Industrial communications subsystem based on ISO 11898 (CAN) for controller-device interfaces —

Part 3: Smart Distributed System (SDS)

The European Standard EN 50325-3:2001 has the status of a British Standard

ICS 43.180

National foreword

This British Standard is the official English language version of EN 50325-3:2001.

The UK participation in its preparation was entrusted to Technical Committee AMT/7, Monitoring and control aspects of AMT, which has the responsibility to:

— aid enquirers to understand the text;

— present to the responsible European committee any enquiries on the interpretation, or proposals for change, and keep the UK interests informed;

— monitor related international and European developments and promulgate them in the UK.

A list of organizations represented on this committee can be obtained on request to its secretary.

Textual errors

The textual errors set out below were discovered when the English language version of EN 50325-3:2001 was adopted as the national standard. They have been reported to CENELEC in a proposal to amend the text of the European Standard.

The ratified text of this document contains extra lettering in Table 3, Table 4, Table 5, Table 6, Table 7, Table 8 and Table 9. In this standard this extra lettering has been removed.

Cross-references

The British Standards which implement international or European publications referred to in this document may be found in the BSI Standards Catalogue under the section entitled "International Standards Correspondence Index", or by using the "Find" facility of the BSI Standards Electronic Catalogue.

A British Standard does not purport to include all the necessary provisions of a contract. Users of British Standards are responsible for their correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

Summary of pages

This document comprises a front cover, an inside front cover, the EN title page, pages 2 to 60, an inside back cover and a back cover.

The BSI copyright date displayed in this document indicates when the document was last issued.

This British Standard, having been prepared under the direction of the DISC Board, was published under the authority of the Standards Committee and comes into effect on 15 July 2001

BSI 07-2001

Amendments issued since publication

Amd. No.	Date	Comments

ISBN 0 580 37947 7

EUROPEAN STANDARD **EN 50325-3**

NORME EUROPÉENNE

EUROPÄISCHE NORM April 2001

ICS 43.180

English version

Industrial communications subsystem based on ISO 11898 (CAN) for controller-device interfaces — Part 3: Smart Distributed System (SDS)

This European Standard was approved by CENELEC on 2000-04-01. 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 only in English. 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 version.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Czech Republic, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and United Kingdom.

CENELEC

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

Central Secretariat: rue de Stassart 35, B - 1050 Brussels

Foreword

This European Standard was prepared by the Technical Committee CENELEC TC 65CX, Fieldbus.

The text of the draft was submitted to the Unique Acceptance Procedure and was approved by CENELEC as EN 50325-3 on 2000-04-01.

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) 2001-10-01

– latest date by which the national standards conflicting with the EN have to be withdrawn (dow) 2003-04-01

EN 50325 is divided into three parts:

Part 1 General requirements
Part 2 DeviceNet
Part 3 Smart Distributed System (SDS)

The specifications for DeviceNet and SDS are based on ISO 11898 *Controller area network (CAN) for high-speed communication*, a broadcast-oriented communications protocol. However, ISO 11898 specifies only part of a complete communication system, and additional specifications are needed for other layers to ensure precise data exchange functionality and support of inter-operating devices. The DeviceNet and SDS specifications build on ISO 11898 to describe a complete industrial communication system.

Contents

Introduction

The Smart Distributed System (SDS) is intended for use in, but is not limited to, industrial automation applications. These applications may include devices such as limit switches, proximity sensors, electro-pneumatic valves, relays, motor starters, operator interface panels, analogue inputs, analogue outputs, and controllers.

SDS provides for the connection of intelligent devices such as sensors, actuators and other components to one or more controllers. SDS functionality may be integrated directly into the devices or be in modules allowing the connection of conventional components to the network.

The SDS network consists of one or more controllers connected to up to 126 Logical Devices. In addition to the process data, SDS allows for the transmission of parameters and diagnostic data. The data exchange may be either event driven, cyclical, multicast or polled. A maximum of 6 bytes of data may be transmitted without fragmentation.

Topology is typically a single trunk with short branches using a cable comprising two shielded, twisted pairs with a common earth wire all within a single jacket.

Data is transmitted at rates of 125 kbit/s, 250 kbit/s, 500 kbit/s and 1Mbit/s with maximum system trunk lengths of 457 m, 182 m, 91 m and 22 m respectively.

Detailed information on the performance is contained in clause 5.

Figure 1 shows an example of an SDS Network.

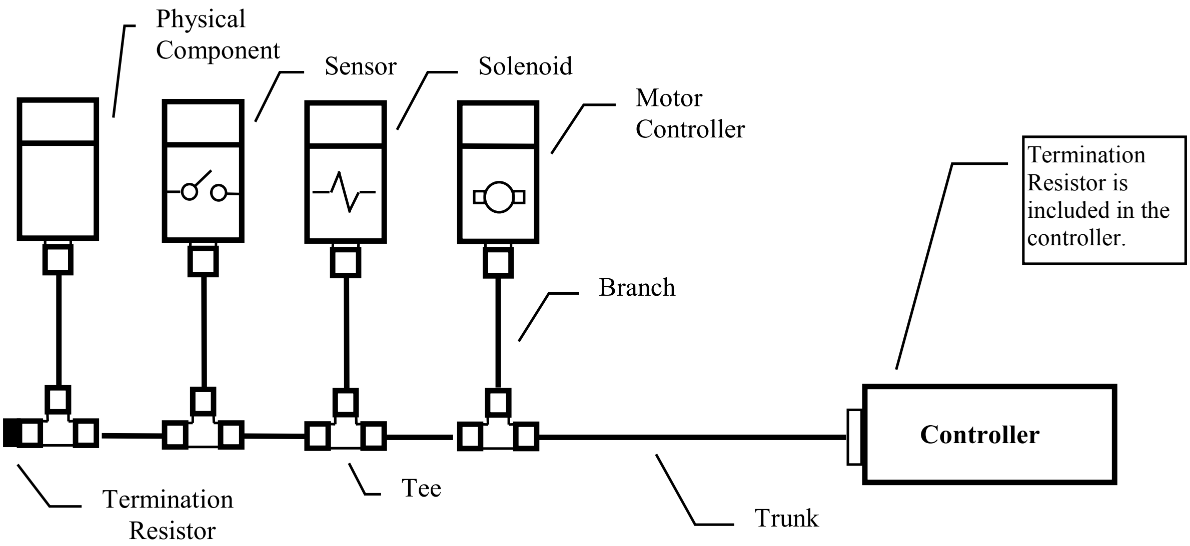


Figure 1 - Example of an SDS Network

General information on licensing

CENELEC calls attention to the fact that patent rights are linked to EN 50325-3 (Part 3: Smart Distributed System). The patent holder, Honeywell Inc., has assured to CENELEC that it is willing to grant a licence under these patents on reasonable and non discriminatory terms and conditions to anyone wishing to obtain such a license, applying the rules of CEN/CENELEC Memorandum 8.

Honeywell's undertakings (policy letter on licensing, the license offer and the form of license) in this respect are on file with CENELEC and available for inspection by all interested parties at the CENELEC Central Secretariat.

The license details may be obtained from:

The Director
(Industrial Marketing and Applied Technology Sensing and Controls Europe)
Honeywell Control Systems Ltd.
Newhouse Industrial Estate,
Motherwell,
Lanarkshire
Scotland ML1 5SB
GB

1 Scope

This Part of EN 50325 specifies the following particular requirements for Smart Distributed System (SDS).

- Requirements for interfaces between control devices and switching elements,
- Normal service conditions for devices,
- Constructional and performance requirements,
- Tests to verify conformance to requirements.

2 Normative references

This Part of EN 50325 incorporates by dated or undated reference, provisions from other publications. These normative references are cited at the appropriate places in the text and the publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications apply to this Part only when incorporated in it by amendment or revision. For undated references the latest edition of the publication referred to applies.

EN 55011	1998	Industrial, scientific and medical (ISM) radio-frequency equipment – Radio disturbance characteristics, limits and methods of measurement (CISPR 11:1997, mod.)
EN 60947-1	1999	Low-voltage switchgear and controlgear Part 1: General rules (IEC 60947-1:1999, mod.)
EN 61000-4-2	1995	Electromagnetic compatibility (EMC) Part 4-2: Testing and measurement techniques - Electrostatic discharge immunity test (IEC 61000-4-2:1995)
EN 61000-4-3	1996	Electromagnetic compatibility (EMC) Part 4-3: Testing and measurement techniques - Radiated, radio-frequency, electromagnetic field immunity test (IEC 61000-4-3:1995, mod.)
EN 61000-4-4	1995	Electromagnetic compatibility (EMC) Part 4-4: Testing and measurement techniques - Electrical fast transient/burst immunity test (IEC 61000-4-4:1995)
EN 61000-4-5	1995	Electromagnetic compatibility (EMC) Part 4-5: Testing and measurement techniques - Surge immunity test (IEC 61000-4-5:1995)
EN 61000-4-6	1996	Electromagnetic compatibility (EMC) Part 4-6: Testing and measurement techniques - Immunity to conducted disturbances, induced by radio-frequency fields (IEC 61000-4-6:1996)
EN 61131-3	1993	Programmable controllers - Part 3: Programming languages (IEC 61131-3:1993)
ISO/IEC 7498-1	1994	Information technology - Open Systems Interconnection Part 1: Basic Reference Model: The Basic Model
ISO TR 8509	1987	Information Processing Systems, Open Systems Interconnection, Service Conventions
ISO 11898	1993	Road vehicles - Interchange of digital information - Controller area network (CAN) for high-speed communication

3 Definitions, abbreviations and symbols

3.1 Definitions

For the purposes of this Part, the following specific terms and definitions apply.

3.1.1 Application Layer

seventh layer of the ISO/OSI Reference Model (see ISO/IEC 7498-1)

3.1.2 Application Layer Protocol (ALP)

rules governing the structure and timing of Application Layer Protocol Data Units that are used to achieve the services provided by the application layer

3.1.3 Application Layer Protocol Data Unit (APDU)

unit of data transfer exchanged between application layers. It is encapsulated within a Data Link Layer Protocol Data Unit (DLPDU) when transmitted from one component to another

3.1.4 Application Layer Service

service provided to the Service User of the application layer. The service may be provided by means of a specified function call

3.1.5 Autobaud

network process for determining the network communication baud rate

3.1.6 Change Of State (COS)

report that a binary device has changed state

3.1.7 Change Of Value (COV)

report that a device input value has changed by a predetermined amount

3.1.8 Confirm

representation of an interaction in which a Service Provider indicates, at a particular service access point, completion of some procedure previously invoked, at that service access point, by an interaction represented by a request Primitive. The Confirm is an Application Layer Primitive service. A Confirm notifies the Service User of the presence of the response (ISO TR 8509)

3.1.9 Data Link Layer Protocol Data Unit (DLPDU)

protocol data unit at the data link layer

3.1.10 Embedded Object

network-addressable entity within a Logical Device. The Embedded Object may be e.g. an embedded interface, an embedded device, an embedded function or an embedded function block

3.1.11 Embedded Object Identifier

5 bit Application Protocol Data Unit field that holds a number used to differentiate between up to 32 Embedded Objects in a Logical Device

3.1.12 Indication

representation of an interaction in which an Application Layer Protocol Service Provider indicates that a procedure has been invoked by the Application Layer Protocol Service User at the peer service access point. The Indication is an Application Layer Primitive service. The Indication notifies the Service User of the presence of a request from another device or controller (ISO TR 8509)

3.1.13 Logical Device

separate addressable entity within a Physical Component. A Logical Device is connected to the communications channel via its logical address

3.1.14 Network

all the physical media, connectors, associated communication elements and a given set of communicating devices interconnected for the purpose of communication

3.1.15 Object Model

description of behaviour and structure that comprises a set of attributes, a set of actions and a set of events

3.1.16 Physical Component

single or modular physical package, including hardware and software and containing one or more Logical Devices, that is connected to the Network

3.1.17 Physical Layer Interface PLI

interface between the Physical Component and the communications media

3.1.18 Primitive

implementation-independent representation of an interaction between a Service User and a Service Provider. Primitive services exist at the Application Layer level. The Primitives are: Request, Response, Indication and Confirm (ISO TR 8509)

3.2 Abbreviations

3.2.1 CAN Controller Area Network

3.2.2 EOID Embedded Object Identifier

3.2.3 FI Fragmentation Indicator

3.2.4 R/R Request/Response

3.2.5 RTR Remote Transmission Request

3.2.6 EUT Equipment under Test

3.3 Symbols

3.3.1 (+)

qualifying suffix used with Result parameter service descriptions. As a Result qualifier, it refers to a successful result in Service Convention diagrams

3.3.2 (-)

qualifying suffix used with Result parameter service descriptions. As a Result qualifier, it refers to an unsuccessful result in Service Convention diagrams

3.3.3 (=)

qualifying prefix used with Indication and Confirm Primitive service descriptions in Service Convention diagrams. It means that the Primitive is the same as one occurring at a previous level

4 Relationship with the OSI Reference Model

The SDS protocol is based on a three-layer architecture.

NOTE 1 These layers constitute a collapsed form of the OSI (Open Systems Interconnection) seven layer architecture, mapping onto the physical, data link, and application layers of the OSI Reference Model (ISO/IEC 7498-1).

The Application Layer uses the services provided by the V2.0A CAN Specification data link layer of ISO 11898.

NOTE 2 Figure 2 compares the architecture of the SDS Protocol Model with the OSI Reference Model.

SDS Protocol Model	**Corresponding ISO Layers**
APPLICATION LAYER	**APPLICATION LAYER**
	PRESENTATION LAYER
	SESSION LAYER
	TRANSPORT LAYER
	NETWORK LAYER
CAN DATA LINK LAYER	**DATA LINK LAYER**
SDS PHYSICAL LAYER	**PHYSICAL LAYER**

Figure 2 - SDS protocol architecture compared with the OSI Reference Model

5 Characteristics

5.1 SDS Network

5.1.1 Network

An SDS Network consists of Physical Components which include input devices, output devices, PLC interfaces, PC interfaces and interfaces to other Networks, etc. Physical Components are modelled as collections of Logical Devices that communicate over a physical medium. Figure 3 shows a logical representation of an SDS Network.

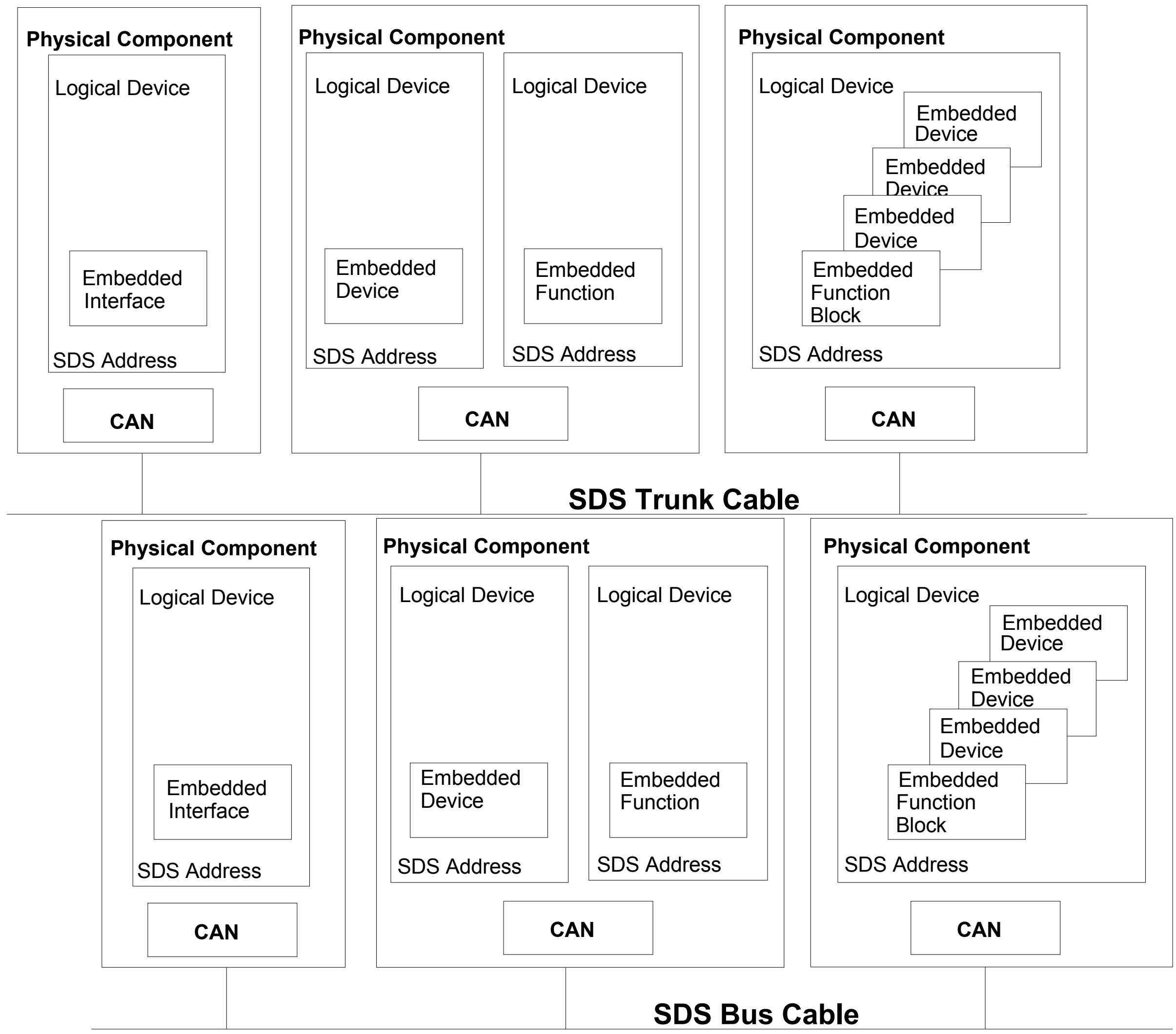


Figure 3 - Logical representation of an SDS Network

5.1.2 Modelling

5.1.2.1 Component Model

Component models represent SDS network visible structure and behaviour of a node. The primary purpose of modelling is to improve the interoperability and interchangeability of SDS components.

The basic SDS component model structure is shown in graphical form in Figure 4. An SDS Physical Component contains at least one Logical Device and provides the connection to the Network. A Logical Device contains at least one and up to 32 SDS Embedded Objects (see 3.1.10). A Physical Component may have several Logical Devices with different SDS Logical Addresses on the Network.

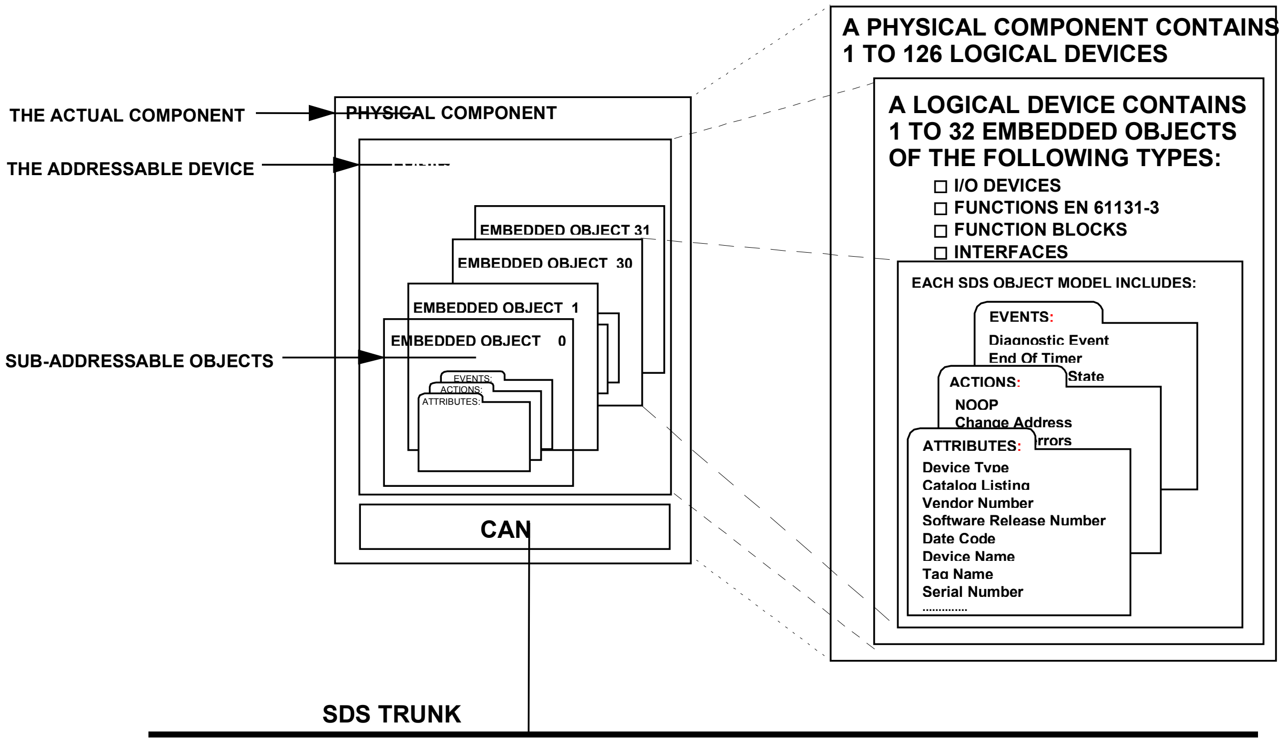


Figure 4 - Graphical representation of Component Model

An SDS Component Model shall include three types of elements: attributes, actions and events. These elements describe the behaviour of an SDS Embedded Object.

5.1.2.2 Attributes

Attributes are containers for the network visible data. Therefore a device network variables shall include the collection of attributes defined by each of its Embedded Objects (figure 4). Each attribute shall have a primitive tag which identifies an associated data type (see 5.3.4) along with data size and/or length specifications. Attributes may be read-only, read-write, or read-write with password protection.

5.1.2.3 Actions

Actions are SDS Embedded Object services which shall be invoked via an action request message which may include data. A device services an action in a manner similar to a subroutine call in a software program. The action response message may also include data.

5.1.2.4 Events

Events are SDS Embedded Object reports. A device shall issue an event as soon as the trigger condition described in the SDS Object Model is true. An event message may include data.

5.1.3 Hierarchy

NOTE SDS defines a mechanism to organize and manage models. These models are separated into major categories and arranged hierarchically.

Figure 5 shows the overview of the standard SDS hierarchy. The SDS Minimum Model specifies a minimum set of features that every device shall support. These features shall be inherited by other models such as I/O Devices, Interface Devices, etc., each of which shall also specify its own set of attributes, actions and events. These may in turn be inherited by other models which add another set of features. A particular model shall support the features of all the models it inherits.

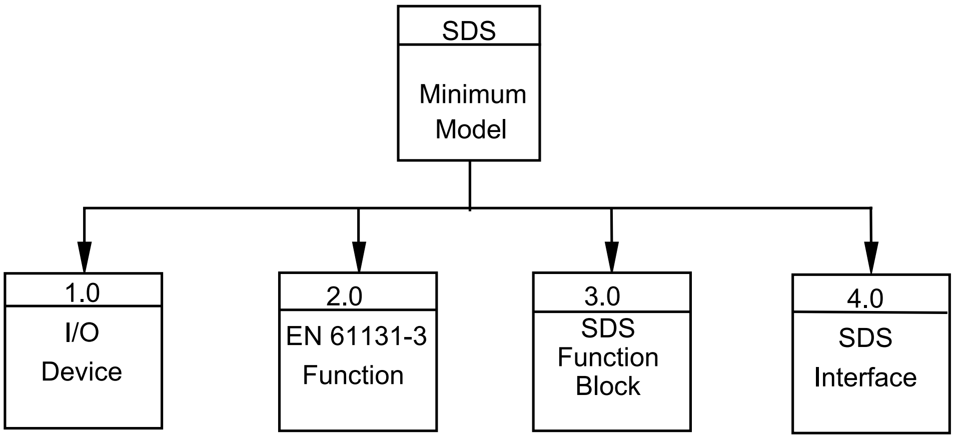


Figure 5 - SDS example of Models in hierarchical form

The topmost level of the hierarchy defines the structure and minimum behaviour of the Physical Component and Logical Device in the SDS Model. Every SDS product shall include this top level. Each lower level of the hierarchy defines additional SDS attributes, actions and events for SDS Embedded Objects at that level. SDS Embedded Objects at each level of the hierarchy shall inherit all of the features from all previous levels. A Logical Device may contain up to 32 Embedded Objects in any combination.

5.2 SDS Application Layer Services

5.2.1 Service conventions

The Application Layer shall provide the services listed in Table 1.

Table 1 - SDS Application Layer Services

Service	Function
Read	Allows the ALP service user to read the value of a device attribute.
Write	Allows the ALP service user to modify the value of a device attribute.
Event	Allows an ALP service user to report an event.
Action	Allows an ALP service user to command a device to execute an action.
Change Of State–ON	Specialized event service that reports a Change Of State to the ON condition of a binary Device Model
Change Of State–OFF	Specialized event service that reports a Change Of State to the OFF condition of a binary Device Model
Write State–ON	Specialized write service that writes an ON state to a binary output device
Write State–OFF	Specialized write service that writes an OFF state to a binary output device

The SDS Service Model uses four primitive functions to provide the Application Layer Services at the CAN Data Link Layer level: Request, Response, Indication and Confirm. The Primitives are shown in Figure 6 for the standard service and in Figure 7 for a fragmented service.

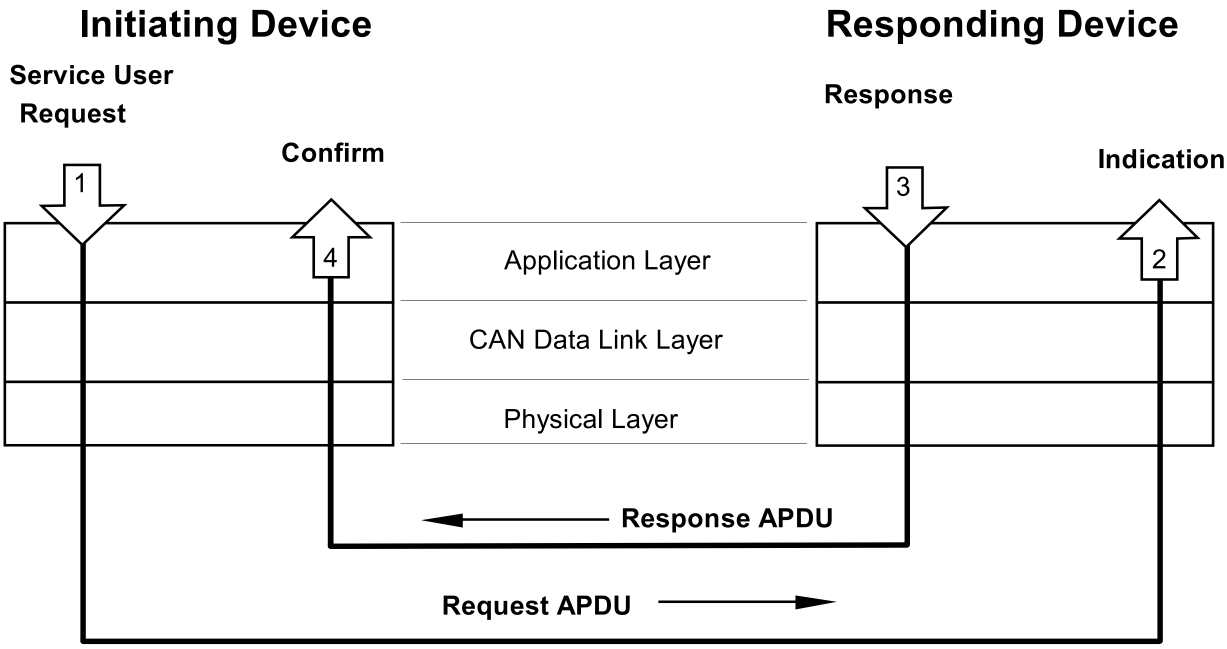


Figure 6 - Representation of the service Primitives (without fragmentation)

The following description corresponds to the service sequence shown in Figure 6. For example, when a sender invokes an application layer “Read” service, the following sequence of events occurs.

a) A **Request** Primitive presented to the Application Layer causes the generation of an Application Layer Protocol Data Unit by the initiating device.

b) The receipt of the Application Layer Protocol Data Unit by the responding device causes an **Indication** of the received message for the Service User at the responding device.

c) The Service User at the responding device presents a **Response** Primitive to the Application Layer that causes the generation of an Application Layer Protocol Data Unit.

d) The receipt of this APDU response by the initiating device causes a **Confirm** Primitive to be delivered to the Service User at the initiating device.

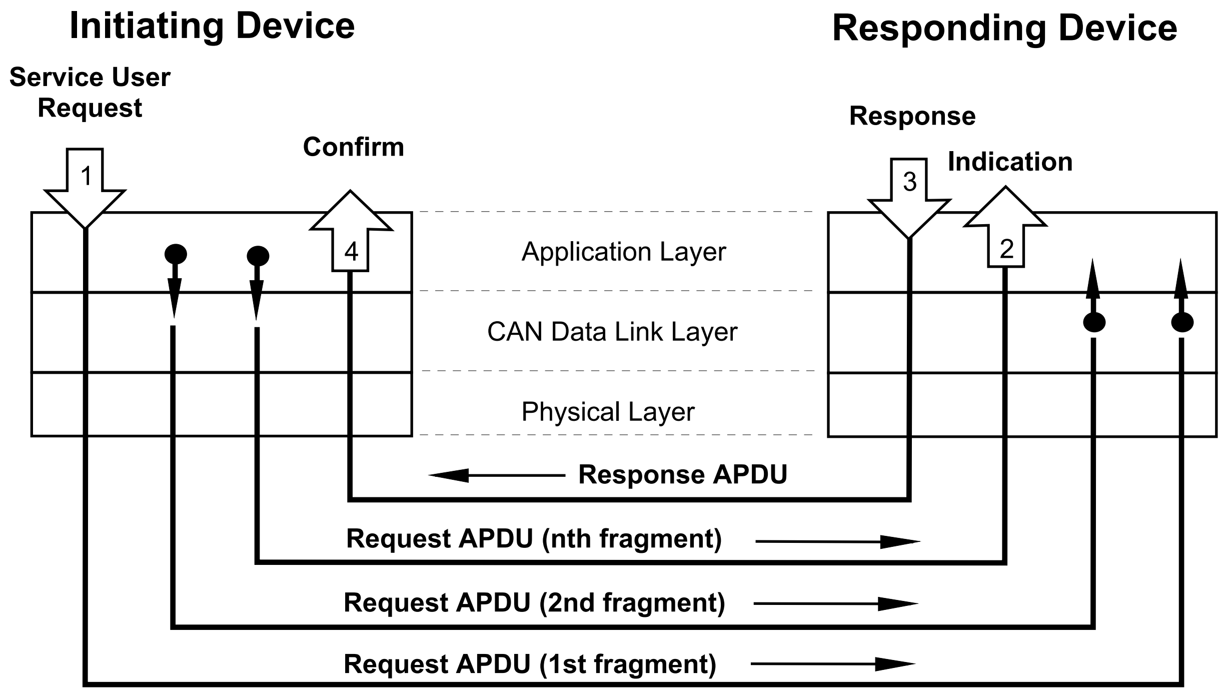


Figure 7 - Representation of the service Primitives with fragmentation

The Fragmented Service is used for information that exceeds the maximum data length of a single Application Layer Protocol Data Unit. The following sequence of events occurs.

a) A **Request** Primitive presented to the Application Layer causes the generation of the individual fragmented APDUs by the initiating device.

b) The receipt of the last APDU fragment by the responding device causes an **Indication** of the received message for the Service User at the responding device.

c) Following receipt of the Indication Primitive, the Service User at the responding device invokes the **Response** Primitive on the Application Layer, which causes the generation of a Response ADPU.

d) The receipt of this Response APDU by the initiating device causes a **Confirm** Primitive to be delivered to the Service User at the initiating device.

5.2.2 Read service

This service shall be used to read an attribute value of an Embedded Object. For example, it could be used to read the present value of a sensor. Table 2 describes the parameters defined for the read service.

Table 2 - Parameters defined for read service

Parameter	Request	Indication	Response	Confirm
Logical Address—EOID	Mandatory	(=) Request received	Mandatory	
Attribute ID	Mandatory	(=) Request received	Mandatory	
Result (+) Attribute ID Attribute Value			Conditional Mandatory Mandatory	 (=) Response received (=) Response received
Result (–) Attribute ID Error Code			Conditional Mandatory Mandatory	 (=) Response received (=) Response received

The functions of the parameters are as follows:

a) **Logical Address** identifies the device from which the attribute data is to be read. This is mandatory in both the Request and the Response.

b) **EOID (Embedded Object ID)** specifies which of the Embedded Objects is to be read. This is mandatory in both the Request and the Response.

c) **Attribute ID** specifies the attribute to be read. The value of the attribute ID is defined in the Object Model of the Embedded Object. This is mandatory in both Request and Response.

d) **Result (+)** The Result (+) parameter is returned if the read was successful. It returns the attribute ID and the value that was read.

e) **Result (–)** The Result (–) parameter is returned if the read fails. It returns the attribute ID and provides an error code specifying why the read request failed.

5.2.3 Write service

Shall be used to modify an attribute of an Embedded Object. For example, this service could be used to set an actuator output to ON or OFF. Table 3 describes the parameters defined for this service.

Table 3 - Parameters defined for write service

Parameter	Request	Indication	Response	Confirm
Logical Address—EOID	Mandatory	(=) Request received	Mandatory	(=) Response received
Attribute ID	Mandatory	(=) Request received	Mandatory	(=) Response received
Attribute Value	Mandatory	(=) Request received		
Result (+) Attribute ID			Conditional Mandatory	 (=) Response received
Result (–) Attribute ID Error Code			Conditional Mandatory Mandatory	 (=) Response received (=) Response received

The functions of the parameters are as follows:

a) **Logical Address** identifies the device to which the attribute data is to be written. This is mandatory in both the Request and the Response.

b) **EOID (Embedded Object ID)** specifies which of the Embedded Objects is to be written.

c) **Attribute ID** specifies the attribute to be written. The value of the ID is defined in the Object Model of the Embedded Object.

d) **Attribute Value** is the value to which the attribute is to be set.

e) **Result (+)** The Result (+) parameter returned with the attribute ID if the write was successful.

f) **Result (–)** The Result (–) parameter is returned if the write failed. It returns the attribute ID and provides an error code specifying why the write request failed.

5.2.4 Event service

Shall be used to report the occurrence of events specified for an Embedded Object. For example, a Logical Device might report a self-test failure. Table 4 describes the parameters defined for this service.

Table 4 - Parameters defined for event service

Parameter	Request	Indication	Response	Confirm
Logical Address—EOID	Mandatory	(=) Request received	Mandatory	(=) Response received
Event ID	Mandatory	(=) Request received	Mandatory	(=) Response received
Event Parameters	Conditional	(=) Request received		
Result (+) Event ID			Conditional Mandatory	 (=) Response received
Result (–) Event ID Error Code			Conditional Mandatory Mandatory	 (=) Response received (=) Response received

The functions of the parameters are as follows:

a) **Logical Address** identifies the device in which the event occurred.

b) **EOID (Embedded Object ID)** specifies which of the Embedded Objects is transmitting.

c) **Event ID** specifies the event that occurred. The value of the event ID is specified in the Object Model of the sending Embedded Object.

d) **Event Parameters** The event parameters are conditional on the type of event. They are defined in the Object Model specifications.

e) **Result (+)** The Result (+) parameter is returned with the event ID if the event was successful.

f) **Result (–)** The Result (–) parameter is returned if the event failed. It returns the event ID and provides an error code specifying why the event failed.

5.2.5 Action service

Shall be used to execute the actions specified for an Embedded Object. For example, the action service may be used to initiate a self-test. Table 5 describes the parameters defined for this service.

Table 5 - Parameters defined for action service

Parameter	Request	Indication	Response	Confirm
Logical Address—EOID	Mandatory	(=) Request received	Mandatory	(=) Response received
Action ID	Mandatory	(=) Request received	Mandatory	(=) Response received
Action Parameters	Conditional	(=) Request received		
Result (+) Action ID Action Results			Conditional Mandatory Conditional	 (=) Response received (=) Response received
Result (–) Action ID Error Code			Conditional Mandatory Mandatory	 (=) Response received (=) Response received

The functions of the parameters are as follows:

a) **Logical Address** identifies the device to which the action request is to be sent. This is mandatory in both the Request and the Response.

b) **EOID (Embedded Object ID)** specifies which of the Embedded Objects is to perform the action.

c) **Action ID** specifies the action to be performed. The value of the action ID is specified in the Object Model.

d) **Action Parameters** are dependent on the type of action. They are defined in the Object Model.

e) **Result (+)** The Result (+) parameter is returned with the action ID and action parameters.

f) **Result (–)** The Result (–) parameter is returned if the action request failed. It returns the action ID and provides an error code specifying why the action request failed.

5.2.6 Change Of State ON (COS ON) service

Shall be used by a Logical Device to report the occurrence of a change of its state to ON. This event service shall be used by a device which contains only a single binary Embedded Object (e.g., a single binary input). Table 6 describes the parameters defined for this service.

Table 6 - Parameters defined for Change Of State ON service

Parameter	Request	Indication	Response	Confirm
Logical Address	Mandatory	(=) Request received	Mandatory	(=) Response received
Result (+) COS ON ACK			Conditional Mandatory	 (=) Response received

The functions of the parameters are as follows:

a) **Logical Address** identifies the device in which the Change Of State to ON occurred.

b) **Result (+)** A Change of State ON Acknowledge (COS ON ACK) is returned when the report is successful.

c) **Result (–)** The Result (–) parameter is not defined for this service.

5.2.7 Change Of State OFF (COS OFF) service

Shall be used by a Logical Device to report the occurrence of a change of its state to OFF. This event service shall be used by a device which contains only a single binary Embedded Object (e.g., a single binary input). Table 7 describes the parameters defined for this service.

Table 7 - Parameters defined for Change Of State OFF service

Parameter	Request	Indication	Response	Confirm
Logical Address	Mandatory	(=) Request received	Mandatory	(=) Response received
Result (+) COS OFF ACK			Conditional Mandatory	(=) Response received

The functions of the parameters are as follows:

a) **Logical Address** identifies the device in which the COS to OFF occurred.

b) **Result (+)** A Change of State OFF Acknowledge (COS OFF ACK) is returned when the report is successful.

c) **Result (–)** The Result (–) parameter is not defined for this service.

5.2.8 Write State ON (WRITE ON) service

Shall be used to write an ON state to a Logical Device. This service shall be used by a device which contains only a single binary Embedded Object (e.g., a single binary output). Table 8 describes the parameters defined for this service.

Table 8 - Parameters defined for Write State ON service

Parameter	Request	Indication	Response	Confirm
Logical Address	Mandatory	(=) Request received	Mandatory	(=) Response received
Result (+) WRITE ON ACK			Conditional Mandatory	(=) Response received

The functions of the parameters are as follows:

a) **Logical Address** identifies the device which is to assume the ON state.

b) **Result (+)** A Write ON Acknowledge (WRITE ON ACK) is returned when the write is successfully received.

c) **Result (–)** The Result (–) parameter is not defined for this service.

5.2.9 Write State OFF (WRITE OFF) service

Shall be used to write an OFF state to a Logical Device. This service shall be used by a Device which contains only a single binary Embedded Object (e.g., a single binary output). Table 9 describes the parameters defined for this service.

Table 9 - Parameters defined for Write State OFF service

Parameter	Request	Indication	Response	Confirm
Logical Address	Mandatory	(=) Request received	Mandatory	(=) Response received
Result (+) WRITE OFF ACK			Conditional Mandatory	(=) Response received

The functions of the parameters are as follows:

a) **Logical Address** identifies the device that is to assume the OFF state.

b) **Result (+)** A Write OFF Acknowledge (WRITE OFF ACK) is returned when the write is successfully received.

c) **Result (–)** The Result (–) parameter is not defined for this service.

5.3 SDS Application Layer Protocol

NOTE This clause specifies the protocol used to implement SDS services, defines the Application Layer Protocol Data Unit (APDU) and describes APDU coding on the data link layer.

5.3.1 Application Protocol Data Unit (APDU)

5.3.1.1 General

NOTE A number of defined APDU types supports all SDS services.

APDU parts which are coded in a fixed format shall be as specified in 5.3.2; other APDU parts contain variable numbers of parameters which shall be either of standard SDS data types or constructed from them. These parameters shall be coded according to the specifications in 5.3.4.

5.3.1.2 Address coding within the APDU

The address field shall be coded in a fixed format. Each Physical Component on the Network shall consist of one or more Logical Devices, each with its own SDS address. Each Logical Device shall contain one or more embedded I/O Device or other Object Models, each with its own 5 bit Embedded Object Identifier.

5.3.1.3 SDS addressing within the CAN protocol

The SDS Application Layer Protocol (ALP) defines that the CAN Identifier bit 10 (ID_{10}) shall be the Direction/Priority bit and CAN Identifier bits 3 through 9 (ID_3 —ID_9) shall be the Logical Address bits. This allows Logical Addresses 0 through 125. The CAN frame format is shown in Figure 8.

	BITS								
	7	6	5	4	3	2	1	0	
BYTES	ID_{10}	ID_9	ID_8	ID_7	ID_6	ID_5	ID_4	ID_3	CAN Header
	ID_2	ID_1	ID_0	RTR	DLC 3	DLC 2	DLC 1	DLC 3	
	Data Byte 1								CAN Data Field
	Data Byte 2								
	Data Byte 3								
	Data Byte 4								
	Data Byte 5								
	Data Byte 6								
	Data Byte 7								
	Data Byte 8								

NOTE 1 Logical Addresses 126 and 127 may not be used because of restrictions in the CAN specification.

NOTE 2 CAN controllers when set to filter messages reduce the number of incoming messages that need to be serviced. In SDS the CAN controller filter is set to its own SDS address (Logical Device address).

Figure 8 - CAN frame format

5.3.2 APDU Forms

5.3.2.1 General

The Application Layer Protocol Data Unit shall take one of two basic forms: short form and long form. Long form APDUs may be non-fragmented or fragmented. A series of fragmented APDUs shall be used to accommodate data which exceeds the maximum size handled by the non-fragmented APDU.

NOTE Throughout this subclause, each field and subfield of each APDU is defined at its first occurrence in this document. Each of these field definitions is preceded by a diagram that shows the primary APDU fields. The applicable primary field will have a bold frame. A primary field that has any previous definitions (i.e. field or subfield) will have a double frame. Unused fields shall be set to a 0 value (see Diagram 1).

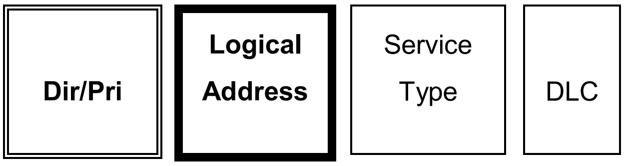


Diagram 1

5.3.2.2 Short form APDU specifications

5.3.2.2.1 Structure and format

The short form APDU shall be 44 bits in length. SDS shall use 16 of these bits (see Figure 9) and the remainder are used by CAN. The first 8 bits of the CAN Identifier shall be coded with the Direction/Priority (Dir/Pri) bit and Logical Address. The next 3 bits shall be used to specify the Service Type. The Remote Transmission Request (RTR) bit shall be set to 0. The Data Length Code (DLC) shall be the number of actual CAN data bytes. For short form APDUs, the DLC shall always be 0, as none of the CAN data bytes are used.

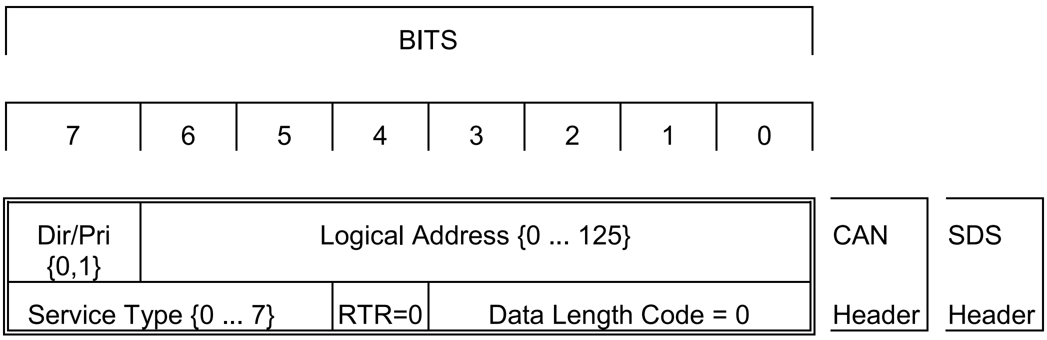


Figure 9 - Short form APDU structure

5.3.2.2.2 Direction/Priority (Dir/Pri) field

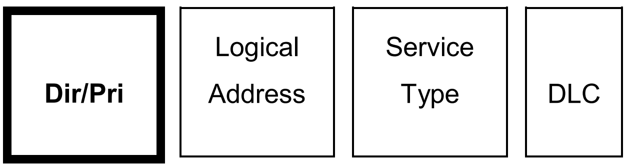


Diagram 2

In short form APDUs, the Dir/Pri bit shall determine the direction with respect to the content of the Logical Address field. If the Dir/Pri bit is 1, the address field contains a source address. If the Dir/Pri bit is 0, the address field contains a destination address. In short form APDUs, a destination addressed message shall always have higher priority than a source addressed message.

5.3.2.2.3 Logical Address field

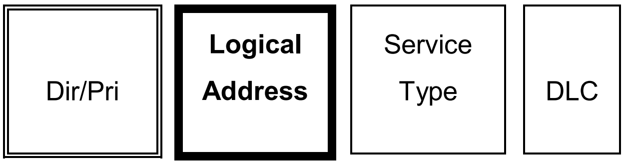


Diagram 3

For the short form APDU, CAN Identifier bits 3 through 9 (ID_3 – ID_9) shall be the Logical Address bits. This provides 7 address bits and permits Logical Addresses 0 to 125. Within the priority level set by the Dir/Pri bit, the logical address field shall determine the priority for access to the SDS network. The lowest address number shall have the highest priority.

5.3.2.2.4 Service type field

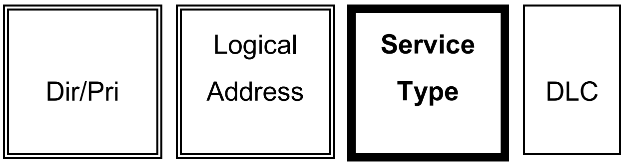


Diagram 4

The second byte of the short form APDU shall include the service type that identifies the short form message type. Devices which contain only a single binary Embedded Object shall use the short form service types. The type values shall be encoded in data bits 5 to 7 of the second byte (CAN ID_0 – ID_2). The short form service types and type values are shown in Table 10.

Table 10 - Short form service type values

Name	Value	Coding		
		ID_2	ID_1	ID_0
Change of State OFF	(0)	0	0	0
Change of State ON	(1)	0	0	1
Change of State OFF ACK	(2)	0	1	0
Change of State ON ACK	(3)	0	1	1
Write State OFF	(4)	1	0	0
Write State ON	(5)	1	0	1
Write State OFF ACK	(6)	1	1	0
Write State ON ACK	(7)	1	1	1

5.3.2.2.5 Data Length Code (DLC) field

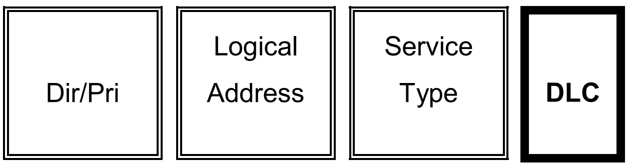


Diagram 5

The DLC field shall always represent the CAN Data field length in bytes. For short form APDUs, the DLC shall always be 0.

5.3.2.3 Generic Long form APDU specification

5.3.2.3.1 Structure and format

The long form APDU shown in Figure 10 shall be used for messages that require more information than the short form APDU may provide. The long form APDU shall be distinguished from the short form in that the long form has a non-zero Data Length Code. The long form shall also used to access objects in devices that contain more than one Embedded Object Identifier. The APDU shall be composed of data fields that hold addressing and service information. The primary information consists of the Dir/Pri bit, Logical Address, Embedded Object Identifier and the service parameters.

The service type field shall hold the type of the APDU. The RTR bit shall be set to 0. The Data Length Code shall indicate the number of CAN data bytes. The service specifiers field shall hold the Request/Response and Fragmentation Indicator subfields. The EOID field shall hold the Embedded Object Identifier. The service parameters field shall hold the service identifier and/or other service parameters. The service parameters field shall be coded in the second CAN data byte.

NOTE Some APDUs may use more than a single byte for their service parameters. The format of the service parameters is dependent on the service type.

	BITS 7	6	5	4	3	2	1	0		
BYTES	Dir/Pri {0,1}	Logical Address { Value 0 ... 125 }							CAN Header	SDS Header
	Service Type {0 ... 7}			RTR=0	Data Length Code {2 ... 8}					
	Service Specifiers			EOID						
	Service Parameters								CAN Data Field	
	Data (6 bytes maximum)									SDS Data Field

Figure 10 - Long form APDU format

5.3.2.3.2 Direction/Priority (Dir/Pri) field (generic)

Dir/Pri	Logical Address	Service Type	DLC	Service Specifiers	EOID	Service Parameters	Data

Diagram 6

In a long form APDU for service types read, write, action and event, this bit shall determine the direction with respect to the content of the Logical Address field. If the Dir/Pri bit is 1, the address field shall contain a source address, and if the Dir/Pri bit is 0, the address field shall contain a destination address.

5.3.2.3.3 Logical Address field (generic)

Dir/Pri	Logical Address	Service Type	DLC	Service Specifiers	EOID	Service Parameters	Data

Diagram 7

For the long form APDU, CAN Identifier bits 3 through 9 (ID_3 – ID_9) shall be the Logical Address bits. This provides 7 address bits and permits Logical Addresses 0 to 125. Within the priority level set by the Dir/Pri bit, the logical address field shall determine the priority for access to the SDS Network. The lowest address number shall have the highest priority.

5.3.2.3.4 Service type field (generic)

Dir/Pri	Logical Address	**Service Type**	DLC	Service Specifiers	EOID	Service Parameters	Data

Diagram 8

The long form service types include: read, write, action and event (see Table 11).

Table 11 - Long form service types

Name	Value	Coding		
		ID_2	ID_1	ID_0
Reserved	(0)	0	0	0
Reserved	(1)	0	0	1
Reserved	(2)	0	1	0
Reserved	(3)	0	1	1
Write	(4)	1	0	0
Read	(5)	1	0	1
Action	(6)	1	1	0
Event	(7)	1	1	1

5.3.2.3.5 Data Length Code (DLC) field (generic)

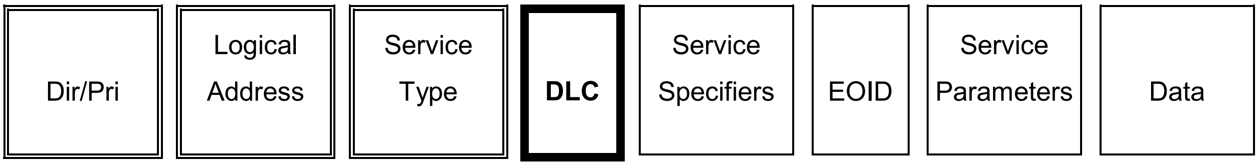


Diagram 9

Data bits 0 through 3 of the second byte of the long form APDU shall form the Data Length Code field that indicates the number of CAN data bytes. A long form APDU shall have at least 2 CAN data bytes (this shall include the service specifiers, EOID, the service parameters and optional data). The valid DLC range in a long form APDU is 2 to 8 inclusive (see Table 12).

Table 12 - Data Length Codes for long form APDUs

Value	Coding (Bits)			
	3	2	1	0
not valid	0	0	0	0
not valid	0	0	0	1
(2)	0	0	1	0
(3)	0	0	1	1
(4)	0	1	0	0
(5)	0	1	0	1
(6)	0	1	1	0
(7)	0	1	1	1
(8)	1	0	0	0

5.3.2.3.6 Service specifiers field (generic)

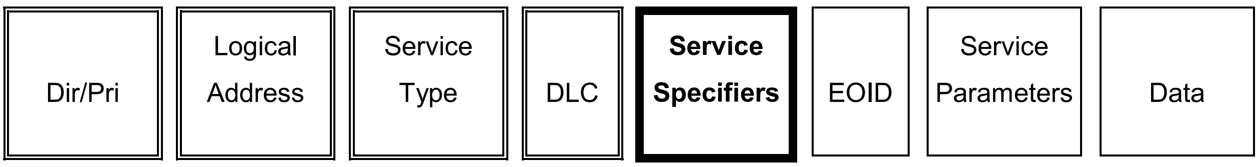


Diagram 10

The service specifiers field shall consist of the Request/Response and Fragmentation Indicator subfields.

The Request/Response subfield, bits 6 and 7 of the service specifiers field, shall specify whether the message is a request, successful response or an error response (see Table 13).

Table 13 - Request/Response subfield for long form APDUs

Request/Response	**Value**	**Bits**	
		7	6
Request	(0)	0	0
Successful Response	(1)	0	1
Error Response	(2)	1	0
Reserved	(3)	1	1

A request message shall request a specific service from a Service Provider.

A successful response message shall notify the initiating device that the requested service has been successfully completed.

An error response message (see Figure 11) shall notify the initiating device that an error has occurred during the processing of the requested service. A separate data field shall carry the error code.

	BITS									
	7	6	5	4	3	2	1	0		
B	Dir/Pri {0,1}	Logical Address { Value 0 ... 125 }							CAN	
Y	Service Type {0 ... 7}			RTR=0	**Data Length Code{2 ... 8}**				Header	SDS
T	**R/R = Error**		FI = 0	EOID {0 ... 31}					CAN	Header
E	**Service Parameters**								Data	
	Error Code									Data

Figure 11 - Error response APDU format

NOTE   The position of the error code in the APDU format may vary with respect to the service type due to variation in the length of the service parameters field.

See 5.3.3, error codes, for SDS error code specifications. The error code data type shall be Unsigned 8.

The Fragmentation Indicator (FI) subfield, bit 5 of the service specifiers field, shall specify whether the APDU format is fragmented or non-fragmented. The Fragmentation Indicator shall have the value 0 for non-fragmented APDUs and value 1 for fragmented APDUs (see Table 14).

Table 14 - Fragmentation Indicator (FI) subfield

Fragmentation Indicator	Value	Bit
		5
Non-fragmented APDU Format	(0)	0
Fragmented APDU Format	(1)	1

5.3.2.3.7 Embedded Object Identifier (EOID) field (generic)

Dir/Pri	Logical Address	Service Type	DLC	Service Specifiers	**EOID**	Service Parameters	Data

Diagram 11

Bits 0 to 4 inclusive of the first CAN Data Byte shall form the Embedded Object Identifier. This identifier shall be used as a selector to distinguish the specific Embedded Object addressed by the particular APDU (see Table 15).

Table 15 - Embedded Object Identifier field

Value	Coding (Bits)				
	4	3	2	1	0
(0)	0	0	0	0	0
(1)	0	0	0	0	1
:	:	:	:	:	:
(30)	1	1	1	1	0
(31)	1	1	1	1	1

5.3.2.3.8 Service Parameters (generic)

Dir/Pri	Logical Address	Service Type	DLC	Service Specifiers	**EOID**	Service Parameters	Data

Diagram 12

The Service Parameters field shall be used to determine how an APDU will be processed. Its definition varies according to the service type.

5.3.3 Error Codes

When the service specifiers indicate the message is an error response, the data portion of the APDU shall contain an error code as shown in Table 16.

Table 16 - APDU error codes

Error Code	Error Code Name	Definition
0	Service Not Available	Service type was {0 ... 3} with DLC (CAN Data Length Code) > 0.
1	Illegal Service Parameters	Requested service parameters does not exist in this object.
2	Read Only Variable	A write action was requested of a read-only variable.
3	Illegal Data	The data for a specific variable is incorrect.
4	EEPROM Read Error	Non-volatile memory error.
5	Time Out	Resource was not available.
6	Unable To Respond	Request could not be honoured at this time, unable to respond.
7	Embedded Object Unavailable	The embedded object addressed is not available.
8	Illegal Object	EOID value out of range for Logical Device.
9	Illegal DLC	DLC value does not match data byte count.
10-253	Reserved	
254	ALP Service Timeout	No response to a request message within the specified time limit.
255	Other Network	Other Network level error.

5.3.4 Data types

5.3.4.1 General

SDS data types shall be defined by Embedded Object specifications for each service parameter. The variable data in the APDU contains parameters that are one of the SDS base data types or constructed from them. The values shall be coded with the most significant byte (MSB) first, and for each byte, the most significant bit first with the entire value least significant byte (LSB) justified. The coding for each data type is described in the following paragraphs.

- Boolean
- Unsigned Integer
- Signed Integer
- Real Number
- Character String

5.3.4.2 Boolean value

Boolean value shall be coded in a single byte, with the value 0x00 for FALSE and 0x01 for TRUE.

5.3.4.3 Unsigned 8 Integer

The data type Unsigned 8 Integer shall be coded as binary numbers in the range 0 to 255. Unsigned Integer data elements shall be coded in one byte (8 bits). Example: The decimal value 70 is coded as 0x46 (see Figure 12).

BITS							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	1	0	0	0	1	1	0

Figure 12 - Unsigned 8 Integer

5.3.4.4 Unsigned 16 Integer

Data elements shall be coded in two bytes (16 bits) in the range of 0 to 65 535. Example: The value 258 is coded as 0x0102 (see Figure 13).

	BITS							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MSB	0	0	0	0	0	0	0	1
LSB	0	0	0	0	0	0	1	0

Figure 13 - Unsigned 16 Integer

5.3.4.5 Unsigned 32 Integer

Data elements shall be coded in four bytes (32 bits) with a maximum value of 4 294 967 295. For example, a value of 8 262 is coded as 0x00002046 (see Figure 14).

	BITS							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MSB	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	1	0	0	0	0	0
LSB	0	1	0	0	0	1	1	0

Figure 14 - Unsigned 32 Integer

5.3.4.6 Signed 8 Integer

Signed Integers shall be coded as binary numbers using 2's complement notation. A Signed 8 Integer value requires 1 byte, and for negative values, data bit 7 shall be 1. Figure 15 shows a Signed 8 Integer value representing -127 decimal, coded with data bit 7 indicating the sign bit value as negative.

BITS							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
1	0	0	0	0	0	0	1

Figure 15 - Signed 8 Integer

5.3.4.7 Signed 16 Integer

A Signed 16 Integer value requires 2 bytes, and for negative values, data bit 7 of the most significant byte (MSB) shall be 1. Figure 16 shows the bit values for a negative 0x46

	BITS							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MSB	1	1	1	1	1	1	1	1
LSB	1	0	1	1	1	0	1	0

Figure 16 - Signed 16 Integer

5.3.4.8 Signed 32 Integer

A Signed 32 integer value requires 4 bytes, and for negative values, data bit 7 of the most significant byte (MSB) shall be 1. Figure 17 shows the bit values for a positive value of 2 105 414, coded as 0x00202046.

	BITS							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MSB	0	0	0	0	0	0	0	0
	0	0	1	0	0	0	0	0
	0	0	1	0	0	0	0	0
LSB	0	1	0	0	0	1	1	0

Figure 17 - Signed 32 Integer

5.3.4.9 Real numbers

Single precision real numbers shall be coded in four bytes. No double precision real numbers shall be supported.

	BITS							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
B	Sign	Exponent						
Y	EXP	Mantissa						
T	Mantissa							
E	Mantissa							

Figure 18 - Real numbers

5.3.4.10 Character strings

Each character shall be coded in one byte using its ASCII value.

5.3.4.11 Date

The date code for the date of manufacture shall be a four byte ASCII string starting with the year tens digit, yy/ww, week of product manufacture (see Figure 19).

The date code for other dates shall be six bytes, beginning with the year tens digit and is in the form of yy/mm/dd (see Figure 20).

	BITS							
	7	6	5	4	3	2	1	0
B	Year tens digit							
Y	Year units digit							
T	Week tens digit							
E	Week units digit							

Figure 19 - Format for ‘manufacturing date’ coding

	BITS							
	7	6	5	4	3	2	1	0
	Year tens digit							
B	Year units digit							
Y	Month tens digit							
T	Month units digit							
E	Day tens digit							
	Day units digit							

Figure 20 - Standard date coding

5.4 SDS APDUs embedded in CAN frames

SDS APDUs shall be embedded in CAN frames as shown in the examples given in Figure 21, Figure 22 and Figure 23.

44 Bits (without STUFF bits)	

SDS Header	SDS Footer	SDS Fields
CAN Header	CAN Footer	CAN Fields

SOF	Dir/ Pri	Logical Address	Service Type	*RTR, RB0*	DLC	*CRC*	*ACK*	*EOF*	FIELD NAME
1	1	7	3	*3*	4	*16*	*2*	*7*	BITS/FIELD
1	{0,1}	{0 ... 125}	{0 ... 7}	0	{0}	---	---	---	VALUE RANGE

	BITS		
	2	1	0
	DLC = 0		
COS OFF	0	0	0
COS ON	0	0	1
COS OFF ACK	0	1	0
COS ON ACK	0	1	1
WRITE OFF	1	0	0
WRITE ON	1	0	1
WRITE OFF ACK	1	1	0
WRITE ON ACK	1	1	1

NOTE 1 Italicized fields may not be changed

NOTE 2 The following abbreviations are defined in the CAN specification ISO 11898.

SOF	Start Of Frame
RTR	Remote Transmission Request
CRC	Cyclical Redundancy Check
ACK	Acknowledge Field
EOF	End of Frame

Figure 21 - SDS short form APDU

108 Bits (without STUFF bits)																
SDS Header								**SDS Data Field**						**SDS Footer**		**SDS Fields**
CAN Header						**CAN Data Field**								**CAN Footer**		**CAN Fields**
SOF	**Dir/ Pri**	**Logical Address**	**Service Type**	***RTR, RB0***	**DLC**	**APDU Modifier**	**Service Parameters**	**Data**	**Data**	**Data**	**Data**	**Data**	**Data**	***CRC***	***ACK***	***EOF***	**FIELD NAME**
1	**1**	**7**	**3**	***3***	**4**	**8**	**8**	**8**	**8**	**8**	**8**	**8**	**8**	***16***	***2***	***7***	**BITS/FIELD**
1	**{0,1}**	**{0 ... 125}**	**{0 ... 7}**	**0**	**{2...8}**	**{0 ... 255}**	**{0 ... 255}**	**{0 ... 255}**	**{0 ... 255}**	**{0 ... 255}**	**{0 ... 255}**	**{0 ... 255}**	**{0 ... 255}**	**---**	**---**	**---**	**VALUE RANGE**

Figure 22 - SDS long form APDU

108 Bits (without STUFF bits)

SDS Header	SDS Data Field	SDS Footer	SDS Fields
CAN Header	CAN Data Field	CAN Footer	CAN Fields

SOF	Dir/ Pri	Logical Address	Service Type	*RTR, RB0*	DLC	APDU Modifier	Service Parameters	Fragment Number	Total Bytes	Data	Data	Data	Data	*CRC*	*ACK*	*EOF*	FIELD NAME
1	1	7	3	*3*	4	8	8	8	8	8	8	8	8	*16*	*2*	*7*	BITS/FIELD
1	{0,1}	{0 ... 125}	{0 ... 7}	0	{2....8}	{0 ... 255}	{0 ... 255}	{0 ... 63}	{0 ... 255}	{0 ... 255}	{0 ... 255}	{0 ... 255}	{0 ... 255}	---	---	---	VALUE RANGE

Service Type – BITS

	2	1	0
	DLC ≠ 0		
RESERVED	0	0	0
RESERVED	0	0	1
RESERVED	0	1	0
RESERVED	0	1	1
WRITE	1	0	0
READ	1	0	1
ACTION	1	1	0
EVENT	1	1	1

Service Parameters – BITS

	7	6
	RRE	
REQUEST	0	0
SUCCESSFUL RESPONSE	0	1
ERROR RESPONSE	1	0
RESERVED	1	1

	5
	FI
NON-FRAGMENTED	0
FRAGMENTED	1

	4	3	2	1	0
	EOID				
OBJECT 0	0	0	0	0	0
OBJECT 1	0	0	0	0	1
OBJECT 2	0	0	0	1	0
OBJECT 3	0	0	0	1	1
- - - - - - - -	-	-	-	-	-
OBJECT 29	1	1	1	0	1
OBJECT 30	1	1	1	1	0
OBJECT 31	1	1	1	1	1

NOTE Italicized fields may not be changed.

Figure 23 - SDS long form fragmented APDU

5.5 Example Short Form APDUs

Example Short Form APDUs are shown in Figure 24 and Figure 25.

SDS Header	
CAN Header	CAN Footer

SOF	Dir/ Pri	Logical Address	Service Type	*RTR*	Reserved RB1,RB0	DLC	CRC	ACK	EOF	FIELD NAME
1	1	7	3	*1*	2	4	16	2	7	BITS/FIELD
---	**1**	**001 1001**	**000**	**0**	---	**0000**	---	---	---	VALUES (BINARY)
---	Dir = From L. Add	25	COS OFF	not used	---	No Data	---	---	---	DESCRIPTION

Figure 24 - COS OFF APDU from Logical Address 25

SDS Header	
CAN Header	CAN Footer

SOF	Dir/Pri	Logical Address	Service Type	*RTR*	Reserved RB1,RB0	DLC	*CRC*	*ACK*	*EOF*	FIELD NAME
1	1	7	3	*1*	2	4	16	2	7	BITS/FIELD
---	**0**	**001 1001**	**010**	**0**	---	**0000**	---	---	---	VALUES (BINARY)
---	Dir = To L. Add	25	COS OFF ACK	not used	---	No Data	---	---	---	DESCRIPTION

Figure 25 - COS OFF ACK APDU to Logical Address 25

5.6 Example Long Form APDUs

Example Long Form APDUs are shown in Figure 26 and Figure 27.

SDS Header													
CAN Header							CAN Data Field			CAN Footer			
SOF	Dir/Pri	Logical Address	Service Type	*RTR*	Reserved RB1,RB0	DLC	Service Specifiers RRE,FI	EOID	Service Parameters	*CRC*	*ACK*	*EOF*	FIELD NAME
1	1	7	3	*1*	2	4	3	5	8	*16*	*2*	*7*	BITS/FIELD
---	**0**	**001 0000**	**101**	**0**	---	**0010**	**000**	**0 0000**	**0000 1000**	---	---	---	VALUES (BINARY)
---	Dir = To L. Add	16	Read	not used	---	2 Bytes	Request	Object 0	Attribute 8	---	---	---	DESCRIPTION

Figure 26 - Read Request to Logical Address 16, Embedded Object 0, Variable 8

SDS Header										SDS Data				
CAN Header							CAN Data Field				CAN Footer			
SOF	Dir/Pri	Logical Address	Service Type	*RTR*	Reserved RB1,RB0	DLC	Service Specifiers RRE,FI	EOID	Service Parameters	Data	*CRC*	*ACK*	*EOF*	FIELD NAME
1	1	7	3	*1*	2	4	3	5	8	8	*16*	2	*7*	BITS/FIELD
---	**1**	**001 0000**	**101**	**0**	---	**0011**	**010**	**0 0000**	**0000 1000**	**0000 0011**	---	---	---	VALUES (BINARY)
---	Dir = From L. Add	16	Read	not used	---	3 Bytes	S. Response	Object 0	Attribute 8	0x03	---	---	---	DESCRIPTION

Figure 27 - Read Response from Logical Address 16, Embedded Object 0, Variable 8

6 Product information

6.1 Instructions for installation, operation and maintenance

Instructions for installation, operation and maintenance shall be in accordance with 6.1 of EN 50325-1.

6.2 Marking

Marking shall be in accordance with 6.2 of EN 50325-1.

7 Normal service, transport and mounting conditions

7.1 Normal service conditions

7.1.1 General

Components of an SDS Network shall be capable of operating under the following conditions.

NOTE If the conditions for operation differ from those given in this standard, the user shall state the deviation from the standard conditions and consult the manufacturer on the suitability for use under such conditions.

7.1.2 Ambient air temperature

SDS components shall operate between the ambient temperatures of -25 °C to +70 °C if not otherwise defined, e.g, in conjunction with a specific actuator or sensor type. The operating characteristics shall be maintained over the permissible range of ambient temperature.

7.1.3 Altitude

SDS components shall be capable of operating at altitude in accordance with 7.1.3 of EN 50325-1

7.1.4 Humidity

SDS components shall be capable of operating in humid conditions in accordance with 7.1.4 of EN 50325-1.

7.1.5 Pollution degree

SDS components shall be capable of operating in polluted conditions in accordance with 7.1.5 of EN 50325-1.

7.1.6 Sealed connectors

Sealed connectors, when provided with SDS components, shall be in accordance with 7.1.6 of EN 50325-1.

7.2 Conditions during transport and storage

Conditions during transport and storage shall be in accordance with 7.2 of EN 50325-1.

7.3 Mounting

SDS components shall be mounted in accordance with 7.3 of EN 50325-1

8 Constructional and performance requirements

NOTE Constructional requirements are verified by inspection and the performance requirements are verified by the tests of clause 9.

8.1 SDS Physical Layer Interface (PLI)

8.1.1 SDS power PLI

8.1.1.1 General

An SDS PLI shall be operational with SDS voltage between 11 and 25 volts dc. An SDS PLI may use SDS power to supply its microcontroller and other circuitry not directly related to SDS. An SDS device may supply SDS power to other devices which are not connected directly to the SDS power lines. Any such power that a node derives from the SDS power lines shall be considered to be SDS power.

8.1.1.2 SDS power requirements PLI

There is no specified limit for device SDS current consumption, however the maximum value shall be specified by the manufacturer.

8.1.1.3 Auxiliary power PLI

In addition to SDS power an SDS node may also have an auxiliary power port. Auxiliary power is not limited to any specific type (e.g., DC, AC, or 3 ∅ AC).

8.1.1.4 Isolation PLI

An SDS device shall provide galvanic isolation as specified in the relevant product standards between the SDS power and any auxiliary power as well as between either power source and the device case.

8.1.1.5 Shield line connection PLI

The shield line shall not be connected to the device SDS power or the device case. The device case shall not be connected to either the SDS power or auxiliary power.

8.1.2 Transceivers

An embedded transceiver shall connect each Physical Component to the SDS Network and gives these devices the capability of both transmitting and receiving to and from each other via the Network.

NOTE 1 The Discrete transceiver is comprised of discrete electronic components. This option provides extremely low power consumption with quiescent supply current typically 0 mA.

NOTE 2 The Integrated transceiver is a single electronic package, typically an integrated circuit component. The benefit of this option is reduced size.

NOTE 3 The Optocoupled transceiver has the additional feature of galvanic isolation of the input/output terminals providing maximum protection from power system faults and ground loops.

8.1.3 Transceiver specifications

8.1.3.1 Discrete transceiver specifications

The discrete transmitter and receiver specifications shall be in accordance with Table 17 and Table 18.

Table 17 - Discrete transmitter specification

Electrical Characteristics	Specifications
Differential Output Level, OFF	-0.5 to 0.05 V
Differential Output Level, ON	1.5 to 4.0 V
Differential Input Impedance (R_{diff}), OFF (minimum)	45 kΩ
$I_{standby}$ (typical)	0 mA
Propagation delay (maximum)	30 ns
Speed (maximum)	1 Mbit/s

Table 18 - Discrete receiver specification

Electrical Characteristics	Specifications
Differential Input Levels, OFF	-0.5 to 0.6 V
Differential Input Levels, ON	0.9 to 5.0 V
Hysteresis (typical)	70 mV
Propagation Delay (maximum)	30 ns

8.1.3.2 Integrated transceiver specification

The integrated transmitter and receiver specifications shall be in accordance with Table 19 and Table 20.

Table 19 - Integrated transmitter specification

Electrical Characteristics	Specifications
Differential Output Levels, OFF	-0.5 V to 50 mV
Differential Output Levels, ON	1.5 V to 3.0 V
Differential Input Impedance (R_{diff}) (min)	20 kΩ
$I_{recessive\ CAN\ state}$ (maximum)	18 mA
Propagation Delay (maximum)	40 ns
Speed (maximum)	1 Mbit/s

Table 20 - Integrated receiver specification

Electrical Characteristics	Specifications
Differential Input Levels, OFF	-1.0 V to 0.4 V
Differential Input Levels, ON	1.0 V to 5.0 V
Hysteresis (typical)	150 mV
Propagation Delay (maximum)	80 ns

8.1.3.3 Optocoupled transceiver specification

The optocoupled transmitter and receiver specifications shall be in accordance with Table 21 and Table 22.

Table 21 - Optocoupled transmitter specification

Electrical Characteristics	Specifications
Differential Output Levels, OFF	-0.5 V to 50 mV
Differential Output Levels, ON	1.5 V to 3.0 V
Differential Input Impedance (R_{diff}) (min)	20 kΩ
$I_{recessive\ CAN\ state}$ (maximum)	18 mA
Propagation Delay (maximum)	100 ns
Speed	1 Mbit/s

Table 22 - Optocoupled receiver specification

Electrical Characteristics	Specifications
Differential Input Levels, OFF	-1.0 V to 0.4 V
Differential Input Levels, ON	1.0 V to 5.0 V
Hysteresis (typical)	150 mV
Propagation Delay (maximum)	140 ns

8.1.4 Indicating means

SDS physical devices may include one or more coloured indicators. If coloured indicators are used they shall have the following meaning:

a) GREEN - communication in progress

b) RED - fault (system or communication)

8.2 SDS Network

8.2.1 Topology

The physical arrangement and spacing of the trunk, branches and nodes (see Figure 1) shall comply with the requirements detailed below. A standard SDS topology is a single trunk with short branches. The minimum separation of the nodes is 300 millimeters for SDS standard cables and 100 millimeters for conventional wiring. With standard cables no more than four (4) tees shall be connected directly together without a section of trunk cable between them. In the case of conventional wiring, multiple connections to the same terminal are not allowed.

One of the main factors affecting Network length and physical layout of an SDS system is voltage drop on the power pair of the SDS cable. Factors other than the physical layout of the Network that determine voltage drop are:

a) Resistance of the conductors and connections

b) Current required by each node

The difference in the power supply voltage between any two SDS Physical Components shall not exceed 2 volts dc.

Table 23 shows the general requirements for SDS bus cable.

Table 23 - SDS cable general requirements

Characteristics	**Specifications**
Cable Construction	Two shielded, twisted pairs, with a common earth wire, all within a single jacket
Keying	Required
Operating Temperature	-30°C to + 70°C
Plating (minimum)	0,76µ m Au over 0,76 µ m Ni
Power & Temperature Cycling	Mated connectors must withstand cycling, i.e.no electrical discontinuities greater than 1 µs, between -30°C and + 70°C ramped at 1°C to 3°C per minute with 30 minute dwell at 70°C, and with power (6.0 mA @ 5 V) continuously cycled on/off (50K times total) concurrent with (35) complete (up/down) temperature cycles.
Storage Temperature	-40°C to + 80°C

Table 24 gives the specifications for SDS standard cables which have either Mini or Micro interface connectors.

Table 24 - SDS standard cable specifications

Category	**MINI Specification**	**MICRO Specification**
Pin/Socket Resistance (typical)	1 mΩ	4 mΩ
Current Rating (minimum)	8 A	3 A
Dielectric Withstanding	1100 Vrms	1100 Vrms
Insulation Resistance (minimum)	10^9 Ω	10^9 Ω
Voltage Rating	300 V	300 V

Table 25 gives maximum system trunk lengths, maximum branch lengths and maximum nodes for various data rates, without taking in account the voltage drop.

Table 25 - Maximum trunk and branch lengths and maximum nodes for various data rates

Data Transmission Rate	**Maximum trunk Length (m)**	**Maximum branch Length (m)**	**Maximum number of nodes**
1 Mbit/s	22	0.3	32
500 kbit/s	91	0.9	64
250 kbit/s	182	1.8	64
125 kbit/s	457	3.6	64

Table 26 gives the wire terminations and colour code for SDS standard cables.

Table 26 - SDS standard cable wire termination and colour code

Mini Pin	**Micro Pin**	**Wire Insulation Colour**	**Function**
1	----	----	Bus Shield
2	1	Brown	Bus Power (V+)
3	2	Blue	Bus Power (GND)
4	4	Black	Bus Communications (CAN_H)
5	3	White	Bus Communications (CAN_L)

8.2.2 SDS power distribution

An SDS power supply shall be connected to the trunk in only one place and the power return line shall only be connected to earth at the power supply.

8.2.3 Auxiliary power ground connection

Figure 28 shows the network grounding connections for a Network which has both SDS power and auxiliary power. There shall be a single connection between the power supply V-. There shall be a single ground connection to this power supply return interconnection.

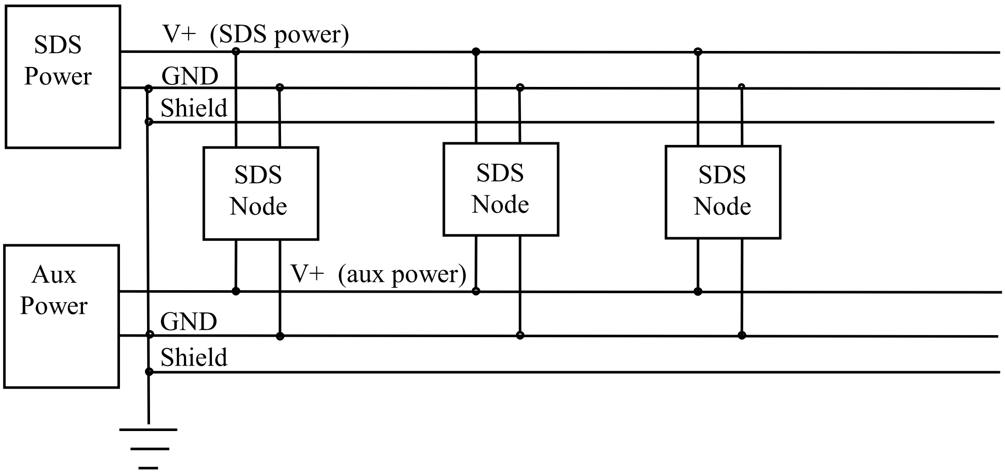


Figure 28 - Network grounding

8.3 Electromagnetic Compatibility (EMC)

8.3.1 General requirements for electromagnetic compatibility tests

The operating characteristics of SDS shall be maintained at the specified levels of electromagnetic interference (EMI).

The SDS device to be tested shall have all the essential design details of the type which it represents and shall be in a clean and new condition. Maintenance or replacement of parts during or after a testing cycle is not permitted.

Compliance with this requirement can be demonstrated using a network comprising one controller, one device and one power supply.

8.3.2 General test conditions for electromagnetic compatibility tests

Unless otherwise stated the tests shall be carried out at an ambient temperature of +23 ± 5° C.

The tests shall be performed under the following conditions:

The SDS device mounted in free air shall be connected to the SDS line and supplied with its rated operational voltage. Terminals and connectors, if any, shall be connected with the intended sensors or actuators according to the manufacture's instructions. The test shall be performed at each baud rate with the branch length specified in Table 27.

Table 27 - Maximum Branch Lengths

Baud Rate (K bits/s)	Branch Length (m)
125	3.6
250	1.8
500	0.9
1000	0.3

8.3.3 Immunity requirements

8.3.3.1 Performance criteria

8.3.3.1.1 Performance criteria A

The test specimen shall continue to operate as intended during and after the test.

No change of operating state or stored data is allowed within the equipment under test (EUT).

a) Sensors, I/O Devices, I/O Devices w/Ext. Power, and Controller:
- Any I/O interface line shall not have a false pulse > 1 μs.
- All input stimulus shall match the resulting output state in node/controller pairs.
- Stimulus may range from 10 to 100 Hz.

b) Allowable degradation in performance:
- audible signals and/or visible indicators not associated with control I/O
- temporary operating point deviations
- excessive data bus traffic (transmission retries, error frames, etc..)

c) Not allowable degradation in performance:
- change in analogue value > 10 X nominal tolerance
- change in digital value
- false I/O data representation in controller
- change of stored data in a device
- system stops operating and requires to be reset
- device reset indication
- device self test failure indication

8.3.3.1.2 Performance criteria B

During the tests a temporary loss of the data communication may occur, thereafter the SDS device shall continue to operate as intended.

No change of actual operating state or stored data shall be allowed.

8.3.3.2 Electrostatic discharge (ESD) immunity

The SDS device, when tested in accordance with EN 61000-4-2, shall

a) withstand the application of 10 positive and 10 negative pulses to each selected point;

b) withstand a voltage of 4 kV applied by the contact discharge method to all conductive surfaces;

c) withstand a voltage of 8 kV applied by the air gap discharge method to all non-conductive surfaces.

Performance criteria B shall be applied.

8.3.3.3 Radiated radio-frequency electromagnetic field immunity

The SDS device shall withstand severity level 3 (10 V/m) when tested in accordance with EN 61000-4-3 at a frequency band of 80 MHz to 1000 MHz, amplitude modulated.

The SDS device shall withstand severity level 3 (10 V/m) when tested in accordance with EN 61000-4-3 at a frequency band of 900 MHz $\pm$ 5 MHz, pulse modulated.

Performance criteria A shall be applied.

8.3.3.4 Conducted radio-frequency disturbance immunity

The SDS device shall withstand a voltage of 10 Vrms when tested in accordance with EN 61000-4-6 at a frequency band of 150 kHz to 80 MHz, amplitude modulated.

Performance criteria A shall be applied.

8.3.3.5 Fast transient/burst immunity

The SDS device shall be capable of withstanding 5/50 Tr/Th ns transients at a 5 kHz repetition frequency when tested in accordance with EN 61000-4-4, modified as detailed in 9.7.2, at the following voltage levels.

a) Ports for signal lines and data buses not involved in process control shall withstand a level of 1 kV (peak).

b) Ports for process, measurements and control lines, and long bus and control lines shall withstand a level of 2 kV (peak).

c) Ports for DC input and output power shall withstand a level of 2 kV (peak).

d) Ports for AC input and output power shall withstand a level of 2 kV (peak).

Performance criteria B shall be applied.

8.3.3.6 Surge immunity

NOTE For target and I/O devices it is not necessary to test for surge immunity. The operating environment of these devices is considered to be well protected against surge voltages caused by lightning strikes.

The SDS device shall be capable of withstanding a 1.2/50 (8/20) Tr/Th μs surge when tested in accordance with EN 61000-4-5 at the following voltage levels.

a) Ports for process, measurement and control lines shall withstand levels of 2 kV for common mode and 1 kV for differential mode.

b) DC input and output power ports shall withstand levels of 0.5 kV for common mode and 0.5 kV for differential mode.

c) AC input and output ports shall withstand levels of 4.0 kV for common mode and 2.0 kV for differential mode.

Performance criteria B shall be applied.

8.3.3.7 Immunity to supply interruptions

The system shall be capable of operating with SDS power supply interruptions of less than or equal to 1 millisecond.

8.3.4 Emission requirements

8.3.4.1 Conducted emissions

SDS devices shall meet the requirements of EN 50081-2 when tested in accordance with EN 55011.

8.3.4.2 Radiated emissions

SDS devices shall not produce radiated emissions that exceed the limits given in Table 28 when tested in accordance with EN 55011 for Class A Group 1 equipment.

Table 28 - Radiated emissions test limits

Frequency Band (MHz)	Measured on a test site at 30 m distance (dBμ V/m)
30 - 230	30
230 - 1000	37

These limits are given for SDS devices exclusively used in an industrial environment. When they may be used in a domestic environment the following warning shall be included in the instructions for use:

ATTENTION

This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

9 SDS Communication channel type tests

9.1 General

The Smart Distributed System communication channel of a product shall be verified by the following tests.

NOTE Standard product addressing conventions are used throughout this clause. User Addresses are normally used in the test descriptions (i.e. the primary address references are User Addresses). However, all references to addresses as they appear on the Network or in actual message fields, are Logical Addresses.

9.2 Product Model

The manufacturer's Product Model, including the Object Model(s) used by the EUT, is the basis for the tests of this clause. The purpose of these tests is to verify the EUT conforms to:

a) the manufacturer's Product Model

b) the Object Model

c) all applicable Smart Distributed System specifications

d) means of connection

e) transceiver type

9.3 Object Model test

9.3.1 General

All attributes, actions and events defined in the Object Model documentation shall be verified with the SDS power supply set to DC 11V, unless otherwise specified.

The device conformance test set-up includes a single controller (Device Conformance Tester), a single device (EUT), a power supply and a network monitor. The set-up is shown in Figure 29.

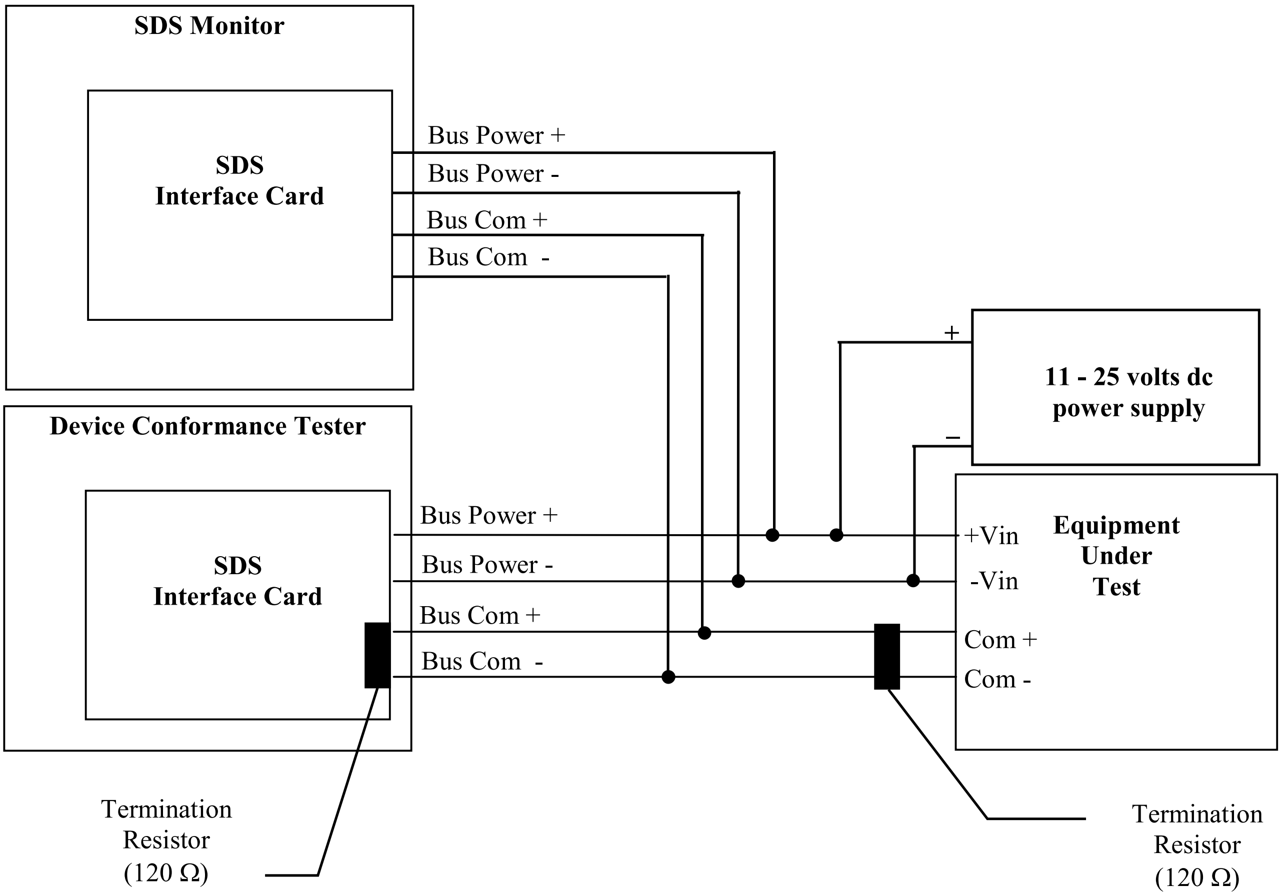


Figure 29 - Device conformance test set-up

9.3.2 Attributes

9.3.2.1 Test procedure

a) Issue a read request message for each attribute identified in the Object Model of the EUT.

b) Issue a write request message for each writeable attribute identified in the Object Model of the EUT.

c) Issue an action request message READ PRIMITIVE TAG for each attribute identified in the Object Model of the EUT.

d) If the EUT has alternative modes which should not effect the product’s response, test the EUT in each mode to verify the specified responses.

e) If the EUT has alternative modes which should effect the product’s response, test the EUT in each mode to verify the specified responses.

9.3.2.2 Test results

a) Record whether the correct response message was produced for each read request.

b) Record whether the correct response message was produced for each write request.

c) Record whether the correct response message was produced for each action request.

9.3.3 Actions

9.3.3.1 Test procedure

a) Issue an action request message for each action identified in the Object Model of the EUT.

b) If the EUT has alternative modes which should not effect the product's response, test the EUT in each mode to verify the specified responses.

c) If the EUT has alternative modes which should effect the product's response, test the EUT in each mode to verify the specified responses.

d) If alternative request message formats are specified for an action, test each format.

9.3.3.2 Test results

Record whether the correct response message was produced for each action request.

9.3.4 Events

9.3.4.1 Test procedure

a) Apply stimulus to the EUT for each event identified in the Object Model of the EUT.

b) If alternative modes to release event messages are specified for the EUT, test each mode.

9.3.4.2 Test results

Record whether the correct event message was produced for each specified condition.

9.3.5 Short form services COS ON and COS OFF

NOTE This test only applies to Object Models which specify a single binary input and/or a single binary output.

9.3.5.1 Test procedure

Stimulate the EUT as specified to produce short form services COS ON and COS OFF.

NOTE If COS ON and COS OFF services are not utilised for the EUT, it shall be verified that there are no such messages issued by the EUT, regardless of the product stimulation.

9.3.5.2 Test results

Record whether the correct COS ON and COS OFF messages are produced by the EUT e.g. the SDS controller responds with the correct COS ON ACK or COS OFF ACK.

9.3.6 Short form services WRITE ON and WRITE OFF

NOTE This test only applies to Object Models which specify a single binary input and/or a single binary output.

9.3.6.1 Test procedure

Issue Short form services WRITE ON and WRITE OFF to the EUT.

9.3.6.2 Test results

a) Record whether the correct WRITE ON ACK and WRITE OFF ACK messages were produced by the EUT.

b) Record whether the EUT correctly performed the specified output state change.

9.4 Physical Layer Test

9.4.1 Transceiver functional test

9.4.1.1 Test procedure

a) Connect the EUT to the controller with the SDS power supply set to DC 18V.

b) Initiate an Autobaud sequence by the controller.

9.4.1.2 Test results

Record whether the EUT performed correctly during the Autobaud sequence.

9.4.2 Transceiver Input Resistance

NOTE This test only applies to EUTs using a discrete transceiver.

9.4.2.1 Test procedure

a) Measure and record the Transceiver Input Resistance $R_{DIFF\ OFF}$.

9.4.2.2 Test results

Record and verify that $R_{DIFF\ OFF}$ is greater than or equal to the minimum specified value (see Figure 30) when the transceiver is in the non-dominant state (i.e. not transmitting).

9.4.3 Transceiver input levels

NOTE This test only applies to EUTs using a discrete transceiver.

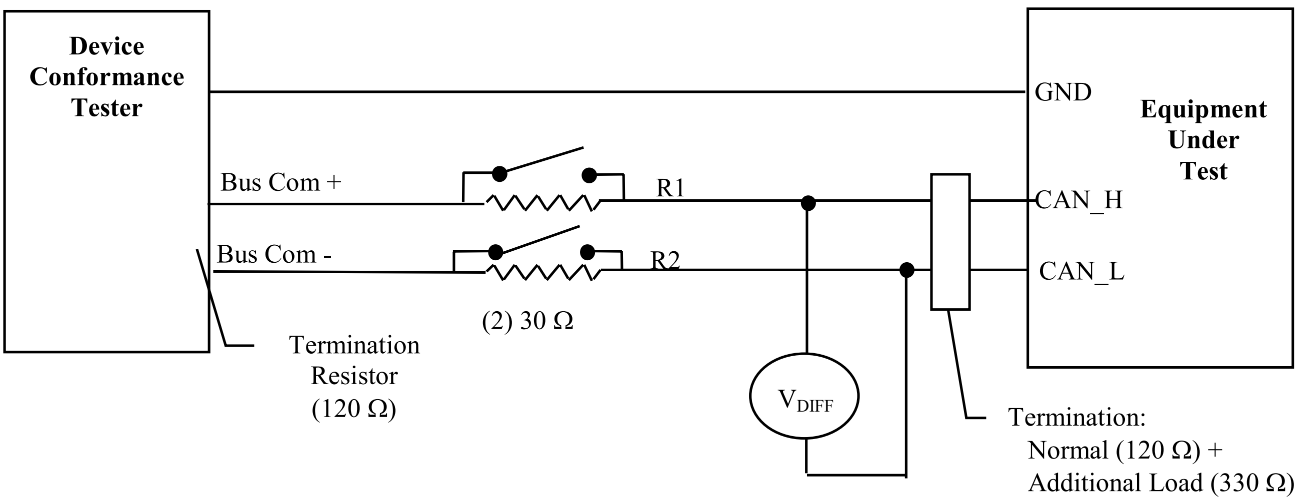


Figure 30 - Input threshold test circuit

9.4.3.1 Test procedure

a) Set the SDS power supply to DC 11V.

b) Set the resistor switches to the positions R1-IN and R2-OUT as shown in Figure 30.

c) Perform an Autobaud sequence by the controller.

d) Issue a read request message to any single attribute identified in the Object Model of the EUT.

e) Measure and record the differential input voltage level V_{DIFF} at the EUT, while the read request message is present.

f) Set the resistor switches to the positions R1-OUT and R2-IN.

g) Issue a read request message to any single attribute identified in the Object Model of the EUT.

h) Measure and record the differential input voltage level V_{DIFF} at the EUT, while the read request message is present.

i) Set the SDS power supply to DC 25V.

j) Repeat test steps b) to I).

9.4.3.2 Test results

Record and verify that the measured values are within the limits as specified in 8.1.3.1.

9.4.4 Transceiver output levels

NOTE This test only applies to EUTs using a discrete transceiver.

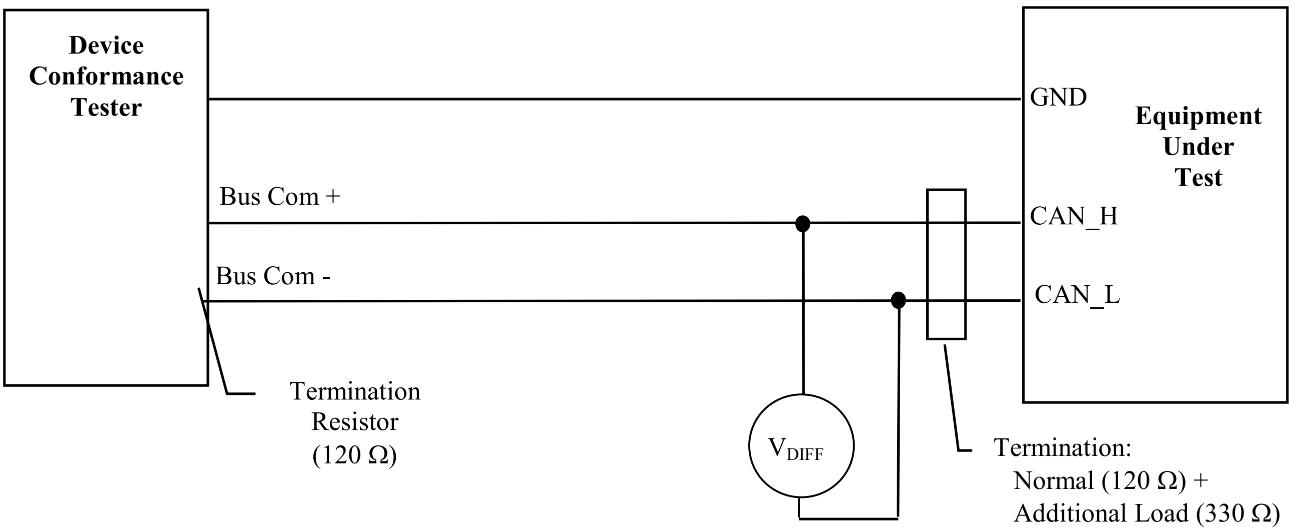


Figure 31 - Output level test circuit

9.4.4.1 Test procedure

a) Set the SDS power supply to DC 25V.

b) Stimulate the EUT to send SDS messages onto the SDS Network..

c) Record the recessive and dominant output signals of the EUT with an oscilloscope.

d) Connect a load resistor of 330Ω into the test circuit as shown in Figure 31.

e) Record the recessive and dominant output signals of the EUT with an oscilloscope.

9.4.4.2 Test results

Record and verify that the measured values are within the limits as specified in 8.1.3.1

9.4.5 SDS power

9.4.5.1 SDS power - voltage range test

9.4.5.1.1 Test procedure

a) Set the SDS power supply to DC 25V.

b) Switch OFF the SDS power supply.

c) Switch ON the SDS power supply.

d) An Autobaud sequence shall be released by the controller.

e) Set the SDS power supply to DC 11V.

f) Switch OFF the SDS power supply.

g) Switch ON the SDS power supply.

h) An Autobaud sequence shall be released by the SDS controller.

9.4.5.1.2 Test results

a) Record and verify that the EUT performs both Autobaud sequences successfully.

b) Record and verify that the correct device address of the EUT is reported by the controller.

9.4.5.2 SDS Power - maximum current test

9.4.5.2.1 Test procedure

a) Connect a current meter in series between the power supply V+ and the EUT V+, with the meter input connected to the power supply.

b) Set the SDS power supply to DC 11V.

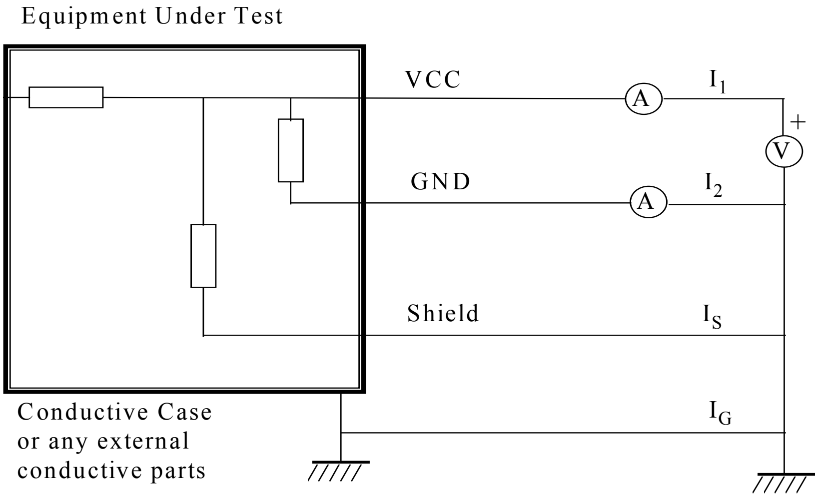
c) Slowly increase the voltage of the SDS power supply up to DC 25V.

d) Record the maximum power supply current.

9.4.5.2.2 Test results

Record and verify that the maximum power supply current is less or equal to the value, specified in the product data sheet.

9.4.5.3 SDS power - current loss test



NOTE If the case is not conductive, make the connection at the connector shell (if standard connector). If the product has no conductive external parts, omit the case connection.

Figure 32 - Current loss test set-up

9.4.5.3.1 Test procedure

a) Set the SDS power supply to DC 24V.

b) Measure the current values I_1 and I_2 as shown in Figure 32.

c) Calculate the difference between I_1 and I_2 (I_1 - I_2).

9.4.5.3.2 Test results

Record and verify that the difference between I_1 and I_2 (I_1 - I_2) is < 0,1mA.

9.5 Application Layer Test

9.5.1 ALP Services

9.5.1.1 General

The following general test instructions apply to all of the ALP tests in this subclause. Verify the data format of all EUT messages observed in conducting the tests in this subclause.

9.5.1.2 Data format

The data format shall comply with the following requirements. Verify that the data, for each message field, is within the range specified in Table 29.

Table 29 - Valid data values

Message	Message field	Valid data values
All Messages	Dir/Pri	0 - 1
	Device Logical Address	0 - 125
	RTR	0
Short Form Messages	Service Type	0 - 7
	DLC	0
Long Form Messages	Service Type	4 - 7
	DLC	> 0
	R/R	0 - 2
	FI	0
	EOID	0 - 15
	Service Parameters	0 - 255
Fragmented Messages	Service Type	4 - 7
	DLC	> 0
	R/R	0 - 2
	FI	1
	EOID	0 - 15
	Service Parameters	0 - 255
	Fragment Number	0 - 63
	Total Fragmented Bytes	7 - 255

9.5.1.3 Message structure checks
Verify proper message structure (e.g. short form response for short form request).

9.5.1.4 Product reaction to double messages
Verify a single response for each request (at least two requests - minimize delay between requests).

9.5.1.5 Multiple product response checks
Verify single response only, from product when presented a single request.

9.5.1.6 Decoding checks
Verify proper decoding of message fields (e.g. attribute ID).

9.5.1.7 Response Time

9.5.1.7.1 Test procedure

a) Issue an action request NOOP message at least every 1 second.

b) Measure the time between completion of the action request NOOP message and the beginning of the action response NOOP message with the SDS monitor.

9.5.1.7.2 Test results

Record and verify that the measured time between the request and the response message is less or equal 5 ms.

9.5.1.8 COS ON

NOTE This test only applies to single binary Object Models.

9.5.1.8.1 Test procedure

Stimulate the EUT as specified to release a COS ON message.

9.5.1.8.2 Test results

Verify that a COS ON message is issued by the EUT.

9.5.1.9 COS OFF

NOTE This test only applies to single binary Object Models.

9.5.1.9.1 Test procedure

Stimulate the EUT as specified to release a COS OFF message.

9.5.1.9.2 Test results

Verify that a COS OFF message is issued by the EUT.

9.5.1.10 Write ON

NOTE This test only applies to single binary Object Models.

9.5.1.10.1 Test procedure

Issue a WRITE ON message to the EUT.

9.5.1.10.2 Test results

Verify that a WRITE ON ACK message is issued by the EUT.

9.5.1.11 Write OFF

NOTE This test only applies to single binary Object Models.

9.5.1.11.1 Test procedure

Issue a WRITE ON message to the EUT.

9.5.1.11.2 Test results

Verify that a WRITE ON ACK message is issued by the EUT.

9.5.1.12 Read

9.5.1.12.1 Test procedure

Issue a read request message to a readable attribute of the EUT.

9.5.1.12.2 Test results

Verify that a read response message is issued by the EUT, with the appropriate attribute ID and data.

9.5.1.13 Write

9.5.1.13.1 Test procedure

Issue a write request message to a writeable attribute of the EUT.

9.5.1.13.2 Test results

a) Verify that a write response message is issued by the EUT.

9.5.1.14 Action

9.5.1.14.1 Test procedure

Issue an action request message to the EUT.

9.5.1.14.2 Test results

Verify that an action response message is issued by the EUT.

9.5.1.15 Event

9.5.1.15.1 Test procedure

Stimulate the EUT, as described in the product documentation, so that it will release an event.

9.5.1.15.2 Test results

Verify that an event request message is issued by the EUT.

9.5.1.16 Fragmented write

9.5.1.16.1 Test procedure

Issue a fragmented write request message to the EUT.

9.5.1.16.2 Test results

Verify that a single write response message is issued by the EUT.

9.5.1.17 Fragmented action

9.5.1.17.1 Test procedure

Issue a fragmented action request message to the EUT.

9.5.1.17.2 Test results

Verify that a single action response message is issued by the EUT.

9.5.1.18 Fragmented event

9.5.1.18.1 Test procedure

Stimulate the EUT, as described in the product documentation, so it will release a fragmented event request message.

9.5.1.18.2 Test results

a) Verify that a multiple fragmented event request message is issued by the EUT.

b) Verify that the single fragmented event messages are issued sequentially, starting with fragment number 0 (zero).

c) Verify that the composite data from the group of fragmented event request messages is the expected product data.

9.5.2 Logical Device functions

9.5.2.1 Address(s)

9.5.2.1.1 Test procedure

Issue read request messages to each Logical Address 0 to 125 inclusive.

9.5.2.1.2 Test results

a) Verify that a read response message is only issued by the EUT when it is addressed by its valid Logical Address.

b) Verify that here are no response message issued by the EUT when it is addressed by addresses other than its valid Logical Address.

9.5.2.2 Attributes

9.5.2.2.1 Test procedure

Issue read request messages to each possible attribute of the EUT, 0 to 255 inclusive.

9.5.2.2.2 Test results

a) Verify that a read response message is issued by the EUT for every read request message.

b) Verify that a positive response (i.e., read response message) is issued by the EUT when the addressed attribute is listed in the EUT’s Object Model.

c) Verify that a negative response (i.e., read response message including error code) is issued by the EUT when the addressed attribute is not listed in the EUT’s Object Model.

9.5.2.3 Actions

9.5.2.3.1 Test procedure

Issue action request messages to each possible action of the EUT, 0 to 255 inclusive.

9.5.2.3.2 Test results

a) Verify that an action response message is issued by the EUT for every action request message.

b) Verify that a positive response (i.e., action response message) is issued by the EUT when the selected action is listed in the EUT’s Object Model.

c) Verify that a negative response (i.e., action response message including error code) is issued by the EUT when the selected action is not listed in the EUT’s Object Model.

9.5.2.4 Events

9.5.2.4.1 Test procedure

Stimulate the EUT, as described in the product documentation, so that it will release each specified event message.

9.5.2.4.2 Test results

Verify that an event request message is issued by the EUT for each specified event condition.

9.5.3 Network functions

9.5.3.1 Autobaud

9.5.3.1.1 Test procedure

a) Configure the controller to a baud-rate of 125 kbit/s.

b) Set the address of the EUT to 1.

c) Issue an Autobaud sequence by the SDS controller.

d) Set the address of the EUT shall be set to a value between 10 and 110.

e) Issue an Autobaud sequence by the SDS controller.

f) Set the address of the EUT to 126.

g) Issue an Autobaud sequence by the SDS controller.

h) Configure the SDS controller to a baud rate of 250 kbit/s and repeat test steps b) to g).

i) Configure the SDS controller to a baud rate of 500 kbit/s and repeat test steps b) to g).

j) Configure the SDS controller to a baud rate of 1 Mbit/s and repeat test steps b) to g).

9.5.3.1.2 Test results

a) Verify that the EUT is recognized by the SDS controller after each Autobaud sequence.

b) Verify that all Autobaud tests are completed without any CAN Error Status being detected at the controller.

9.5.3.2 Heartbeat

NOTE This subclause only applies when the heartbeat option is specified in the EUT product documentation.

9.5.3.2.1 Test procedure

a) Enable the EUT to perform heartbeat behaviour.

b) Stimulate the EUT, as specified in the product documentation, so that a heartbeat condition occurs.

9.5.3.2.2 Test results

a) Verify that the EUT issues heartbeat messages which are long form Action NOOP request messages.

b) Verify that the heartbeat messages repeat as specified in product documentation.

9.6 System test

9.6.1 System test set-up

Figure 33 shows the test set-up which is used for the tests described in 9.6.2, 9.6.3 and 9.6.4.

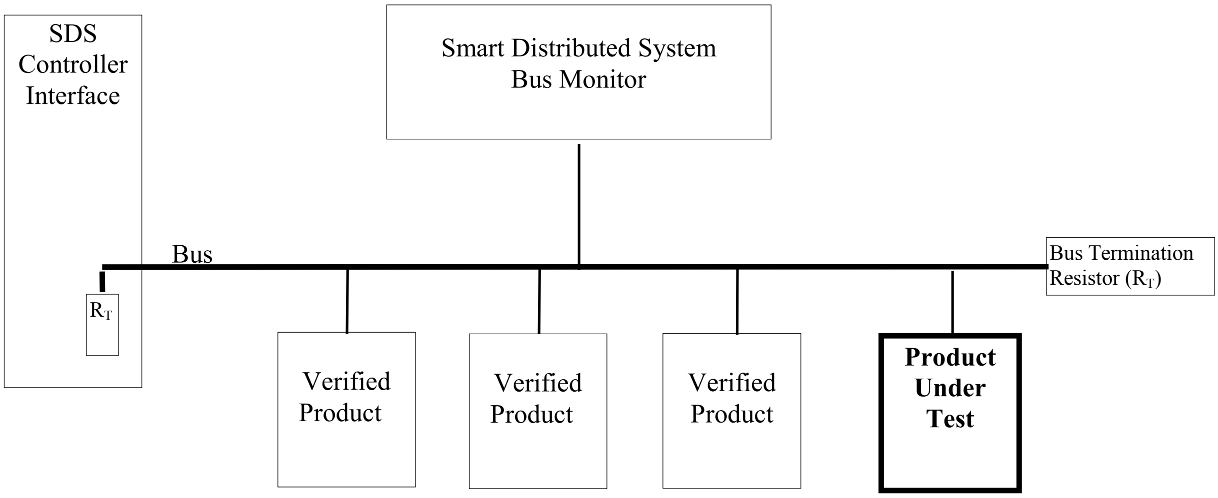


Figure 33 - System test set-up

9.6.2 Non-participative system testing

9.6.2.1 Test Procedure

a) Place the EUT in a test system as described in Figure 33.

b) Do not include the SDS address of the EUT in the application program of the SDS controller.

c) Issue the Autobaud sequence from the SDS controller and start the application program.

9.6.2.2 Test results

a) Verify that the EUT does not cause any disruption on the SDS Network.

b) Verify that the EUT does not communicate any messages in response to other SDS products.

9.6.3 Participative system testing

9.6.3.1 Test procedure

a) Place the EUT in a test system as described in Figure 33.

b) Include the SDS address of the EUT in the application program of the SDS controller.

c) Issue the Autobaud sequence from the SDS controller and start the application program.

9.6.3.2 Test results

a) Verify that the EUT does not cause any disruption on the SDS Network.

b) Verify that the EUT does not communicate any messages in response to other SDS products.

c) Verify that the EUT responds to messages with its own SDS address.

9.6.4 Other basic system testing

9.6.4.1 Power ON Test procedure

a) Switch ON the SDS.

b) Do not issue an Autobaud sequence from the SDS controller.

c) Observe the bus monitor for at least 10 s.

9.6.4.2 Power ON Test results

Verify that no messages are issued by the EUT.

9.7 Electromagnetic Compatibility Test

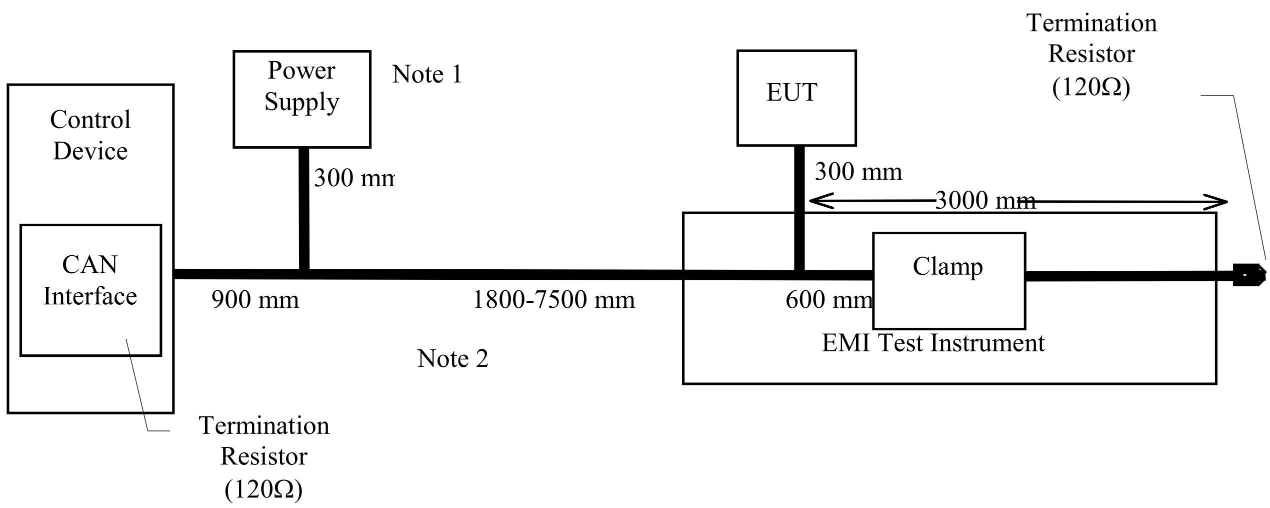
9.7.1 General

Unless otherwise noted the tests shall be carried out in accordance with 8.3.2

9.7.2 Fast transient/burst immunity

9.7.2.1 Test conditions

In accordance with EN 61000-4-4 and 8.3.3.5, modified as specified in this subclause. Except for the ambient temperature, the general requirements of 8.3.2 for EMC tests do not apply to this test.



NOTE 1 Power Supply ground (green ground) and Bus Shield shall be connected to earth ground. The Bus Shield to earth ground connection shall not exceed 75 mm in length (keep as short as possible).

NOTE 2 The cable lengths separating the EUT and the EMI Test Instrument (Clamp) are critical - the overall cable length shall be 900 mm exactly. The other cable lengths are maximum values or ranges.

Figure 34 - EMI test set-up

9.7.2.2 Test procedure

a) Set up the test as described in Figure 34. All bus cabling shall be by connector. If the EUT does not have a connector use an adapter cable of not longer than 600 mm.

b) Set the SDS controller to a baud rate of 125 kbit/s.

c) Set devices with an embedded input Object Model to unsolicited mode.

d) Subject the EUT to a Fast Transient/Burst level of 2 kV with a 5 ns rise-time and a 50 ns pulse width with a 5 kHz repetition frequency, for a period of 60 seconds.

e) Monitor the EUT and record any extraneous I/O signals.

f) If the EUT is a input type device, perform the test twice - once with the input activated and once with it not activated.

9.7.2.3 Test results

a) Verify that the EUT does not drop or add any SDS messages.

b) Verify that the SDS controller operates normally.

c) Verify that the EUT does not issue any unsolicited COS ON, COS OFF or COV APDUs.

d) Verify that any false pulse of an output type EUT has a width of < 1 μs.

e) Verify that an analogue type EUT does not change it's analogue value.

f) Verify that a digital type EUT does not change it's digital value.

g) Verify that the EUT does not indicate reset condition.

h) Verify that the EUT does not indicate self test failure condition.

i) Verify that the EUT does not issue error response APDUs.

The following degradation in performance is allowed:

- audible signals (not associated with control I/O)
- visible indicators (not associated with control I/O)
- temporary operating point deviations
- excessive data bus traffic (transmission retries, error frames, etc.)

blank

BSI — British Standards Institution

BSI is the independent national body responsible for preparing British Standards. It presents the UK view on standards in Europe and at the international level. It is incorporated by Royal Charter.

Revisions

British Standards are updated by amendment or revision. Users of British Standards should make sure that they possess the latest amendments or editions.

It is the constant aim of BSI to improve the quality of our products and services. We would be grateful if anyone finding an inaccuracy or ambiguity while using this British Standard would inform the Secretary of the technical committee responsible, the identity of which can be found on the inside front cover. Tel: 020 8996 9000. Fax: 020 8996 7400.

BSI offers members an individual updating service called PLUS which ensures that subscribers automatically receive the latest editions of standards.

Buying standards

Orders for all BSI, international and foreign standards publications should be addressed to Customer Services. Tel: 020 8996 9001. Fax: 020 8996 7001. Standards are also available from the BSI website at http://www.bsi-global.com.

In response to orders for international standards, it is BSI policy to supply the BSI implementation of those that have been published as British Standards, unless otherwise requested.

Information on standards

BSI provides a wide range of information on national, European and international standards through its Library and its Technical Help to Exporters Service. Various BSI electronic information services are also available which give details on all its products and services. Contact the Information Centre. Tel: 020 8996 7111. Fax: 020 8996 7048.

Subscribing members of BSI are kept up to date with standards developments and receive substantial discounts on the purchase price of standards. For details of these and other benefits contact Membership Administration. Tel: 020 8996 7002. Fax: 020 8996 7001. Further information about BSI is available on the BSI website at http://www.bsi-global.com.

Copyright

BSI
389 Chiswick High Road
London
W4 4AL