SLOVENSKI STANDARD

SIST EN 50090-3-3:2009

01-oktober-2009

Stanovanjski in stavbni elektronski sistemi (HBES) - 3-3. del: Vidiki uporabe - Vzajemno delovanje modela HBES in splošni tipi podatkov HBES

Home and Building Electronic Systems (HBES) -- Part 3-3: Aspects of application - HBES Interworking model and common HBES data types

Elektrische Systemtechnik für Heim und Gebäude (ESHG) - Teil 3-3: Anwendungsaspekte - ESHG-Interworking-Modell und übliche ESHG-Datenformate

Systèmes électroniques pour les foyers domestiques et les bâtiments (HBES) -- Partie 3-3 : Aspects de l'application - Modèle d'inter-fonctionnement des HBES et types de données communes

Ta slovenski standard je istoveten z: EN 50090-3-3:2009

ICS:

97.120	Avtomatske krmilne naprave za dom	Automatic controls for household use

SIST EN 50090-3-3:2009 **en**

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN 50090-3-3

May 2009

ICS 97.120

English version

Home and Building Electronic Systems (HBES) - Part 3-3: Aspects of application - HBES Interworking model and common HBES data types

Systèmes électroniques pour les foyers domestiques et les bâtiments (HBES) - Partie 3-3: Aspects de l'application - Modèle d'inter-fonctionnement des HBES et types de données communes

Elektrische Systemtechnik für Heim und Gebäude (ESHG) - Teil 3-3: Anwendungsaspekte - ESHG-Interworking-Modell und übliche ESHG-Datenformate

This European Standard was approved by CENELEC on 2008-12-01. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the Central Secretariat or to any CENELEC member.

This European Standard exists in two official versions (English, French). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the Central Secretariat has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Cyprus, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and the United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

Central Secretariat: Avenue Marnix 17, B - 1000 Brussels

Ref. No. EN 50090-3-3:2009 E

Foreword

This European Standard was prepared by the Technical Committee CENELEC TC 205, Home and Building Electronic Systems (HBES), joined by the co-operating partner KNX Association.

The text of the draft was submitted to the Unique Acceptance Procedure and was approved by CENELEC as EN 50090-3-3 on 2008-12-01.

CENELEC takes no position concerning the evidence, validity and scope of patent rights.

KNX Association as Cooperating Partner to CENELEC confirms that to the extent that the standard contains patents and like rights, the KNX Association's members are willing to negotiate licenses thereof with applicants throughout the world on fair, reasonable and non-discriminatory terms and conditions.

KNX Association
De Kleetlaan 5, bus 11
B - 1831 Diegem

Tel.: + 32 2 775 85 90
Fax.: + 32 2 675 50 28
e-mail: info@knx.org
www.knx.org

Attention is drawn to the possibility that some of the elements of this standard may be the subject of patent rights other than those identified above. CENELEC shall not be held responsible for identifying any or all such patent rights.

The following dates were fixed:

– latest date by which the EN has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2009-12-01

– latest date by which the national standards conflicting with the EN have to be withdrawn (dow) 2011-12-01

EN 50090-3-3 is part of the EN 50090 series of European Standards, which will comprise the following parts:

Part 1: Standardization structure

Part 2: System overview

Part 3: Aspects of application

Part 4: Media independent layers

Part 5: Media and media dependent layers

Part 6: Interfaces

Part 7: System management

Part 8: Conformity assessment of products

Part 9: Installation requirements

Contents

Figures

Tables

Introduction

Interworking between devices signifies that these products send and receive datagrams and are able to properly understand and react on them. This ability is provided without additional equipment (like translators or gateways).

NOTE Media couplers are needed if different media are used in an installation.

The market requires Interworking for a multi-vendor approach, this is, products from different manufacturers can interwork in a certain application segment or domain as well as across different applications (cross discipline Interworking).

Such an Interworking is only possible if a set of requirements is complied with as defined in an Interworking model. For this, Functional Blocks need to be defined, which in turn specify Datapoints and the communication mechanisms to be used. Such a set of requirements is referred to as "Application Interworking Specifications" (AIS).

AIS allow Interworking independent of the implementation by a manufacturer. Besides the advantages for the user (multi-vendor offer) Interworking also allows a broad OEM market and easy market access for niche-products providers. Furthermore Interworking allows the establishment of a common market infrastructure (i.e. common configuration tool, training, etc.)

1 Scope

This European Standard gives general guidelines and recommendations to ensure interworking between HBES devices made by different manufacturers. It also contains design guidelines for the design of Functional Blocks and new datapoint types, the building blocks of HBES interworking.

In this way, the standard can be used as a basis to design application specifications relative to an Application Domain. If designed and supported by a large group of manufacturers, such application specifications will ensure to end customers a high degree of interoperability between products based on the HBES Communication System of different manufacturers.

This European Standard is used as a product family standard. It is not intended to be used as a stand-alone standard.

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.

EN 50090-1 [1)]	Home and Building Electronic Systems (HBES) – Part 1: Standardization structure
EN 50090-3-2:2004	Home and Building Electronic Systems (HBES) – Part 3-2: Aspects of application – User process for HBES Class 1
EN 50090-4-1:2004	Home and Building Electronic Systems (HBES) – Part 4-1: Media independent layers – Application Layer for HBES Class 1
EN 50090-4-2: 2004	Home and Building Electronic Systems (HBES) – Part 4-2: Media independent layers – Transport layer, network layer and general parts of data link layer for HBES Class 1

1) Under consideration.

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the definitions given in EN 50090-1 apply.

3.2 Abbreviations

AIS	Application Interworking Specifications
COV	Change of Value
DMA	Direct Memory Access
DP	Data Point
DPT	Datapoint type
DPT ID	Datapoint type identifier
FB	Functional Block
GO	Group Object
IO	Interface Object
lsb	least significant bit
M	Mandatory
msb	most significant bit
MSB	Most Significant Byte
NA	Not Applicable
O	Optional
PDT	Property Data Type
PID	Property Identifier
OEM	Original Equipment Manufacturer

4 HBES Interworking model

HBES Interworking can be applied in many application domains. Each Application Domain can encompass one or more Applications.

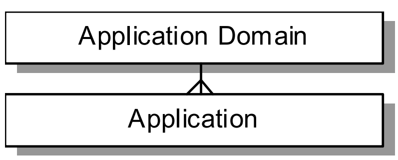

Figure 1 - The HBES Interworking Model
An Application Domain can contain one or more Applications

The Application Modelling or the process of analysing, deciding on the solution and specifying the model for such an Application needs to be agreed upon amongst manufacturers in Application Specification Groups.

Applications shall not be defined in terms of products, but analysed and split up into *Functional Blocks*, which communicate with one another. The term *Distributed Application* indicates this approach: the total functionality of an Application is spread over a number of Functional Blocks implemented in various devices in the network.

A Functional Block transports its data over the bus via one or more Datapoints (these are Inputs, Outputs, Parameters and Diagnostic Data).

A Functional Block thus describes the standard specification of the chosen solution for one given task of an application.

These Datapoints and their described functionality are implemented by the product developer.

The following picture shows the Interworking model as defined so far.

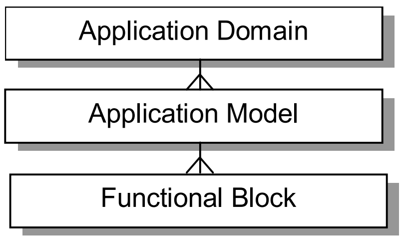

Figure 2 - The HBES Interworking Model
An Application Model may contain one or more Functional Blocks

The Functional Blocks shall be described as objects; this is a set of Datapoints and a well-defined behaviour.

The standard graphical representation for a Functional Block shall be the following:

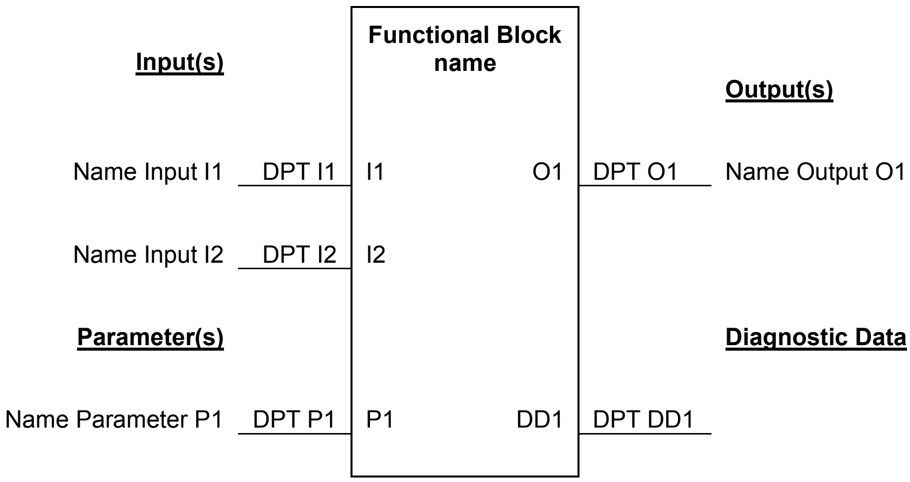

Figure 3 - Standard representation for Functional Blocks

The Datapoints that are typically Inputs shall be put on the left, the Outputs on the right, the Parameters on the left below the Inputs and the Diagnostic Data on the right below the Outputs. For each Datapoint, a name shall be given, an indication of its Datapoint Type and an abbreviation.

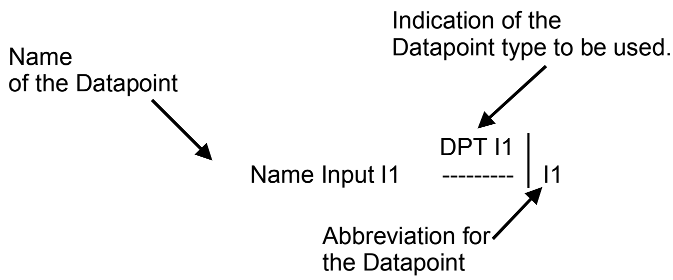

Figure 4 - Datapoints indicated in Functional Blocks

A manufacturer may group one or more of these Functional Blocks, of the same or of different Applications, to build a device.

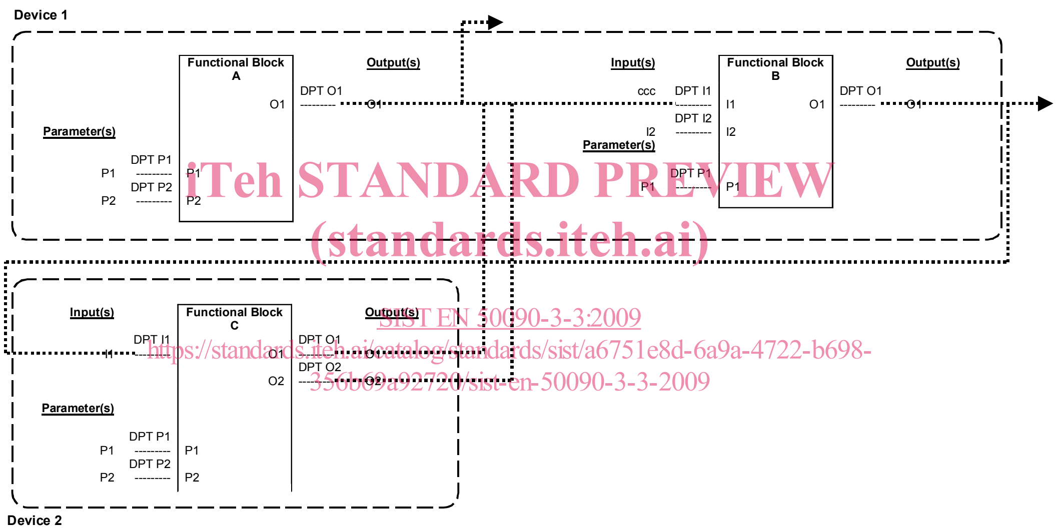

Figure 5 - Functional Blocks grouped in devices and linked

For every Functional Block its behaviour shall be specified. This fixes its handling of its Datapoints and physical inputs and outputs (e.g. a state machine of a dimming actuator).

A Datapoint is any interface over which data in the Functional Block can be set or received and/or transmitted (for its run-time operation). Every Functional Block may have one or more such Datapoints.

From the **communication** point of view, 4 classes of Datapoints can be distinguished (for more information see 5.3.3):

NOTE These classes of Datapoints are defined in EN 50090-4-2, EN 50090-4-1 and EN 50090-3-2:

- Group Object Datapoint
- Interface Object Property Datapoint
- Polling Value Datapoint
- Memory Mapped Datapoint

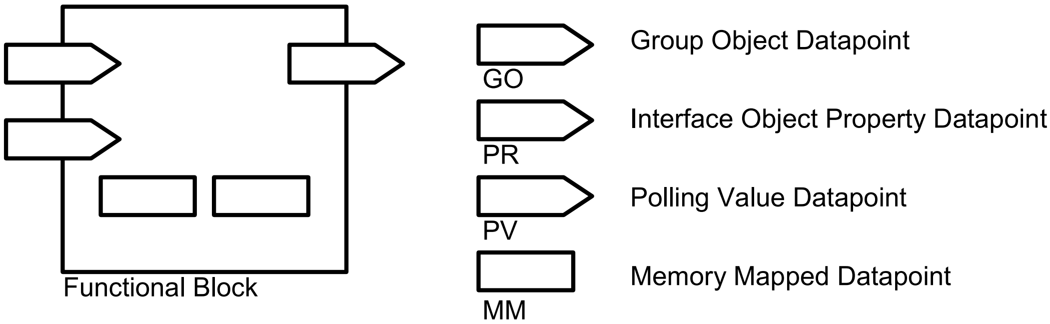

Figure 6 - Functional Block with 5 Datapoints

From the **access** point of view another differentiation of Datapoints is possible:

- Input: a value that is received and processed by the Functional Block.
- Output: the value resulting from the process of the Functional Block and to be provided to at least one other Functional Block.
- Parameter: a value that controls the process of handling the Input(s) and generating the values of the Output(s). This access type is typically non-volatile or saved at reset. It is usually set by management functions.
- Diagnostic Data: a value that represents the local or internal status information of a Functional Block. It is not used for runtime communication to Inputs or Outputs of other FBs but serves for visualisation on a central unit or operating station and, during installation, service and maintenance.

The data exchanged through Datapoints shall be standard in format, encoding, range and unit. They are defined in *Datapoint Types* (DPT).

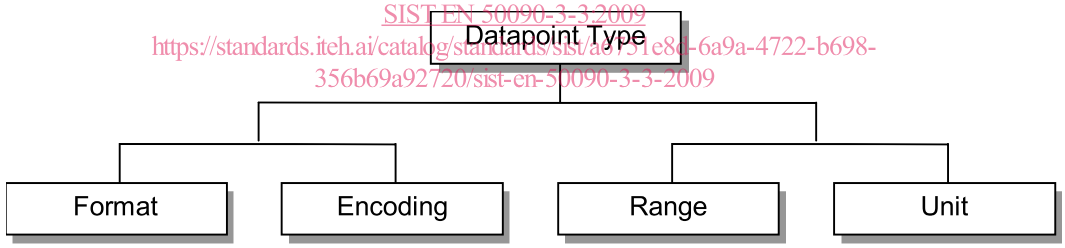

Figure 7 - The information contained in a Datapoint Type definition

When a given Datapoint Type is used to represent the value of a Datapoint in a Functional Block, it additionally obtains a specific <u>semantic</u> meaning.

EXAMPLE A Datapoint Type "Temperature" obtains the semantic value "Input Temperature Setpoint Value" when it is used in the Functional Block "Boiler Control".

The definition of a Datapoint Type shall consist of the following elements:

- *Format*: describes the sequence and length of the fields, each consisting of one or more bits, of which the Datapoint Type is built up.
- *Encoding:* describes how the data, that shall be transported using this Datapoint Type, is coded using the given format, possibly for each field.
- *Range:* describes the limitation of the values that may be encoded in this Datapoint Type, possibly for each field. This may be a minimum/maximum indication or an explicit list.
- *Unit*: indicates the unit of the information that can be transported, possibly for each field.

An example of this all is given in Figure 8.

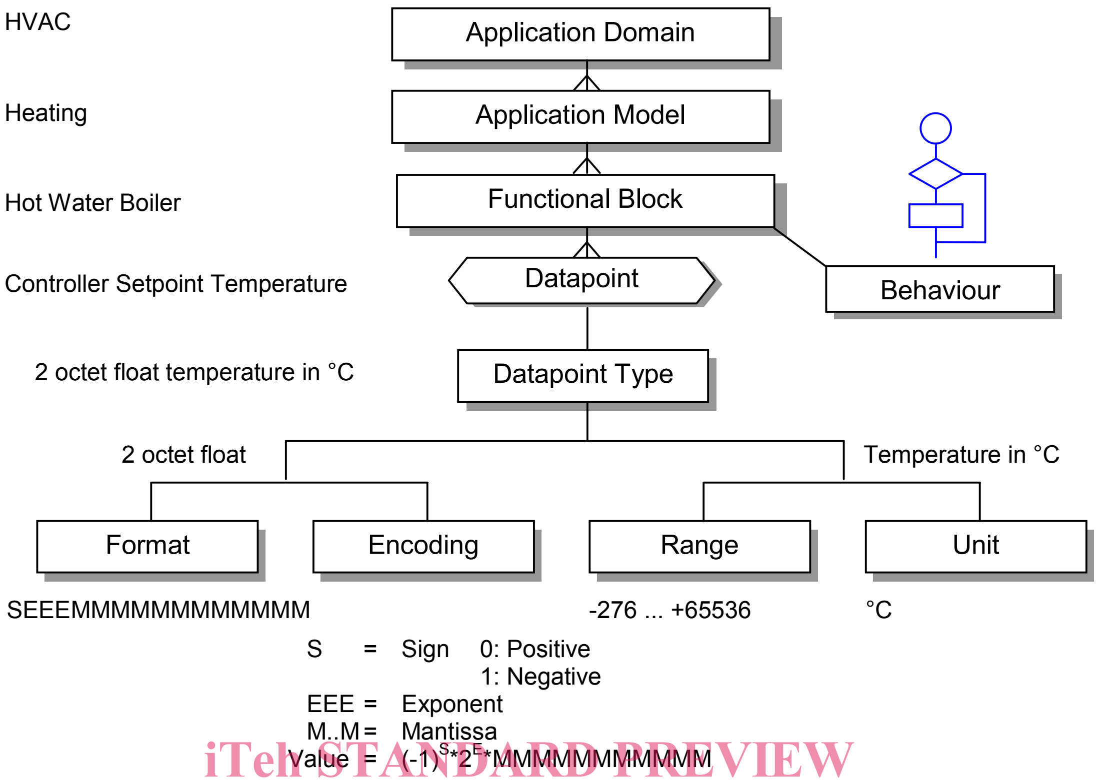

Figure 8 - Example of an Interworking specification

4.1 Principles of HBES Interworking

- HBES Interworking is primarily ensured by the use of standard Datapoint Types, which specify how to encode data that is exchanged between devices by using identical *communication mechanisms*.
- Functional Blocks can be used to **describe** the used Datapoint Types, communication mechanisms and behaviour for a given functionality.
- Functional Blocks may oblige the use of a specific class of Datapoints (in most cases Group Objects), implementation of an additional class of Datapoint (e.g. Interface Objects Property) may be optional.
- A list of the commonly used HBES DPTs is given in Annex A to this standard. A DPT can only be considered compliant to the one given in the Annex when the specified encoding, range and unit really complies with what is realised.

EXAMPLE For 8 bit unsigned integers, value shall indeed be numerical (e.g. no bit-field) and the entire specified range shall be supported without gaps.

- Reserved fields shall be set to the value as specified in the DPT specification. If reserved fields are set to a value other than allowed by the specification, the implementation of the DPT can not be considered as compliant. In case of designing a new DPT, unused (i.e. reserved fields) shall be set to 0b.
- It may be possible to use some of the DPTs in Annex A only in specific applications, others are more general purpose.
- Fields of an even octet size, e.g. 2 octets, should be positioned on even octet number positions within the DPT.
- For parameterization during device set-up and for diagnostic purposes, DMA parameters or HBES Interface Objects should be implemented instead of HBES Group objects.
- Datapoints should not be used bidirectionally.
- If an application provides status information, one (or more) separate Datapoint(s) should be used.
- It is recommended, that necessary time delays are located in the actuators or the application controller and not in the sensors.

4.2 Busload

4.2.1 Repetition rate

The repetition rate shall be selected very carefully as it influences the busload (risk of overload).

NOTE At the planning stage of an installation the possible busload shall be assessed.

The following aspects shall be covered.

4.2.2 Change of value and Delta value

If a device generates information on the bus depending on a Change Of Value (COV) condition, a limitation-mechanism (either by means of hardware [strap] or software features [parameter] shall be provided.

The application shall transmit the value only after the (sensor) value changes at least for the (minimum) Delta value. For fast changing sensor values, the application shall not transmit a new value before a minimum repetition time has elapsed.

4.2.3 Message on request

If this mode is selected, the same requirement as for delta value applies for the application sending the request.

4.2.4 Heart-beat

For sensor values with little changes, smaller than the Delta value or no change at all, the application shall transmit the value after the maximum repetition time heart-beat. Heart-beat communication denotes that

– the sender shall periodically send its data, and
– the receiver shall maintain a time-out timer to check for this periodic transmission and act in a specified way when no sensor signal is received within the given time-out.

Heart-beat can be used to increase application reliability as well as for life-check of a communication partner.

Table 1 specifies the use requirements of heart-beat.

Table 1 – Use of heart-beat

Signal	Sender / receiver	Required ?
sensor values	heart-beat at sender: time-out at receiver:	Y O
calculated values [a]	heart-beat at sender: time-out at receiver:	Y O
safety relevant values	heart-beat at sender: time-out at receiver:	Y Y
life check [b]	heart-beat at sender: time-out at receiver:	Y Y
trigger signal [c]	heart-beat at sender: time-out at receiver:	NA NA
user triggered signals [d]	heart-beat at sender: time-out at receiver:	O NA

Y = mandatory, O = optional, NA = not applicable.

[a] By an automation process, scheduler, building management station. Whether time-out and heart-beat is required may depend on the application domain.

[b] Datapoints used to check the presence of a partner device.

[c] Datapoints using DPT_Trigger.

[d] Signals sent exclusively on user interaction.

4.2.5 Transmission priorities

The priorities should be selected very carefully to ensure fair bus access. The priorities for frames are defined in EN 50090-4-2.

The maximum priority that shall be used for run-time communication is “normal” (01b).

Usage of the priority “urgent” shall be reserved for specific applications only.

Usage of the priority “system” is reserved for communication for system configuration and Management Procedures.

4.2.6 Bus load considerations for Property Clients

When Property values in a Property Server are accessed (write/read) by a Property Client, then the bus load generated by this communication is fully controlled by the Property Client.

At runtime, the Property Client shall therefore guard the following rules to keep the bus load within limits:

– The Property Client shall not access a next Property value before the Property Server has responded to the previous Property access (A_PropertyValue_Response-PDU).

– While waiting for the response of one Property Server, the Property Client shall not address another Property Server. During configuration, this requirement does not apply: a Configuration Client may access more than one device at the same time.

– In subsequent accesses to Property values, in between the response from the Property Server and the next access to a Property value, the Property Client shall guard a longer interframe time than for low priority data. This will allow normal process data to access the bus meanwhile. This may in the Application in the Property client either be given automatically by the delays in processing the received property values, or may be handled explicitly by introducing small wait times of e.g. 1 ms.

4.3 Datapoint Type error handling

4.3.1 Scope

DPT error handling denotes the capability to properly handle and react on not-allowed or unexpected values of DPTs or fields of DPTs.

Senders – devices that provide the data – shall in general not send erroneous data. Receivers may react in several ways to unexpected values.

The rules for DPT error handling listed below apply to entire DPTs in case the DPT exists only of a single field and for each individual field in case the DPT is composed of two or more fields.

Consistency of data does not have to be tested and handled across different fields of structured DPTs.

EXAMPLE It is not required to check whether February 29 is a possible date in DPT_DateTime.