

The Single-drop Digital Communication Interface (SDCI) and system technology (IO-Link™[1])) for low-cost sensors and actuators is standardized within IEC 61131-9 [1]. The technology is an answer to the need of these digital/analog sensors and actuators to exchange process data, diagnosis information and parameters with a controller (PC or PLC) using a low-cost, digital communication technology while maintaining backward compatibility with the current DI/DO signals as defined in IEC 61131-2.

Any SDCI compliant Device can be attached to any available interface port of an SDCI Master. SDCI compliant devices perform physical to digital conversion in the device, and then communicate the result directly in a standard 24 V I/O digital format, thus removing the need for different DI, DO, AI, AO modules and a variety of cables.

Physical topology is point-to-point from each Device to the Master using 3 wires over distances up to 20 m. The SDCI physical interface is backward compatible with the usual 24 V I/O signalling specified in IEC 61131-2. Transmission rates of 4,8 kbit/s, 38,4 kbit/s and 230,4 kbit/s are supported.

Tools allow the association of Devices with their corresponding electronic I/O device descriptions (IODD) and their subsequent configuration to match the application requirements [2].

This document describes a common part of a sensor model that should be valid for future Device profiles and a more specific part for so-called Smart Sensors.

This document follows the IEC 62390 [3] to a certain extent.

Terms of general use are defined in IEC 61131-1 or in [4]. Specific SDCI terms are defined in this part.

0.2 Patent declaration

There are no known patents related to the content of this document.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The IO-Link Consortium shall not be held responsible for identifying any or all such patent rights.

---

1 IO-Link™ is a trade name of the "IO-Link Consortium". This information is given for the convenience of users of this specification. Compliance to this specification does not require use of the registered logos for IO-Link™. Use of the registered logos for IO-Link™ requires permission of the "IO-Link Consortium".

PROGRAMMABLE CONTROLLERS —

Profile for Smart Sensor Devices according IEC 61131-9 (Single-drop Digital Communication Interface – SDCI)

1 Scope

The single-drop digital communication interface (SDCI) technology described in part 9 of the IEC 61131 series focuses on simple sensors and actuators in factory automation, which are nowadays using small and cost-effective microcontrollers. With the help of the SDCI technology, the existing limitations of traditional signal connection technologies such as switching 0/24 V, analog 0 to 10 V, etc. can be turned into a smooth migration. Classic sensors and actuators are usually connected to a fieldbus system via input/output modules in so-called remote I/O peripherals. The (SDCI) Master function enables these peripherals to map SDCI Devices onto a fieldbus system or build up direct gateways. Thus, parameter data can be transferred from the PLC level down to the sensor/actuator level and diagnosis data transferred back in turn by means of the SDCI communication. This is a contribution to consistent parameter storage and maintenance support within a distributed automation system. SDCI is compatible to classic signal switching technology according to part 2 of the IEC 61131 series.

This document defines the common characteristics of SDCI Device profiles before it defines the model of a so-called Smart Sensor. This model comprises process data structures, identification objects, binary switching thresholds and hysteresis, best practice handling of quantity measurements with or without associated units, diagnosis objects, and teaching commonalities.

This document contains statements on conformity testing for Smart Sensor Devices and profile specific IODD features.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61131-2*, Programmable controllers – Part 2: Equipment requirements and tests*

IEC 61131-9, *Programmable controllers – Part 9: Single-drop digital communication interface for small sensors and actuators (SDCI)*

3 Terms, definitions, symbols, abbreviated terms and conventions

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions in addition to those given in IEC 61131-1, IEC 61131-2, and IEC 61131-9 apply.

3.1.1
BinaryDataChannel (BDC)
threshold information for binary signals in switching mode

3.1.2
block parameter
consistent parameter access via multiple Indices or Subindices

3.1.3
coded switching
SDCI communication, based on the standard binary signal levels of IEC 61131-2

3.1.4
communication channel
logical connection between Master and Device

NOTE Four communication channels are defined: process channel, page and ISDU channel (for parameters) and diagnosis channel.

3.1.5
Device
single passive peer to a Master such as a sensor or actuator

NOTE Uppercase "Device" is used for SDCI equipment, while lowercase "device" is used in a generic manner.

3.1.6
event
an instance of a change of conditions

NOTE An event is indicated via the event flag within the Device's status cyclic information, then acyclic transfer of event data (typically diagnosis information) is conveyed through the diagnosis communication channel.

[IEC 61158-5-x, modified]

3.1.7
FunctionClass
particular function within a Device profile

NOTE A profile Device can use one or several FunctionClasses one or several times.

3.1.8
ISDU
indexed service data unit used for acyclic acknowledged transmission of parameters that can be segmented in a number of F-sequences

3.1.9
Master
active peer connected through ports to one up to n Devices and which provides an interface to the gateway to the upper level communication systems or PLCs

NOTE Uppercase "Master" is used for SDCI equipment, while lowercase "master" is used in a generic manner.

3.1.10
on-request data (OD)
acyclically transmitted data upon request of the Master application consisting of parameters or event data

3.1.11
port
communication medium interface of the Master to one Device

3.1.12
process data (PD)
input or output values from or to a discrete or continuous automation process cyclically transferred with high priority and in a configured schedule automatically after start-up of a Master

3.1.13
ProcessDataVariable (PDV)
representation of a measurement value

3.1.14
process variable input descriptor (PVinD)
descriptor for position and offset of variables within the Process (input) Data entity

3.1.15
process variable output descriptor (PVoutD)
descriptor for position and offset of variables within the Process (output) Data entity

3.1.16
ProfileIdentifier
list of supported profiles and function classes

3.1.17
Setpoint
threshold measurement value of a sensor for the rising or falling edge of a binary output signal

3.1.18
single point mode
evaluation method with one single Setpoint where the binary output signal changes whenever the sensor signal passes above or below this Setpoint

3.1.19
SIO
port operation mode in accordance with digital input and output defined in IEC 61131-2 that is established after power-up or fallback or unsuccessful communication attempts

3.1.20
switching mode
one out of a set of possible operational modes for binary signals such as 'deactivated', 'Single Point Mode', 'Window Mode', or 'Two Point Mode'

NOTE Vendor specific modes are possible.

3.1.21
switching signal
binary signal from or to a Device when in SIO mode (as opposed to the "coded switching" SDCI communication)

3.1.22
switchpoint
measurement value of a sensor where the binary output signal changes its value

3.1.23
Teach Flag
indication for the success of a Teachpoint setting

3.1.24
Teachpoint
trigger to set a threshold value or the border value of a range

3.1.25
Teach State
indication of the current state of the teach-in procedure

3.1.26
two point mode
evaluation method with two Setpoints where the binary output signal only changes if the sensor measurement value comes from above the highest Setpoint and passes the lowest Setpoint or if it comes from below the lowest Setpoint and passes the highest Setpoint

3.1.27
wake-up
procedure for causing a Device to change its mode from SIO to COMx

3.1.28
window mode
evaluation with two Setpoints where the binary output signal depends on the measurement value of the sensor being between the Setpoints or either above the highest or below the lowest Setpoint

3.2 Symbols and abbreviated terms

DI	Digital input	
DO	Digital output	
I/O	Input / output	
OD	On-request Data	
PD	Process Data	
PLC	Programmable logic controller	
SDCI	Single-drop digital communication interface	
SIO	Standard Input Output (binary switching signal)	[IEC 61131-2]
SP1	Setpoint 1 (rising edge)	
SP2	Setpoint 2 (falling edge)	
TP1	Teachpoint 1 (rising edge or lower border)	
TP2	Teachpoint 2 (falling edge or higher border)	

3.3 Conventions

3.3.1 Behavioral descriptions

For the behavioral descriptions, the notations of UML 2 [4] are used, mainly state diagrams. The layout of the associated state-transition tables is following IEC 62390 [3].

The state diagrams shown in this document are entirely abstract descriptions. They do not represent a complete specification for implementation.

3.3.2 Memory and transmission octet order

Figure 1 demonstrates the order that shall be used when transferring WORD based data types from memory to transmission and vice versa (Figure 1).

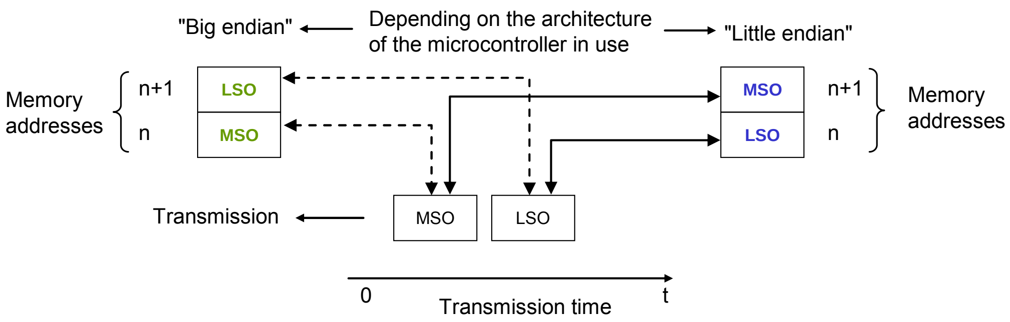

Figure 1 – Memory and transmission octet order

4 Overview of sensor devices

4.1 Smart Sensors

In factory automation, sensors nowadays are using a broad spectrum of transducers based on many different physical or chemical effects. They are converting one or more physical or chemical quantities (for example position, pressure, temperature, substance, etc.) and propagate them in an appropriate form to data processing units such as for example PLCs.

Due to the built-in microcontrollers these sensors are able to not only provide the conversion of the quantities but also to provide some preprocessing. Most of these sensors are "switch-point sensors". With the help of an individual parameterization or teaching process ("teach-in"), the sensors receive information on their "switching mode" and the threshold values ("Set-points"). This can result in one or more binary information about the measured quantity. Depending on functionality, those sensors provide the following outputs

- Binary information to transfer a switching state and/or
- Analog information to transfer measurement values such as pressure or temperature

This widespread sensor type is called "Smart Sensor". It has been somewhat "handicapped" so far by the restrictions of conventional digital and analog interfaces defined in IEC 61131-2.

4.2 Sensors migrating to SDCI

It is the purpose of SDCI to overcome the limitations of the classic sensor interfaces DI, DO, AI, and AO via a point-to-point digital communication that allows transmitting not only binary and/or analog information but additional information also. Very often, the changes to the core sensor application ("sensor technology") are very little during the migration to SDCI. However, the user realizes a dramatic increase in comfort and flexibility through the identification, parameterization, and diagnosis features.

5 Device profiles related to IEC 61131-9

5.1 SDCI technology specified in IEC 61131-9

Figure 2 shows the domain of the SDCI technology within the automation hierarchy.

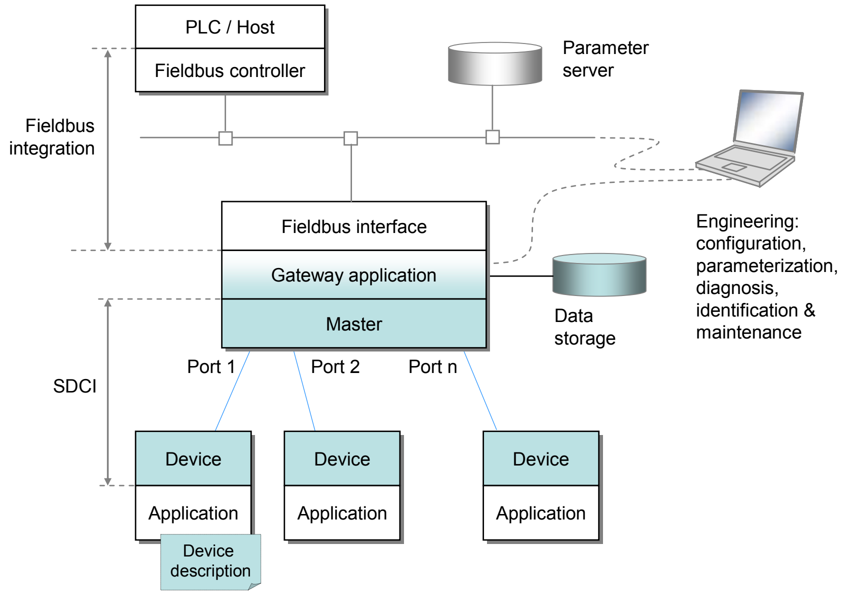

Figure 2 – Domain of the SDCI technology within the automation hierarchy

The SDCI technology defines a point-to-point digital communication interface for connecting "digital" or "analog" type sensors and actuators to a Master unit, which can be combined with gateway capabilities to become a fieldbus remote I/O node. The technology is specified in [1] and [2].

5.2 Profile classification

Figure 3 shows an overview of the SDCI technologies and profiles.

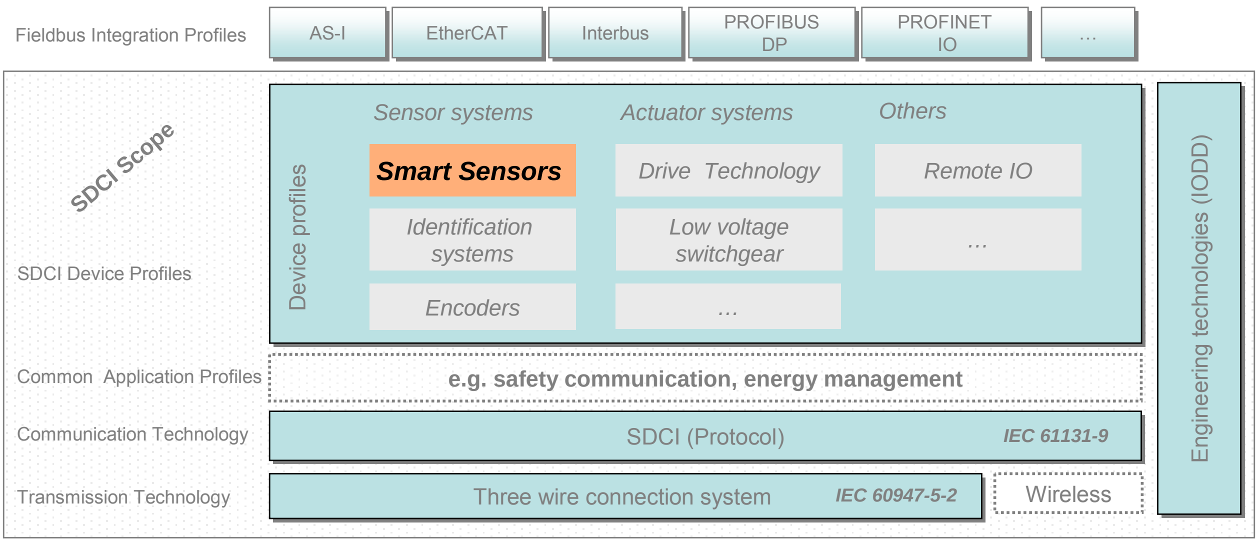

Figure 3 – Overview of SDCI technologies and profiles

The "SDCI Device Profiles" represent specifications of common functionality of particular Device type families/classes such as drives, low voltage switch gears, encoders, etc. These profiles primarily focus on the structure and behavior of the Device technology and secondarily on the data mapping on SDCI. Thus, the user recognizes a "generic" Device to a certain extent even when he switches from one brand to another.

The "Common Application Profiles" represent specifications that several Device type families/ classes can use. Examples are safety communication protocols or energy management.

The "Fieldbus Integration Profiles" specify the adaptation of the SDCI technology to particular fieldbuses. These specifications are outside the responsibility of the organization listed in Annex B. However, this organization is interested in harmonizing the "views" of SDCI users through the different fieldbuses.

5.3 Profile related Index space

The SDCI technology holds Indices and Subindices within Devices to store and/or retrieve parameter objects. Table 1 shows the profile related Indices defined in [1]. This profile specification overwrites some of the definitions in the standard, for example the ProductID, which is mandatory for Smart Sensors.

Table 1 – Excerpt of the SDCI Indices related to profiles

Index (dec)	Object name	Access	Length	Data type	M/O/C	Smart Sensor profile definitions
...						
0x0002 (2)	System Command	W	1 octet	UIntegerT	M/O	See Table 13
...						
0x000D (13)	Profile Charac-teristic	R	variable	ArrayT of UIntegerT16	M	See Table 3
0x000E (14)	PDInput Descriptor	R	variable	ArrayT of OctetStringT3	M	See Table 7
0x000F (15)	PDOutput Descriptor	R	variable	ArrayT of OctetStringT3	M	Not used in this Smart Sensor profile
...						
0x0013 (19)	Product ID	R	Max. 64 octets	StringT	M	See Table 9
...						
0x0017 (23)	Firmware Revision	R	Max. 64 octets	StringT	M	See Table 9
0x0018 (24)	Application Specific Tag	R/W	Min. 16, max. 32 octets	StringT	M	See Table 9
...						
0x0024 (36)	Device Status	R	1 octet	UIntegerT	O	See clause 11
0x0025 (37)	Detailed Device Status	R	variable	ArrayT of OctetStringT3	O	See clause 11
...						
0x0031 to 0x003F (49 to 63)	Reserved for profiles					Teach-in Channel and Teach-in Status BDC1 and BDC2 Index space See clause 12 and Table 11
...						
0x4000 to 0x4FFF (16384 to 20479)	Profile specific Index					Index space for BDC3 to BDC128. See 9.3.2
Key M = mandatory; O = optional; C = conditional						

5.4 Profile characteristics

All SDCI Device Profiles shall be characterized within the parameter object "Profile Characteristic" in Index 0x000D via ProfileIdentifiers (PID) listed within an array. Normally, an SDCI device supports only one SDCI Device Profile (e.g. this Smart Sensor Profile). It is also possible for an SDCI device to support several Common Application Profiles as well as several FunctionClasses (see 6.4). FunctionClasses defined in this profile can also be inherited to other SDCI device profiles, for example to SDCI actuators.

The individual PID describes a particular profile and its supported functions via the following IDs:

- DeviceProfileID
- CommonApplicationProfileID
- FunctionClassID

The parameter object "Profile Characteristic" supports up to 32 ID entries. Each and every supported profile and FunctionClass shall be indicated and coded as specified in Table 2.

Table 2 – Coding of ProfileIdentifiers (PID)

Parameter object name	Data type	Value range	Profile type
ProfileIdentifier (PID)	UIntegerT16	0x0000	No profile supported
		0x0001 to 0x3FFF	DeviceProfileID
		0x4000 to 0x7FFF	CommonApplicationProfileID
		0x8000 to 0xBFFF	FunctionClassID
		0xC000 to 0xFFFF	Reserved

The following rules apply:

1) Whenever a Device profile is supported such as for example "Smart Sensors", it shall be indicated via a DeviceProfileID entry

2) Whenever 1 to n common application profiles are supported, they shall be indicated via 1 to n CommonApplicationProfileIDs

3) Whenever 1 to n functions are supported, they shall be indicated via 1 to n FunctionClassIDs

4) The IDs shall be listed in ascending order (DeviceProfileIDs → CommonApplicationProfileIDs → FunctionClassIDs

Figure 4 – Indication rules for ProfileIdentifiers

The different profile identifers shall be ordered within the array of the parameter object "Profile Characteristic" in a sequence shown in Table 2 using the SDCI Subindices as a reference.

Table 3 shows the example for the "Profile Characteristic" of a Smart Sensor

Table 3 – Example of the profile identification for a Smart Sensor

Index	Sub-index	ProfileID	R/W	Data Type	Example ID
0x000D	1	Profile Identifier (DeviceProfileID)	R	UIntegerT16	0x0001: Smart Sensor Profile
	2	Profile Identifier (FunctionClassID)	R		0x8001: Binary data channel
	3	Profile Identifier (FunctionClassID)	R		0x8002: Process value
	4	Profile Identifier (FunctionClassID)	R		0x8004: Teach Channel

5.5 User benefits

As already mentioned in 5.2 the user recognizes from the Masters point of view a "generic" Device through the communication interface even when he switches from one brand to another. The exchange of process data and the behavior of the profile Device are the same, at least for "basic" functions. That means he is not forced to change his user program within the controller (PLC) in this case and he can expect the same basic behavior of the Device (for example process data, diagnosis, and identification). However, due to the objectives for the individual Device profiles, the interoperability levels can be different and the compatibility between the profile Devices can be partly limited. A good compromise is the possibility of reading the profile features out of the Device via the PLC program and adjusting the user program accordingly. Such a solution is the Smart Sensor profile defined in the following.

6 Smart Sensor profile

6.1 Objectives for the Smart Sensor profile

As mentioned in 5.5 the user expects a common "view" on a profile device and therefore he requires standardized functions. On the other hand, he expects innovations and customer specific adaptations to a certain extent. With this background, Device profiles are always a challenge and they are striving for good compromises. The Smart Sensor Device group compiled the following requirements and objectives for the profile:

- Manufacturer/vendor specific extensions (functions) shall always be possible
- The standardized profile functions (FunctionClasses) specified within this document are optional. If a manufacturer/vendor indicates particular FunctionClasses they shall be implemented and behave in the specified manner
- Each Smart Sensor shall provide its manufacturer/vendor specific Device description file (IODD). There shall be no "Profile-IODDs".
- The Smart Sensor profile does not focus on particular measurement technologies such as pressure, temperature, and alike. It focuses on common technology-independent features.
- The Device model shall describe the switching behavior of the Smart Sensor ("Switching model")
- Representation and transmission of the measurement information shall be based on ProcessDataVariables (PDV) and BinaryDataChannels (BDC)
- Necessary parameters for the profile shall be defined, for example setpoints, switching modes, etc.
- A uniform profile identification shall be specified (which parameter objects are mandatory)
- A uniform diagnosis information shall be defined

6.2 Measurement categories for Smart Sensors

The Smart Sensor profile definitions are independent from the physical or chemical quantities to be measured. Table 4 contains a list of typical physical and chemical measurement quantities for Smart Sensors. The list is far from being complete.

Table 4 – Typical physical and chemical measurement quantities

Geometry	Movement	Force	Heat	Optic	Chemistry
Position Distance Angle Direction Strain Level	Travel Speed Rotation Displacement Acceleration Vibration	Force Pressure Tension Torque Acceleration	Temperature Heat Heat conductivity Specific heat	Refractivity Irradiance Light density Luminance Chrominance	Substances Volume fraction Mass fraction Humidity Conductivity pH value

Smart Sensors are independent from the measurement quantities and represent the measurement results in a uniform manner

- as ProcessDataVariables (PDV) and/or
- switch information as BinaryDataChannels (BDC)

6.3 Smart Sensor model

The Smart Sensor profile defines a so-called function-driven Device model instead of for example an architectural model. That means it only defines independent and consistent functions (FunctionClasses). This allows the manufacturers/vendors to create a large variety of subsets from basic switching sensors using only the BinaryDataChannel (BDC) up to complex sensors with several measurement values using the ProcessDataVariables (PDV).

A Smart Sensor Device shall only support the indicated profiles and FunctionClasses.

Each and every FunctionClass consists of a communication dependent function and an associated mapping on the SDCI communication. FunctionClasses are represented and referenced by profile identifiers, for example FunctionClassID = 0x8001, as shown in Figure 5.

The measurement technology (transducer) is manufacturer/vendor specific and not part of this profile.

Figure 5 shows the FunctionClasses defined by the Smart Sensor profile.

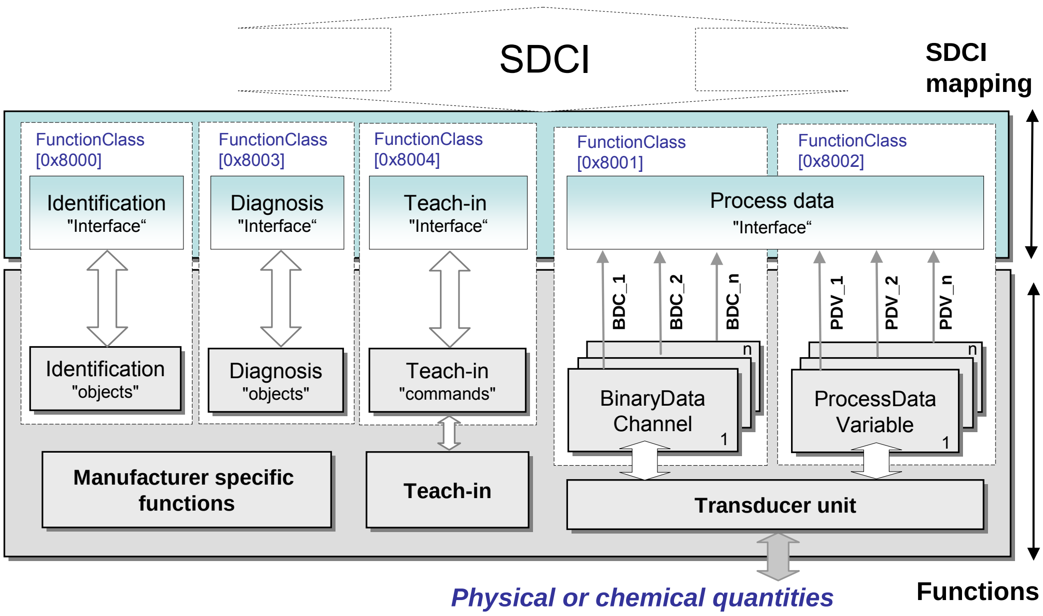

Figure 5 – Overview of FunctionClasses

The *Device Identification* (FunctionClass [0x8000]) extends the standard SDCI Device identification by some additional identification objects. This FunctionClass is mandatory for the Smart Sensor profile.

The *BinaryDataChannel* (FunctionClass [0x8001]) uses the measurement values out of the transducer unit and creates binary information (BDC_n), whenever certain thresholds are passed. These thresholds are defined via parameters as defined in clause 9.

The *ProcessDataVariables* (FunctionClass [0x8002]) uses the measurement values out of the transducer unit and creates data structures (PDV_n) representing the physical or chemical quantity, for example pressure or temperature. These data structures within the ProcessDataVariables are standardized to a maximum extent as shown in clause 10.

The *Device Diagnosis* (FunctionClass [0x8003]) extends the standard SDCI Device diagnosis by some additional diagnosis objects. This FunctionClass is optional for the Smart Sensor profile.

The *Teach-in Commands* (FunctionClass [0x8004]) allow the user to remotely teach certain threshold levels in the automation process via the user program in a controller. Manufacturer/vendor specific teach-in procedure specialties are not within the scope of this profile.

The mapping of BDCs and PDVs into SDCI communication messages is specified in clause 7. These data structures are designed for simplicity and highest efficiency.

6.4 Smart Sensor object model

The profile for Smart Sensors provides standardized functions that are encapsulated within Smart Sensor objects. Figure 6 shows the defined FunctionClasses of this Smart Sensor profile. Besides the classes for identification, and diagnosis, it contains the classes ProcessDataVariable and BinaryDataChannel. These classes are showing the associated attributes, whereas the class TeachChannel shows its defined methods (commands).

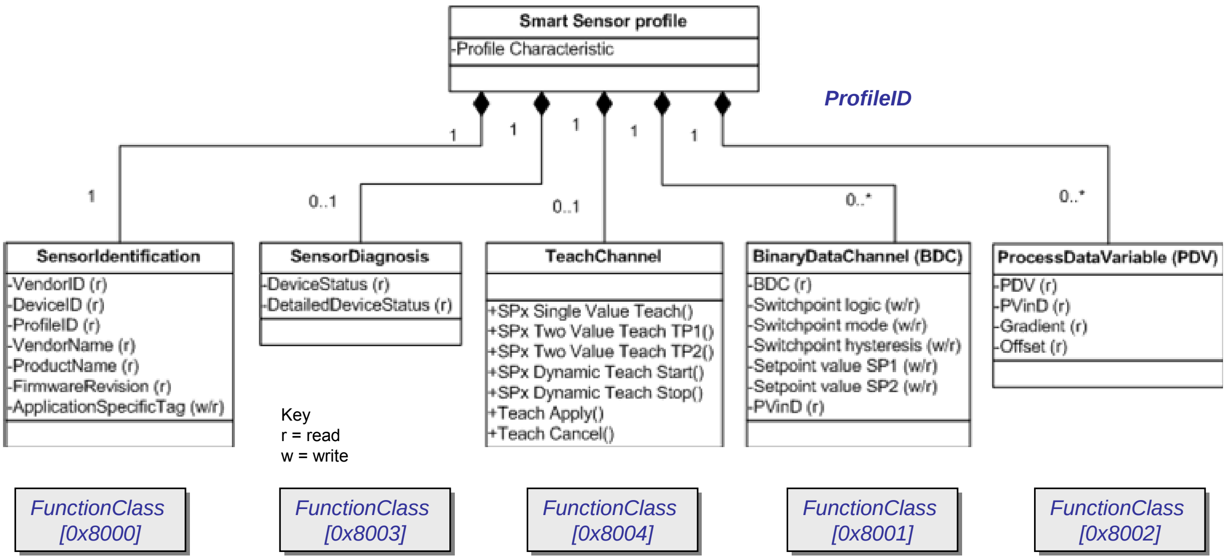

Figure 6 – Smart Sensor object model

The classes BDC and PDV can be instantiated 0 to * (n) times as shown in the example in Figure 7, depending on the complexity of the sensor. The FunctionClass for identification is mandatory. The FunctionClass for diagnosis is optional.

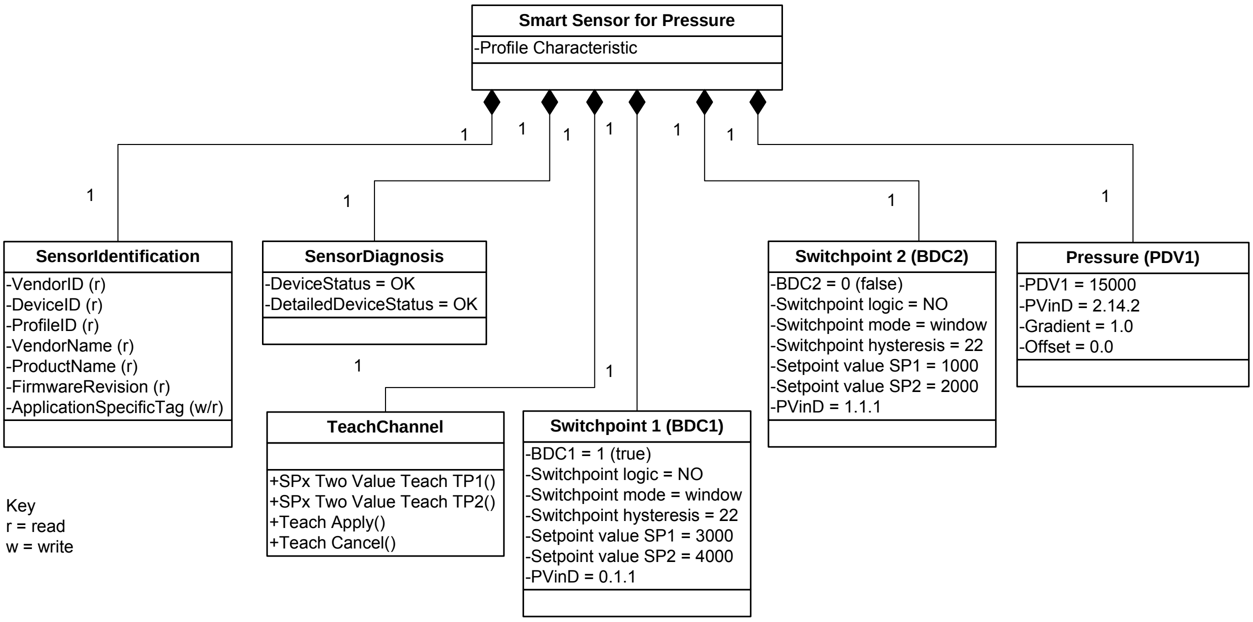

Figure 7 – Example for a pressure sensor

The example sensor in Figure 7 demonstrates:

- Each BinaryDataChannel (binary output) is represented by its own instance of the class (object) with the instantiated attributes. In the particular example two BDC instances "Switchpoint 1" and "Switchpoint 2" are available.
- Each measurement value in the "ProcessDataVariable is represented by its own instance of the class (object) with the instantiated attributes. In the particular example one PDV instance "Pressure" is available.
- The TeachChannel offers four commands for remote teach-in (clause 12):
  - SPx Two Value Teach TP1
  - SPx Two Value Teach TP2
  - Teach Apply
  - Teach Cancel

A user program ("client") for example in a PLC can access the BDC and PDV objects via corresponding functions or methods respectively (Table 5).

Table 5 – Abstract notation for BDC and PDV access of a PLC client

Read/Write access	Description
Read Sensor1.Pressure.PDV1	Readout of the pressure value with corresponding scale: PDinD, gradient, offset
Read Sensor1.switch point 1.BDC1	Readout of the switching signal state: PDinD
Write Sensor1.switch point 1.SetPointValue SP1	Write SetPointValueSP1

The parameter set of a FunctionClass can be classified into two groups:

- Operating parameters, which can be modified during production
- Configuration parameters (static data), which are only set/modified during commissioning

6.5 How to use the Smart Sensor profile

The different FunctionClasses are either mandatory or optional depending on the sensor type. Table 6 shows the possible FunctionClass combinations for different sensor types.

Table 6 – FunctionClass combinations for different sensor types

Smart Sensor type	Identification FunctionClass [0x8000]	BDC FunctionClass [0x8001]	PDV FunctionClass [0x8002]	Diagnosis FunctionClass [0x8003]	Teach-in FunctionClass [0x8004]
"Binary" sensor	M	1 to n	-	O	O
"Analog" sensor	M	-	1 to n	O	O
"Binary + analog" sensor	M	1 to n	1 to n	O	O
Key M = mandatory	O = optional	- = not relevant			

7 Process Data mapping (PD)

7.1 Process Data and its transmission

Depending on the particular type, a Smart Sensor arranges binary information (BDC) and/or 0 to n ProcessDataVariables (PDV) for the cyclic transmission to the Master via SDCI in a so-called "PDinput data stream".

Each and every Smart Sensor provides an input Process Data description (PDinputDescriptor) indicating the composition (mapping) of the BinaryDataChannels (BDC) and ProcessData-Variables (PDV) in the "PDinput data stream" with the necessary number of octets.

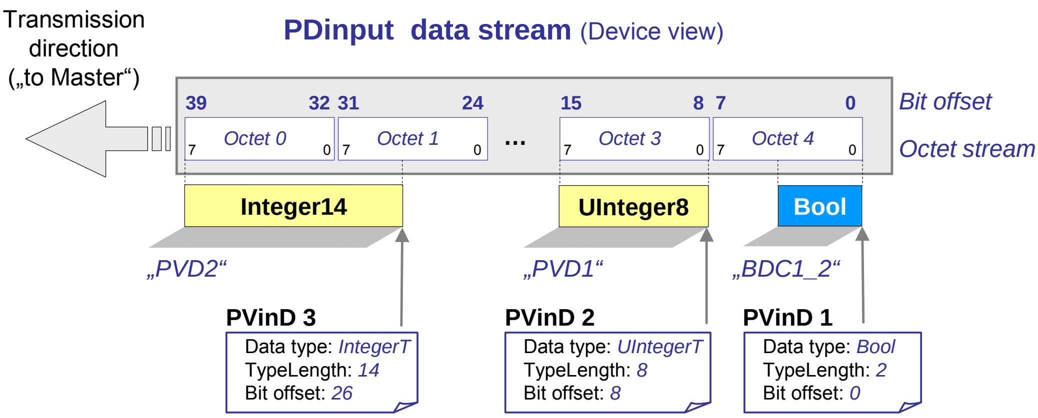

Figure 8 – Example PDinput data stream

The "PDinput data stream" example shown in Figure 8 comprises 5 octets (octet 0, 1, 2, 3, 4) to be transmitted to the Master. The Smart Sensor technology (application) maps BDCs and PDVs into the data stream. The location of each of these data elements within the data stream is described in a process variable descriptor (PVinD). Basis for this description is the "Bit offset" reaching from the last transmitted bit to the first one as defined in Annex E "Data types" in [1].

NOTE From the user program perspective, usage of standard data types such as UInteger16, or Integer16 would be the preferred way of mapping. However, due to performance reasons "packed" data structures cannot be avoided.

For Smart Sensors the following information is relevant and it will be specified in the subsequent clauses:

- The content of the PVinD process variable descriptor defining the data type and location of BDC and PDV within the data stream
- The content of the PDinputDescriptor describing "what" is to be transmitted and "how"

7.2 Process variable descriptors (PVinD, PVoutD)

7.2.1 Coding

The content of the process variable descriptors PVinD or PVoutD shall be available

- in the user manual of the Smart Sensor,
- in the IODD Device description file, and
- within the Device in the corresponding Index.

Each and every PVD or BDC respectively is described unambiguously via its descriptor PVinD. Subsequent Boolean variables can be described within one PVinD. The following information shall be provided within a PVinD:

- the data type (DataType) of the particular process variable. "Set of BoolT" describes here combined BinaryDataChannels (BDCs)
- the length of the data type (TypeLength) in bit, for example 6 for UInteger6
- the bit offset (Bit offset) as the beginning of the variable in the data stream
- any manufacturer/vendor specific data structures, which cannot be described via the standard BDC or PDV descriptors, are described via a process variable descriptor (e.g. additional output data)

Table 7 presents the coding of the process variable descriptors PVinD or PVoutD.

Table 7 – Coding of PVinD or PVoutD

Bit	Item	Coding
Octet 1	DataType	0: OctetStringT 1: Set of BoolT 2: UIntegerT 3: IntegerT 4: Float32T 5 to 255: reserved
Octet 2	TypeLength	0 to 255 Bit
Octet 3	Bit offset	0 to 255 Bit

NOTE The abstract notation in this profile specification of a PVinD is: DataType.TypeLength.Bit_offset

7.2.2 PDInputDescriptor

Smart Sensor Devices shall use the standard Device parameter "PDInputDescriptor" in Index 0x000E to store the description information according to Table B.7 in [1]. The user program within a controller (e.g. PLC) can thus read this information. The "PDInputDescriptor" contains a descriptor (PVinD) for each and every process variable. Exception: Subsequent Boolean variables can be described within one PVinD. Table 8 presents an example "PDInputDescriptor" with two PDVs and two combined BDCs.

Table 8 – Example "PDInputDescriptor"

Index (dec)	Subindex (dec)	Access	PDInputDescriptor	Coding	Data type
0x000E (14)	1	R	PVinD (BDC1,BDC2)	See Table 7	OctetStringT3
	2	R	PVinD (PDV1)		
	3	R	PVinD (PDV2)		

7.3 Profile specific PD structures

7.3.1 General

In order to avoid a large variety of data structures and descriptors and as a consequence complexity, this profile specification specifies and recommends only a few variable descriptions.

7.3.2 One or more BDCs (recommended)

It is highly recommended for pure binary Smart Sensors without additional PDVs to use the data structure demonstrated in Figure 9. The number of supported BDCs, four in Figure 9, defines the size of the bit field. The BDCs are right-aligned in ascending order without gaps.

The PVinD in this case is: Set of BoolT.4.0

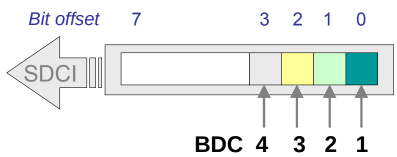

Figure 9 – Recommended data structure for pure BDCs

7.3.3 One PDV

It is highly recommended for Smart Sensors with one PDV to use the data structure demonstrated in Figure 10. The example shows, that a Smart Sensor can cast an 8, 10, or 14 bit value into a UIntegerT16 data type, thereby using only part of the space. The leading bits shall be "0". Variables of type Integer < 16 bit shall also be casted into variables of type IntegerT16. Type casting rules are specified in [1], Annex E2.3 or E2.4.

The PVinD in this case is: UIntegerT.16.0

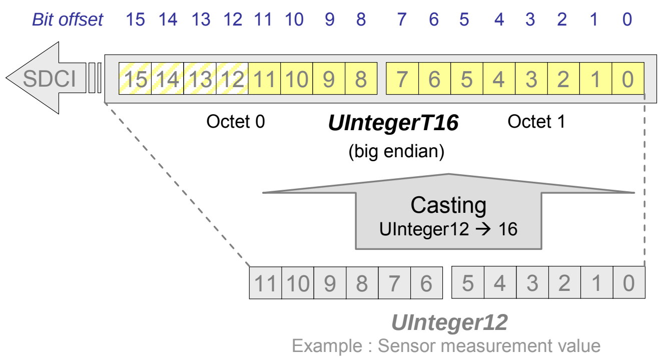

Figure 10 – Recommended data structure for one PDV

7.3.4 PD lengths up to two octets

Exceptions exist for PD lengths up to two octets. Especially for bit offsets up to 16 other than octet aligned data types may be used ("packed format"). For PD with more than two octets the rules in 7.3.5 apply.

7.3.4.1 One PDV and several BDCs

It is highly recommended for Smart Sensors with one PDV and one to two BDCs to use the data structure demonstrated in Figure 11.

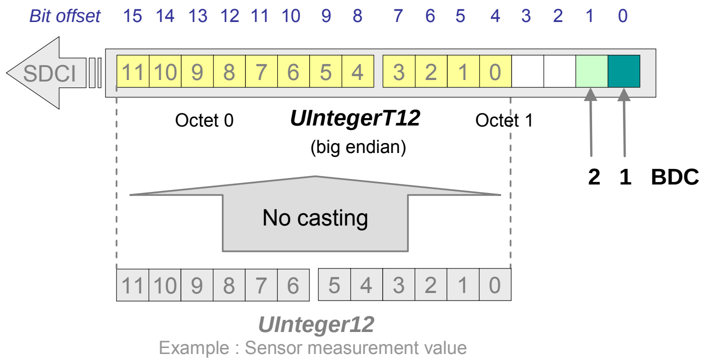

Figure 11 – Recommended data structure for a PDV and up to two BDCs

The following rules apply:

- BDCs are right-aligned in ascending order (always at bit offset 0)
  PVinD in this case is: Set of BoolT.2.0
- PDV with e.g. UIntegerT12 is left-aligned mapped to bit offset 4
  PVinD in this case is: UIntegerT.12.4

7.3.4.2 One PDV, several BDCs, and auxiliary variables

It is highly recommended for Smart Sensors with one PDV, one to two BDCs, and auxiliary variables such as qualifiers to use the data structure demonstrated in Figure 12.

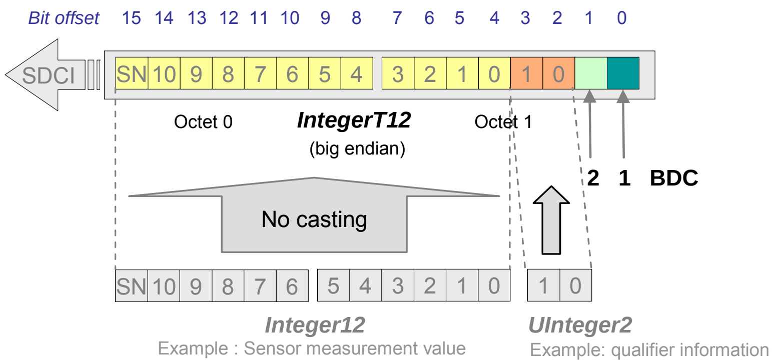

Figure 12 – Recommended data structure for a PDV, BDCs, and auxiliary variables

The following rules apply:

- BDCs are right-aligned in ascending order (always at bit offset 0)
  PVinD in this case is: Set of BoolT.2.0
- PDV with IntegerT12 (e.g. measurement value) is mapped left-aligned to bit offset 4
  PVinD in this case is: IntegerT.12.4
- Auxiliary variables (e.g. qualifier information) shall be right-aligned to the BDCs
  PVinD in this case is: UIntegerT.2.2

7.3.5 PD lengths larger than two octets

It is highly recommended for Smart Sensors with 0 or more BDCs, 2 or more PDVs, and manufacturer/vendor specific process data (outside the scope of this profile specification) to use the data structure demonstrated in the example in Figure 13. The following rules shall be observed (mandatory):

- Within the first two octets the rules of 7.3.4 apply. Especially the BDCs are always starting at bit offset 0.
- All variables starting at bit offset 16 shall be mapped octet aligned. Potential waste of bits is accepted. Variables shall be casted to SDCI data structures if necessary. See [1], Annex E2.3 and E2.4 for casting rules.

In addition, it is highly recommended to observe the following rules (recommended):

- Best practice for PDVs is the usage of UInteger16 or Integer16 respectively (easier data processing)
- UIntegerT to be favored over IntegerT
- Manufacturer/vendor specific process data can use their own rules. However, it is highly recommended to observe the rules within this profile

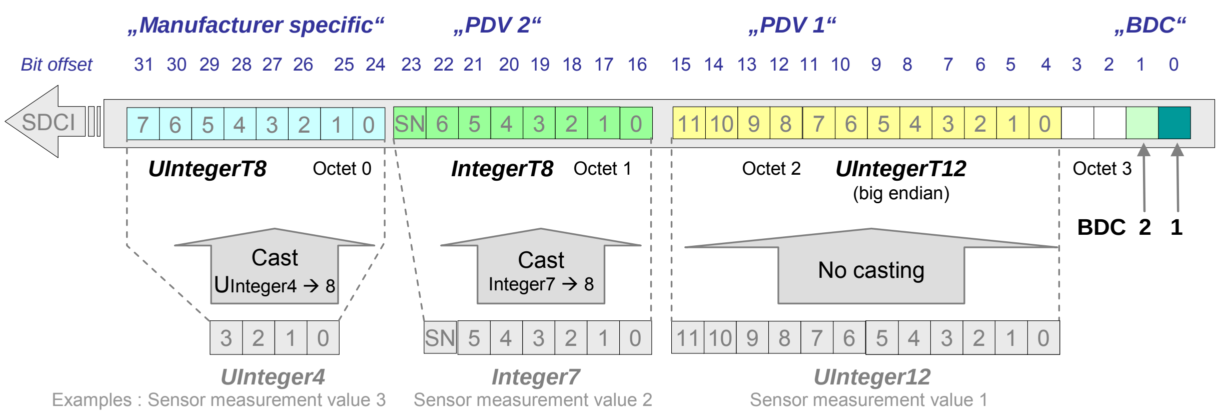

Figure 13 – Recommended data structure for multi PDVs and zero or more BDCs

The PVinDs in Figure 13 are:

PVinD 1	Set of BoolT.2.0	(BDC2 and BDC1)
PVinD 2	UInteger.12.4	(PDV1)
PVinD 3	Integer.8.16	(PDV2)
PVinD 4	UInteger.8.24	(Manufacturer/vendor specific)

8 Device identification objects [0x8000]

Table 9 shows the deviating definitions in this profile as opposed to the standard [1].

Table 9 – Deviating definitions for identification data objects

Index (dec)	Object name	Access	Length (octets)	Data Type	Mandatory/ Optional	Comment
0x0013 (19)	Product_ID	R	Max. 64	StringT	M	Herein mandatory
0x0017 (23)	Firmware Revision	R	Max. 64	StringT	M	Herein mandatory
0x0018 (24)	ApplicationSpecificTag	R/W	Min. 16, max. 32	StringT	M	Herein mandatory
Keys R = read W = write						

9 BinaryDataChannel [0x8001]

9.1 Characteristic of the BDC

This FunctionClass represents as process data the binary switching state information (BDC). It requires configuration and parameterization.

9.2 Configuration and parameterization of the BDC

9.2.1 General

This profile specification defines several best-practices BDCs. Manufacturer/vendor specific BDCs are always possible.

The following 4 parameters define the switching behavior of a BDC:

- Switchpoint logic
- Switchpoint hysteresis
- Switchpoint mode
- Setpoints SP1 and SP2

The parameters are specified in the subsequent clauses.

The Setpoint parameters are defined in detail in 9.2.5. The coding of the Setpoint and Switchpoint parameters is specified in 9.2.6.

9.2.2 Switchpoint logic

The parameter "Switchpoint logic" defines whether the switching information is transmitted in inverted or not inverted manner.

9.2.3 Switchpoint hysteresis

The parameter "Switchpoint hysteresis" defines whether a hysteresis is associated with the Setpoints SP1 and SP2. The layout of the hysteresis in respect to SP1 and SP2, for example symmetrical, right-aligned, or left-aligned, etc. is manufacturer/vendor specific. It cannot be defined in the FunctionClass.

The interpretation of the hysteresis values (relative or absolute) is also manufacturer/vendor specific.

9.2.4 Switchpoint mode

9.2.4.1 Overview

The parameter "Switchpoint mode" defines how the binary switching information is created depending on Setpoint parameters (SP1, SP2) and the current measurement value.

The Switchpoint Mode does not define the switching function itself. The different sensor types are using different switching functions depending on the various manufacturer/vendor specific applications.

The quiescent state of sensors for presence detection (e.g. optical proximity sensors or ultrasonic sensors) is a measurement value of "infinite". An approaching object will cause the switching information of the sensor to change at the setpoint (measurement value). The departing object will cause the switching information of the sensor to switch back at a larger measurement value than the setpoint (see Figure 14)

The quiescent state of sensors for quantity detection (e.g. pressure or temperature sensors) is a measurement value of "zero". An increasing measurement value will cause the switching information of the sensor to change at the setpoint. A decreasing measurement value will cause the switching information of the sensor to switch back at a smaller measurement value than the setpoint (see Figure 15).

The associated FunctionClass comprises 4 different modes:

- Deactivated
- Single Point Mode
- Window Mode
- Two Point Mode

If a Smart Sensor implements a BDC, it shall support at least one of these Switchpoint modes. Additional modes are optional. In case a Smart Sensor does not support any other of the additional optional modes, the general rule for not supported parameters applies (9.3.3). It is possible for a manufacturer/vendor to supplement the above defined modes by his own specific modes.

9.2.4.2 Single Point Mode

Figure 14 demonstrates the switching behavior in Single Point Mode. The switching information changes, when the current measurement value passes the threshold defined in Setpoint SP1. This change occurs with raising or falling measurement values. If a hysteresis is defined for SP1, the switching behavior shall observe the hysteresis as shown in Figure 14. This behavior is typical for "presence detection of objects" with none symmetrical hysteresis in respect to SP1 and not inverted switching.

The Setpoint SP2 is not relevant for this mode.

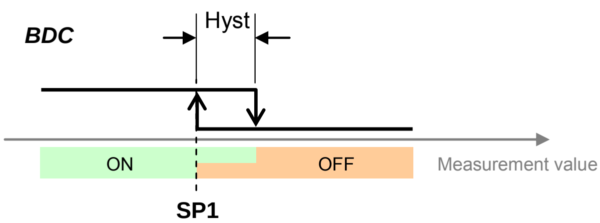

Figure 14 – Example of a Single Point Mode for presence detection

The behavior shown in Figure 15 is typical for "quantity (level) detection of materials (liquids)" with none symmetrical hysteresis in respect to SP1 and not inverted switching.

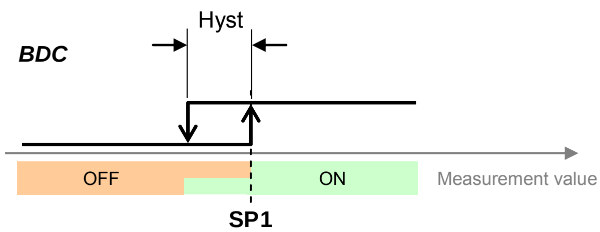

Figure 15 – Example of a Single Point Mode for quantity detection

9.2.4.3 Window Mode

Figure 16 demonstrates the switching behavior in Window Mode. The switching information changes, when the current measurement value passes the thresholds defined in Setpoint SP1 and Setpoint SP2. This change occurs with raising or falling measurement values.

If hysteresis is defined for SP1 and SP2, the switching behavior shall observe the hysteresis as shown in Figure 16. This behavior shows symmetrical hysteresis in respect to SP1 and SP2 and not inverted switching.

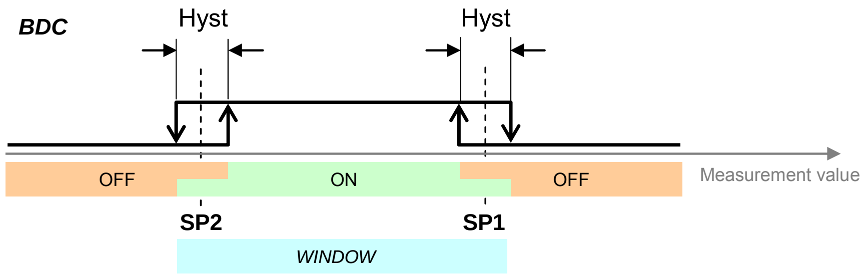

Figure 16 – Example for the Window Mode

9.2.4.4 Two Point Mode (without hysteresis)

Figure 17 demonstrates the switching behavior in Two Point Mode. The switching information changes, when the current measurement value passes the threshold defined in Setpoint SP1. This change occurs only with raising measurement values. The switching information changes also, when the current measurement value passes the threshold defined in Setpoint SP2. This change occurs only with falling measurement values. Hysteresis shall be ignored in this case.

If the measurement value is in between SP1 and SP2 at power-on of the Smart Sensor, the behavior depends on the manufacturer/vendor specific design of the Device.

The behavior shown in Figure 17 is typical for "presence detection of objects" with no hysteresis in respect to SP1 and SP2 and not inverted switching.

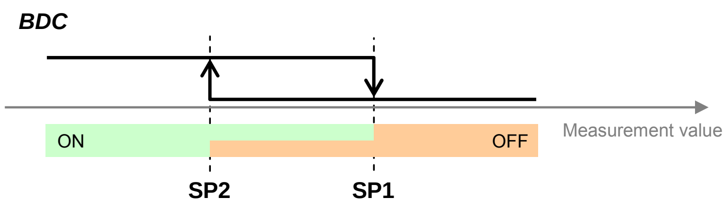

Figure 17 – Example for the Two Point Mode of presence detection

The behavior shown in Figure 18 is typical for "quantity (level) detection of materials (liquids)" with no hysteresis in respect to SP1 and SP2 and not inverted switching.

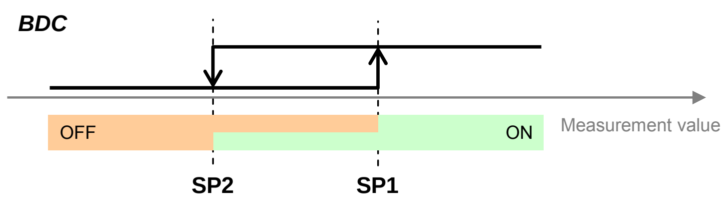

Figure 18 – Example for the Two Point Mode of quantity detection

9.2.5 Setpoint parameters (SP1, SP2)

A Smart Sensor deploys at least the Setpoint SP1 or both Setpoints SP1 and SP2 per BDC. However, it always shall provide both Setpoint parameters of this FunctionClass BDC. That means, even if the Smart Sensor does not use SP2 in its swiching functions, it shall support read and write access to both parameters. In case a Smart Sensor does not support any parameters, the general rule for not supported parameters applies (see 9.3.3).

The interpretation of the Setpoints SP1 and SP2 depends on the implementation of the manufacturer/vendor. However, if the measurement value for the definition of switching information (BDC) is also provided as a ProcessDataVariable (PDV), the Setpoints shall be represented in the same manner, for example with Gradient and Offset and octet granular data types (≥ 1 octets). See 10.2 for details.

The Smart Sensor Device shall support all the necessary plausibility checks described in clause 10 ("Device") of the standard [1] and the following:

- Setpoint SP2 shall be outside the hysteresis range of SP1
- Setpoints SP1 and SP2 are within the measurement value range

In case one or more checks failed, the Smart Sensor shall behave in the following manner:

- During acyclic data exchange (via ISDU), the Device shall return a negative response and restore the previous values
- During cyclic data exchange, the Device shall send valid Process Data based on previous valid parameter data

In order to avoid inconsistent configuration data it is important to note,

- that SP1 and SP2 data are written together via Subindex 0 (one record) guaranteeing that a changed value of SP1 or SP2 cannot cause a plausibility check error, or
- that the option Block Parameter [1] is used for a change of configuration guaranteeing a plausibility check and activation of the written parameters not before the termination of the entire transmission.

9.2.6 Setpoint and Switchpoint parameter coding

Table 10 shows the parameter coding of the Setpoint and Switchpoint parameters.

Table 10 – Setpoint and Switchpoint parameter coding

Object name	Length	Data Type	Coding	Definition
Setpoint SP1/2	8/16/32/64	UIntegerT IntegerT Float32T	Manufacturer/vendor specific	Typically corresponding to the PDV. However, the data structure is extended to octet granular data types and right-aligned.
Switchpoint logic	8	UIntegerT	0x00 : Value as specified Optional values: 0x01 : Inverted value 0x02 … 0x7F : Reserved 0x80 … 0xFF : Vendor specific	Binary value of the switching information ("1" = true, "0" = false) within the BDC (BinaryDataChannel)

Object name	Length	Data Type	Coding	Definition
Switchpoint mode	8	UIntegerT	0x00 : Deactivated 0x01 : Single point mode 0x02 : Window mode 0x03 : Two point mode 0x04 to 0x7F : Reserved 0x80 to 0xFF : Vendor specific	One of the defined modes shall be supported
Switchpoint hysteresis	16	UIntegerT	0x0000 : mandatory, if no hysteresis or vendor specific default Optional values: 0x0001 to 0xFFFF: Vendor specific definition	-

9.3 BDC mapping

9.3.1 Concepts

The binary switching information of the BDCs is mapped into the PDinput data stream (Figure 8) as defined in 7.3. The configuration and the parameterization of the BDCs are mapped in the profile related Index space as illustrated in Table 1.

The BDC FunctionClass [0x8001] can be parameterized via the standardized parameter objects described within the subsequent clause.

9.3.2 BDC Index space

Each and every BDC features a parameter set to define its switching behavior (Switchpoints) and an additional parameter set to define the thresholds (Setpoints). The mapping of these parameter sets for BDC1 and BDC 2 is shown in Table 11. The coding of the parameters is defined in Table 10.

Table 11 – Index space for BDC1 and BDC2

Index (dec)	Subindex (dec)	Access	Parameter name	Coding	Data type	Comment
0x003C (60)	01	R/W	Setpoint SP1	See Table 10	UIntegerT IntegerT Float32T	BDC1
	02	R/W	Setpoint SP2		UIntegerT IntegerT Float32T	
0x003D (61)	01	R/W	Switchpoint logic		UIntegerT	
	02	R/W	Switchpoint mode		UIntegerT	
	03	R/W	Switchpoint hysteresis		UIntegerT	
0x003E (62)	01	R/W	Setpoint SP1		UIntegerT IntegerT Float32T	BDC2
	02	R/W	Setpoint SP2		UIntegerT IntegerT Float32T	
0x003F (63)	01	R/W	Switchpoint logic		UIntegerT	
	02	R/W	Switchpoint mode		UIntegerT	
	03	R/W	Switchpoint hysteresis		UIntegerT	

Index space for additional 126 BDCs is available in the ProfileSpecificIndex space (Table 1). Thus, BDC3 is located in 0x4000 and 0x4001 and BDC128 in 0x407B and 0x407C.

9.3.3 Access behavior of not supported Subindices

The parameters for each and every supported BDC shall be readable and writeable as already indicated in Table 11. In detail the following rules apply:

- Parameters of a BDC not functionally supported by the Smart Sensor shall also be readable and writeable
- Those parameters can be written with the default value
- If other than default values are written, the Smart Sensor shall respond with the ErrorCode 0x8030 (PAR_VALOUTOFRNG = parameter value out of range)
- In case of a readout of a functionally not supported parameter, the Smart Sensor shall respond with the default value
- In case of access to not supported BDCs, the Smart Sensor shall respond with the ErrorCode 0x8011 (IDX_NOTAVAIL = Index not available)

10 ProcessDataVariable [0x8002]

10.1 Scaling and dimensions

Normally, the ProcessDataVariable of a Smart Sensor carries a measurement value of a physical or chemical quantity within the data structures (PDV) defined by the manufacturer/vendor of the Device. See clause 7 for details.

The transmitted value can be converted into a dimensioned value (°F, °C, inch, m, etc.) via a linear equation y = m·x + b. "m" represents the slope and "b" the intercept with the y coordinate. Within this profile, "slope" is called "gradient" and the value of the intercept is called "Offset". Figure 19 illustrates the relationships.

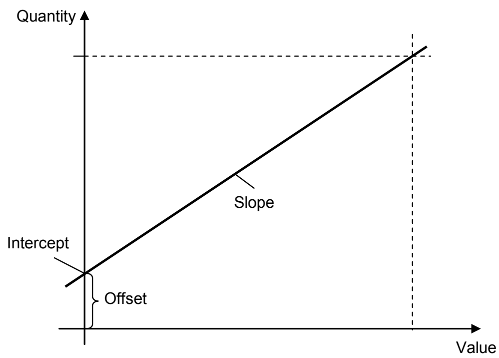

Figure 19 – Value to quantity conversion via linear equation

The manufacturer/vendor is responsible for the provision of the "Gradient" and the "Offset" values for the conversion equation (1).

$$\text{Variable} = \text{Gradient} \times \text{PDV} + \text{Offset} \quad \textbf{(1)}$$

Usually the data type for Gradient and Offset is Float32T. With the help of this information any computer software or PLC can calculate the dimensioned variable out of the transmitted PDV. Figure 20 illustrates two conversion examples for pressure and temperature.

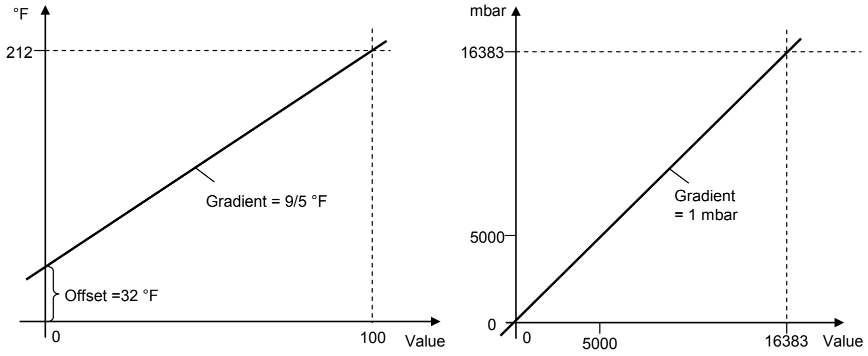

Figure 20 – Conversion examples

Usually, the transmitted PDV value is based on a dimensioned measurement value as shown in the right example of Figure 20 (pressure in mbar). In the left example a dimensioned temperature measurement value (°C) is converted in °F.

10.2 Recommended PDV representation

Objective of the recommendations within this clause is to demonstrate the data processing of PDVs in PLCs. It is highly recommended to observe the following rules in order to simplify the programming and to increase performance:

- PDVs of size > 16 bit should be represented in octet granular data types (16, 24, 32), preferably UIntegerT
- For data < 16 bit the data type UIntegerT should be used that is easily extendable to octet granular data types
- Preferred data lengths are 8, 12, 14, 16, 32, or 64 bit
- PDVs should carry dimensioned measurement values as shown in Figure 20 and Figure 21

Figure 21 illustrates the relationship between a dimensioned PDV and its PLC variable.

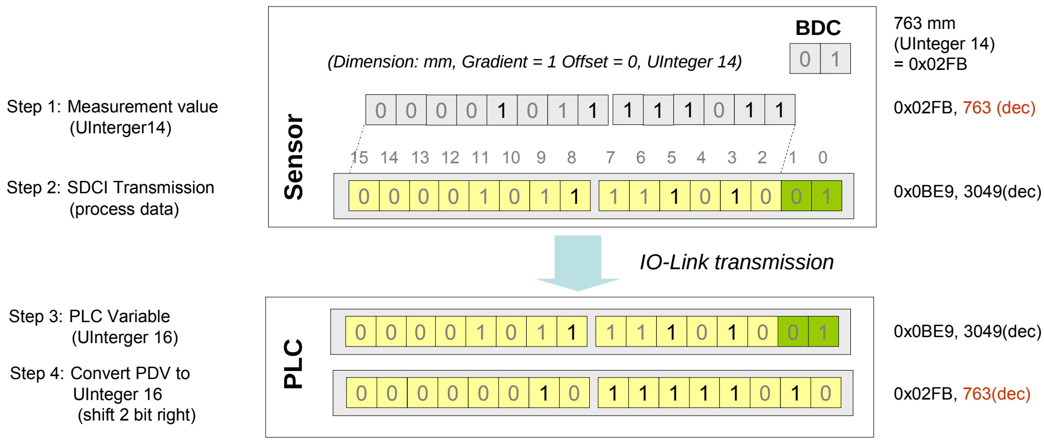

Figure 21 – Relationship between a dimensioned PDV and its PLC variable

Figure 22 demonstrates a typical PLC user program for a measurement value conversion. A PLC user program transforms the PDV via shift operations into a 16 bit UInteger variable.

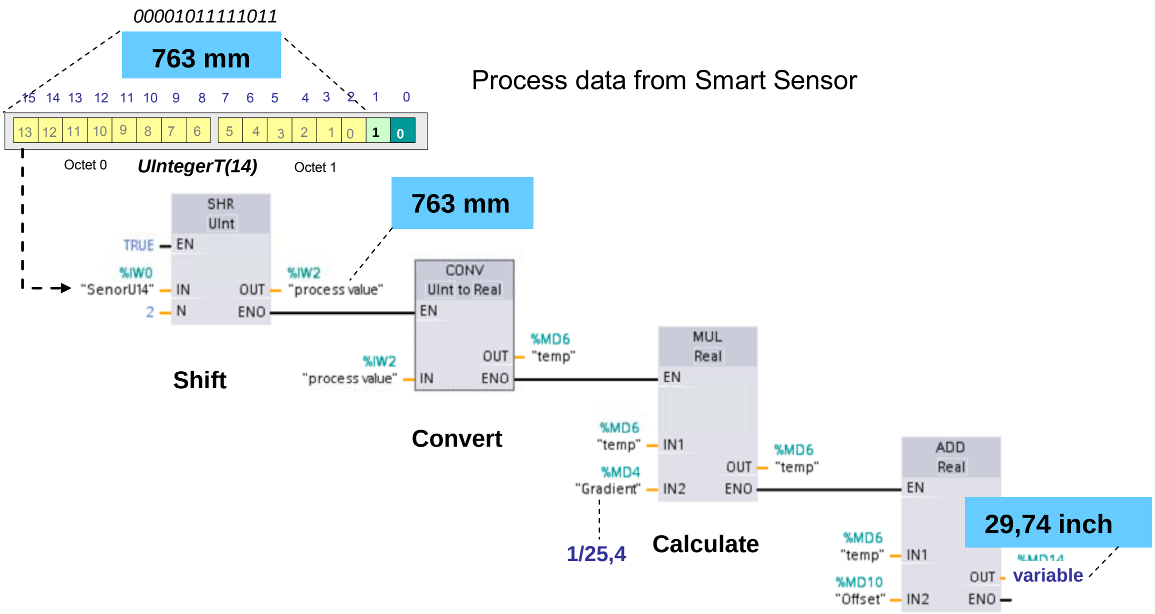

Figure 22 – Example PLC program for a measurement value conversion

11 Diagnosis [0x8003]

11.1 DeviceStatus and DetailedDeviceStatus

Each Smart Sensor Device shall feature a hierarchical diagnosis status within the parameter objects DeviceStatus and DetailedDeviceStatus as shown in Table 1 and defined in [1].

The DeviceStatus and DetailedDeviceStatus are already defined in [1] and need no further profile specific definitions.

However, Smart Sensors shall meet the following requirements for DetailedDeviceStatus:

- Only entries of Events of type "appears"/"disappears"
- Each appeared Event (Event Qualifier and EventCode) shall be entered
- Each disappeared Event shall be entered and lead to a deletion of the corresponding entry (identical EventCode) in DetailedDeviceStatus or to an overwriting with "0". This way, the current diagnosis status is always represented within the DetailedDeviceStatus.
- The DetailedDeviceStatus contains a maximum of 64 entries and thus can keep 64 current diagnosis statements at a time. The actual size is manufacturer/vendor dependent.
- The DetailedDeviceStatus buffer shall be cleared at each start-up of the Smart Sensor Device and refilled with diagnosis statements based on faults still in place
- Implementation hint: The dynamic/static strategy for the entries is manufacturer/vendor specific:
  - static: one fixed diagnosis statement within a particular Subindex
  - dynamic: an occurring diagnosis information will be entered in the next free Subindex (revolving system)

11.2 Smart Sensor EventCodes

The IEC 61131-9 [1] reserves in Annex D the EventCode range from 0xB000 to 0xBFFF for profiles. This profile for Smart Sensors does not define any profile specific EventCode.

12 TeachChannel [0x8004]

12.1 Teach-in concepts for Smart Sensors

The FunctionClass "TeachChannel" defines an interface for remote teach-in functions via SDCI communication and standardized commands for the most common basic teach-in mechanisms. Thus, the Smart Sensor profile provides a uniform and flexible interface for several teach-in methods. Instead of defining all kinds of teach-in methods, this FunctionClass defines a set of universal commands that can be used in various sequences to realize many individual methods. This includes the calculation algorithms for the associated parameters such as the thresholds for the Setpoints SP1 and SP2. The FunctionClass provides a "music instrument"; the "music" to play is defined by the manufacturer/vendor.

Two parameters are defined to control the teach-in procedure. The "Teach-in Channel" parameter (12.2 and Table 13) allows to select the BDC to be taught. This is required, if several BDCs are assigned to a teach-in procedure and the adjustment of the threshold values. It is default behavior, that teach-in commands are automatically active for the BDC with teach-in capability defined by the manufacturer/vendor. It is highly recommended for basic Smart Sensors to assign teach-in capability to BDC1 in order to avoid explicite addressing of a BDC.

Several commands are defined for the second parameter "Teach-in Command" (12.3 and Table 13). Each individual command enables the user to start one out of several standardized teach-in procedures. The commands are described within the context of a possible application within the subsequent clauses.

The FunctionClass [0x8004] provides also feedback on the status and the results of the teach-in activities. A universal state machine with common states (Idle, Busy, Wait-on-command, Success, and Error) for the different teach-in procedures is defined in 12.4.2. The parameter "Teach-in Status" holds the information about the current state of the activated teach-in procedure (12.4). The parameter provides two different types of information:

- Teach Flags: Feedback, whether the Device determined a certain "Teachpoint" successfully or not
- Teach State: Feedback on the current state of the particular teach-in procedure

12.2 Parameter 1: "Teach-in Channel"

The parameter "Teach-in Channel" allows addressing the particular BDC or a set of BDCs for which the teach-in commands apply. A maximum of 128 BDCs can be addressed.

12.3 Parameter 2: "Teach-in Command"

12.3.1 General

The parameter "Teach-in Command" allows teaching of a teachpoint (TP) or controlling of the teach-in procedure. Manufacturer/vendor specific extensions are possible. The commands of the FunctionClass [0x8004] are described within the context of a possible application in the subsequent clauses.

12.3.2 "Single Value Teach"

A threshold is defined by one "Teachpoint" (TP). The teach-in procedure is "static", which means, the measurement value is constant during the teach-in procedure.

The associated commands "0x41" and "0x42" are specified in Table 14.

Figure 23 illustrates an example for "Single Value Teach" in "Single Point Mode".

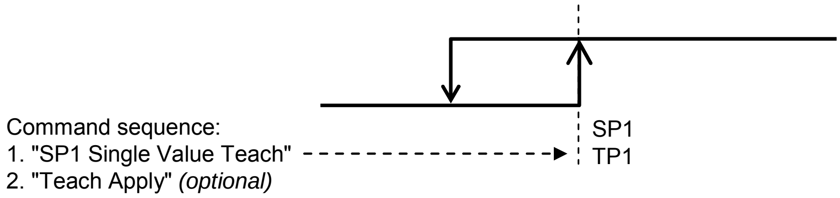

Figure 23 – "Single Value Teach" (Single Point Mode)

Figure 24 illustrates an example for "Single Value Teach" in "Window Mode".

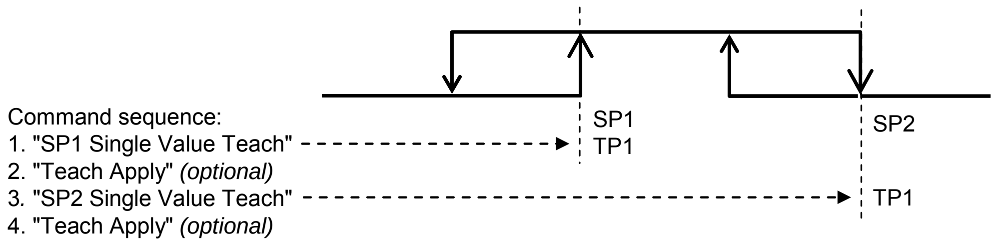

Figure 24 – "Single Value Teach" (Window Mode)

12.3.3 "Two Value Teach"

A threshold is defined by two "Teachpoints" (TP). The teach-in procedure is "static", which means, the measurement value is constant during the teach-in procedure.

The associated commands "0x43" to "0x46" are specified in Table 14.

NOTE The calculation method to determine SP from TP1 and TP2 is manufacturer/vendor specific.

Figure 25 illustrates an example for "Two Value Teach" in "Single Point Mode".

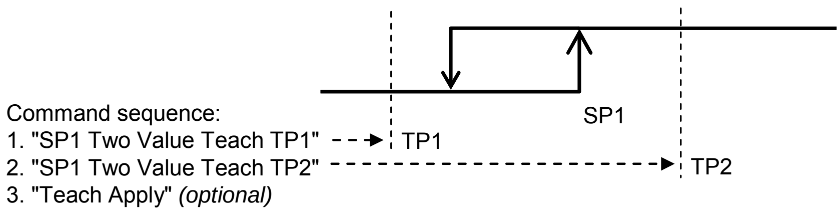

Figure 25 – "Two Value Teach" (Single Point Mode)

Figure 26 illustrates an example for "Two Value Teach" in "Two Point Mode".

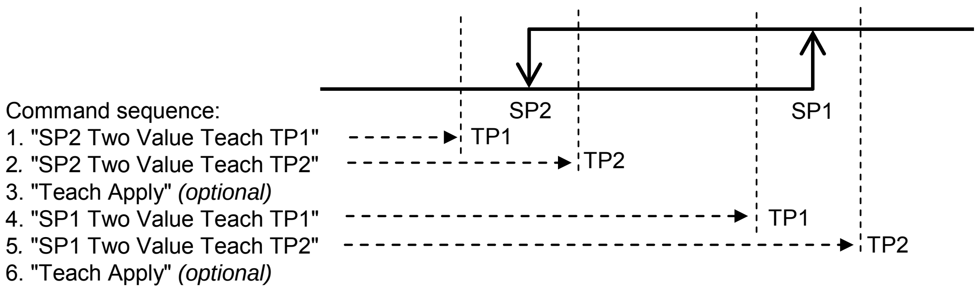

Figure 26 – "Two Value Teach" (Two Point Mode)

12.3.4 "Dynamic Teach" (within a time period)

One single threshold or both thresholds of a BDC are set-up via captured measurement values during a certain period of time. The teach-in procedure is "dynamic", which means, the measurement value is not constant during the teach-in procedure. Usually, the minimum and maximum values within this time frame are taken to define the thresholds. The associated commands "0x47" to "0x4A" are specified in Table 14 .

Figure 27 illustrates an example for "Dynamic Teach" in "Window Mode", where commands "0x47" and "0x4A" are used for the determination of both Setpoints SP1 and SP2 (see NOTE in Table 14). It is the responsibility of the manufacturer to describe the required commands for the "Dynamic Teach" procedure.

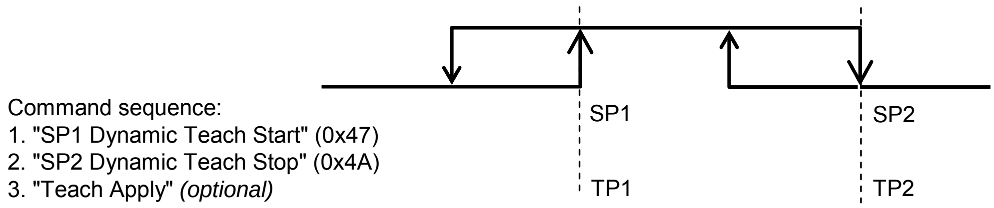

Figure 27 – "Dynamic Teach" (Window Mode)

Figure 28 illustrates an example for "Dynamic Teach" in "Two Point Mode", where commands "0x47" and "0x4A" are used for the determination of both Setpoints SP1 and SP2 (see NOTE in Table 14).

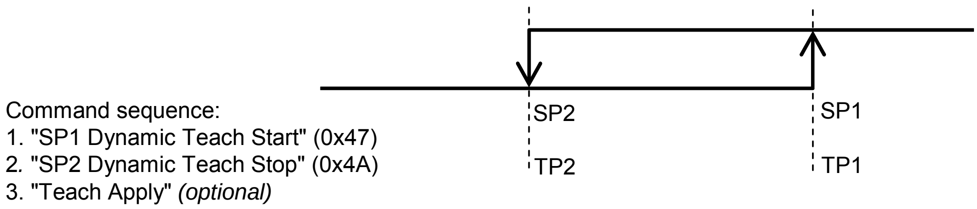

Figure 28 – "Dynamic Teach" (Two Point Mode)

12.3.5 "Teach Apply"

The command "Teach Apply" can be used optionally to terminate the teach-in procedure with the calculation of the thresholds. In this case, the thresholds will be accepted only after "Teach Apply".

12.3.6 "Teach Cancel"

The command "Teach Cancel" can be used to cancel the teach-in procedure without calculation of the thresholds. In this case, the previously taught thresholds will be established.

12.4 Parameter 3: "Teach-in Status"

12.4.1 Status types

The parameter "Teach-in Status" provides feedback on the status and the results of the teach-in activities. This status information is split into "Teach State" and "Teach Flags" (Figure 30).

The following "Teach States" are defined:

- IDLE
- BUSY
- WAIT FOR COMMAND
- SPxSUCCESS
- ERROR

See Table 12 for definitions of these status types reported via the "Teach-in Status" parameter, and Table 13 for the mapping of this parameter. The reported status information can be extended manufacturer/vendor specific.

In order to differentiate the teach-in status information "Teach Flags" are available, only indicating the result of the Teachpoint (TP) capture (Figure 30 and Table 17).

12.4.2 Teach-in dynamics

Figure 29 shows a state machine for the common teach-in procedure.

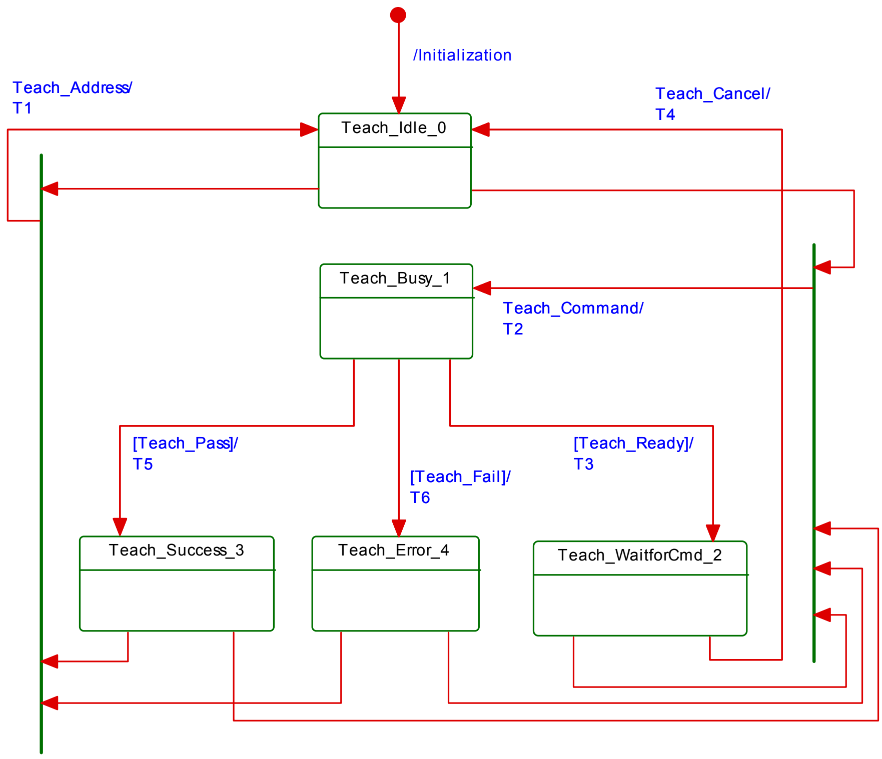

Figure 29 – State machine of the common teach-in procedure

A taken state depends on the received particular teach-in command. Thus, a reported "Teach-in Status" depends on the actual state of the state machine for the teach-in procedure.

Table 12 shows the state transition tables of the teach-in procedure.

Table 12 – State transition tables of the teach-in procedure

STATE NAME	STATE DESCRIPTION
Teach_Idle_0	In this state the Device is waiting for a new teach-in channel address or a teach-in command. The Device operates with the initial or last valid Setpoint settings for the selected teach-in channel.
Teach_Busy_1	In this state the acquisition of Teachpoint values and/or calculation of Setpoint values take place. The state is left on a ready signal of either of these actions. Depending on Device implementation acquisition of Teachpoints and calculation of Setpoints may be executed in one single sequence, without requirement for further teach-in commands.
Teach_WaitForCmd_2	In this state the Device is waiting for a new teach-in command. The state is left on receiving any valid teach-in command or a teach-in cancel command.
Teach_Success_3	In this state the Device operates with the newly acquired and calculated Setpoint values for the selected teach-in channel. The state is left on receiving a new teach-in channel address or a teach-in command
Teach_Error_4	In this state the Device operates with the last valid Setpoint settings for the selected teach-in channel.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0,3,4	0	The teach-in channel address is set to the selected value. The Teach Flags are reset. The reported Teach State is "IDLE".
T2	0,2,3,4	1	The acquisition of a single or several Teachpoints is started for the selected teach-in channel. The reported Teach State is "BUSY".
T3	1	2	The acquisition of a single or several Teachpoints is ready and the Device requires further teach-in commands. The Teach Flags for the acquired Teachpoints are set. The reported Teach State is "WAIT FOR COMMAND".
T4	2	0	Teach Flags are reset. The last valid Setpoint settings are restored. The reported Teach State is "IDLE".
T5	1	3	Teach Flags are reset. The new set point values are activated. The reported Teach State is "SP1SUCCESS", "SP2SUCCESS" or "SP12SUCCESS", depending on the already executed Setpoint calculations since selection of the teach-in channel.
T6	1	4	Teach Flags are reset. The last valid set point values are restored. The reported Teach State is "ERROR".
Initialization	-	0	The teach-in channel address is initialized with the default value. Teach Flags are reset. The reported Teach State is "IDLE".

INTERNAL ITEMS	TYPE	DEFINITION
Teach Flags	-	See Figure 30
Teach State	-	See Figure 30
Teach_Pass	-	Setpoint successfully calculated from Teachpoints
Teach_Fail	-	Teachpoints inconsistent or Setpoint calculation impossible
Teach_Ready	-	A single teach-in action terminated

12.5 Mapping to SDCI communication

Table 13 shows, how the "Teach-in Command", the "Teach-in Channel", and "Teach-in Status" parameters are mapped into the SDCI Index space. The SystemCommand parameter is used as a vehicle to convey the "Teach-in Commands". The table references the individual coding tables Table 14, Table 15, Table 16, and Table 17.

Table 13 – Teach-in related parameter objects (Index)

Index (dec)	Object name	Access	Length	Data type	Comment
0x0002 (2)	SystemCommand	Write		UIntegerT (8)	See Table 14
0x003A (58)	Teach-In Channel	Read/ Write		UIntegerT (8)	See Table 15
0x003B (59)	Teach-In Status	Read		UIntegerT (8)	See Figure 30, Table 16, and Table 17

Table 14 shows the "Teach-in Command" coding. These commands are transmitted using the SystemCommand parameter.

Table 14 – "Teach-in Command" coding

Teach-in Command	Value	Comment
Teach Apply	0x40	Calculate and apply SP1,2 from Teachpoint(s)
SP1 Single Value Teach	0x41	Determine Teachpoint 1 for Setpoint1
SP2 Single Value Teach	0x42	Determine Teachpoint 1 for Setpoint2
SP1 Two Value Teach TP1	0x43	Determine Teachpoint 1 for Setpoint1
SP1 Two Value Teach TP2	0x44	Determine Teachpoint 2 for Setpoint1
SP2 Two Value Teach TP1	0x45	Determine Teachpoint 1 for Setpoint2
SP2 Two Value Teach TP2	0x46	Determine Teachpoint 2 for Setpoint2
SP1 Dynamic Teach Start	0x47	Start dynamic teach-in for Setpoint1 NOTE
SP1 Dynamic Teach Stop	0x48	Stop dynamic teach-in for Setpoint1
SP2 Dynamic Teach Start	0x49	Start Dynamic teach-in for Setpoint2
SP2 Dynamic Teach Stop	0x4A	Stop Dynamic teach-in for Setpoint2 NOTE
Manufacturer Teach	0x4B to 0x4E	For manufacturer specific use
Teach Cancel	0x4F	Abort Teach-in sequence
NOTE These commands shall be applied for the determination of both Setpoints SP1 and SP2 in one single teach-in procedure		

Table 15 shows the "Teach-in Channel" coding into Index 0x003A (reserved for profiles).

Table 15 – "Teach-in Channel" coding

Teach-in Channel	Definition
0	Address of the manufacturer/vendor specific pre-defined (default) BDC
1 to 128	Address of the BDC1 to BDC128
129 to 191	Reserved
192-254	Different manufacturer/vendor specific BDC sets
255	Addressing of all implemented BDCs

Figure 30 defines the data structure of the "Teach Flags" and the "Teach State" to be used in the "Teach-in Status" coding in Table 13.

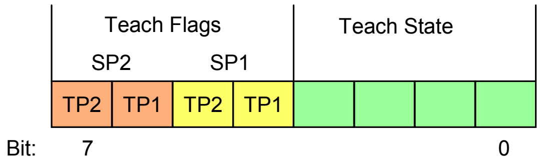

Figure 30 – Structure of the "Teach Flags" and the "Teach State"

Table 16 shows the "Teach State" coding.

Table 16 – "Teach State" coding

Teach State	Definition
0	IDLE
1	SP1 SUCCESS
2	SP2 SUCCESS
3	SP12 SUCCESS
4	WAIT FOR COMMAND
5	BUSY
6	RESERVED
7	ERROR
8 to 11	Reserved
12 to 15	Manufacturer/vendor specific

Table 17 shows the "Teach Flag" coding.

Table 17 – "Teach Flag" coding

Teach Flags	Comment
0	Teachpoint x not taught or not successful
1	Teachpoint x successfully taught

Annex A
(normative)
Profile testing and conformity

A.1 General

A.1.1 Overview

It is the responsibility of the vendor/manufacturer of a Smart Sensor profile Device to perform a conformity testing and to provide a document similar to the manufacturer declaration defined in [1] or based on the template downloadable from the IO-Link website (www.io-link.com).

A.1.2 Issues for testing/checking

- Identification complete and correct?
- Descriptors available and correct?
- All rules observed?
- Switching behavior conform to the specification?
- FunctionClasses available and correct?
  - Indices available and correct?
  - Read/write correct?
  - Data structures: Record? Value ranges?
  - Behavior of the FunctionClass conforms to the specification?
- Extract BDCs (switching functions) from user manual or IODD and check conformity with the specification
- Checklist: checkbox "relevant" and checkbox "verified"
- IODD: see [6]

Annex B
(informative)
Information on conformity testing of profile Devices

Information about testing profile Devices for conformity with this document can be obtained from the following organization:

IO-Link Consortium
Haid-und-Neu-Str. 7
76131 Karlsruhe
Germany
Phone: +49 (0) 721 / 96 58 590
Fax: +49 (0) 721 / 96 58 589
E-mail: info@io-link.com
Web site: http://www.io-link.com

Bibliography

[1] IO-Link Consortium, *IO-Link Interface and System,* V1.1, November 2010, Order No. 10.002 or IEC 61131-9, *Programmable controllers – Part 9: Single-drop digital communication interface for small sensors and actuators (SDCI)*

[2] IO-Link Consortium, *IO Device Description (IODD),* V1.1, July 2011, Order No. 10.012

[3] IEC/TR 62390:2005, *Common automation device profile guideline*

[4] IEC 60050 (all parts), *International Electrotechnical Vocabulary*

NOTE See also the IEC Multilingual Dictionary – Electricity, Electronics and Telecommunications (available on CD-ROM and at <http://domino.iec.ch/iev>).

[5] IO-Link Consortium, *IO-Link Communication,* V1.0, January 2009, Order No. 10.002

[6] IO-Link Consortium, *IO-Link Test Specification,* V1.1, May 2011, Order No. 10.032

___

IO-Link Consortium
Haid-und-Neu-Str. 7
76131 Karlsruhe
Germany
Phone: +49 (0) 721 / 96 58 590
Fax: +49 (0) 721 / 96 58 589
e-mail: info@io-link.com
http://www.io-link.com/

IO-Link