

# **IO-Link**

# **Smart Sensor Profile**

Draft  
**Specification**  
(Not for implementation)

**Version 1.0.2**  
**July 2011**

**Order No: 10.042**

File name: **IOL-Smart-Sensor-Profile-Spec\_10042\_V102\_Jul11.doc**

Prepared by the IO-Link Smart Sensor profile group for review within the IO-Link consortium and in PI until **September 16<sup>th</sup>, 2011**.

Any comments, proposals, requests on this document are appreciated. Please use [www.io-link-projects.com](http://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](http://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](http://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](http://www.io-link.com).


**Disclaimer:**

The attention of adopters is directed to the possibility that compliance with or adoption of IO-Link Consortium specifications may require use of an invention covered by patent rights. The IO-Link Consortium shall not be responsible for identifying patents for which a license may be required by any IO-Link Consortium specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. IO-Link Consortium specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

The information contained in this document is subject to change without notice. The material in this document details an IO-Link Consortium specification in accordance with the license and notices set forth on this page. This document does not represent a commitment to implement any portion of this specification in any company's products.

WHILE THE INFORMATION IN THIS PUBLICATION IS BELIEVED TO BE ACCURATE, THE IO-LINK CONSORTIUM MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL INCLUDING, BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR PARTICULAR PURPOSE OR USE.

In no event shall the IO-Link Consortium be liable for errors contained herein or for indirect, incidental, special, consequential, reliance or cover damages, including loss of profits, revenue, data or use, incurred by any user or any third party. Compliance with this specification does not absolve manufacturers of IO-Link equipment, from the requirements of safety and regulatory agencies (TÜV, BIA, UL, CSA, etc.).

 **IO-Link** ® is registered trade mark. The use is restricted for members of the IO-Link Consortium. More detailed terms for the use can be found in the IO-Link Consortium Rules on [www.io-link.com](http://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](mailto:info@io-link.com)

Web site: [www.io-link.com](http://www.io-link.com)

© No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

## CONTENTS

0	Introduction .....	6
0.1	General .....	6
0.2	Patent declaration .....	6
1	Scope .....	7
2	Normative references .....	7
3	Terms, definitions, symbols, abbreviated terms and conventions .....	7
3.1	Terms and definitions .....	7
3.2	Symbols and abbreviated terms .....	10
3.3	Conventions .....	10
3.3.1	Behavioral descriptions .....	10
3.3.2	Memory and transmission octet order .....	10
4	Overview of sensor devices .....	11
4.1	Smart Sensors .....	11
4.2	Sensors migrating to SDCI .....	11
5	Device profiles related to IEC 61131-9 .....	11
5.1	SDCI technology specified in IEC 61131-9 .....	11
5.2	Profile classification .....	12
5.3	Profile related Index space .....	13
5.4	Profile characteristics .....	13
5.5	User benefits .....	14
6	Smart Sensor profile .....	15
6.1	Objectives for the Smart Sensor profile .....	15
6.2	Measurement categories for Smart Sensors .....	15
6.3	Smart Sensor model .....	16
6.4	Smart Sensor object model .....	17
6.5	How to use the Smart Sensor profile .....	18
7	Process Data mapping (PD) .....	19
7.1	Process Data and its transmission .....	19
7.2	Process variable descriptors (PVinD, PVoutD) .....	20
7.2.1	Coding .....	20
7.2.2	PDInputDescriptor .....	20
7.3	Profile specific PD structures .....	21
7.3.1	General .....	21
7.3.2	One or more BDCs (recommended) .....	21
7.3.3	One PDV .....	21
7.3.4	PD lengths up to two octets .....	21
7.3.5	PD lengths larger than two octets .....	22
8	Device identification objects [0x8000] .....	23
9	BinaryDataChannel [0x8001] .....	24
9.1	Characteristic of the BDC .....	24
9.2	Configuration and parameterization of the BDC .....	24
9.2.1	General .....	24
9.2.2	Switchpoint logic .....	24
9.2.3	Switchpoint hysteresis .....	24

9.2.4	Switchpoint mode.....	24
9.2.5	Setpoint parameters (SP1, SP2).....	27
9.2.6	Setpoint and Switchpoint parameter coding .....	27
9.3	BDC mapping .....	28
9.3.1	Concepts .....	28
9.3.2	BDC Index space .....	28
9.3.3	Access behavior of not supported Subindices .....	29
10	ProcessDataVariable [0x8002].....	29
10.1	Scaling and dimensions .....	29
10.2	Recommended PDV representation.....	30
11	Diagnosis [0x8003].....	31
11.1	DeviceStatus and DetailedDeviceStatus.....	31
11.2	Smart Sensor EventCodes .....	31
12	TeachChannel [0x8004].....	32
12.1	Teach-in concepts for Smart Sensors.....	32
12.2	Parameter 1: "Teach-in Channel" .....	32
12.3	Parameter 2: "Teach-in Command" .....	32
12.3.1	General .....	32
12.3.2	"Single Value Teach" .....	32
12.3.3	"Two Value Teach".....	33
12.3.4	"Dynamic Teach" (within a time period).....	34
12.3.5	"Teach Apply" .....	34
12.3.6	"Teach Cancel" .....	35
12.4	Parameter 3: "Teach-in Status" .....	35
12.4.1	Status types.....	35
12.4.2	Teach-in dynamics.....	35
12.5	Mapping to SDCI communication .....	36
Annex A (normative)	Profile testing and conformity .....	39
A.1	General.....	39
A.1.1	Overview .....	39
A.1.2	Issues for testing/checking.....	39
Annex B (informative)	Information on conformity testing of profile Devices .....	40
Figure 1	– Memory and transmission octet order.....	11
Figure 2	– Domain of the SDCI technology within the automation hierarchy.....	12
Figure 3	– Overview of SDCI technologies and profiles.....	12
Figure 4	– Indication rules for ProfileIdentifiers.....	14
Figure 5	– Overview of FunctionClasses.....	16
Figure 6	– Smart Sensor object model.....	17
Figure 7	– Example for a pressure sensor .....	18
Figure 8	– Example PDinput data stream.....	19
Figure 9	– Recommended data structure for pure BDCs.....	21
Figure 10	– Recommended data structure for one PDV.....	21
Figure 11	– Recommended data structure for a PDV and up to two BDCs .....	22
Figure 12	– Recommended data structure for a PDV, BDCs, and auxiliary variables.....	22
Figure 13	– Recommended data structure for multi PDVs and zero or more BDCs.....	23

Figure 14 – Presence detection.....	25
Figure 15 – Example of a Single Point Mode for quantity detection .....	25
Figure 16 – Example for the Window Mode.....	26
Figure 17 – Example for the Two Point Mode of presence detection.....	26
Figure 18 – Example for the Two Point Mode of quantity detection.....	26
Figure 19 – Value to quantity conversion via linear equation .....	29
Figure 20 – Conversion examples .....	30
Figure 21 – Relationship between a dimensioned PDV and its PLC variable.....	30
Figure 22 – Example PLC program for a measurement value conversion.....	31
Figure 23 – "Single Value Teach" (Single Point Mode).....	33
Figure 24 – "Single Value Teach" (Window Mode) .....	33
Figure 25 – "Two Value Teach" (Single Point Mode).....	33
Figure 26 – "Two Value Teach" (Two Point Mode) .....	34
Figure 27 – "Dynamic Teach" (Window Mode) .....	34
Figure 28 – "Dynamic Teach" (Two Point Mode) .....	34
Figure 29 – State machine of the common teach-in procedure .....	35
Figure 30 – Structure of the "Teach Flags" and the "Teach State" .....	38
Table 1 – Excerpt of the SDCI Indices related to profiles .....	13
Table 2 – Coding of ProfileIdentifiers (PID).....	14
Table 3 – Example of the profile identification for a Smart Sensor.....	14
Table 4 – Typical physical and chemical measurement quantities .....	15
Table 5 – Abstract notation for BDC and PDV access of a PLC client.....	18
Table 6 – FunctionClass combinations for different sensor types .....	19
Table 7 – Coding of PVinD or PVoutD .....	20
Table 8 – Example "PDInputDescriptor".....	20
Table 9 – Deviating definitions for identification data objects .....	23
Table 10 – Setpoint and Switchpoint parameter coding.....	27
Table 11 – Index space for BDC1 and BDC2 .....	28
Table 12 – State transition tables of the teach-in procedure.....	36
Table 13 – Teach-in related parameter objects (Index) .....	37
Table 14 – "Teach-in Command" coding .....	37
Table 15 – "Teach-in Channel" coding.....	37
Table 16 – "Teach State" coding .....	38
Table 17 – "Teach Flag" coding.....	38

## **0 Introduction**

### **0.1 General**

The Single-drop Digital Communication Interface (SDCI) and system technology (IO-Link™<sup>1</sup>) 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.

---

<sup>1</sup> IO-Link™ is a trade name of the "IO-Link Consortium". This information is given for the convenience of users of this international Standard and does not constitute an endorsement by IEC of the trade name holder or any of its products. Compliance to this standard 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

- 44 **3.1.3**  
45 **coded switching**  
46 SDCI communication, based on the standard binary signal levels of IEC 61131-2
- 47 **3.1.4**  
48 **communication channel**  
49 logical connection between Master and Device
- 50 NOTE Four communication channels are defined: process channel, page and ISDU channel (for parameters) and  
51 diagnosis channel.
- 52 **3.1.5**  
53 **Device**  
54 single passive peer to a Master such as a sensor or actuator
- 55 NOTE Uppercase "Device" is used for SDCI equipment, while lowercase "device" is used in a generic manner.
- 56 **3.1.6**  
57 **event**  
58 an instance of a change of conditions
- 59 NOTE An event is indicated via the event flag within the Device's status cyclic information, then acyclic transfer of  
60 event data (typically diagnosis information) is conveyed through the diagnosis communication channel.  
61 [IEC 61158-5-x, modified]
- 62 **3.1.7**  
63 **FunctionClass**  
64 particular function within a Device profile
- 65 NOTE A profile Device can use one or several FunctionClasses one or several times.
- 66 **3.1.8**  
67 **ISDU**  
68 indexed service data unit used for acyclic acknowledged transmission of parameters that can  
69 be segmented in a number of F-sequences
- 70 **3.1.9**  
71 **Master**  
72 active peer connected through ports to one up to n Devices and which provides an interface  
73 to the gateway to the upper level communication systems or PLCs
- 74 NOTE Uppercase "Master" is used for SDCI equipment, while lowercase "master" is used in a generic manner.
- 75 **3.1.10**  
76 **on-request data (OD)**  
77 acyclically transmitted data upon request of the Master application consisting of parameters  
78 or event data
- 79 **3.1.11**  
80 **port**  
81 communication medium interface of the Master to one Device
- 82 **3.1.12**  
83 **process data (PD)**  
84 input or output values from or to a discrete or continuous automation process cyclically trans-  
85 ferred with high priority and in a configured schedule automatically after start-up of a Master
- 86 **3.1.13**  
87 **ProcessDataVariable (PDV)**  
88 representation of a measurement value

- 89 **3.1.14**  
90 **process variable input descriptor (PViND)**  
91 descriptor for position and offset of variables within the Process (input) Data entity
- 92 **3.1.15**  
93 **process variable output descriptor (PVoutD)**  
94 descriptor for position and offset of variables within the Process (output) Data entity
- 95 **3.1.16**  
96 **ProfileIdentifier**  
97 list of supported profiles and function classes
- 98 **3.1.17**  
99 **Setpoint**  
100 threshold measurement value of a sensor for the rising or falling edge of a binary output sig-  
101 nal
- 102 **3.1.18**  
103 **single point mode**  
104 evaluation method with one single Setpoint where the binary output signal changes whenever  
105 the sensor signal passes above or below this Setpoint
- 106 **3.1.19**  
107 **SIO**  
108 port operation mode in accordance with digital input and output defined in IEC 61131-2 that is  
109 established after power-up or fallback or unsuccessful communication attempts
- 110 **3.1.20**  
111 **switching mode**  
112 one out of a set of possible operational modes for binary signals such as 'deactivated', 'Single  
113 Point Mode', 'Window Mode', or 'Two Point Mode'
- 114 NOTE Vendor specific modes are possible.
- 115 **3.1.21**  
116 **switching signal**  
117 binary signal from or to a Device when in SIO mode (as opposed to the "coded switching"  
118 SDCI communication)
- 119 **3.1.22**  
120 **switchpoint**  
121 measurement value of a sensor where the binary output signal changes its value
- 122 **3.1.23**  
123 **Teach Flag**  
124 indication for the success of a Teachpoint setting
- 125 **3.1.24**  
126 **Teachpoint**  
127 trigger to set a threshold value or the border value of a range
- 128 **3.1.25**  
129 **Teach State**  
130 indication of the current state of the teach-in procedure

131 **3.1.26**  
132 **two point mode**  
133 evaluation method with two Setpoints where the binary output signal only changes if the sen-  
134 sor measurement value comes from above the highest Setpoint and passes the lowest Set-  
135 point or if it comes from below the lowest Setpoint and passes the highest Setpoint

136 **3.1.27**  
137 **wake-up**  
138 procedure for causing a Device to change its mode from SIO to COMx

139 **3.1.28**  
140 **window mode**  
141 evaluation with two Setpoints where the binary output signal depends on the measurement  
142 value of the sensor being between the Setpoints or either above the highest or below the low-  
143 est Setpoint

144

## 145 **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)	

146

## 147 **3.3 Conventions**

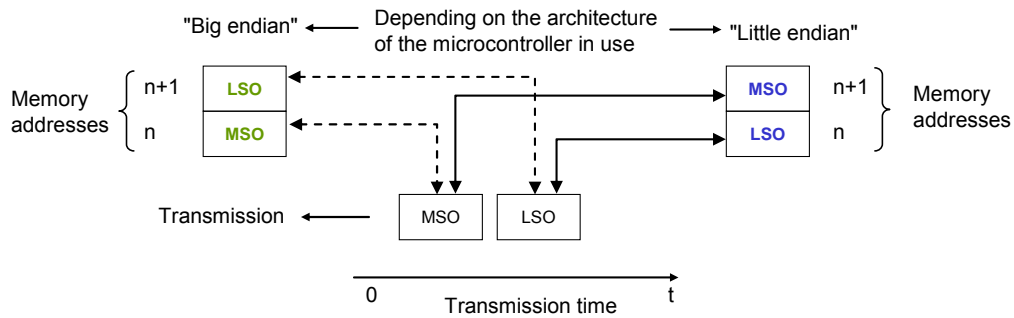
### 148 **3.3.1 Behavioral descriptions**

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

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

### 153 **3.3.2 Memory and transmission octet order**

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



156

157

**Figure 1 – Memory and transmission octet order**

## 158 4 Overview of sensor devices

### 159 4.1 Smart Sensors

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

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

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

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

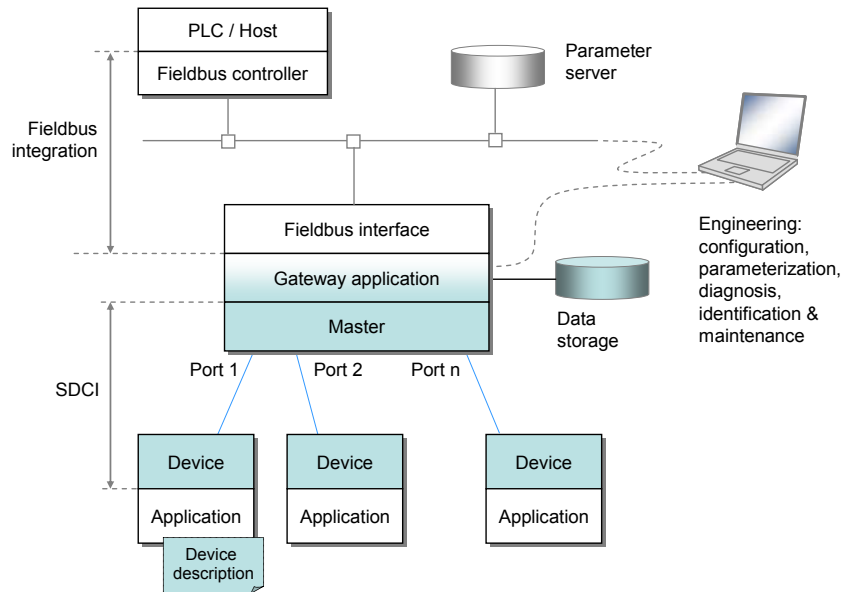
### 174 4.2 Sensors migrating to SDCI

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

## 181 5 Device profiles related to IEC 61131-9

### 182 5.1 SDCI technology specified in IEC 61131-9

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



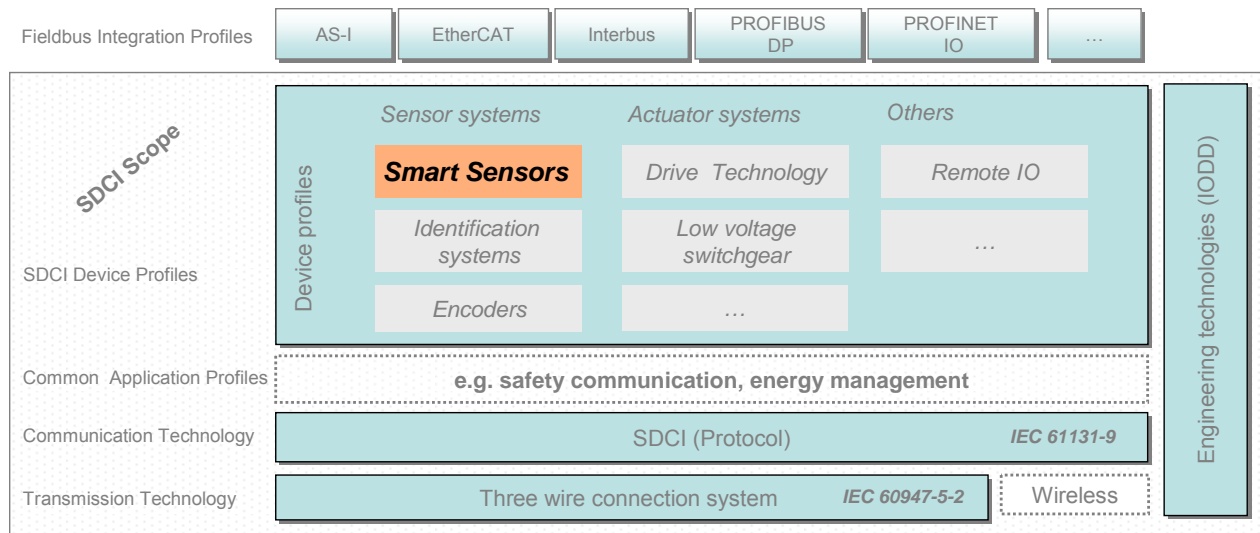
184

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

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

190 **5.2 Profile classification**

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



192

193 **Figure 3 – Overview of SDCI technologies and profiles**

194 The "SDCI Device Profiles" represent specifications of common functionality of particular De-  
 195 vice type families/classes such as drives, low voltage switch gears, encoders, etc. These pro-  
 196 files primarily focus on the structure and behavior of the Device technology and secondarily  
 197 on the data mapping on SDCI. Thus, the user recognizes a "generic" Device to a certain ex-  
 198 tent even when he switches from one brand to another.

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

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

### 205 5.3 Profile related Index space

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

210 **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 Characteristic	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	O	See Table 9
...						
0x0017 (23)	Firmware Revision	R	Max. 64 octets	StringT	O	See Table 9
0x0018 (24)	Application Specific Tag	R/W	Min. 16, max. 32 octets	StringT	O	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						

211

### 212 5.4 Profile characteristics

213 All SDCI Device Profiles shall be characterized within the parameter object "Profile Character-  
 214 istic" in Index 0x000D via ProfileIdentifiers (PID) listed within a record. Normally, an SDCI de-  
 215 vice supports only one SDCI Device Profile (e.g. this Smart Sensor Profile). It is also possible  
 216 for an SDCI device to support several Common Application Profiles as well as several Func-

217 tionClasses (see 6.4). FunctionClasses defined in this profile can also be inherited to other  
218 SDCI device profiles, for example to SDCI actuators.

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

- 221 • DeviceProfileID
- 222 • CommonApplicationProfileID
- 223 • FunctionClassID

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

226 **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

227 The following rules apply:

- 228 1) Whenever a Device profile is supported such as for example "Smart Sensors", it shall be indicated via a De-  
DeviceProfileID entry
- 229 2) Whenever 1 to n common application profiles are supported, they shall be indicated via 1 to n CommonAp-  
ApplicationProfileIDs
- 230 3) Whenever 1 to n functions are supported, they shall be indicated via 1 to n FunctionClassIDs
- 231 4) The IDs shall be listed in ascending order (DeviceProfileIDs → CommonApplicationProfileIDs → Function-  
ClassIDs)

232 **Figure 4 – Indication rules for ProfileIdentifiers**

233 The different profile identifiers shall be ordered within the array of the parameter object "Pro-  
234 file Characteristic" in a sequence shown in Table 2 using the SDCI Subindices as a reference.

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

236 **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

237

## 238 5.5 User benefits

239 As already mentioned in 5.2 the user recognizes from the Masters point of view a "generic"  
240 Device through the communication interface even when he switches from one brand to an-  
241 other. The exchange of process data and the behavior of the profile Device are the same, at  
242 least for "basic" functions. That means he is not forced to change his user program within the  
243 controller (PLC) in this case and he can expect the same basic behavior of the Device (for

244 example process data, diagnosis, and identification). However, due to the objectives for the  
 245 individual Device profiles, the interoperability levels can be different and the compatibility be-  
 246 tween the profile Devices can be partly limited. A good compromise is the possibility of read-  
 247 ing the profile features out of the Device via the PLC program and adjusting the user program  
 248 accordingly. Such a solution is the Smart Sensor profile defined in the following.

249 **6 Smart Sensor profile**

250 **6.1 Objectives for the Smart Sensor profile**

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

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

272

273 **6.2 Measurement categories for Smart Sensors**

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

277 **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

278

279 Smart Sensors are independent from the measurement quantities and represent the meas-  
 280 urement results in a uniform manner

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

### 283 6.3 Smart Sensor model

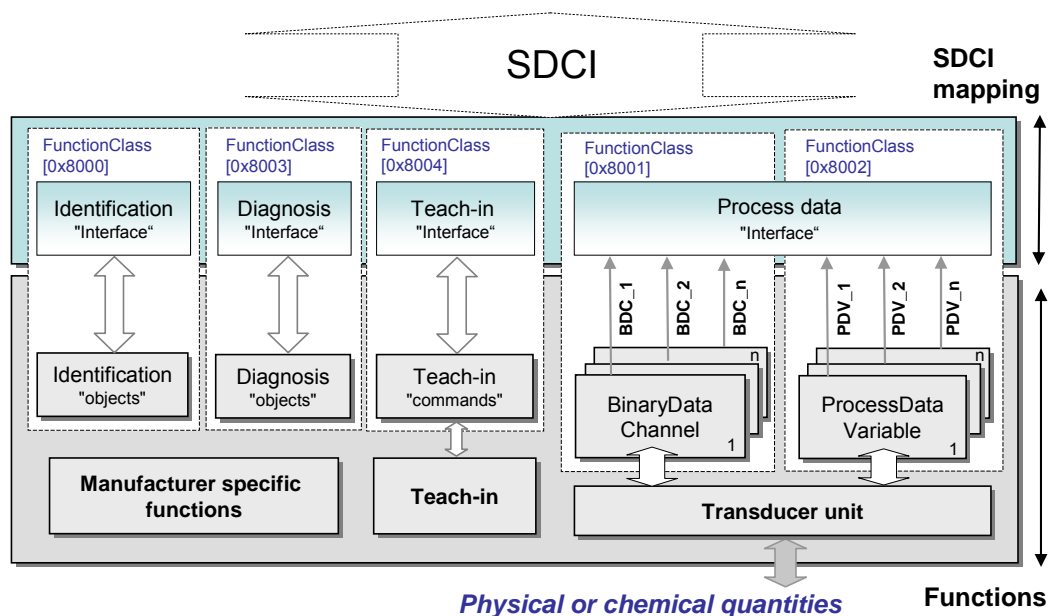
284 The Smart Sensor profile defines a so-called function-driven Device model instead of for ex-  
 285 ample an architectural model. That means it only defines independent and consistent func-  
 286 tions (FunctionClasses). This allows the manufacturers/vendors to create a large variety of  
 287 subsets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

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

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

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

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



296

297

**Figure 5 – Overview of FunctionClasses**

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

301 The *BinaryDataChannel* (FunctionClass [0x8001]) uses the measurement values out of the  
 302 transducer unit and creates binary information (BDC<sub>n</sub>), whenever certain thresholds are  
 303 passed. These thresholds are defined via parameters as defined in clause 9.

304 The *ProcessDataVariables* (FunctionClass [0x8002]) uses the measurement values out of the  
 305 transducer unit and creates data structures (PDV<sub>n</sub>) representing the physical or chemical  
 306 quantity, for example pressure or temperature. These data structures within the Process-  
 307 DataVariables are standardized to a maximum extent as shown in clause 10.

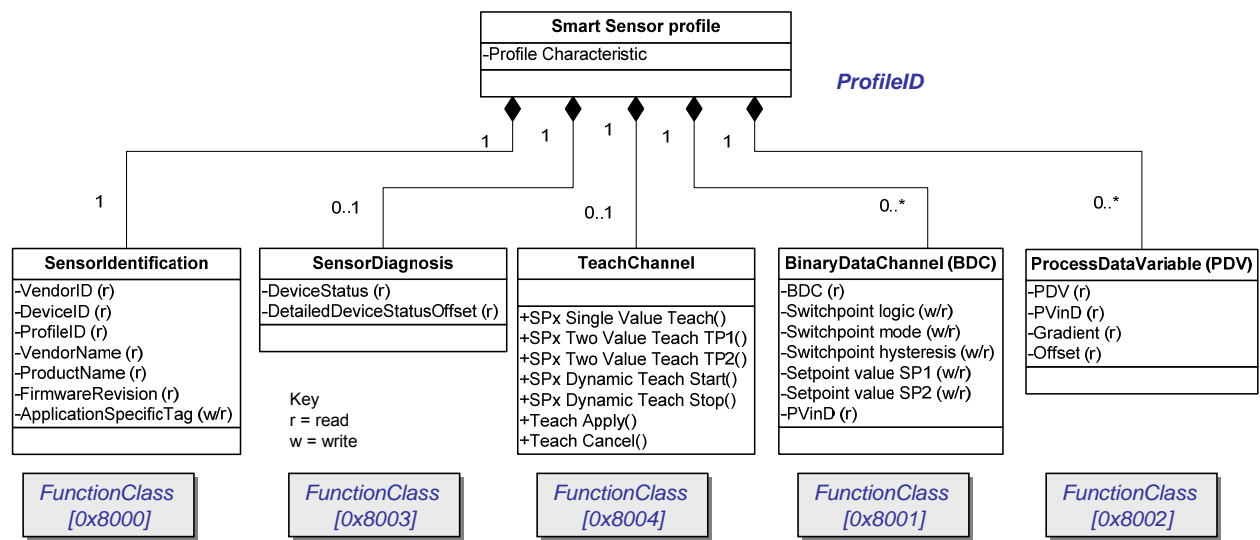
308 The *Device Diagnosis* (FunctionClass [0x8003]) extends the standard SDCI Device diagnosis  
 309 by some additional diagnosis objects. This FunctionClass is mandatory for the Smart Sensor  
 310 profile.

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

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

316 **6.4 Smart Sensor object model**

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

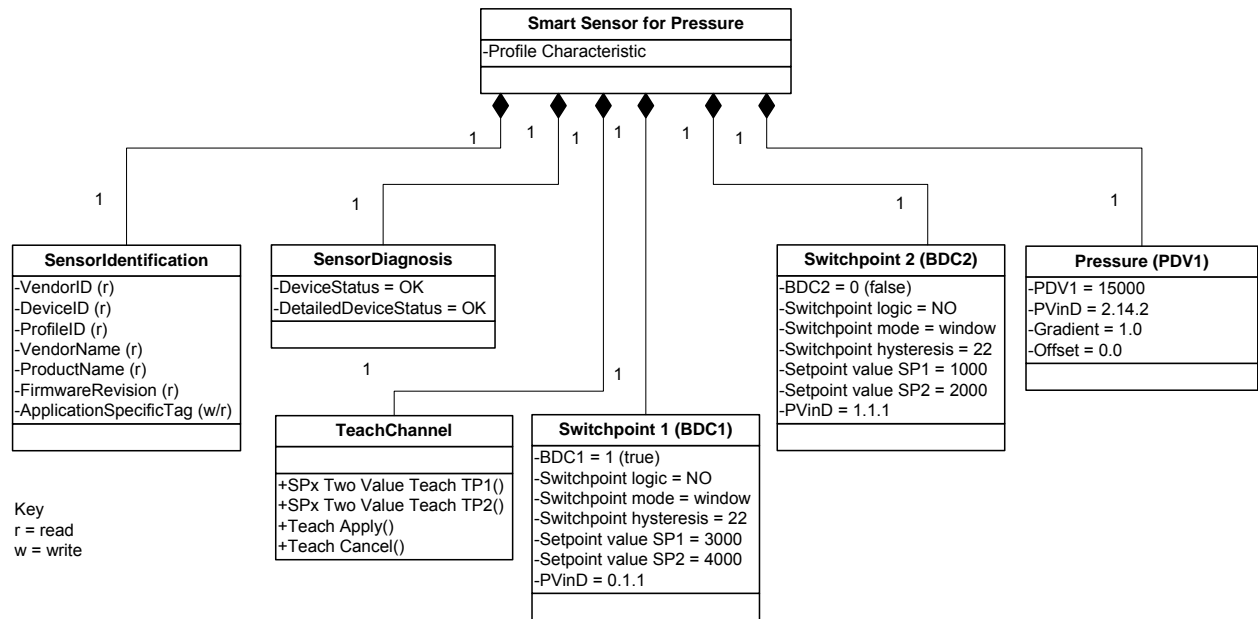


322

323 **Figure 6 – Smart Sensor object model**

324 The classes BDC and PDV can be instantiated 0 to \* (n) times as shown in the example in  
 325 Figure 7, depending on the complexity of the sensor. The FunctionClasses for identification  
 326 and diagnosis are mandatory.

327



328

329

**Figure 7 – Example for a pressure sensor**

330 The example sensor in Figure 7 demonstrates:

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

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

344 **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

345

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

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

349

350 **6.5 How to use the Smart Sensor profile**

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

353

**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					

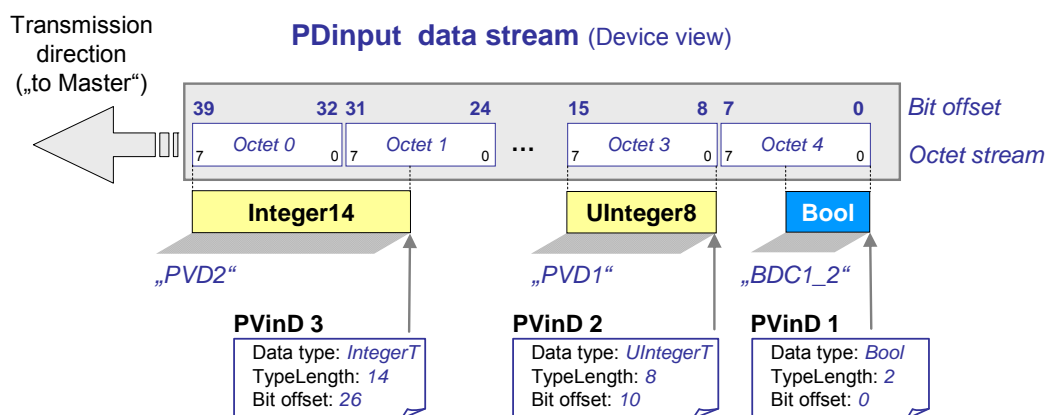
354

355 **7 Process Data mapping (PD)**

356 **7.1 Process Data and its transmission**

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

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



363

364 **Figure 8 – Example PInput data stream**

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

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

374 For Smart Sensors the following information is relevant and it will be specified in the subse-  
 375 quent clauses:

- 376 • The content of the PVD process variable descriptor defining the data type and location  
 377 of BDC and PDV within the data stream
- 378 • The content of the PInputDescriptor describing "what" is to be transmitted and "how"

379

380 **7.2 Process variable descriptors (PVinD, PVoutD)**381 **7.2.1 Coding**

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

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

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

- 389 • the data type (DataType) of the particular process variable. "Set of BoolT" describes here  
 390 combined BinaryDataChannels (BDCs)  
 391 • the length of the data type (TypeLength) in bit, for example 6 for UInteger6  
 392 • the bit offset (Bit offset) as the beginning of the variable in the data stream  
 393 • any manufacturer/vendor specific data structures, which cannot be described via the stan-  
 394 dard BDC or PDV descriptors, are described via a process variable descriptor (e.g. addi-  
 395 tional output data)

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

397

**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

398

399 NOTE The abstract notation in this profile specification of a PVinD is: DataType.TypeLength.Bit\_offset

400 **7.2.2 PDInputDescriptor**

401 Smart Sensor Devices shall use the standard Device parameter "PDInputDescriptor" in Index  
 402 0x000E to store the description information according to Table B.7 in [1]. The user program  
 403 within a controller (e.g. PLC) can thus read this information. The "PDInputDescriptor" contains  
 404 a descriptor (PVinD) for each and every process variable. Exception: Subsequent Boolean  
 405 variables can be described within one PVinD. Table 8 presents an example "PDInputDescrip-  
 406 tor" with two PDVs and two combined BDCs.

407

**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)		

408

409

410 **7.3 Profile specific PD structures**

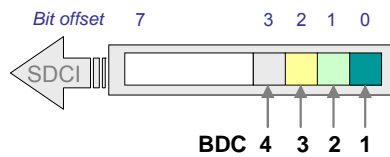
411 **7.3.1 General**

412 In order to avoid a large variety of data structures and descriptors and as a consequence  
 413 complexity, this profile specification specifies and recommends only a few variable descrip-  
 414 tions.

415 **7.3.2 One or more BDCs (recommended)**

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

419 The PViND in this case is: Set of BoolT.4.0



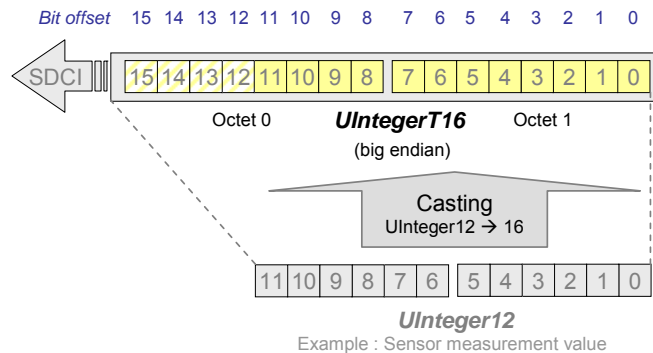
420

421 **Figure 9 – Recommended data structure for pure BDCs**

422 **7.3.3 One PDV**

423 It is highly recommended for Smart Sensors with one PDV to use the data structure demon-  
 424 strated in Figure 10. The example shows, that a Smart Sensor can cast an 8, 10, or 14 bit  
 425 value into a UIntegerT16 data type, thereby using only part of the space. The leading bits  
 426 shall be "0". Variables of type Integer < 16 bit shall also be casted into variables of type Inte-  
 427 gerT16. Type casting rules are specified in [1], Annex E2.3 or E2.4.

428 The PViND in this case is: UIntegerT.16.0



429

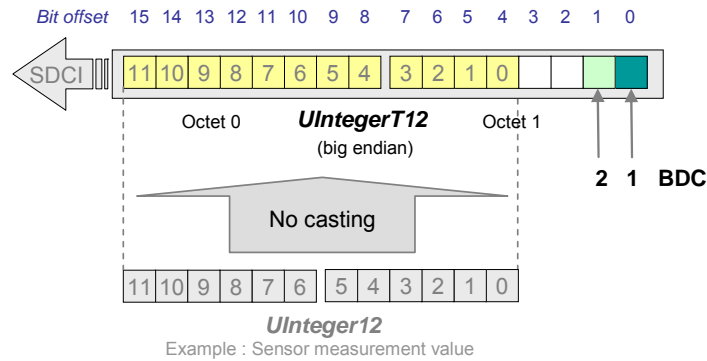
430 **Figure 10 – Recommended data structure for one PDV**

431 **7.3.4 PD lengths up to two octets**

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

435 **7.3.4.1 One PDV and several BDCs**

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



438

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

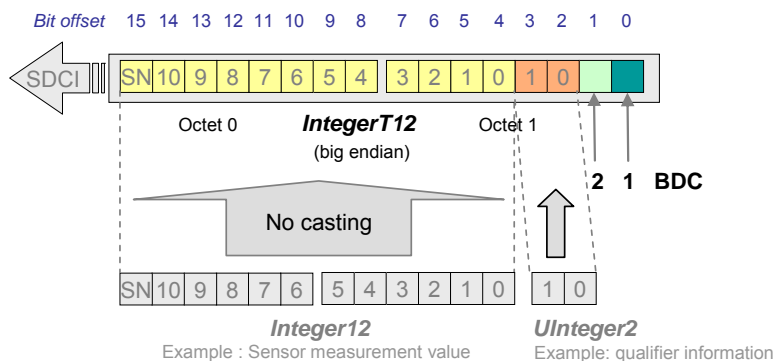
440 The following rules apply:

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

445

446 **7.3.4.2 One PDV, several BDCs, and auxiliary variables**

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



449

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

451

452 The following rules apply:

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

459

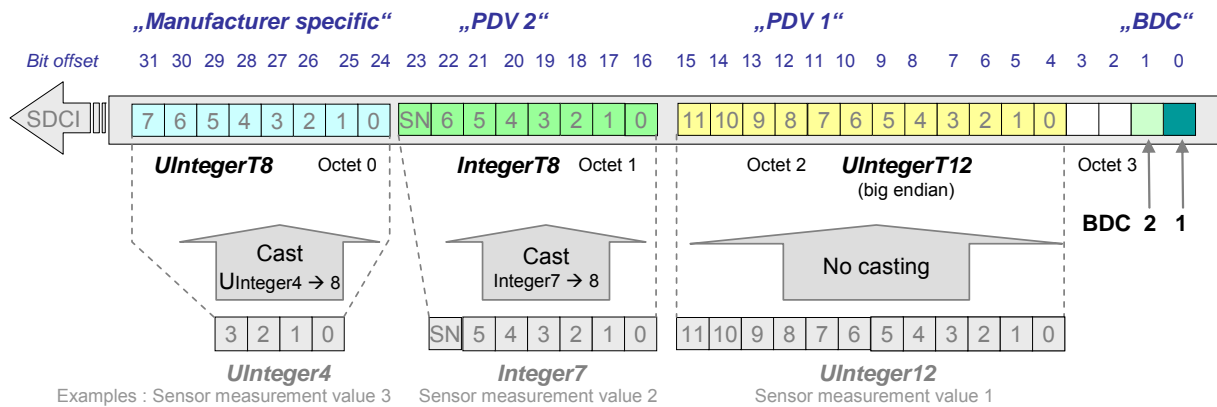
460 **7.3.5 PD lengths larger than two octets**

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

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

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

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



476 Examples : Sensor measurement value 3    Sensor measurement value 2    Sensor measurement value 1

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

- 478 The PViDs in Figure 13 are:
- 479 PViD 1        Set of BoolT.2.0        (BDC2 and BDC1)
  - 480 PViD 2        UInteger.12.4        (PDV1)
  - 481 PViD 3        Integer.8.16        (PDV2)
  - 482 PViD 4        UInteger.8.24        (Manufacturer/vendor specific)

483

484 **8 Device identification objects [0x8000]**

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

486 **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						

## 489 **9 BinaryDataChannel [0x8001]**

### 490 **9.1 Characteristic of the BDC**

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

### 493 **9.2 Configuration and parameterization of the BDC**

#### 494 **9.2.1 General**

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

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

- 498 • Switchpoint logic
- 499 • Switchpoint hysteresis
- 500 • Switchpoint mode
- 501 • Setpoints SP1 and SP2

502 The Switchpoint parameters describe the switching behavior of a BDC. They are defined in  
503 the subsequent clauses.

504 The Setpoint parameters are defined in detail in 9.2.5. The coding of the Setpoint and Switch-  
505 point parameters is specified in 9.2.6.

#### 506 **9.2.2 Switchpoint logic**

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

#### 509 **9.2.3 Switchpoint hysteresis**

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

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

#### 516 **9.2.4 Switchpoint mode**

##### 517 **9.2.4.1 Overview**

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

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

523 The quiescent state of sensors for presence detection (e.g. optical proximity sensors or ultra-  
524 sonic sensors) is a measurement value of "infinite" (in case of non-inverted switching informa-  
525 tion). An approaching object will cause the switching information of the sensor to change at  
526 the setpoint (measurement value). The departing object will cause the switching information of  
527 the sensor to switch back at a larger measurement value than the setpoint (see Figure 14)

528 The quiescent state of sensors for quantity detection (e.g. pressure or temperature sensors)  
 529 is a measurement value of "zero" (in case of non-inverted switching information). An increas-  
 530 ing measurement value will cause the switching information of the sensor to change at the  
 531 setpoint. A decreasing measurement value will cause the switching information of the sensor  
 532 to switch back at a smaller measurement value than the setpoint (see Figure 15).

533 The associated FunctionClass comprises 4 different modes:

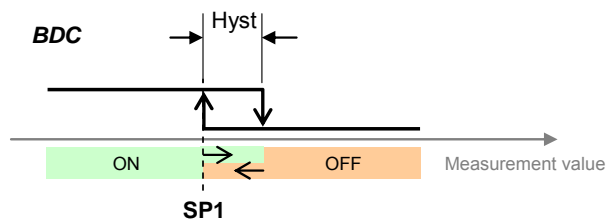
- 534 • Deactivated
- 535 • Single Point Mode
- 536 • Window Mode
- 537 • Two Point Mode

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

#### 543 9.2.4.2 Single Point Mode

544 Figure 14 demonstrates the switching behavior in Single Point Mode. The switching informa-  
 545 tion changes, when the current measurement value passes the threshold defined in Setpoint  
 546 SP1. This change occurs with raising or falling measurement values. If a hysteresis is defined  
 547 for SP1, the switching behavior shall observe the hysteresis as shown in Figure 14. This be-  
 548 havior is typical for "presence detection" of material with none symmetrical hysteresis in re-  
 549 spect to SP1 and not inverted switching.

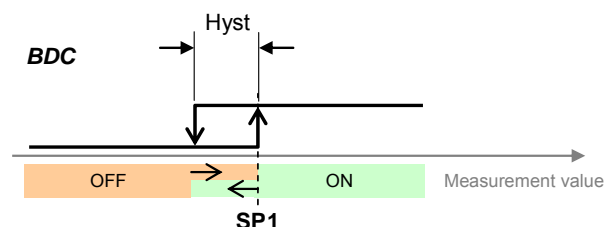
550 The Setpoint SP2 is not relevant for this mode.



551

552 **Figure 14 – Presence detection**

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



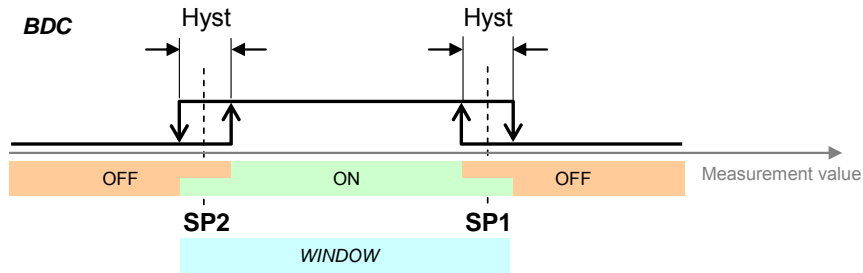
555

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

557 **9.2.4.3 Window Mode**

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

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



564

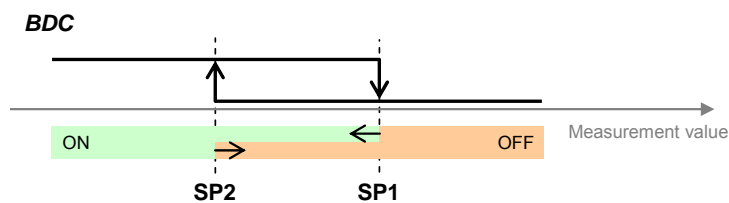
565 **Figure 16 – Example for the Window Mode**

566 **9.2.4.4 Two Point Mode (without hysteresis)**

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

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

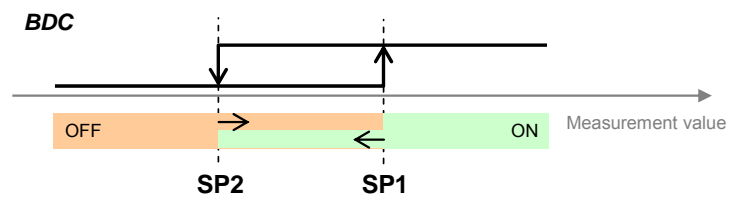
574 The behavior shown in Figure 17 is typical for "presence detection" of material with no hys-  
 575 teresis in respect to SP1 and SP2 and not inverted switching.



576

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

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



580

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

582 **9.2.5 Setpoint parameters (SP1, SP2)**

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

588 The interpretation of the Setpoints SP1 and SP2 depends on the implementation of the manu-  
 589 facturer/vendor. However, if the measurement value for the definition of switching information  
 590 (BDC) is also provided as a ProcessDataVariable (PDV), the Setpoints shall be represented in  
 591 the same manner, for example with Gradient and Offset and octet granular data types (≥ 2  
 592 octets). See 10.2 for details.

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

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

597  
 598 In case one or more checks failed, the Smart Sensor shall behave in the following manner:

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

603  
 604 In order to avoid inconsistent configuration data it is important to note,  
 605 • that SP1 and SP2 data are written together via Subindex 0 (one record) guaranteeing that  
 606 a changed value of SP1 or SP2 cannot cause a plausibility check error, or  
 607 • that the option Block Parameter [1] is used for a change of configuration guaranteeing a  
 608 plausibility check and activation of the written parameters at the termination of the entire  
 609 transmission.

610  
 611 **9.2.6 Setpoint and Switchpoint parameter coding**

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

613 **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 (Binary-DataChannel)

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	-

614

615 **9.3 BDC mapping**616 **9.3.1 Concepts**

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

620 The BDC FunctionClass [0x8001] can be parameterized via the standardized parameter ob-  
621 jects described within the subsequent clause.

622 **9.3.2 BDC Index space**

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

627

**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		

628

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

### 631 9.3.3 Access behavior of not supported Subindices

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

- 634 • Parameters of a BDC not functionally supported by the Smart Sensor shall also be read-  
635 able and writeable
- 636 • Those parameters can be written with the default value
- 637 • If other than default values are written, the Smart Sensor shall respond with the ErrorCode  
638 0x8030 (PAR\_VALOUTOFRNG = parameter value out of range)
- 639 • In case of a readout of a functionally not supported parameter, the Smart Sensor shall re-  
640 spond with the default value
- 641 • In case of access to not supported BDCs, the Smart Sensor shall respond with the Error-  
642 Code 0x8011 (IDX\_NOTAVAIL = Index not available)

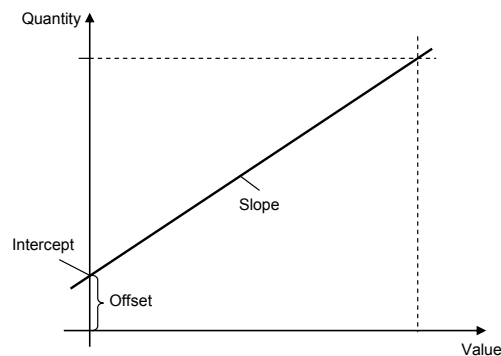
643

## 644 10 ProcessDataVariable [0x8002]

### 645 10.1 Scaling and dimensions

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

649 The transmitted value can be converted into a dimensioned value (°F, °C, inch, m, etc.) via a  
650 linear equation  $y = m \cdot x + b$ . "m" represents the slope and "b" the intercept with the y coordi-  
651 nate. Within this profile, "slope" is called "gradient" and the value of the intercept is called  
652 "Offset". Figure 19 illustrates the relationships.



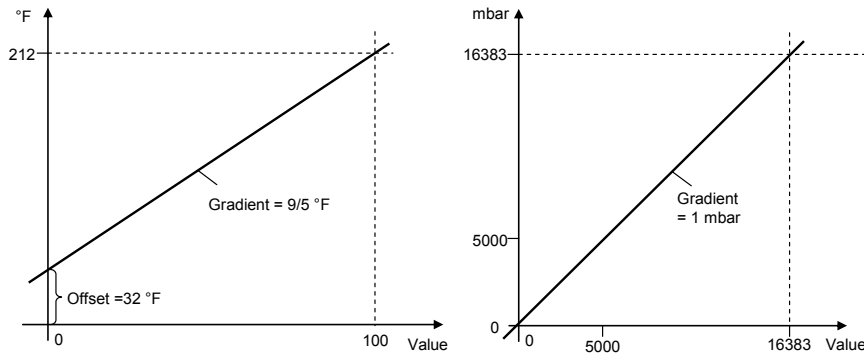
653

654 **Figure 19 – Value to quantity conversion via linear equation**

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

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

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



660

661

**Figure 20 – Conversion examples**

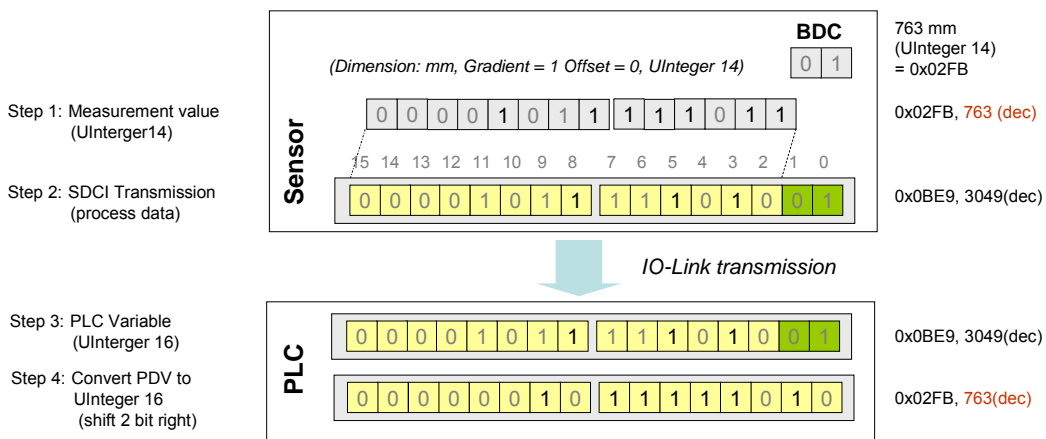
662 Usually, the transmitted PDV value is based on a dimensioned measurement value as shown  
 663 in the right example of Figure 20 (pressure in mbar). In the left example a dimensioned tem-  
 664 perature measurement value (°C) is converted in °F.

665 **10.2 Recommended PDV representation**

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

- 669 • PDVs of size > 16 bit should be represented in octet granular data types (16, 24, 32),  
 670 preferably UIntegerT
- 671 • For data < 16 bit the data type UIntegerT should be used that is easily extendable to octet  
 672 granular data types
- 673 • Preferred data lengths are 8, 12, 14, 16, 32, or 64 bit
- 674 • PDVs should carry dimensioned measurement values as shown in Figure 20 and Figure  
 675 21

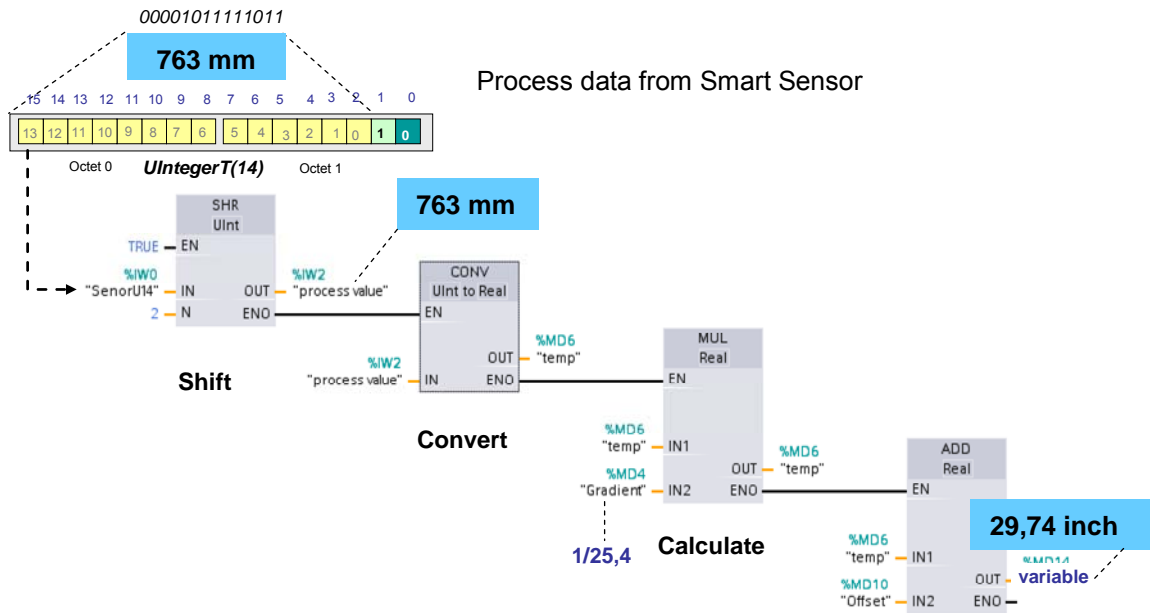
676 Figure 21 illustrates the relationship between a dimensioned PDV and its PLC variable.



677

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

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



681

682

Figure 22 – Example PLC program for a measurement value conversion

683

684 **11 Diagnosis [0x8003]**

685 **11.1 DeviceStatus and DetailedDeviceStatus**

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

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

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

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

705 **11.2 Smart Sensor EventCodes**

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

## 708 **12 TeachChannel [0x8004]**

### 709 **12.1 Teach-in concepts for Smart Sensors**

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

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

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

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

- 733 • Teach Flags: Feedback, whether the Device determined a certain "Teachpoint" success-  
734 fully or not
- 735 • Teach State: Feedback on the current state of the particular teach-in procedure

### 736 **12.2 Parameter 1: "Teach-in Channel"**

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

### 739 **12.3 Parameter 2: "Teach-in Command"**

#### 740 **12.3.1 General**

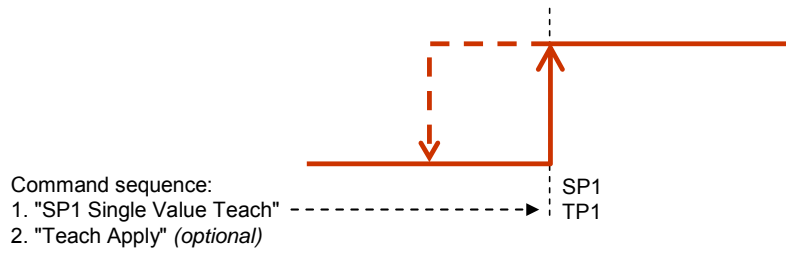
741 The parameter "Teach-in Command" allows teaching of a teachpoint (TP) or controlling of the  
742 teach-in procedure. Manufacturer/vendor specific extensions are possible. The commands of  
743 the FunctionClass [0x8004] are described within the context of a possible application in the  
744 subsequent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

#### 745 **12.3.2 "Single Value Teach"**

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

748 The associated command is "SPx Single Value Teach", with x = 1 for threshold 1 or x = 2 for  
749 threshold 2.

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



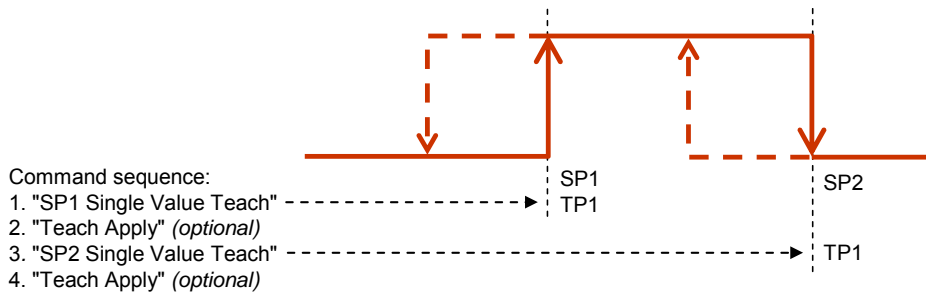
751

752

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

753

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



755

756

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

757

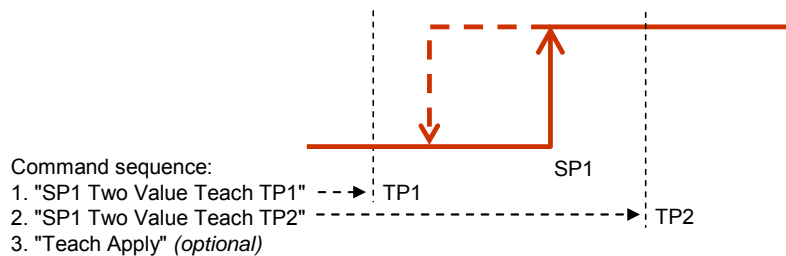
758 **12.3.3 "Two Value Teach"**

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

761 The associated commands are "SPx Two Value Teach TP1", with x = 1 for threshold 1 or x =  
 762 2 for threshold 2 and "SPx Two Value Teach TP2", with x = 1 for threshold 1 or x = 2 for  
 763 threshold 2.

764 NOTE The calculation method for TP1, TP2, and SP is manufacturer/vendor specific.

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



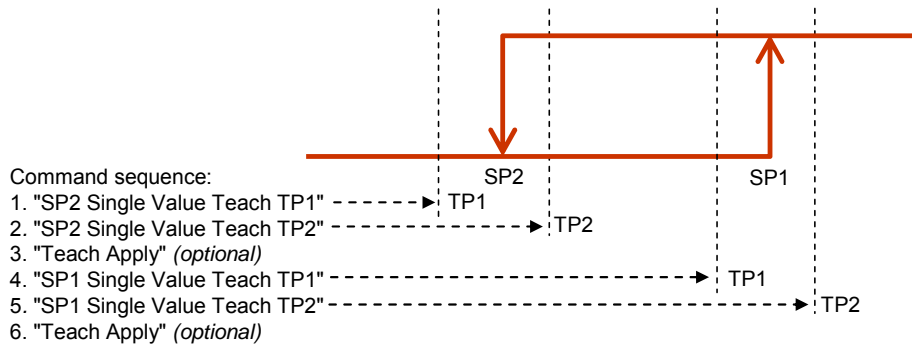
766

767

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

768

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



770

771

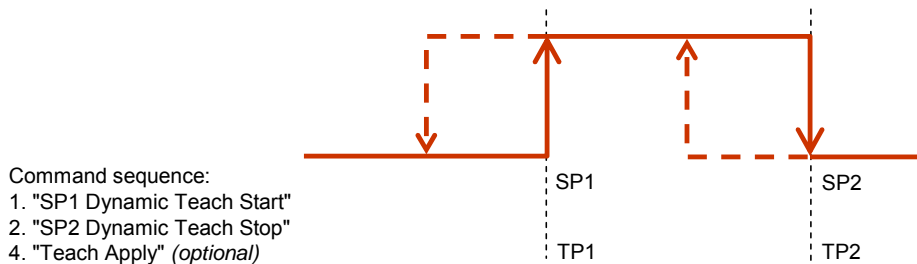
**Figure 26 – "Two Value Teach" (Two Point Mode)**

772

**773 12.3.4 "Dynamic Teach" (within a time period)**

774 One single threshold or both thresholds of a BDC are set-up via captured measurement val-  
 775 ues during a certain period of time. The teach-in procedure is "dynamic", which means, the  
 776 measurement value is not constant during the teach-in procedure. Usually, the minimum and  
 777 maximum values within this time frame are taken to define the thresholds. The associated  
 778 commands are "SPx Dynamic Teach Start", with x = 1 for threshold 1 or x = 2 for threshold 2  
 779 and "SPx Dynamic Teach Stop", with x = 1 for threshold 1 or x = 2 for threshold 2.

780 Figure 27 illustrates an example for "Dynamic Teach" in "Window Mode".



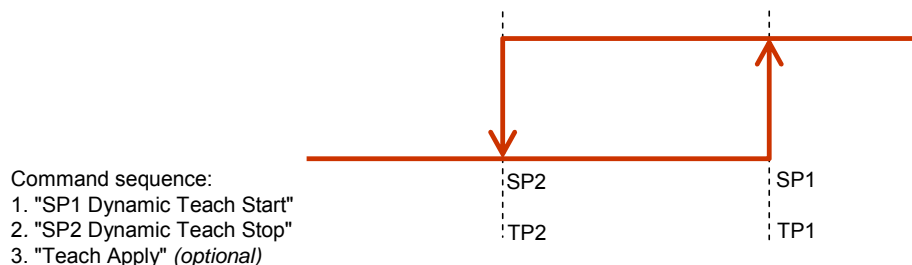
781

782

**Figure 27 – "Dynamic Teach" (Window Mode)**

783

784 Figure 28 illustrates an example for "Dynamic Teach" in "Two Point Mode".



785

786

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

**787 12.3.5 "Teach Apply"**

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

791 **12.3.6 "Teach Cancel"**

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

794 **12.4 Parameter 3: "Teach-in Status"**

795 **12.4.1 Status types**

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

798 The following "Teach States" are defined:

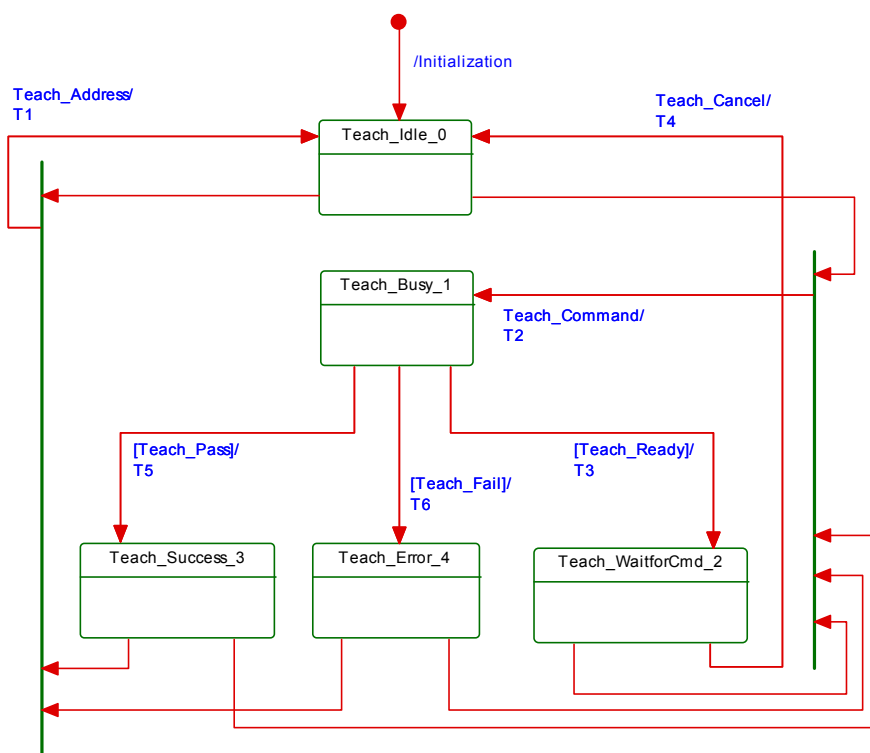
- 799 • IDLE
- 800 • BUSY
- 801 • WAIT FOR COMMAND
- 802 • SPxSUCCESS
- 803 • ERROR

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

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

809 **12.4.2 Teach-in dynamics**

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



811

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

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

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

816 **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	

819

## 820 12.5 Mapping to SDCI communication

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

825

**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

826

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

829

**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 for Setpoint1
SP2 Single Value Teach	0x42	Determine Teachpoint 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
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
Manufacturer Teach	0x4B to 0x4E	For manufacturer specific use
Teach Cancel	0x4F	Abort Teach-in sequence

830

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

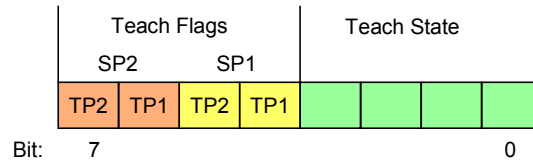
832

**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

833

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



836

837

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

838 Table 16 shows the "Teach State" coding.

839

**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

840

841 Table 17 shows the "Teach Flag" coding.

842

**Table 17 – "Teach Flag" coding**

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

843

844  
845  
846

## **Annex A** (normative) **Profile testing and conformity**

### 847 **A.1 General**

#### 848 **A.1.1 Overview**

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

#### 853 **A.1.2 Issues for testing/checking**

- 854 • Identification complete and correct?
- 855 • Descriptors available and correct?
- 856 • All rules observed?
- 857 • Switching behavior conform to the specification?
- 858 • FunctionClasses available and correct?
  - 859 - Indices available and correct?
  - 860 - Read/write correct?
  - 861 - Data structures: Record? Value ranges?
  - 862 - Behavior of the FunctionClass conforms to the specification?
- 863 • Extract BDCs (switching functions) from user manual or IODD and check conformity with  
864 the specification
- 865 • Checklist: checkbox "relevant" and checkbox "verified"
- 866 • IODD: see [6]

867

868

869

870

871  
872  
873

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

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

876 **IO-Link Consortium**  
877 Haid-und-Neu-Str. 7  
878 76131 Karlsruhe  
879 Germany  
880 Phone: +49 (0) 721 / 96 58 590  
881 Fax: +49 (0) 721 / 96 58 589  
882 E-mail: [info@io-link.com](mailto:info@io-link.com)  
883 Web site: <http://www.io-link.com>  
884

885

## Bibliography

- 886 [1] IO-Link Consortium, *IO-Link Interface and System*, V1.1, November 2010, Order No.  
887 10.002 or IEC 61131-9, *Programmable controllers – Part 9: Single-drop digital communi-*  
888 *cation interface for small sensors and actuators (SDCI)*
- 889 [2] IO-Link Consortium, *IO Device Description (IODD)*, V1.1, July 2011, Order No. 10.012
- 890 [3] IEC/TR 62390:2005, *Common automation device profile guideline*
- 891 [4] IEC 60050 (all parts), *International Electrotechnical Vocabulary*  
892
- 893 NOTE See also the IEC Multilingual Dictionary – Electricity, Electronics and Telecommunications (available  
894 on CD-ROM and at <<http://domino.iec.ch/iev>>).
- 895 [5] IO-Link Consortium, *IO-Link Communication*, V1.0, January 2009, Order No. 10.002
- 896 [6] IO-Link Consortium, *IO-Link Test Specification*, V1.1, May 2011, Order No. 10.032

897

898

© Copyright by:

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](mailto:info@io-link.com)

<http://www.io-link.com/>

