structure and behavior onto the network for each SEMI specified SDM (referred to as the “SDM mapping”).

2.4 *Adding SDM Mappings* — SDM mappings added to part two of this document are considered document additions and are balloted as such. An SDM mapping may only be balloted for addition to this document if the corresponding SEMI SDM has been standardized.

NOTICE: This standard does not purport to address safety issues, if any, associated with its use. It is the responsibility of the users of this standard to establish appropriate safety and health practices and determine the applicability of regulatory or other limitations prior to use.

3 Limitations

3.1 This document specifies a semiconductor equipment SAN based solely on EtherNet/IP and is a companion document to the EtherNet/IP specification; thus a complete specification of this standard necessarily includes the EtherNet/IP specifications. There are other semiconductor equipment SAN communications options. The specifications for these options are not included here.

3.2 This standard specifies enhancements that provide additional capabilities over and above those currently required by EtherNet/IP. In order to avoid document consistency problems, information in the EtherNet/IP specification that relates to this standard is not repeated in this document. This document is limited to describing enhancements or limitations to the EtherNet/IP specification that are imposed by this standard.

3.3 A complete specification of the conformance testing procedure shall include the EtherNet/IP conformance testing specification. Conformance testing shall also include enhancements and limitations to the EtherNet/IP specification required by this standard.

4 Referenced Standards

4.1 *SEMI Standards*

SEMI E30 — Generic Model for Communications and Control of Manufacturing Equipment (GEM)

SEMI E39 — Object Services Standard: Concepts, Behavior, and Services (OSS)

SEMI E54 — Sensor/Actuator Network Standard

SEMI E54.1 — Standard for Sensor/Actuator Network Common Device Model

SEMI E54.2 — Guide for Writing Sensor/Actuator Network (SAN) Standard Ballots

4.2 *ISO Standards*[2]

ISO 7498 — Basic Reference Model for Open Systems Interconnection.

ISO/IEC 8802-3 — Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications (Ethernet IEEE 802.3).

4.3 *Other Documents*

ODVA[3] EtherNet/IP Specification — Volume I and Volume II, Release 1.0.

NOTICE: Unless otherwise indicated, all documents cited shall be the latest published versions.

5 Terminology

5.1 Terminology that is common to all of the documents in this SAN standard may also be defined in SEMI E54. Terminology may be reproduced here which is defined in other SEMI documents.

5.2 *Abbreviations and Acronyms*

5.2.1 *CDM* — Common Device Model

5.2.2 *CIP* — Control and Information Protocol

5.2.3 *CM* — Connection Manager object

5.2.4 *DM* — Device Management object

5.2.5 *EIP* — EtherNet/IP

5.2.6 *EL* — Ethernet Link object

5.2.7 *IP* — Internet Protocol

5.2.8 *MR* — Message Router object

5.2.9 *NCS* — Network Communication Standard

5.2.10 *OSI* — Open Systems Interconnect

5.2.11 *OSS* — Object Services Standard

5.2.12 *PDU* — Protocol Data Unit

5.2.13 *SAC* — Sensor, Actuator, Controller object

5.2.14 *SAN* — Sensor/Actuator Network

5.2.15 *SDM* — Specific Device Model

5.2.16 *S-DS* — S-Device Supervisor object

5.2.17 *TCP* — Transport Control Protocol

5.3 *Device Component Definitions*

[2] International Organization for Standardization. ISO Central Secretariat, 1, rue de Varembé, Case postale 56, CH-1211 Geneva 20, Switzerland. Telephone: 41.22.749.01.11; Fax: 41.22.733.34.30. Website: /www.iso.ch

[3] Open DeviceNet Vendor Association, 20423 State Road 7 #F6, Boca Raton, FL 33498-6797; Phone: (1) 561-477-7966; Website: www.odva.org.

5.3.1 As this standard defines the presentation or mapping of CDM data structure and behavior over a network, it makes use of many of the terms in the CDM document. Table 1 provides a mapping of fundamental terminology of the CDM document into this document and the EtherNet/IP specification. Note that Column 2 contains an equal sign "=" if the definition is used exactly as specified in the CDM specification.

Table 1 Mapping of CDM to NCS Terminology

CDM Term	*NCS Equivalent*	*EtherNet/IP Equivalent*
Device	=	=
Device Model	=	Device Profile
Object	=, Class	=, Class
Instance	=	=
Attribute	=	=
Behavior	=	=
Service	=	=
Behavior State Diagram	=	State Transition Diagram
State Transition Matrix	=	State Event Matrix

5.4 *EtherNet/IP Specific Definitions*

5.4.1 *class* — A set of objects that all represent the same kind of system component. A class is a generalization of an object. All objects in a class are identical in form and behavior, but may contain different attribute values.

5.4.2 *Control and Information Protocol (CIP)* — The common network, transport and application layers shared by EtherNet/IP and DeviceNet.

5.4.3 *device profile* — An EtherNet/IP specification for a device that contains an object model for the device type, the I/O data format for the device type, and the configuration data and the public interface(s) to that data.

5.4.4 *encapsulation* — The technique used by layered protocols in which a layer adds header information to the protocol data unit (PDU) from the layer above. As an example, in Internet terminology, a packet would contain a header from the data link layer, followed by a header from the network layer (IP), followed by a header from the transport layer (TCP), followed by the application protocol data.

5.4.5 *Ethernet* — A 10/100-Mb/s standard for LANs, initially developed by Xerox, and later refined by Digital, Intel and Xerox (DIX). All hosts are connected to the network media where they contend for network access using a Carrier Sense Multiple Access with Collision Detection (CSMA/CD).

5.4.6 *EtherNet/IP* — EtherNet/IP (Ethernet/Industrial Protocol) is a networked communications protocol that specifies the encapsulation of CIP over TCP/IP.

5.4.7 *Explicit Message Connections* — Connections over an EtherNet/IP network that provide generic, multi-purpose communication paths between two devices. These connections often are referred to as just *Messaging* Connections. Explicit Messages provide the typical request/response–oriented network communications.

5.4.8 *Input/Output Connections* — Connections over an EtherNet/IP network that provide dedicated, special-purpose communication paths between a producing application and one or more consuming applications. Application-specific I/O data moves though these ports.

6 Communication Protocol High Level Structure

6.1 Though the protocol specification is called "EtherNet/IP", Ethernet is technically not required. EtherNet/IP may be used on any media that supports transmission of the Internet Protocol (IP). When Ethernet is used, it shall comply with the IEEE 802.3 specification.

NOTE 1: Conformance testing of EtherNet/IP products is limited to those products implemented on Ethernet.

6.2 The following sections give a brief description of the EtherNet/IP specification in terms of the seven layer architecture described by the ISO Basic Reference Model for Open Systems Interconnection (the OSI model).

6.3 *Physical Layer* — The physical layer requirements include specifications for topology, media, connectors, shielding and grounding. Where copper media is used, the EtherNet/IP specification is based on the ANSI/TIA/EIA-568-B Category 5/5E cable/connector specifications. A Commercial grade and Industrial grade level of conformance are specified. The signaling and coupling are as specified in the IEEE 802.3 / TP-PMD standard.

6.4 *Data Link Layer* — For compatibility with EtherNet/IP conformance testing capabilities, the device must comply with the Ethernet Data Link Layer Specifications as specified by IEEE 802.3. Otherwise, any data link layer that supports the IP is acceptable.

6.5 *Network Layer* — The network layer for the EtherNet/IP protocol is defined by the Internet Protocol (IP version 4) (RFC 791).

6.6 *Transport Layer* — The transport layer requires the support of Transmission Control Protocol (TCP) (RFC 793) as well as User Datagram Protocol (UDP) (RFC 768).

6.7 *Session Layer* — Many of the services and functions of the session layer for the EtherNet/IP protocol are divided and handled, in varying capacities, by the TCP/UDP protocol specifications and the EtherNet/IP Application Layer specifications. The first defines transport level connection sessions, while the latter defines application level sessions in terms of Explicit Connection and I/O Connection management.

6.8 *Presentation Layer* — The presentation layer is defined in terms of standard data types and specific encoding definitions for application layer data.

6.9 *Application Layer* — The device shall comply with the EtherNet/IP application layer specification for defining and addressing objects, including their attributes and services, and enabling specified network behavior. The device shall comply with the object model specifications provided in the EtherNet/IP specification.

6.10 *Object models* — EtherNet/IP provides an object-oriented specification for creating, defining and addressing objects explicitly, including their attributes and services (i.e., explicit messaging), and creating, defining and communicating object attribute assemblies in an application dependent format (i.e., input/output messaging). The device shall comply with the object model specifications provided in the EtherNet/IP documentation. In addition, the device shall comply with the object specifications defined in Section 7 of this document.

7 Required Object Types

7.1 The EtherNet/IP specification identifies and describes objects (i.e., classes) that must exist in all EtherNet/IP compliant devices. The Common Device Model (CDM) specification additionally identifies two objects, the Device Management (DM) and Sensor Actuator Controller (SAC) objects, that must exist in all SEMI compliant SAN devices. The required object types for a SEMI compliant SAN device utilizing the network communication specification described herein, necessarily comprises the union of the above two requirements.

7.2 A list of required and optional object types is given in Table 2. Note that the Sensor, Actuator, and Controller object types are not required, and are indicated as optional in the CDM specification. These objects are aggregated together to form a SEMI and EtherNet/IP compliant device.

7.3 The two required object types from the CDM (the DM and the SAC) are implemented as mapped components in EtherNet/IP; they are mapped to the SEMI-specific S-Device Supervisor object class as described in this section.

7.4 The listed object types are:

- Identity,
- Message Router (MR),
- Connection (Conn),
- Connection Manager (CM),
- Ethernet Link (EL),
- TCP/IP Interface (TCP/IP),
- S-Device Supervisor (S-DS),
- Device Management (DM),
- Sensor Actuator Controller (SAC),
- Sensor,
- Actuator, and
- Controller.

7.5 Also, although not a network addressable object, each device shall support the Unconnected Message Manager (UCMM) as defined by EtherNet/IP.

Table 2 Object Types

Object	*EIP Class ID (See Note 1.)*	*CDM Tag (See Note 2.)*	*Required by*		
			EIP (See Note 1.)	*CDM (See Note 2.)*	*NCS*
Identity	01	—	Yes	No	Yes
MR	02	—	Yes	No	Yes
Conn	05	—	Yes	No	Yes
CM	06	—	Conditional (See Note 1.)	No	Conditional (See Note 1.)
EL	F5	—	Conditional (See Note 1.)	No	Conditional (See Note 1.)
TCP/IP	F6	—	Yes	No	Yes
S-DS	30	—	No	No	Yes
DM	—	DmI0	No	Yes	Yes (See Note 4.)
SAC	—	SACI0	No	Yes	Yes (See Note 4.)
Sensor	31	SenIn	No	No	No
Actuator	32	ActIn	No	No	No

Object	*EIP Class ID (See Note 1.)*	*CDM Tag (See Note 2.)*	*Required by*		
			EIP (See Note 1.)	*CDM (See Note 2.)*	*NCS*
Controller	33	CntIn	No	No	No
(Other)	***	—	No	No	No

NOTE 1: See EtherNet/IP specification for further information; values are hexadecimal.

NOTE 2: See CDM specification for further information.

NOTE 3: Application Dependent.

NOTE 4: The DM and SAC Objects are implemented as mappings to the S-Device Supervisor object for EtherNet/IP.

7.6 See the EtherNet/IP specification for the implementation detail of the following object classes:

- Identity
- Message Router (MR)
- Connection (Conn)
- Connection Manager (CM)
- Ethernet Link (EL)
- TCP/IP Interface (TCP/IP)
- S-Device Supervisor (S-DS)

7.7 The Sensor, Actuator and Controller object types are utilized collectively to model the type specific structure and behavior of the device. The requirement and number of each of these object types in a device model is device type specific. Further, the attributes, services and behavior associated with each of these object classes and instances in a device is also device type specific, but must be compliant with both SEMI and EtherNet/IP specifications. The specification of these object types for a specific device type can be found in the appropriate SDM. The method of presentation of object structure and behavior to the EtherNet/IP network for objects defined for and associated with a specific device type can be found in Section 9 of this document.

7.8 The following sections describe the implementation detail for the two required objects from the CDM. The EtherNet/IP specification essentially combines the Device Management (DM) Object and the Sensor Actuator Controller (SAC) Object into a single Object: the S-Device Supervisor (S-DS) Object.

7.9 *Device Management (DM) Object* — The DM object is the device component responsible for managing and consolidating the device operation. The DM object as well as its common required and optional attributes, services and behavior are described in the CDM standard. The presentation of object attributes and services to the EtherNet/IP network shall be as indicated in Table 3 and Table 4 respectively. Each attribute and service is shown with its corresponding identifier (ID) for the S-DS Object.

Table 3 Network Presentation of DM Object Attributes

Device Management Object Attributes		
CDM Tag	*Name*	*S-DS Attribute ID (See Note 1.)*
DmA1	Device Type	03
DmA2	Standard Revision Level	04
DmA3	Device Manufacturer Identifier	05
DmA4	Manufacturer Model Number	06
DmA5	Software or Firmware Revision Level	07
DmA6	Hardware Revision Level	08
DmA7	Serial Number	09
DmA8	Device Configuration	0A
DmA9	Device Status	0B
DmA10	Reporting Mode	N.A. (See Note 2.)
DmA11	Exception Status Timer	N.A. (See Note 2.)
DmA12	Exception Status	0C
DmA13	Exception Detail Alarm	0D
DmA13	Exception Detail Warning	0E

NOTE 1: Attribute Identifier Values are hexadecimal.

NOTE 2: Reporting is controlled in EtherNet/IP by the configuration of connection related object instances. See the EtherNet/IP specification for detail.

Table 4 Network Presentation of DM Object Services

Device Management Object Services			
CDM Tag	*Name (SEMI)*	*Name (S-DS)*	*S-DS Service ID (See Note 1.)*
DmS1	Reset	Reset	05
DmS2	Abort	Abort	4B
DmS3	Recover	Recover	4C
DmS4	Get Attribute	Get_Attribute _Single	0E
DmS5	Set Attribute	Set_Attribute _Single	10
DmS6	Execute	Start	06
DmS7	Perform Diagnostics	Perform _Diagnostics	4E

NOTE 1: Service Identifier Values are hexadecimal.

7.10 *Sensor, Actuator, Controller (SAC) Object* — The SAC object is the device component responsible for coordinating the interaction of the device with the sensory/actuation/control environment. The SAC object as well as its common required and optional attributes, services and behavior are described in the CDM standard. The presentation of object attributes and services to the EtherNet/IP network shall be as indicated in Table 5 and Table 6 respectively. Each attribute and service is shown with its corresponding identifier (ID) for the S-DS Object.

Table 5 Network Presentation of SAC Object Attributes

SAC Object Services		
CDM Tag	*Name*	*S-DS Attribute ID (See Note 1.)*
SacA1	Last Calibration Date	13
SacA2	Next Calibration Date	14
SacA3	Expiration Timer	15
SacA4	Expiration Warning Enable	16
SacA5	Run Hours	17

NOTE 1: Attribute Identifier Values are hexadecimal.

Table 6 Network Presentation of SAC Object Services

SAC Object Services			
CDM Tag	*Name (SEMI)*	*Name (S-DS)*	*S-DS Service ID (See Note 1.)*
SacS1	Reset	Reset	05
SacS2	Abort	Abort	4B
SacS3	Recover	Recover	4C
SacS4	Get Attribute	Get_Attribute _Single	0E
SacS5	Set Attribute	Set_Attribute _Single	10
SacS6	Operate	Start	06
SacS7	Restore Default	N.A. (See Note 2.)	—
SacS8	Publish Attribute	N.A. (See Note 3.)	—

NOTE 1: Service Identifier Values are hexadecimal.

NOTE 2: Restoring defaults in EtherNet/IP is managed at the device level with services defined for the Identity Object.

NOTE 3: Reporting is controlled in EtherNet/IP by the configuration of connection related object instances. See the EtherNet/IP specification for detail.

8 Protocol Compliance

8.1 A method of testing protocol compliance is required to verify implementation conformance to the standard.

8.2 The compliance test suite for this protocol is documented along with the EtherNet/IP specification. ODVA has established an independent test center that perform a complete EtherNet/IP protocol compliance test suite. Visit <http://www.odva.org> for more information.

9 Specific Device Model Mappings

9.1 The following sections specify mappings for Sensor Actuator Network Specific Device Models.

9.2 *Mass Flow Device* — Reference SEMI E54.3 for a complete specification of the SDM for Mass Flow Devices. Accordingly, the following mapping rules apply to the identification tags for the Objects, Attributes and Services of this model.

9.2.1 *Objects* — Table 7 shows the mapping of the SDM objects specified in SEMI E54.3 and the EtherNet/IP objects.

Table 7 MFD Object Identifiers

SDM Object Name	*SDM Object ID*	*EIP Object Name*	*EIP ID (See Note 1.)*
Sensor-AI-MF	MFD3	S-Analog Sensor	31
Sensor-AI-AT	MFD4	S-Analog Sensor	31
Assembly-MFM	MFD5	Assembly	04
Sensor-AI-Aux	MFD6	S-Analog Sensor	31
Actuator-AO-MF	MFD7	S-Analog Actuator	32
Controller	MFD8	S-Single Stage Controller	33
Local Link	MFD9	N.A. (See Note 2.)	—
SISO	MFD10	N.A. (See Note 2.)	—
SISO Setpoint	MFD11	N.A. (See Note 2.)	—
Assembly-MFC	MFD12	Assembly	04

NOTE 1: Object Identifier Values are hexadecimal.

NOTE 2: The EtherNet/IP device model provides a different mechanism for data flow based on the configuration of EPATH type attributes. See the EtherNet/IP specification for detail.

9.2.1.1 Additional objects may be defined by the manufacturer in the Device Profile for a given device.

9.2.1.2 All objects listed in Table 7 provide a one-to-one correlation, with respect to Attributes and Services, between the SEMI SAN and EtherNet/IP. Therefore, no mapping is necessary for attributes and services here in this standard.

NOTICE: SEMI makes no warranties or representations as to the suitability of the standards set forth herein for any particular application. The determination of the suitability of the standard is solely the responsibility of the user. Users are cautioned to refer to manufacturer' s instructions, product labels, product data sheets, and other relevant literature, respecting any materials or equipment mentioned herein. These standards are subject to change without notice.

By publication of this standard, Semiconductor Equipment and Materials International (SEMI) takes no position respecting the validity of any patent rights or copyrights asserted in connection with any items mentioned in this standard. Users of this standard are expressly advised that determination of any such patent rights or copyrights, and the risk of infringement of such rights are entirely their own responsibility.

SEMI E54.14-0305
SPECIFICATION FOR SENSOR/ACTUATOR NETWORK COMMUNICATIONS FOR PROFINET

This specification was technically approved by the Global Information & Control Committee and is the direct responsibility of the European Equipment Automation Committee. Current edition approved by the European Regional Standards Committee on November 10, 2004. Initially available at www.semi.org February 2005; to be published March 2005.

1 Purpose

1.1 This specification is part of the SEMI Sensor/Actuator Network (SAN) suite of standards and defines a specific communications protocol based on the PROFINET standard. This Network Communication Standard (NCS) taken together with the SEMI Sensor/Actuator Network standard suite and the PROFINET standard completely and unambiguously defines an open standard providing an industry specific solution to off-the-shelf interoperability of networked devices in semiconductor manufacturing equipment.

1.2 PROFINET is a vendor independent, open field bus standard for a wide range of applications in manufacturing, process and building automation.

1.3 The application model of PROFINET is compatible with the model of PROFIBUS. The same mapping rules apply.

1.4 PROFINET is optimized for high speed and inexpensive connectivity between automation control systems and distributed I/O at the device level.

2 Scope

2.1 This document specifies a SAN communications standard based on the PROFINET specification that is in compliance with SEMI E54.1. As such, it specifies the protocol, services, and behavior that compliant intelligent devices must support in order to interchange information over this SAN in a method compatible with SEMI E39.

2.2 In conjunction with a SEMI standard SAN Common Device Model (CDM) specification and one or more SEMI standard Specific Device Model (SDM) specifications (e.g., for a mass flow controller), this Network Communication Standard (NCS) with the related PROFINET standard describe the data structures, interactions, and behavior that are characteristic of the various devices on the network. This composite model forms a complete interoperability standard for communications among intelligent sensors, actuators, and controllers in semiconductor manufacturing equipment.

NOTICE: This standard does not purport to address safety issues, if any, associated with its use. It is the responsibility of the users of this standard to establish appropriate safety and health practices and determine the applicability of regulatory or other limitations prior to use.

3 Limitations

3.1 This document specifies a semiconductor equipment SAN based solely on PROFINET and is a companion document to the PROFINET specification, including, by reference, the PROFINET standard; thus, a complete specification of this standard necessarily includes the PROFINET specifications. There are other semiconductor equipment SAN communications options. The specifications for these options are not included here.

3.2 The specifications within are strictly enhancements that provide additional capabilities over and above those currently required by PROFINET. Included throughout this document, primarily in §6, is information paraphrased from the PROFINET specifications such as: protocol structure, capabilities, options, and limitations. This information is provided here for reference only and is not intended to provide specification definitions. In all such areas, refer to the PROFINET specification documents for information. This document is limited to describing enhancements or limitations to the PROFINET specification that are imposed by this standard.

3.3 A complete specification of the conformance testing procedure shall include the PROFINET protocol conformance testing specification. Conformance testing shall also include enhancements and limitations to the PROFINET specification required by this standard.

4 Referenced Standards

4.1 *SEMI Standards*

SEMI E39 — Object Services Standard: Concepts, Behavior, and Services

SEMI E54.1 — Standard for Sensor/Actuator Network Common Device Model

SEMI E54.3 — Specification for Sensor/Actuator Network Specific Device Model for Mass Flow Device

SEMI E54.10 — Specification for Sensor/Actuator Network Specific Device Model for an In-Situ Particle Monitor Device

SEMI E54.11 — Specific Device Model for Endpoint Devices

4.2 *IEC Standards*[1]

IEC 61158-5 — Digital data communication for measurement and control – Fieldbus for use in industrial control systems – Application Layer service definition (reference standard for the model in use)

IEC 61158-6 — Digital data communication for measurement and control – Fieldbus for use in industrial control systems – Application Layer protocol specification (reference standard for the model in use)

4.3 *IEEE Standards*[2]

IEEE 802.3 — Telecommunications and information exchange between systems — Local and metropolitan area networks — Part 3:Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications

IEEE 802.1D — Telecommunications and information exchange between systems — Local and metropolitan area networks — Common specifications — Media Access Control (MAC) Bridges

IEEE 802.1Q — Telecommunications and information exchange between systems — Local and metropolitan area networks — Common specifications —Virtual Bridged Local Area Networks

4.4 *ISO Standards*[3]

ISO 7498 Information Technology — Open Systems Interconnection — Basic Reference Model

4.5 *PROFINET Standards*[4]

PROFINET IO — Application Layer protocol specification

PROFINET IO — Application Layer service definition

PROFIBUS Profile Guidelines — Part 1: Identification & Maintenance Functions

PROFINET — Discovery and Configuration Protocol

PROFINET — Installation Guideline

GSD Specification for PROFINET

NOTICE: Unless otherwise indicated, all documents cited shall be the latest published versions.

5 Terminology

5.1 *Abbreviations and Acronyms*

5.1.1 *APDU* — Application Protocol Data Unit

[1] International Electrotechnical Commission 3, rue de Varembé, Case Postale 131, CH-1211 Geneva 20, Switzerland. Telephone: 41.22.919.02.11; Fax: 41.22.919.03.00, Website: www.iec.ch

[2] IEEE , The Institute of Electrical and Electronics Engineers, Inc., 3 Park Avenue, New York, NY 10016-5997, USA, Website: www.ieee.org

[3] International Organization for Standardization, ISO Central Secretariat, 1, rue de Varembé, Case postale 56, CH-1211 Geneva 20, Switzerland. Telephone: 41.22.749.01.11; Fax: 41.22.733.34.30, Website: www.iso.ch

[4] Profibus International, Haid-und-Neu-Str. 7,76131 Karlsruhe, Germany, Telephone: 49 721 9658 590; Fax: 49 721 9658 589, Website: www.profibus.com

5.1.2 *AREP* — Application Reference Endpoint

5.1.3 *ASE* — Application Service Element

5.1.4 *CDM* — Common Device Model

5.1.5 *DCE* — Distributed Computing Environment

5.1.6 *DCP* — Discovery and Configuration Protocol

5.1.7 *DHCP* — Dynamic Host Configuration Protocol

5.1.8 *DM* — Device Manager

5.1.9 *GSD* — Generic station description

5.1.10 *IM* — Identification and Maintenance

5.1.11 *IO* — Input Output

5.1.12 *IP* — Internet Protocol

5.1.13 *NCS* — Network Communication Standard

5.1.14 *OSI* — Basic Reference Model for Open Systems Interconnection (ISO 7498)

5.1.15 *PDU* — Protocol Data Unit

5.1.16 *PHY* — Physical Layer

5.1.17 *RPC* — Remote Procedure Call

5.1.18 *SAC* — Sensor, actor, controller (object)

5.1.19 *SAN* — Sensor/actor network

5.1.20 *SAP* — Service Access Point

5.1.21 *SDM* — Specific Device Model

5.1.22 *UDP* — User Datagram Protocol

5.2 *Terminology Defined in Sensor/Actuator Network Common Device Model (SEMI E54.1)*

5.2.1 *attribute*

5.2.2 *behavior*

5.2.3 *byte*

5.2.4 *common device model*

5.2.5 *device*

5.2.6 *Device Manager (DM) Object*

5.2.7 *device model*

5.2.8 *instance*

5.2.9 *network communication standard*

5.2.10 *object*

5.2.11 *Sensor, Actuator and Controller (SAC) Object*

5.2.12 *service*

5.2.13 *specific device model*

5.2.14 *state diagram*

5.3 *Terminology Mapping*

5.3.1 As this standard defines the mapping of CDM data structure and behavior over a network it makes use of many of the terms in SEMI E54.1. Table 1 provides a mapping of fundamental terminology of the CDM document into this document which uses the terminology of PROFINET IO.

5.4 *PROFINET Specific Definitions*

5.4.1 *Context Management* — network-accessible information (communication objects) that supports managing the operation of the system, including the application layer.

NOTE 1: Managing includes functions such as controlling, monitoring, and diagnosing.

Table 1 Mapping of CDM to NCS Terminology

CDM Term	*NCS Equivalent*
Device	IO Device
Object	=(with ASE as class specification)
Instance	=
Attribute	=
Behavior	=
Service	=
State Diagram	Protocol Machine, State Machine
Byte	=, Unsigned8
Nibble	Specific field coding
Character String	Visible String

5.4.2 *Device Data Base* — an electronic file that provides a clear and comprehensive description of the characteristics of a device type in a precisely defined format. Also called a GSD File.

5.4.3 *Device Profile* — a Device Data Base Sheet, which specifies the characteristic features of a device, and a GSD File.

5.4.4 *channel* — single physical or logical link of an input or output application object of a server to the process.

5.4.5 *channel related diagnosis* — information concerning a specific element of an input or output application object, provided for maintenance purposes.

5.4.6 *diagnosis data object* — object(s) which contains diagnosis information referenced by device/slot/subslot/diagnosis identifier.

5.4.7 *GSD File* — see Device Data Base.

5.4.8 *Index* — address of a record data object.

5.4.9 *IO Controller* — a device that manages its assigned IO Devices and handles user data exchange; usually a programmable controller.

5.4.10 *IO data object* — object designated to be transferred cyclically for the purpose of processing and referenced by device/slot/Subslot.

5.4.11 *IO Device* — a device that is configured and managed by IO Controllers and IO Supervisors; an IO Device initiates no unsolicited communications.

5.4.12 *IO Supervisor* — a device that interacts as a configuration or diagnostic tool; usually a programming device.

5.4.13 *module* — hardware or logical component of a physical device.

5.4.14 *provider* — node or source sending data to one or many consumers.

5.4.15 *record data object* — object(s) which are already pre-processed and transferred acyclically for the purpose of information or further processing and referenced by device/slot/subslot/index.

5.4.16 *resource* — processing or information capability.

5.4.17 *server* — a) role of an AREP in which it returns a confirmed service response APDU to the client that initiated the request — b) object which provides services to another (client) object.

5.4.18 *service* — operation or function that an object and/or object class performs upon request from another object and/or object class.

5.4.19 *Service Access Point* — an addressable location in a device for the directing of service requests.

5.4.20 *slot* — address of a structural unit within an IO device.

NOTE 2: Within a modular device, a slot typically addresses a physical module. Within compact devices, a slot typically addresses a logical function or virtual module.

5.4.21 *slot related diagnosis* — information dedicated to modules for maintenance purpose.

5.4.22 *submodule* — hardware or logical component of a module.

5.4.23 *subslot* — address of a structural unit within a slot.

NOTE 3: A subslot may address a physical interface for submodules within a module. Generally, a subslot is a second level to structure data within a device.

5.4.24 *vendor ident number* — central administrative number assigned by the PNO.

6 Communication Protocol High Level Structure

6.1 The PROFINET IO protocol constitutes a collapsed form of the OSI seven layer architecture. PROFINET IO uses physical, data link, network, transport, and application layers of the Reference Model. For real time communication a three layer approach with physical, data link and application is specified. This section has been formatted to be aligned with the Basic Reference Model for OSI. Figure 1 gives an overview of the architecture.

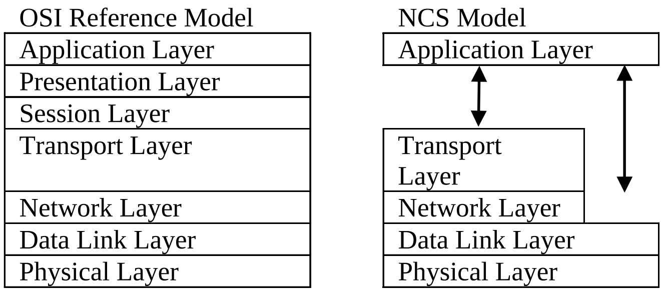


Figure 1
Layering of PROFINET in Relation to OSI

6.2 *General Behavior*

6.2.1 In a typical remote I/O configuration, single IO controller architectures are used to optimize response times. In lower speed applications, multi IO controller architectures are also possible.

6.2.2 Message transfer is organized in cycles. A message cycle mainly consists of a set of Output Data Frames of the IO controller and a set of Input Data Frames of several IO devices. Every Output Data Frame is associated to an Input Data Frame. The monitoring of an IO application relationship is done by the receiver of that frames.

6.2.3 A brief description of the PROFINET protocol as it relates to the ISO 7498 OSI model follows in the sections below. For protocol efficiency, PROFINET does not define layers 3 to 7 for cyclic IO data transfer and alarms. Layer 7 is the interface between the Application Process and the communication stack.

NOTE 4: The information contained in this section is for reference only. It in no way represents specifications for PROFINET. See related documentation for these specifications.

6.3 *Physical Layer — Layer 1*

6.3.1 The Physical Layer of IEEE 802.3 is adopted. There are two recommendations specified for the Physical Layer (PHY): twisted pair (100 Base TX) and optical (100 Base FX) in PROFINET. See IEEE 802.3 standard for more information about these options. More Details are specified in the PROFINET Installation Guideline. If an accepted physical standard other than IEEE 802.3 is being used it has be clearly specified in the product documentation.

6.4 *Data Link Layer — Layer 2*

6.4.1 *Data Transfer*

6.4.1.1 The Data Link Layer provides the functions for sending and receiving data over the network. Protocol Data Units (PDU) are packaged, delivered, and checked. Checks are used to guard against Line Protocol Errors (e.g., frame, overrun, and coding violations). IEEE 802.3 Data Link MAC sublayer is the preferred Data link technology of PROFINET. If an accepted data link standard other than IEEE 802.3 is being used it has to be clearly specified in the product documentation.

6.4.1.2 A PDU is restricted to 1500 bytes. A protocol overhead of 28 Bytes is needed for addressing, error detection and protocol selection.

6.4.2 *Conveyance of Data*

6.4.2.1 In IEEE 802 networks bridges (switches) are used to transport data. These bridge functions are defined in IEEE 802.1D and IEEE 802.1Q (priority options). Real time extensions can be used to enhance performance and availability of bridged networks.

6.5 *Network Layer — Layer 3*

6.5.1 While real time data transfer uses a protocol with no network layer, all other services are using IP (Internet Protocol) as network layer.

6.6 *Transport Layer — Layer 4*

6.6.1 All non-real time services are using UDP (User Datagram Protocol) as transport layer.

6.7 *Session Layer — Layer 5*

6.7.1 There is no distinct Session Layer in this CDM.

6.8 *Presentation Layer — Layer 6*

6.8.1 There is no distinct Presentation Layer in this CDM. The Encoding is part of the application layer.

6.9 *Application Layer — Layer 7*

6.9.1 The PROFINET IO application layer is structured in a so called service definition and protocol specification. The service definition uses an object orientated approach and specifies the services for remote access and local functions together with their objects (the ASE is a class definition of these objects). The protocol specification includes both coding and state machines.

6.9.2 *Service Definition*

6.9.2.1 Application layer services are structured to reflect the needs of flexible configurable automation devices. A Device consists of a set of modules that are placed in slots (see Figure 2). Modules are addressed uniquely by the slot number. The module view can be a hardware oriented or reflects the software structure of the IO Device.

6.9.2.2 Each module contains submodules that contain objects of different classes. Submodules are the addresses of the submodules.

6.9.2.3 Slot 0 is used to address the IO Device itself. Subslot 0 represents the module and contains no IO Data. The other object classes can have instances scattered over the modules and their submodules addressed by slots and their subslots respectively. Each submodule (except submodule 0) can contain IO Data, Context parameter, Diagnosis information, Record Data and Alarms. Record Data is a generic class which can contain different application specific parameters accessible by read and write services. There is a set of identification and maintenance parameters defined in a PROFIBUS Guideline that are mandatory within PROFINET.

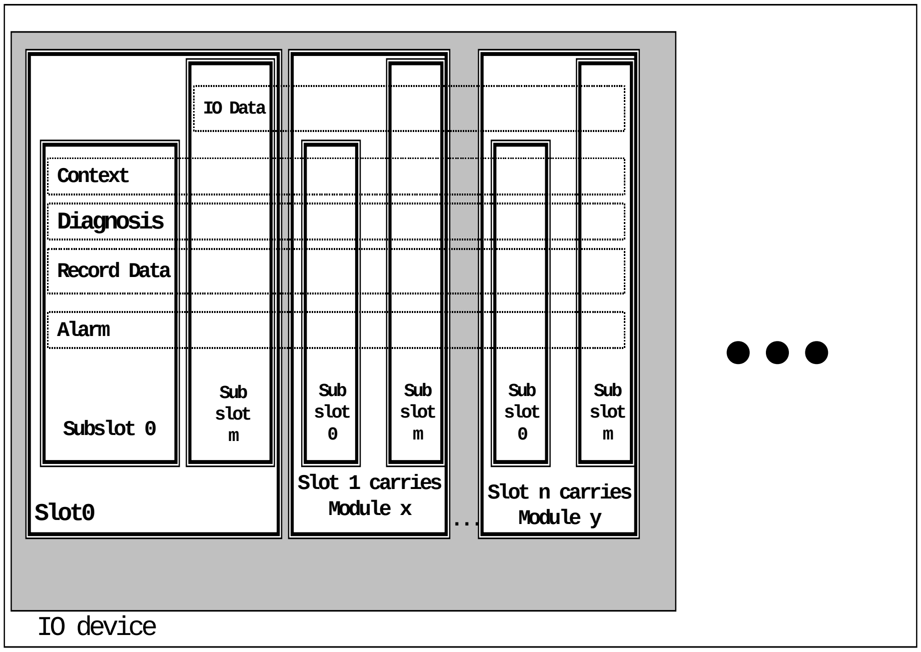


Figure 2
Device Model

6.9.2.4 Object classes are defined for:

- IO Data for periodic reporting,
- Context for configuration,
- Diagnosis for event collection,
- Record Data for polled access, and
- Alarms for asynchronous event reporting.

A set of services are defined for these object classes.

6.9.2.5 IO Data are handled mainly by buffered services which allow decoupling between application and communication. Client/Server service structure (request/response) is used for Context for configuration, Diagnosis for event collection, Record Data for polled access and Alarms for asynchronous event reporting.

6.9.2.6 Cyclic functionalities offered in PROFINET IO:

- exchange of IO data with related IO devices

6.9.2.7 Acyclic functionalities offered in PROFINET IO:

- read diagnosis from IO devices
- configuration of IO devices
- write parameter data to IO devices (startup or application parameter)
- treatment of configuration and diagnosis requests of an engineering device
- initiate connections to IO devices by means of context management
- acyclic access to record data of IO devices
- treatment of alarms from IO devices

- sending of alarms to IO devices

6.9.3 *Protocol Specification*

6.9.3.1 An IO device has no IP-address and logical name when it is shipped.

6.9.3.2 A set up procedure has to allocate name and IP-address prior to operation.

6.9.3.3 To access an IO Device a controller has to establish the context first with a connect service. A check is done to ensure that the appropriate device type with the required resources is accessed. Resources can be locked with the connect service.

6.9.3.4 After a positive confirmation of the connect service the IO device may be loaded with the parameter required by the controller application (e.g. warning limits, measure range, filter time). A control service is issued at the end of parameterization.

6.9.3.5 There is a control service indicating that the IO Device application is ready to enter the operate state. The start up is completed with this service.

6.9.3.6 After this start up procedure the data exchange of IO data can be done and alarms can be signaled from the IO Device to the IO Controller. The IO Controller can invoke services to read and write record data that contain all kind of information e.g. produced units, calibration information, and batch information.

6.9.3.7 A Release service will terminate the context.

6.9.3.8 A context is monitored by the receipt of Input Data at the IO controller and by the receipt of Output Data at the IO device site. A timeout error of one timer unit is allowed which makes it possible to use a single timer resource with granularity in the range of a millisecond.

6.9.3.9 Figure 3 shows a sequence diagram of the start up procedure.

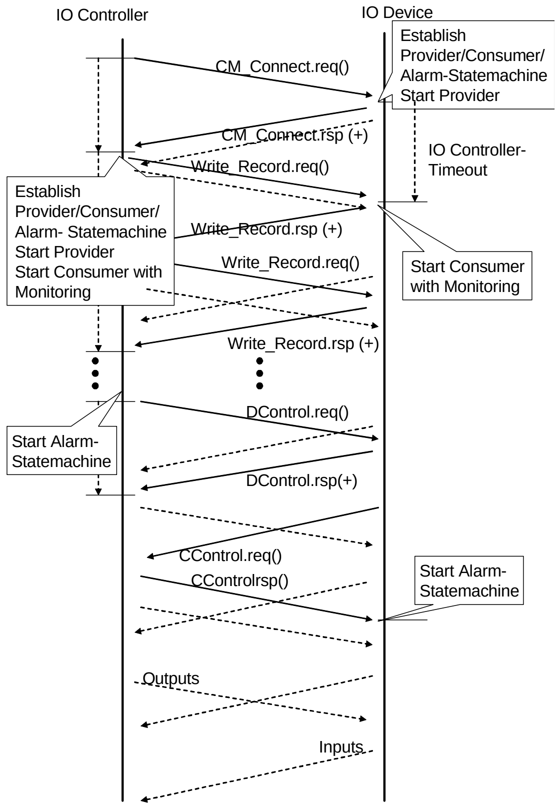


Figure 3
Start Up Sequence

6.9.4 *Communication Management*

6.9.4.1 DCP (Discovery and Configuration Protocol) is used for basic management of stations (especially setting of IP-address and Device name). DHCP (Dynamic Host Configuration Protocol) can be used as configuration option.

6.9.4.2 Connectionless DCE RPC is used as an interface between transport and application.

6.9.4.3 "Identification & Maintenance Functions" (I&M functions) define general parameters and protocols. The main purpose of the I&M functions is to support the end user during various scenarios of a device's life cycle be it configuration, commissioning, parameterization, diagnosis, repair, firmware update, asset management, audit trailing, and alike. Well-defined uniform parameters and rules should enable the manufacturers to offer devices that behave in a uniform manner These profile guidelines take into account requirements from FDA (Food & Drug Administration) and others. The basic information offered can be characterized as "Type Plate" or "Boiler Plate". It consists of Manufacturer ID, Order ID, Serial Number, Hardware Revision, Software Revision, Revision Counter, Profile ID.

7 Required Object Types

7.1 This section describes a general mapping of the SEMI SAN Object Model to the PROFINET environment. Component definitions are clarified and the mapping of Attributes, Services, and Behaviors are specified.

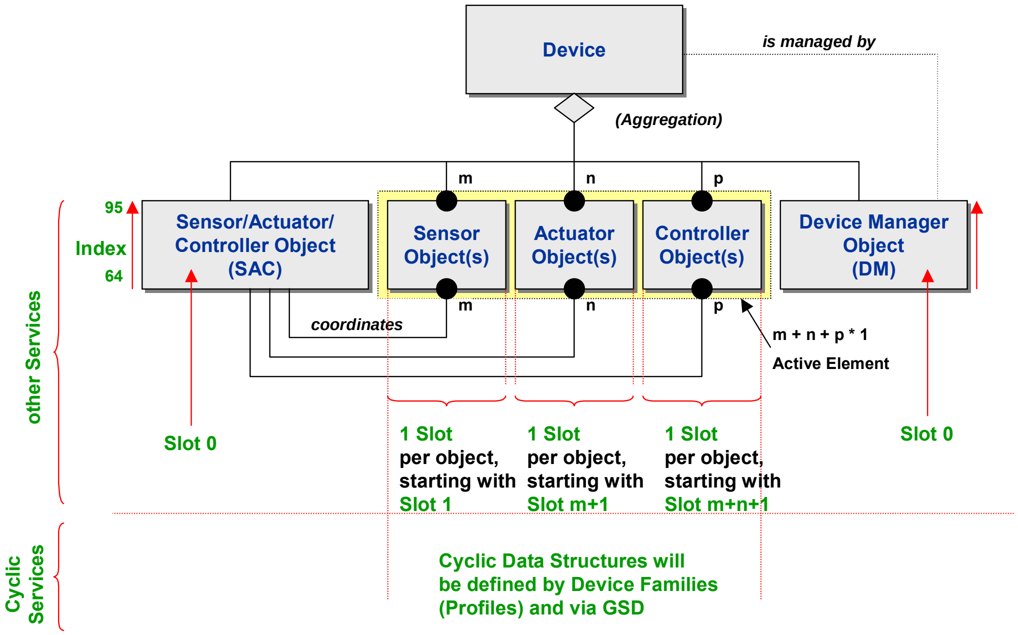
7.2 *Object Model*

7.2.1 The Object Model defined in the CDM is represented in the PROFINET NCS. Specifically, the DM and SAC objects are mapped.

7.2.2 The CDM Objects are mapped to slots. These slots have a unique address space within the device. The Specific Application Objects are mapped in PROFINET Device Data Base. §9 specifies the mapping of SDM Objects in PROFINET.

7.3 *Component Mapping Summary*

7.3.1 Figure 4 provides a summary of the components of the CDM object model as they relate to the components of PROFINET.



Device
is managed by
(Aggregation)
m
n
p
95
Index
64
Sensor/Actuator/ Controller Object (SAC)
Sensor Object(s)
Actuator Object(s)
Controller Object(s)
Device Manager Object (DM)
m
n
p
coordinates
m + n + p * 1
Active Element
other Services
Slot 0
1 Slot per object, starting with Slot 1
1 Slot per object, starting with Slot m+1
1 Slot per object, starting with Slot m+n+1
Slot 0
Cyclic Services
Cyclic Data Structures will be defined by Device Families (Profiles) and via GSD

Figure 4
Component Mapping Summary

7.4 *Objects*

7.4.1 The required objects of the CDM are identified here. Additional objects that are contained in the SDM are given identifiers in the Device Profile. §9 specifies additional mapping information.

7.4.2 Table 2 lists the Object Identifiers specified for use in protocol messages. Slot numbers is the mechanism used for addressing Objects within PROFINET. Subslot IDs can be used for sub addressing. Within this NCS the subslot 1 shall be used as default subslot.

Table 2 Object Identifiers

Slot ID	*Index*	*Object*
0	IM0-17 96	DM Object Attributes DM Control
0	64 65-95	SAC Control SAC Object Attributes
1–z	64 65-254	Application Object Control Application Objects Attributes as specified in §7 and §9

7.4.3 Assembly Objects and Local Link Objects which are objects embedded in the SAC Object are not explicitly mapped in PROFINET.

7.5 *Attributes*

7.5.1 *General*

7.5.1.1 All attributes are accessible via Get_Attribute and Set_Attribute services defined in the sections below. The Get_Attribute and Set_Attribute services shall be mapped to PROFINET Read and Write services for Record Data objects. Additionally, attributes are accessible via different PROFINET defined methods that are mapped in this document based on attribute type.

7.5.2 *DM Attributes*

7.5.2.1 All attributes are communicated with Read and Write Service and mapped to IM Blocks. Status attributes are modeled additionally as Diagnosis. See Table 17 for detailed mapping.

7.5.2.2 Table 3 shows the DM status attribute mapping to Diagnosis.

Table 3 Diagnosis Mapping of DM Status

Parameter	*Value*
Slot	0 = Device
Subslot	1 = Device
Channel	0 = Global
Channel Type	0 = unspecified
IO Type	0 = unspecified
Error Type	9 = Error with Bit0 - 6 set in Common Exception Detail 1 21 = Error Calibration set in Common Exception Detail Attribute 1 17 = Error with Bit set in Common Exception Detail 2

7.5.3 *SAC Attributes*

7.5.3.1 The attributes of SAC are mapped to Record data objects of PROFINET.

7.5.3.2 The Slot number used is 0. The index is the numeric value of the SAC Attribute identifier with an offset of 64 (see Table 4).

Table 4 SAC Object Attribute Identifiers

SEMI CDM Attribute ID	*PROFINET Index*	*Attribute*	*Detail*
SacA1	65	Last Calibration Date	
SacA2	66	Next Calibration Date	
SacA3	67	Expiration Timer	
SacA4	68	Expiration Warning Enable	
SacA5	69	Run Hours	

7.5.4 *Active Element Attributes*

7.5.4.1 The attributes of Active Elements are mapped to Record data objects of PROFINET. These attributes are contained in any Sensor, Actuator and Controller Object.

7.5.4.2 The Slot number used is the instance number in the device. The index is the numeric value of the Active Element Attribute identifier with an offset of 64 (see Table 5).

Table 5 Active Element Object Attribute Identifiers

SEMI CDM Attribute ID	*PROFINET Index*	*Attribute*	*Detail/Alternative Access Method*
nA1	65	Name	
nA2	66	Status	
nA3	67	Alarm Enable	
nA4	68	Warning Enable	

7.5.5 *Sensor Attributes*

7.5.5.1 The attributes of Sensors are mapped to Record data objects of PROFINET. These attributes are contained in any Sensor-AI, Sensor-EI and Sensor-BI Object.

7.5.5.2 The Slot number used is the instance number in the device. The index is the numeric value of the Sensor Attribute identifier with an offset of 64 (see Table 6).

Table 6 Sensor Object Attribute Identifiers

SEMI CDM Attribute ID	*PROFINET Index*	*Attribute*	*Detail/Alternative Access Method*
nA16	80	Value	Can be mapped as Input Data Object
nA17	81	Report Inhibit Time	
nA18	82	Enable Report Rate	
nA19	83	Report Rate	

7.5.6 *Actuator Attributes*

7.5.6.1 The attributes of Actuators are mapped to Record data objects of PROFINET. These attributes are contained in any Actuator-AO, Actuator-EO and Actuator-BO Object.

7.5.6.2 The Slot number used is the instance number in the device. The index is the numeric value of the Actuator Attribute identifier with an offset of 64 (see Table 7).

Table 7 Actuator Object Attribute Identifiers

SEMI CDM Attribute ID	*PROFINET Index*	*Attribute*	*Detail/Alternative Access Method*
nA16	80	Setting	Can be mapped as Output Data Object
nA17	81	Safe State	
nA18	82	Watch Rate	
nA19	83	Watch Dog	The value of this variable reflects the WD_Factor para-meter of PROFINET

7.5.7 *Controller Attributes*

7.5.7.1 The attributes of Controllers are mapped to Record data objects of PROFINET.

7.5.7.2 The Slot number used is the instance number in the device. The index is the numeric value of the Controller Attribute identifier with an offset of 64 (see Table 8).

Table 8 Controller Object Attribute Identifiers

SEMI CDM Attribute ID	*PROFINET Index*	*Attribute*	*Detail/Alternative Access Method*
CA16	80	Setpoint	
CA17	81	Process Variable	
CA18	82	Control Variable	
CA19	83	Data Type	
CA20	84	Data Unit	
CA21	85	Alarm Settle Time	
CA22	86	Alarm Error Band	
CA24	88	Warning Settle Time	
CA25	89	Warning Error Band	

7.5.8 *Sensor Analog Input Attributes*

7.5.8.1 The attributes of Sensor Analog Inputs are mapped to Record data objects of PROFINET.

7.5.8.2 The Slot number used is the instance number in the device. The index is the numeric value of the Sensor Analog Input Attribute identifier with an offset of 64 (see Table 9).

Table 9 Sensor Analog Input Object Attribute Identifier

SEMI CDM Attribute ID	*PROFINET Index*	*Attribute*	*Detail/Alternative Access Method*
SaiA64	128	Offset	
SaiA65	129	Gain	
SaiA66	130	Data Type	
SaiA67	131	Data Units	
SaiA68	132	Safe State	
SaiA69	133	Enable Report Delta	
SaiA70	134	Report Delta	
SaiA71	135	Enable Report ROC	
SaiA72	136	Report ROC	
SaiA73	137	Alarm Trip Point High	
SaiA74	138	Alarm Trip Point Low	
SaiA75	139	Alarm Hysteresis	
SaiA76	140	Warning Trip Point High	
SaiA77	141	Warning Trip Point Low	
SaiA78	142	Warning Hysteresis	

7.5.9 *Sensor Binary Input Attributes*

7.5.9.1 The attributes of Sensor Binary Inputs are mapped to Record data objects of PROFINET.

7.5.9.2 The Slot number used is the instance number in the device. The index is the numeric value of the Sensor Binary Input Attribute identifier with an offset of 64 (see Table 10).

Table 10 Sensor Binary Input Object Attribute Identifiers

SEMI CDM Attribute ID	*PROFINET Index*	*Attribute*	*Detail/Alternative Access Method*
SbiA64	128	Debounce Control	
SbiA65	129	Alarm State	
SbiA66	130	Warning State	

7.5.10 *Sensor Enumerated Input Attributes*

7.5.10.1 The attributes of Sensor Enumerated Inputs are mapped to Record data objects of PROFINET.

7.5.10.2 The Slot number used is the instance number in the device. The index is the numeric value of the Sensor Enumerated Input Attribute identifier with an offset of 64 (see Table 11).

Table 11 Sensor Enumerated Input Object Attribute Identifiers

SEMI CDM Attribute ID	*PROFINET Index*	*Attribute*	*Detail/Alternative Access Method*
SeiA64	128	Debounce Control	
SeiA65	129	Alarm State	
SeiA66	130	Warning State	

7.5.11 *Actuator Analog Output Attributes*

7.5.11.1 The attributes of Actuator Analog Outputs are mapped to Record data objects of PROFINET.

7.5.11.2 The Slot number used is the instance number in the device. The index is the numeric value of the Actuator Analog Output Attribute identifier with an offset of 64 (see Table 12).

Table 12 Actuator Analog Output Object Attribute Identifiers

SEMI CDM Attribute ID	*PROFINET Index*	*Attribute*	*Detail/Alternative Access Method*
AaoA64	128	Offset	
AaoA65	129	Gain	
AaoA66	130	Data Type	
AaoA67	131	Data Units	

7.5.12 *Sensor Binary Input Threshold Attributes*

7.5.12.1 The attributes of Sensor Binary Input Thresholds are mapped to Record data objects of PROFINET.

7.5.12.2 The Slot number used is the instance number in the device. The index is the numeric value of the Sensor Binary Input Threshold Attribute identifier with an offset of 80 (see Table 13). The different mapping has to be done because there is an Overlap to the Sensor Binary Input Attributes enumeration part.

Table 13 Sensor Binary Input Threshold Object Attribute Identifiers

SEMI CDM Attribute ID	*PROFINET Index*	*Attribute*	*Detail/Alternative Access Method*
SbithA64	144	Reading Valid	
SbithA65	145	State	
SbithA66	146	Status	

7.6 *Services*

7.6.1 *DM Services*

7.6.1.1 Get_Attribute and Set_Attribute are mapped to Read and Write services to the IM Objects.

7.6.1.2 The other services are mapped to a write at an UINT Object addressed with Index 96 as described in Table 2 and Table 14.

7.6.1.3 A read to Index 96 will return the last command executed successfully.

Table 14 DM Service Mapping

SEMI CDM Service ID	*Service*	*Write Value to Index 96*
DmS4	Get Attribute	Mapped as Read from Attribute Index
DmS5	Set Attribute	Mapped as Write to Attribute Index
DmS1	Reset	0
DmS2	Abort	1
DmS3	Recover	2
DmS6	Execute	3
DmS7	Perform Diagnostics	4
DmS8 – DmS12		Optional Services are not mapped in PROFINET

7.6.2 *SAC Services*

7.6.2.1 Get_Attribute and Set_Attribute are mapped to Read and Write services with Slot and Index as described in ¶¶ 7.2 and 7.4.

7.6.2.2 The other services are mapped to a write at an UINT Object addressed with Index 64 as described in Table 2 and Table 15.

7.6.2.3 A read to Index 64 will return the last command executed successfully (Exception: Restore Default is not mirrored).

Table 15 SAC Service Mapping

SEMI CDM Service ID	*Service*	*Write Value to Index 64*
SacS4	Get Attribute	Mapped as Read from Attribute Index
SacS5	Set Attribute	Mapped as Write to Attribute Index
SacS1	Reset	0
SacS2	Abort	1
SacS3	Recover	2
SacS6	Operate	3
SacS7	Restore Default	4
SacS8	Publish Attribute	Optional Service not mapped in PROFINET

7.6.3 *Active Element Services*

7.6.3.1 Get_Attribute and Set_Attribute are mapped to Read and Write services with Slot and Index as described in ¶¶ 7.2 and 7.4.

7.6.3.2 The other services are mapped to a write at an UINT Object addressed with Index 64 as described in Table 2 and Table 16.

7.6.3.3 A read to Index 64 will return the last command executed successfully (Exception: Restore Default is not mirrored).

Table 16 Active Element Service Mapping

SEMI CDM Service ID	*Service*	*Write Value to Index 64*
nS4	Get Attribute	Mapped as Read from Attribute Index
nS5	Set Attribute	Mapped as Write to Attribute Index
nS1	Reset	0
nS2	Abort	1
nS3	Recover	2
nS6	Operate	3
nS7	Restore Default	4

7.7 *PROFINET Device Data Base*

7.7.1 The specification of the Device Data Base (or GSD file) for a given SDM is beyond the scope of this document. The PROFIBUS Trade Organization is responsible for the management of these files.

Table 17 DM Object Attribute Identifiers

SEMI CDM Attribute ID	*PROFINET Attribute ID*	*Attribute*	*Detail/Alternative Access Method*
DmA1	IM0, DEVICE_ID	Device Type	8 Octets significant
DmA2	IM0, REVISION_COUNTER	Standard Revision Level	Shall be an UINT type
DmA3	IM0, MANUFACTURER_ID	Device Manufacturer Identifier	Shall be an UINT type, the number is resolved via file from www.profibus.com to supply with web pages and phone numbers
DmA4	IM0, ORDER_ID	Manufacturer Model Number	
DmA5	IM0, SOFTWARE_REVISION	Software or Firmware Revision Level	
DmA6	IM0, HARDWARE_REVISION	Hardware Revision Level	
DmA7	IM0, SERIAL_NUMBER	Serial Number	16 Octets maximum
DmA8	IM3	Device Configuration	
DmA9	IM17, octet 10-11	Device Status	Additional as Status Diagnosis
DmA10	IM16, octet 10-11	Reporting Mode	
DmA11	IM16, octet 12-13	Exception Status Report Interval	
DmA12	IM17, octet 12-13	Exception Status	Additional as Status Diagnosis
DmA13	Slot 0 Index 0xe00b	Exception Detail Alarm	Additional as Diagnosis Alarm
DmA14	Slot 0 Index 0xe00b	Exception Detail Warning	Additional as Diagnosis Alarm
DmA15	Optional Attribute not mapped	Visual Indicator	N/A
DmA16	Context	Alarm Enable	Put into a Parameter Data Record
DmA17	Context	Warning Enable	Put into a Parameter Data Record
DmA18	Optional Attribute not mapped	Exception Detail Type	N/A
DmA19	Optional Attribute not mapped	Exception Detail Alarm Queue	N/A
DmA20	Optional Attribute not mapped	Exception Detail Warning Queue	N/A
DmA21	Optional Attribute not mapped	Date and Time	N/A
DmA22	Optional Attribute not mapped	Date and Time Type	N/A
DmA23-DmA31	Reserved	Reserved	N/A

8 Protocol Compliance

8.1 PROFIBUS International has established a qualified certification system, with test laboratories in Europe, Asia and North America, which includes conformance testing and interoperability testing (address list can be found at http://www.PROFIBUS.com/support.html[5]). Certified products are listed with their certificate number in the Electronic Product Guide.

8.2 GSD files of all PROFINET devices that are tested for their conformity to the PROFIBUS standard are available in the GSD library on the World Wide Web Server of the PROFIBUS User Organization at http://www.PROFIBUS.com.

9 Specific Device Model Mappings

9.1 Every type of device shall have an identifier number. Vendors must apply for an identifier number from the PROFIBUS User Organization. A Device Profile shall be submitted in the form of a GSD File and Device Data Base Sheet.

9.2 The Device Profile must specify the identifiers for Objects, Attributes and Services for CDM and SDM components, including data formats and bit mappings for specified parameters, as represented in this document.

9.3 The following sections specify mappings for Sensor Actuator Network Specific Device Models.

9.4 *Mass Flow Device*

9.4.1 Reference SEMI E54.3 for a complete specification of the SDM for Mass Flow Devices (MFD).

9.4.2 *Objects*

9.4.2.1 Consistent with SEMI E54.3 and ¶7.4 above, the DM and SAC objects are identified as Object 1 and Object 2, respectively.

9.4.2.2 Notice that references for the Local Link Objects are not included; the existence of these objects are implied by behavior and not explicitly included. Therefore, these objects are not accessible from the network.

9.4.2.3 Table 18 shows the mapping of the SDM.

9.4.2.4 Objects specified in SEMI E54.3 are listed under the heading NCS Module ID. Every instance is allocated to a slot number. The configuration can be modeled in the GSD by using the Module ID as key for the module description.

Table 18 Mass Flow Device Object Identifiers

SDM Object Name	*SDM Object ID*	*NCS Module ID*
Sensor-AI-MF	MFD3	103
Sensor-AI-AT	MFD4	104
Assembly-MFM	MFD5	105
Sensor-AI-Aux	MFD6	106
Actuator-AO-MF	MFD7	107
Controller	MFD8	108
Local Link	MFD9	109
SISO	MFD10	110
SISO-Setpoint	MFD11	111
Assembly-MFC	MFD12	112

9.4.3 *Attributes*

9.4.3.1 The mapping of Attribute Tags and Identifiers is defined in §7 for SAC. The same method applies here for the SDM.

9.4.3.2 Tables 19–23 show the mapping of the specific SDM attributes.

[5] Website maintained by PROFIBUS International.

Table 19 Sensor AI-MF Object Attribute Identifiers

SEMI SDM Attribute ID	*PROFINET Index*	*Attribute*
A1	160	Flow Totalizer
A2	161	Flow Hours
A5	164	Zero Offset Mode
A6	165	Zeroing Status
A7	166	Autorange Status

Table 20 Actuator AO-MF Object Attribute Identifiers

SEMI SDM Attribute ID	*PROFINET Index*	*Attribute*
A1	160	Valve Type
A2	161	Override

Table 21 Controller Object Attribute Identifiers

SEMI SDM Attribute ID	*PROFINET Index*	*Attribute*
A1	160	Valve Type
A2	161	Override

Table 22 SISO Object Attribute Identifiers

SEMI SDM Attribute ID	*PROFINET Index*	*Attribute*
A1	65	Input
A2	66	Output
A3	67	Data Type

Table 23 SISO Setpoint Object Attribute Identifiers

SEMI SDM Attribute ID	*PROFINET Index*	*Attribute*
A33	97	Ramp Type
A34	98	Ramp Rate
A35	99	Ratio

9.4.4 *Services*

9.4.4.1 The mapping of Service Tags and Identifiers is defined in §7 for SAC. The same method applies here for the SDM.

9.4.4.2 The additional Services of Sensor-AI-MF are mapped as described in Table 24.

Table 24 Sensor-AI-MF Additional Service Mapping

SEMI SDM Service ID	*Service*	*Write Value to Index 64*
S1	Perform Zero Offset	11
S2	Query Supported Gas Type	12
S3	Select Programmed Gas Type	13
S4	Insert Gas Type	14
S5	Delete Gas Type	15
S6	Get Gas Calibration Data Value	16
S7	Set Gas Calibration Data Value	17
S8	Autorange	18

9.5 *In-Situ Particle Monitor Device*

9.5.1 Reference SEMI E54.10 for a complete specification of the SDM for In-Situ Particle Monitor Device (ISPM).

9.5.2 *Objects*

9.5.2.1 Consistent with SEMI E54.10 and ¶7.4 above, the DM and SAC objects are identified as Object 1 and Object 2, respectively.

9.5.2.2 Notice that references for the Local Link Objects are not included; the existence of these objects are implied by behavior and not explicitly included. Therefore, these objects are not accessible from the network.

9.5.2.3 Table 25 shows the mapping of the SDM.

9.5.2.4 Objects specified in SEMI E54.10 are listed under the heading NCS Module ID. Every instance is allocated to a slot number. The configuration can be modeled in the GSD by using the Module ID as key for the module description.

Table 25 In-Situ Particle Monitor Device Object Identifiers

SDM Object Name	*SDM Object ID*	*NCS Module ID*
Sensor-AI-LCS	ISPMD03	203
Sensor-AI-SLS	ISPMD04	204
Sensor-AI- MNS	ISPMD05	205
Sensor-AI-Counter	ISPMD16	216
Assembly-ISPM#1	ISPMD17	217
Assembly-ISPM#2	ISPMD18	218
Assembly-ISPM#3	ISPMD19	219
Assembly-ISPM#4	ISPMD20	220
Assembly-ISPM#5	ISPMD21	221
Assembly-ISPM#6	ISPMD22	222
Assembly-ISPM#7	ISPMD23	223
Assembly-ISPM#8	ISPMD24	224
Assembly-ISPM#9	ISPMD25	225

9.5.3 *Attributes*

9.5.3.1 The mapping of Attribute Tags and Identifiers is defined in §7 for SAC. The same method applies here for the SDM.

9.5.3.2 Tables 26–31 show the mapping of the specific SDM attributes.

Table 26 DM Object Additional Attribute Identifiers

SEMI SDM Attribute ID	*PROFINET Index*	*Attribute*
DmA33	96	Gain
DmA34	97	Filter Bandwidth
DmA35	98	Tool State
DmA36	99	Laser Status
DmA37	100	Flow Path
DmA38	101	Volume
DmA39	102	Volume Units
DmA40	103	Leak Status
DmA41	104	Time Stamp

Table 27 SAC Object Additional Attribute Identifiers

SEMI SDM Attribute ID	*PROFINET Index*	*Attribute*
SacA65	129	Number of Bins
SacA66	130	Count Mode
SacA67	131	Duration

Table 28 Sensor AI-LCS Object Attribute Identifiers

SEMI SDM Attribute ID	*PROFINET Index*	*Attribute*
LcsA1	160	Reading Valid
LcsA2	161	Full Scale
LcsA3	162	Alarm Settling Time
LcsA4	163	Warning Settling Time

Table 29 Sensor AI-SLS Object Attribute Identifiers

SEMI SDM Attribute ID	*PROFINET Index*	*Attribute*
SlsA1	160	Reading Valid
SlsA2	161	Full Scale
SlsA3	162	Alarm Settling Time
SlsA4	163	Warning Settling Time

Table 30 Sensor AI-MNS Object Attribute Identifiers

SEMI SDM Attribute ID	*PROFINET Index*	*Attribute*
MnsA1	160	Reading Valid
MnsA2	161	Full Scale
MnsA3	162	Alarm Settling Time
MnsA4	163	Warning Settling Time

Table 31 Sensor AI-Counter Object Attribute Identifiers

SEMI SDM Attribute ID	*PROFINET Index*	*Attribute*
CounterA1	160	Reading Valid
CounterA2	161	Full Scale
CounterA3	162	Alarm Settling Time
CounterA4	163	Warning Settling Time
CounterA5	164	Upper Size
CounterA6	165	Lower Size

9.5.4 *Services*

9.5.4.1 The mapping of Service Tags and Identifiers is defined in §7 for SAC. The same method applies here for the SDM.

9.5.4.2 The additional Services of DM are mapped as described in Table 32.

Table 32 DM Additional Service Mapping

SEMI SDM Service ID	*Service*	*Write Value to Index 96*
DmS1	Laser On	11
DmS2	Laser Off	12

9.5.4.3 The additional Services of SAC are mapped as described in Table 33.

Table 33 SAC Additional Service Mapping

SEMI SDM Service ID	*Service*	*Write Value to Index 64*
SacS1	Clear Counts	11

9.6 *Endpoint Device*

9.6.1 Reference SEMI E54.11 for a complete specification of the SDM for Endpoint Devices.

9.6.2 Date-and-Time data structure shall be mapped to PROFINET data type Time Of Day. An Offset of 4383 has to be added to Time Of Day day value to obtain the number of days since 1/1/72 (Time Of Day base is 1/1/84).

9.6.3 *Objects*

9.6.3.1 Consistent with SEMI E54.11 and ¶7.4 above, the DM and SAC objects are identified as Object 1 and Object 2, respectively.

9.6.3.2 Notice that references for the Local Link Objects are not included; the existence of these objects are implied by behavior and not explicitly included. Therefore, these objects are not accessible from the network.

9.6.3.3 Table 34 shows the mapping of the SDM.

9.6.3.4 Objects specified in SEMI E54.11 are listed under the heading NCS Module ID. Every instance is allocated to a slot number. The configuration can be modeled in the GSD by using the Module ID as key for the module description.

9.6.3.5 The Sensor-BI-TH-EP can occur more than once. This shall be modeled in GSD in that way, that this Module ID can be placed in several slots.

Table 34 Endpoint Device Object Identifiers

SDM Object Name	*SDM Object ID*	*NCS Module ID*
Sensor-BI-TH-EP	EPD3	203
Assembly-EPD#1	EPD4	204
Assembly- EPD#2	EPD5	205
Assembly- EPD#3	EPD6	206
Assembly- EPD#4	EPD7	207

9.6.4 *Attributes*

9.6.4.1 The mapping of Attribute Tags and Identifiers is defined in §7 for SAC. The same method applies here for the SDM.

9.6.4.2 Tables 35 and 36 show the mapping of the specific SDM attributes.

Table 35 DM Object Additional Attribute Identifiers

SEMI SDM Attribute ID	*PROFINET Index*	*Attribute*
SacA65	129	Number of Endpoint Objects

Table 36 Sensor-BI-TH-EP Object Additional Attribute Identifiers

SEMI SDM Attribute ID	*PROFINET Index*	*Attribute*
EpA1	160	Minimum Time
EpA2	161	Maximum Time
EpA3	162	Target Time
EpA4	163	Elapsed Time
EpA5	164	Time Stamp
EpA6	165	Recipe Identifier
EpA7	166	Step Identifier

9.6.5 *Services*

9.6.5.1 The mapping of Service Tags and Identifiers is defined in §7 for SAC. The same method applies here for the SDM.

9.6.5.2 The additional Services of SAC are mapped as described in Table 37.

Table 37 SAC Additional Service Mapping

SEMI SDM Service ID	*Service*	*Write Value to Index 64*
SacS33	Reset Endpoint	33
SacS34	Download Recipe	34
SacS35	Upload Recipe	35
SacS36	Calibrate	36

9.6.5.3 The additional Services of Sensor BI-EP are mapped as described in Table 38.

Table 38 Sensor BI-EP Additional Service Mapping

SEMI SDM Service ID	*Service*	*Write Value to Index 64*
EpS1	Endpoint On	11
EpS2	Endpoint Off	12
EpS3	Endpoint Start	13
EpS4	Endpoint Suspend	14
EpS5	Endpoint Resume	15

10 Related Documents

10.1 *IEEE Standards*[6]

IEEE 802.3 — Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications

IEEE 802.1D — Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Common Specification – Media Access Control (MAC) Bridges

IEEE 802.1Q — Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Common Specification – Virtual Bridged Local Area Networks

10.2 *Internet Engineering Task Force Standards*[7]

RFC 768 — UDP: User Datagram Protocol; IETF, available at <http://www.ietf.org>

RFC 791 — IP: Internet Protocol; IETF, available at <http://www.ietf.org>

RFC 1541 — DHCP: Dynamic Host Configuration Protocol; IETF, available at <http://www.ietf.org>

10.3 *Other Documents*

C706, CAE Specification DCE1.1: Remote Procedure Call (RPC); OSF available at <http://www.opengroup.org/onlinepubs/9629399/toc.htm>

NOTICE: SEMI makes no warranties or representations as to the suitability of the standards set forth herein for any particular application. The determination of the suitability of the standard is solely the responsibility of the user. Users are cautioned to refer to manufacturer's instructions, product labels, product data sheets, and other relevant literature, respecting any materials or equipment mentioned herein. These standards are subject to change without notice.

By publication of this standard, Semiconductor Equipment and Materials International (SEMI) takes no position respecting the validity of any patent rights or copyrights asserted in connection with any items mentioned in this standard. Users of this standard are expressly advised that determination of any such patent rights or copyrights, and the risk of infringement of such rights are entirely their own responsibility.

[6] Institute of Electrical and Electronics Engineers, IEEE Operations Center, 445 Hoes Lane, P.O. Box 1331, Piscataway, New Jersey 08855-1331, USA. Telephone: 732.981.0060; Fax: 732.981.1721, Website: www.ieee.org

[7] Internet Engineering Task Force, c/o Corporation for National Research Initiatives 1895 Preston White Drive, Suite 100 Reston, VA 20191-5434, USA. Telephone: 1.703.620.8990, Fax 1.703.620.9071, Website: www.ietf.org

SEMI E54.15-0305
SENSOR/ACTUATOR NETWORK COMMUNICATION SPECIFICATION FOR SafetyBUS p

This specification was technically approved by the Global Information & Control Committee and is the direct responsibility of the North American Information & Control Committee. Current edition approved by the North American Regional Standards Committee on December 10, 2005. Initially available at www.semi.org February 2005; to be published March 2005.

1 Purpose

1.1 This standard defines a communication specification based on the SafetyBUS p protocol to enable communications between intelligent devices on a sensor/actuator network (SAN) that operate according to SEMI specified device models (common and device specific) in a semiconductor manufacturing equipment.

1.2 *Background and Motivation*

1.2.1 SafetyBUS p is a device level network which provides a simple, inexpensive, fast, and deterministic means of exchanging data among industrial devices (e.g., sensors and actuators) and higher level devices such as controllers (e.g., programmable control systems) while at the same time providing for the communication of information at a high priority level to support reliable operations. SafetyBUS p provides:

- A solution to low-level device networking,
- Access to intelligence present in low-level devices,
- Networking between higher level controllers,
- Master/Slave and Peer-to-Peer communication capabilities, and
- Serial communication of device-critical information.

1.2.2 SafetyBUS p specifies a communication model and protocol. The Physical and Data Link Layer definitions are defined by the Controller Area Network (CAN) technology. The CAN specification provides the Media Access Control (MAC) methodology and physical signaling characteristics.

1.2.3 This document enables communications between intelligent devices on a SEMI compliant SAN by providing a presentation mapping of common and specific device network visible structure and behavior to SafetyBUS p network.

1.2.4 This Network Communication Standard specification for SafetyBUS p is not intended to be a safety guideline for using SafetyBUS p technology. This document specifies a SAN communications standard specification based on the Object Communication Specification (OCS) for the SafetyBUS p Protocol. This SAN is structured in compliance with SEMI E54.1.

2 Scope

2.1 This document specifies the protocol and services that compliant intelligent devices shall support to exchange information over the SafetyBUS p semiconductor equipment sensor/actuator network.

2.2 This document specifies the utilization of the SafetyBUS p protocol to present externally visible device structure and behavior, specified in the Common Device Model (CDM) and appropriate Specific Device Models (SDMs), on a SafetyBUS p network.

2.3 This document is used in conjunction with a SEMI standard SAN Common Device Model specification, one or more SEMI standard Specific Device Model (SDM) specifications (e.g. for a mass flow device), the SafetyBUS p Object Communications Specification Reference Guide. Together, they describe the SafetyBUS p protocol, the externally visible data structures and behaviors of devices utilizing the SafetyBUS p networking capability in a SEMI compliant SAN system.

NOTICE: This standard does not purport to address safety issues, if any, associated with its use. It is the responsibility of the users of this standard to establish appropriate safety and health practices and determine the applicability of regulatory or other limitations prior to use.

3 Limitations

3.1 This document specifies a semiconductor equipment SAN based solely on SafetyBUS p and is a companion document to the Object Communication Specification for the SafetyBUS p Protocol; thus a complete specification of this standard necessarily includes the SafetyBUS p protocol specifications. There are other semiconductor equipment SAN communications options. The specifications for these options are not included.

3.2 This standard specifies enhancements that provide additional capabilities over and above those currently required by SafetyBUS p. In order to avoid document consistency problems, information in the Object Communication Specification for the SafetyBUS p Protocol that relate to this standard is not repeated in this document. This document is limited to describing enhancements or limitations to the SafetyBUS p specifications that are imposed by this standard specification.

4 Referenced Standards

4.1 *SEMI Standards*

SEMI E39 — Object Services Standard: Concepts, Behavior, and Services (OSS)

SMI E54 — Sensor/Actuator Network Standard

SEMI E54.1 — Standard For Sensor/Actuator Network Common Device Model

SEMI E54.3 — Specification For Sensor/Actuator Network Specific Device Model For Mass Flow Device

SEMI E54.10 — Specification For Sensor/Actuator Network Specific Device Model For An In-Situ Particle Monitor Device

SEMI E54.11 — Specification For Sensor/Actuator Network Specific Device Model For An Endpoint Device

4.2 *ISO Standards*[1]

ISO 7498 — Basic Reference Model for Open Systems Interconnection

4.3 *SafetyBUS p Club International*[2]

Specification — Object Communication Specification (OCS) for the SafetyBUS p Protocol Version 1.1 April 09, 2004 may be accessed on the SafetyBUS p Club International web site http://www.safetybus.com/semi/

Reference Guide — Pilz Automation Technology – Guide to Programmable Safety Systems, February 2002 Volume 2, 1st Edition

NOTICE: Unless otherwise indicated, all documents cited shall be the latest published versions.

5 Terminology

5.1 Terminology that is common to all of the documents in this SAN standard may also be defined in the Sensor /Actuator Network Standard (see reference SEMI E54 ¶4.1). Terminology may be reproduced here which is defined in other SEMI documents.

5.2 *Abbreviations and Acronyms*

5.2.1 *CAN* — Controller Area Network

5.2.2 *CDM* — Common Device Model

5.2.3 *DM* — Device Manager (object)

[1] International Organization for Standardization, ISO Central Secretariat, 1, rue de Varembé, Case postale 56, CH-1211 Geneva 20, Switzerland. Telephone: 41.22.749.01.11; Fax: 41.22.733.34.30, Website: /www.iso.ch

[2] SafetyBUS p Club International Partner Association, Robert-Bosch-Str. 30, 73760 Ostfildern, Germany ,Telephone (+49) 711 3409-118, Fax: (+49) 711 3409-449, website: www.safetybus.com

5.2.4 *ISO* — International Standards Organization

5.2.5 *MAC* — Media Access Control

5.2.6 *NCS* — Network Communication Standard

5.2.7 *OCS* — Object Communications Specification

5.2.8 *OSI* — Open Systems Interconnect

5.2.9 *SAC* — Sensor, Actuator, Controller (Object)

5.2.10 *SAN* — Sensor/Actuator Network

5.2.11 *SDM* — Specific Device Model

5.3 *Device Component Definitions*

5.3.1 As this standard specification defines the presentation or mapping of CDM data structure and behavior over a network, it makes use of many of the terms in the SEMI E54.1 CDM document. Table 1 provides a mapping of fundamental terminology of the CDM document into this document and the SafetyBUS p standard specifications. Note that Column 2 contains an equal sign "=" if the definition is used exactly as specified in the CDM specification.

Table 1 Mapping of Common Device Model to NCS Terminology

CDM Term	*NCS Equivalent*	*SafetyBUS p Equivalent*
Device	=	=
Device Model	=	=
Class	=	Server Class
Object	=, Class, Instance	=, Server Class, Instance
Instance	=	=
Attribute	=	=
Behavior	=	=
Service	=	=
State Diagram	=	=
Byte	=	=
Nibble	=	=
Character String	=	=

5.4 *SafetyBUS p Specific Definitions*

5.4.1 *Server Class* — subset of devices that offer similar functions and provide fixed defined functionality in a uniform way.

5.4.2 *master/slave* — communication over a SafetyBUS p network provides exclusive control of data by a "master" or "host" device. All network input data is reported exclusively to the host when requested by the host, and the host has exclusive control over the states of all network output signals of all nodes acting as its "slaves". Master/Slave communication provides the typical request/response oriented network communications.

5.4.3 *peer-to-peer* — On SafetyBUS p networks, messages formatted according to the SafetyBUS p protocol are embedded into the SafetyBUS p packet structure that is used on the CAN network. The SafetyBUS p protocol over CAN supports the asynchronous or unsolicited bi-directional transmission of data between nodes. This type of communication is referred to as peer-to-peer.

5.4.4 *SafetyBUS p* — an open protocol maintained by Pilz GmbH & Co and distributed by SafetyBUS p Club International as a reliable and standard means of interconnection for simple field devices. The SafetyBUS p standard wraps a communication model and protocol as well as CAN specifications for OSI reference model layers 1 and 2, to provide a complete network definition. The OSI reference model layer 7 specifies the application layer.

6 Communication Protocol High Level Structure

6.1 The SafetyBUS p protocol is loosely based on three layer architecture. These three layers constitute a collapsed form of the seven layer OSI architecture which map into the physical, data link, and application layers of the OSI Basic Reference Model (see ¶4.2). The high-level protocol architecture is shown in Figure 1.

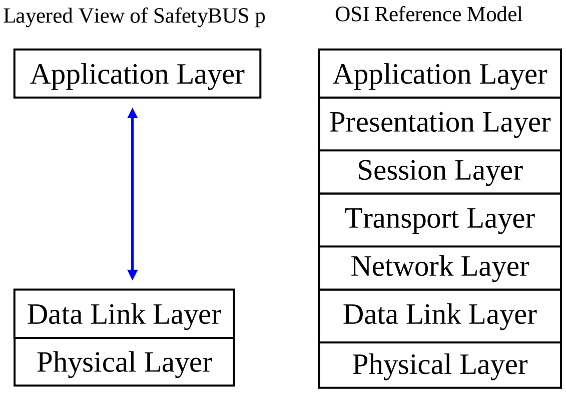


Figure 1
Layered View of SafetyBUS p

6.1.1 Note that Figure 1 represents a conceptual view of the technology architecture. Conforming implementations must implement the services defined in this specification at each layer and must appear (from the network) to have implemented this architecture, however an internal modular partitioning is not required. Implementations may sacrifice modularity in order to achieve high performance.

6.1.2 The application layer is specified in the SafetyBUS p OCS (see reference ¶4.3) and provides for the definition of SafetyBUS p applications as a collection of addressable objects. A subset of these objects may be addressed over the network (as defined by the implementation).

6.1.3 In the remainder of this section the protocol structure is described in more detail in terms of the OSI seven layer reference model, the object model environment and network management specifications.

6.2 *Physical Layer* — The device shall comply with a physical layer specification identified in the Controller Area Network (CAN) specification. Physical layer specification includes physical signaling (levels and data rates), transceivers, node isolation, media topology, cable specifications, network connectors and taps, and power considerations (load limits, system tolerances, and power supply options).

6.3 *Data Link Layer* — The device shall comply with a data link layer specification the Controller Area Network (CAN) specification. Data link layer specification includes the media access control mechanism and the logical link control mechanism.

6.4 *Network Layer* — There is no distinct Network layer.

6.5 *Transport (Messaging) Layer* — There is no distinct Transport layer.

6.6 *Session Layer* — There is no distinct Session layer.

6.7 *Presentation Layer* — There is no distinct Presentation layer.

6.8 *Application Layer* — The device shall comply with the SafetyBUS p application layer specification for defining and addressing objects, including their attributes and services, and enabling specified network behavior. The device shall comply with the object messaging and object model specifications included in the SafetyBUS p OCS. In addition the device shall comply with the object specifications defined in §7 of this document.

6.8.1 *Object Models* — The SafetyBUS p protocol includes an object-oriented specification for addressing objects explicitly, including their attributes and services, and communicating object attributes in an application dependent format. The device shall comply with the object messaging and object model specifications included in the

SafetyBUS p OCS documentation. In addition the device shall comply with the object specifications defined in §7 of this document.

6.9 *Network Management* — The device shall comply with the SafetyBUS p and CAN network management specifications detailed in the SafetyBUS p standard and SafetyBUS p Guide to Programmable Safety Systems Standard Specifications (e.g., physical layer bit rate, master/slave and peer-to-peer network management, etc.). No (additional) network management functions are specified in this document.

7 Required And Optional Object Types

7.1 The SafetyBUS p standard specification does not require any specific objects to exist in a SafetyBUS p device in order to be a compliant SafetyBUS p device. The SafetyBUS p standard specification is extended in this standard to identify and describe objects (i.e. classes) that shall exist in devices that are to be interoperable and interchangeable on a SafetyBUS p SEMI compliant SAN network.

7.1.1 The Common Device Model (CDM) specification (see reference SEMI E54.1 ¶4.1) identifies two objects (namely the Device Manager (DM) and Sensor Actuator Controller (SAC) objects) that shall exist in all SEMI compliant SAN devices.

7.1.2 The required object types for a SEMI compliant SAN device utilizing the network communication specification described herein necessarily comprises, at minimum, the union of the SafetyBUS p object type requirements and the CDM specification requirements.

7.1.3 A list of required and optional object types is given in Table 2. Additional objects that are specified in a particular SDM are given identifiers in that SDM specification; SafetyBUS p specific presentation information for these identifiers is given in Section 9 of this document.

Table 2 Required and Optional Object Types

Object Name	*SafetyBUS p Class ID / Instance ID* [#1]	*CDM Tag* [#2]	*Required by SafetyBUS p* [#1]	*Required by CDM* [#2]	*Required by NCS*
Device Manager	1/1	DmI0	No	Yes	Yes
Sensor/ Actuator/ Controller	2/1	SacI0	No	Yes	Yes
Assembly	3/1 through i	Asm	No	No	No
Local Link	4/1 through j	Lnk	No	No	No
Sensor – AI	33/1 through k	Sai	No	No	No
Sensor – EI	34/1 through l	Sei	No	No	No
Sensor – BI	35/1 through m	Sbi	No	No	No
Actuator – AO	36/1 through n	Aao	No	No	No
Actuator – EO	37/1 through o	Aeo	No	No	No
Actuator – BO	38/1 through p	Abo	No	No	No
Controller	39/1 through q	C	No	No	No
Sensor – BI-TH	40/I through s	Sbith	No	No	No
Application Objects	129 through x/1 through r	(3)	No	No	No

[#1] See SafetyBUS p specification for further information; values are decimal; 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r' and 's' represent arbitrary numbers (greater than or equal to 1) indicating that more than one instance may be supported. 'x' is a number greater than or equal to 129 indicating that one or more application object classes may be supported.

[#2] See CDM specification for further information

[#3] Application Dependent objects' tags as specified in SDM

7.1.4 An embodiment of a specific device type represented as an aggregation of the object types listed in Table 2 that is compliant with both the CDM specification and the SafetyBUS p specification, is a candidate for a SEMI SDM as well as a SafetyBUS p device definition. Conversely, all SEMI SDMs and SafetyBUS p device definitions specified for operation over a SEMI compliant SafetyBUS p network must be an aggregation of the object types listed in Table 2, and be compliant with both the CDM specification and the SafetyBUS p standard specifications.

7.1.5 In the following sections the presentation to the network of object addressing, object attributes and object services for each of the object types listed in Table 2 is described in detail. Refer to the CDM standard to determine if the object instance attribute and service is specified as required or optional. Unless otherwise noted, all attributes and services described are instance level attributes (as opposed to class level attributes). A class level attribute and service is accessed as instance number zero.

7.1.6 Note that the formats of object attributes and services are detailed in the CDM document; the presentation of object attributes and services to the SafetyBUS p network is detailed in the tables contained in the following sub-sections and in the SafetyBUS p standard specifications.

7.2 *Device Manager (DM) Object* — The DM object instance is the device component responsible for managing and consolidating the device operation. Each device must support one (and only one) DM object. The DM object as well as its common required and optional attributes, services and behavior are described in the CDM standard. The presentation of object instance attributes and services to the SafetyBUS p network shall be as indicated in Table 3. Note that all service ID values identified refer to the ID of the request or notification component of that service. Corresponding reply components to request/reply services shall have a service ID value equal to the request component ID plus one.

Table 3 DM Object Instance Attributes and Services

Device Manager Object (DM) *Class ID = 01, Instance ID = 01*		
Attributes		
ID	*Attribute Name*	*CDM Tag*
01	Device Type	DmA1
02	Standard Revision Level	DmA2
03	Device Manufacturer Identifier	DmA3
04	Manufacturer Model Number	DmA4
05	Software or Firmware Revision Level	DmA5
06	Hardware Revision Level	DmA6
07	Serial Number	DmA7
08	Device Configuration	DmA8
09	Device Status	DmA9
10	Reporting Mode	DmA10
11	Exception Status Report	DmA11
12	Exception Status	DmA12
13	Exception Detail Alarm	DmA13
14	Exception Detail Warning	DmA14
15	Visual Indicator	DmA15
16	Alarm Enable	DmA16
17	Warning Enable	DmA17
18	Exception Detail Type	DmA18
19	Exception Detail Alarm Queue	DmA19
20	Exception Detail Warning Queue	DmA20
21	Date and Time	DmA21
22	Date and Time Type	DmA22

Services		
ID	*Service Name*	*CDM Tag*
01	Reset	DmS1
03	Abort	DmS2
05	Recover	DmS3
07	Get Attribute	DmS4
09	Set Attribute	DmS5
11	Execute	DmS6
13	Perform Diagnostics	DmS7
15	Publish Attribute	DmS8
17	Lock	DmS9
19	Unlock	DmS10
21	Get Exception Queue	DmS11
23	Clear Exception Queue	DmS12

7.3 *Sensor, Actuator, Controller (SAC) Object* — The SAC object instance is the device component responsible for coordinating the interaction of the device with the sensory/actuation/control environment. Each device must support one (and only one) SAC object instance. The SAC object instance as well as its common required and optional attributes, services and behavior are described in the CDM standard. The presentation of object instance attributes and services to the SafetyBUS p network shall be as indicated in Table 4.

Table 4 Object Instance Attributes and Services

Sensor, Actuator, Controller Object (SAC) *Class ID = 02, Instance ID = 01*		
Attributes		
ID	*Attribute Name*	*CDM Tag*
01	Last Calibration Date	SacA1
02	Next Calibration Date	SacA2
03	Expiration Timer	SacA3
04	Expiration Warning Enable	SacA4
05	Run Hours	SacA5
Services		
ID	*Service Name*	*CDM Tag*
01	Reset	SacS1
03	Abort	SacS2
05	Recover	SacS3
07	Get Attribute	SacS4
09	Set Attribute	SacS5
25	Operate	SacS6
27	Restore Default	SacS7
29	Publish Attribute	SacS8

7.4 *Assembly Object (Asm)* — The Assembly (Asm) object instances may be used to provide for grouping more than one attribute from one or more object instances in a device into a single data structure for communication over the SafetyBUS p Network. The presentation of object instance attributes and services shall be as indicated in Table 5.

Table 5 Assembly Object Instance Attributes and Services

ASSEMBLY Object (Asm) *Class ID = 03, Instance ID = 01 through i*		
Attributes		
ID	*Attribute Name*	*CDM Tag*
01	Data	AsmA1
Services		
ID	*Service Name*	*CDM Tag*
07	Get Attribute	AsmS4
09	Set Attribute	AsmS5

7.5 *Local Link Object (Lnk)* — The Local Link (Lnk) object instances may be used to 'link" an attribute of one object instance to an attribute of another object instance. Refer to the CDM for further explanation and use of this object. The presentation of object instance attributes and services are as indicated in Table 6.

Table 6 Local Link Object Instance Attributes and Services

Local Link Object (Lnk) *Class ID = 04, Instance ID = 01 through j*		
Attributes		
ID	*Attribute Name*	*CDM Tag*
01	Source Object Class	LnkA1
02	Source Object Instance	LnkA2
03	Source Object Attribute	LnkA3
04	Destination Object Class	LnkA4
05	Destination Object Instance	LnkA5
06	Destination Object Attribute	LnkA6
07	Commit	LnkA7
Services		
ID	*Service Name*	*CDM Tag*
--	No services defined	--

7.6 *Sensor-AI Object (Sai)* — The presentation of the Sensor Analog Input (Sensor-AI) object instance attributes and services are as indicated in Table 7.

Table 7 Sensor-AI Object Instance Attributes and Services

Sensor-AI (Sai) *Class ID = 33, Instance ID = 01 through k*		
Attributes		
ID	*Attribute Name*	*CDM Tag*
01	Name	SaiA1
02	Status	SaiA2
03	Alarm Enable	SaiA3
04	Warning Enable	SaiA4
16	Value	SaiA16
17	ReportInhibitTimer	SaiA17
18	EnableReportRate	SaiA18
19	ReportRate	SaiA19
64	Offset	SaiA64
65	Gain	SaiA65
66	DataType	SaiA66
67	DataUnits	SaiA67
68	SafeState	SaiA68
69	EnableReportDelta	SaiA69
70	ReportDelta	SaiA70
71	EnableReportROC	SaiA71
72	ReportROC	SaiA72
73	AlarmTripPointHigh	SaiA73
74	AlarmTrippointLow	SaiA74
75	AlarmHystersis	SaiA75
76	WarningTripPointHigh	SaiA76
77	WarningTripPointLow	SaiA77
78	WarningHystersis	SaiA78
Services		
ID	*Service Name*	*CDM Tag*
01	Reset	SaiS1
03	Abort	SaiS2
05	Recover	SaiS3
25	Operate	SaiS4
07	GetAttribute	SaiS5
09	SetAttribute	SaiS6
27	RestoreDefault	SaiS7

7.7 *Sensor-EI Object (Sei)* — The presentation of the Sensor Enumerated Input (Sensor-EI) object instance attributes and services are as indicated in Table 8.

Table 8 Sensor-EI Object Instance Attributes and Services

Sensor-EI (Sei) *Class ID = 34, Instance = 01 through l*		
Attributes		
ID	*Attribute Name*	*CDM Tag*
01	Name	SeiA1
02	Status	SeiA2
03	Alarm Enable	SeiA3
04	Warning Enable	SeiA4
16	Value	SeiA16
17	ReportInhibitTimer	SeiA17
18	EnableReportRate	SeiA18
19	ReportRate	SeiA19
64	DebounceControl	SeiA64
65	AlarmState	SeiA65
66	WarningState	SeiA66
Services		
ID	*Service Name*	*CDM Tag*
01	Reset	SeiS1
03	Abort	SeiS2
05	Recover	SeiS3
25	Operate	SeiS4
07	GetAttribute	SeiS5
09	SetAttribute	SeiS6
27	RestoreDefault	SeiS7

7.8 *Sensor-BI Object (Sbi)* — The presentation of the Sensor Binary Input (Sensor-BI) object instance attributes and services are as indicated in Table 9.

Table 9 Sensor-BI Object Instance Attributes and Services

Sensor-BI (Sbi) *Class ID = 35, Instance ID = 01 through m*		
Attributes		
ID	*Attribute Name*	*CDM Tag*
01	Name	SbiA1
02	Status	SbiA2
03	Alarm Enable	SbiA3
04	Warning Enable	SbiA4
16	Value	SbiA16
17	ReportInhibitTimer	SbiA17
18	EnableReportRate	SbiA18
19	ReportRate	SbiA19
64	DebounceControl	SbiA64
65	AlarmState	SbiA65
66	WarningState	SbiA66

Services		
ID	*Service Name*	*CDM Tag*
01	Reset	SbiS1
03	Abort	SbiS2
05	Recover	SbiS3
25	Operate	SbiS4
07	GetAttribute	SbiS5
09	SetAttribute	SbiS6
27	RestoreDefault	SbiS7

7.9 *Sensor-BI-TH Object (Sbith)* — The presentation of the Sensor Binary Input Threshold (Sensor-BI-TH) object instance attributes and services are as indicated in Table 10.

Table 10 Sensor-BI-TH Object Instance Attributes and Services

Sensor-BI (Sbith) *Class ID = 40, Instance ID = 01 through m*		
Attributes		
ID	*Attribute Name*	*CDM Tag*
01	Name	SbiA1
02	Status	SbiA2
03	Alarm Enable	SbiA3
04	Warning Enable	SbiA4
16	Value	SbiA16
17	ReportInhibitTimer	SbiA17
18	EnableReportRate	SbiA18
19	ReportRate	SbiA19
64	DebounceControl	SbiA64
65	AlarmState	SbiA65
66	WarningState	SbiA66
67	Reading Valid	SbithA64
68	State	SbithA65
69	Status	SbithA66
Services		
ID	*Service Name*	*CDM Tag*
01	Reset	SbiS1
03	Abort	SbiS2
05	Recover	SbiS3
25	Operate	SbiS4
07	GetAttribute	SbiS5
09	SetAttribute	SbiS6
27	RestoreDefault	SbiS7

7.10 *Actuator-AO Object (Aao)* — The presentation of the Actuator Analog Output (Actuator-AO) object instance attributes and services are as indicated in Table 11.

Table 11 Actuator-AO Object Instance Attributes and Services

Actuator-AO (Aao) *Class ID = 36, Instance = 01 through n*		
Attributes		
ID	*Attribute Name*	*CDM Tag*
01	Name	AaoA1
02	Status	AaoA2
03	Alarm Enable	AaoA3
04	Warning Enable	AaoA4
16	Setting	AaoA16
17	SafeState	AaoA17
18	WatchRate	AaoA18
19	Watchdog	AaoA19
64	Offset	AaoA64
65	Gain	AaoA65
66	DataType	AaoA66
67	DataUnits	AaoA67
Services		
ID	*Service Name*	*CDM Tag*
01	Reset	AaoS1
03	Abort	AaoS2
05	Recover	AaoS3
25	Operate	AaoS4
07	GetAttribute	AaoS5
09	SetAttribute	AaoS6
27	RestoreDefault	AaoS7

7.11 *Actuator-EO Object (Aeo)* — The presentation of the Actuator Enumerated Output (Actuator-EO) object instance attributes and services are as indicated in Table 12.

Table 12 Actuator-EO Object Instance Attributes and Services

Actuator-EO (Aeo) *Class ID = 37, Instance ID = 01 through o*		
Attributes		
ID	*Attribute Name*	*CDM Tag*
01	Name	AeoA1
02	Status	AeoA2
03	Alarm Enable	AeoA3
04	Warning Enable	AeoA4
16	Setting	AeoA16
17	SafeState	AeoA17
18	WatchRate	AeoA18
19	Watchdog	AeoA19

Services		
ID	*Service Name*	*CDM Tag*
01	Reset	AeoS1
03	Abort	AeoS2
05	Recover	AeoS3
25	Operate	AeoS4
07	GetAttribute	AeoS5
09	SetAttribute	AeoS6
27	RestoreDefault	AeoS7

7.12 *Actuator-BO Object (Abo)* — The presentation of the Actuator Binary Output (Actuator-BO) object instance attributes and services are as indicated in Table 13.

Table 13 Actuator-BO Object Instance Attributes and Services

Actuator-BO (Abo) *Class ID = 38, Instance ID = 01 through p*		
Attributes		
ID	*Attribute Name*	*CDM Tag*
01	Name	AboA1
02	Status	AboA2
03	Alarm Enable	AboA3
04	Warning Enable	AboA4
16	Setting	AboA16
17	SafeState	AboA17
18	WatchRate	AboA18
19	Watchdog	AboA19
Services		
ID	*Service Name*	*CDM Tag*
01	Reset	AboS1
03	Abort	AboS2
05	Recover	AboS3
25	Operate	AboS4
07	GetAttribute	AboS5
09	SetAttribute	AboS6
27	RestoreDefault	AboS7

7.13 *Controller Object (C)* — The presentation of the Controller (C) object instance attributes and services are as indicated in Table 14.

Table 14 Controller-C Instance Object Attributes and Services

Controller (C) *Class ID = 39, Instance ID = 01 through q*		
Attributes		
ID	*Attribute Name*	*CDM Tag*
01	Name	CA1
02	Status	CA2
03	Alarm Enable	CA3
04	Warning Enable	CA4
16	Setpoint	CA16
17	ProcessVariable	CA17
18	ControlVariable	CA18
19	DataType	CA19
20	DataUnits	CA20
21	AlarmSettleTime	CA21
22	AlarmErrorBand	CA22
24	WarningSettleTime	CA24
25	WarningErrorBand	CA25
Services		
ID	*Service Name*	*CDM Tag*
01	Reset	CS1
03	Abort	CS2
05	Recover	CS3
25	Operate	CS4
07	GetAttribute	CS5
09	SetAttribute	CS6
27	RestoreDefault	CS7

8 Protocol Compliance

8.1 A method of testing protocol compliance is required to verify implementation conformance to the standard. SafetyBUS p Club International e.V. has established and maintains a mechanism for compliance certification of devices on a SafetyBus p network. This certification includes procedures and reporting mechanisms to demonstrate conformance testing and interoperability testing of SafetyBUS p devices. The SafetyBUS p Club International e.V. organization can be contacted at the web site http://www.safetybus.com to obtain the latest certification procedures.

9 Specific Device Model Mappings

9.1 This section provides for the mapping of network visible specific device structure and behavior, specified in a SEMI standard SDM specification, to the SafetyBUS p network. Each subsection is devoted to a single Specific Device Model (SDM) specification. Additional SDM mappings are added as sub-sections to this NCS specification according to SEMI guidelines and the guidelines of the SEMI SAN Interoperability standard. Unless otherwise noted, all attributes and services described are instance level attributes (as opposed to class level attributes).

9.1.1 Note that the formats of object instance attributes and services are detailed in the associated SDM specification; the presentation of object attributes and services to the SafetyBUS p network is detailed in the tables contained in the following sub-sections and in the SafetyBUS p standard specifications. Note that relationships between object classes, including inheritance are defined in the associated SDM specification and the CDM specification.

9.1.2 The instance identifier of 1 through s, assigned to an object type, refers to the possibility of multiple instantiations of the object type. Refer to Table 2 of this document and the CDM document for a further explanation of object instance identifier assignments.

9.2 *Specific Device Model for Mass Flow Device*

9.2.1 These sections detail the network mapping required to support the Specific Device Model for Mass Flow Device (see SEMI E54.3 ¶4.1). Table 15 summarizes the Mass Flow Device Object types. Subsequent Tables 16 to 25 details the instance attributes and services associated with each Mass Flow Device object type.

Table 15 Mass Flow Device Object Types

SDM Object Identifier	*Object Name*	*SafetyBUS p Class ID*
MFD1 (DM)	Device Manager	1
MFD2 (SAC)	Sensor Actuator Controller	2
MFD3	Sensor-AI-MF	129
MFD4	Sensor-AI-AT	130
MFD5	Assembly-MFM	131
MFD6	Sensor-AI-Aux	132
MFD7	Actuator-AO-MF	133
MFD8	Controller	39
MFD9	Local Link	4
MFD10	SISO	134
MFD11	SISO-Setpoint	135
MFD12	Assembly-MFC	136

9.2.2 *Sensor-AI-MF* — The presentation of the Sensor Analog Input Mass Flow (Sensor-AI-MF) object instance attributes and services are as indicated in Table 16.

Table 16 Sensor-AI-MF Object Instance Attributes and Services

Sensor-AI-MF Class ID = 129, Instance ID = 01 through r		
Attributes		
ID	*Attribute Name*	*SDM Tag*
128	Flow Totalizer	A1
129	Flow Hours	A2
130	Zero Offset Mode	A5
131	Zeroing Status	A6
132	Autorange Status	A7
Services		
ID	*Service Name*	*SDM Tag*
129	Perform Zero Offset	S1
131	Query-Supported Gas Types	S2
133	Selected Programmed Gas Type	S3
135	Insert Gas Type	S4
137	Delete Gas Type	S5
139	Get Gas Calibration Data Value	S6
141	Set Gas Calibration Data Value	S7
143	Autorange	S8

9.2.3 *Sensor-AI-AT* — The presentation of the Sensor Analog Input Ambient Temperature (Sensor-AI-AT) object instance attributes and services are as indicated in Table 17.

Table 17 Sensor-AI-AT Object Instance Attributes and Services

Sensor-AI-AT *Class ID = 130, Instance ID = 01 through r*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
--	No additional attributes defined	--
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.2.4 *Assembly-MFM* — The presentation of the Assembly Mass Flow Meter (Assembly-MFM) object instance attributes and services are as indicated in Table 18.

Table 18 Assembly-MFM Object Instance Attributes and Services

Assembly-MFM *Class ID = 131, Instance ID = 01*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
01	Data	A1
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.2.5 *Sensor-AI-Aux* — The presentation of the Sensor Analog Input Auxiliary (Sensor-AI-Aux) object instance attributes and services are as indicated in Table 19.

Table 19 Sensor-AI-Aux Object Instance Attributes and Services

Sensor-AI-Aux *Class ID = 132, Instance ID = 01 through r*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
--	No additional attributes defined	--
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.2.6 *Actuator-AO-MF* — The presentation of the Actuator Analog Output Mass Flow (Actuator-AO-MF) object instance attributes and services are as indicated in Table 20.

Table 20 Actuator-AO-MF Object Instance Attributes and Services

Actuator-AO-MF *Class ID = 133, Instance ID = 01 through r*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
128	Valve Type	A1
129	Override	A2

Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.2.7 *Controller* — The presentation of the extended Controller (C) object instance attributes and services are as indicated in Table 21.

Table 21 Controller Object Instance Attributes and Services

Controller *Class ID = 39, Instance ID = 01*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
128	Alarm Settling Time	CA21
129	Warning Settling Time	CA24
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.2.8 *Local Link* — The presentation of the extended Local Link (Lnk) object instance attributes and services are as indicated in Table 22.

Table 22 Local Link Object Instance Attributes and Services

Local Link *Class ID = 4, Instance ID = 01 through j*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
--	No additional attributes defined	--
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.2.9 *SISO* — The presentation of the Single Input Single Output (SISO) object instance attributes and services are as indicated in Table 23.

Table 23 SISO Object Instance Attributes and Services

SISO *Class ID = 134, Instance ID = 01 through r*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
128	Input	A1
129	Output	A2
130	Data Type	A3
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.2.10 *SISO-Setpoint* — The presentation of the Single Input Single Output Setpoint (SISO-Setpoint) object instance attributes and services are as indicated in Table 24.

Table 24 SISO-Setpoint Object Instance Attributes and Services

SISO-Setpoint Class ID = 135, Instance ID = 01 through r		
Attributes		
ID	*Attribute Name*	*SDM Tag*
161	Ramp Type	A33
162	Ramp Rate	A34
163	Ratio	A35
Services		
ID	*Service Name*	*SDM Tag*
---	No additional services defined	--

9.2.11 *Assembly-MFC* — The presentation of the Assembly Mass Flow Controller (Assembly-MFC) object instance attributes and services are as indicated in Table 25.

Table 25 Assembly-MFC Object Instance Attributes and Services

Assembly-MFC Class ID = 136, Instance ID = 01 through r		
Attributes		
ID	*Attribute Name*	*SDM Tag*
01	Data	A1
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3 *Specific Device Model For In-Situ Particle Monitor Device* — These sections detail the network mapping required to support the Specific Device Model for In-Situ Particle Monitor (ISPM) Devices (see SEMI E54.10 ¶4.1). Table 26 summarizes the In-Situ Particle Monitor Device Object types. Subsequent Tables 27 to 42 details the attributes and services associated with each In-Situ Particle Monitor Device object type.

Table 26 In-Situ Particle Monitor Device Object Types

SDM Object Identifier	*Object Name*	*SafetyBUS p Class ID*
ISPMD1 (DM)	Device Manager	1
ISPMD2 (SAC)	Sensor Actuator Controller	2
ISPMD3	Sensor-AI-LCS	137
ISPMD4	Sensor-AI-SLS	138
ISPMD5	Assembly-AI-MNS	139
ISPMD16	Sensor-AI-Counter	140
ISPMD17	Assembly-ISPM#1	141
ISPMD18	Assembly-ISPM#2	142
ISPMD19	Assembly-ISPM#3	143
ISPMD20	Assembly-ISPM#4	144
ISPMD21	Assembly-ISPM#5	145
ISPMD22	Assembly-ISPM#6	146
ISPMD23	Assembly-ISPM#7	147
ISPMD24	Assembly-ISPM#8	148
ISPMD25	Assembly-ISPM#9	149
ISPMD64	Assembly-ISPM#48	150

9.3.1 *Device Manager* — The presentation of the extended ISPM Device Manager (DM) object instance attributes and services are as indicated in Table 27.

Table 27 DM Object Instance Attributes and Services

Device Manager Object (DM) *Class ID = 01, Instance ID = 01*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
128	Gain	A33
129	Filter Bandwidth	A34
130	Tool State	A35
131	Laser Status	A36
132	Flow Path	A37
133	Volume	A38
134	Volume Units	A39
135	Leak Status	A40
136	Time Stamp	A41
Services		
ID	*Service Name*	*SDM Tag*
33	Laser On	S1
35	Laser Off	S2

9.3.2 *Sensor Actuator Controller (SAC)* — The presentation of the extended ISPM Sensor Actuator Controller (SAC) object attributes and services are as indicated in Table 28.

Table 28 SAC Object Instance Attributes and Services

Sensor, Actuator, Controller Object (SAC) *Class ID = 02, Instance ID = 01*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
65	Number of Bins	SacA65
66	Count Mode	SacA66
67	Duration	SacA67
Services		
ID	*Service Name*	*SDM Tag*
33	Clear Counts	S33

9.3.3 *Sensor-AI-LCS* — The presentation of the Sensor Analog Input Laser Current Sensor (Sensor-AI-LCS) object instance attributes and services are as indicated in Table 29.

Table 29 Sensor-AI-LCS Object Instance Attributes and Services

Sensor-AI-LCS *Class ID = 137, Instance ID = 01*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
128	Reading Valid	LcsA1
129	Full Scale	LcsA2
130	Alarm Settling Time	LcsA3
131	Warning Settling Time	LcsA4

Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3.4 *Sensor-AI-SLS* — The presentation of the Sensor Analog Input Stray Light Sensor (Sensor-AI-SLS) object instance attributes and services are as indicated in Table 30.

Table 30 Sensor-AI-SLS Object Instance Attributes and Services

Sensor-AI-SLS *Class ID = 138, Instance ID = 1*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
128	Reading Valid	SlsA1
129	Full Scale	SlsA2
130	Alarm Settling Time	SlsA3
131	Warning Settling Time	SlsA4
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3.5 *Sensor-AI-MNS* — The presentation of the Sensor Analog Input Medium Noise Sensor (Sensor-AI-MNS) object instance attributes and services are as indicated in Table 31.

Table 31 Sensor-AI-MNS Object Instance Attributes and Services

Sensor-AI-MNS *Class ID = 139, Instance ID = 01*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
128	Reading Valid	MnsA1
129	Full Scale	MnsA2
130	Alarm Settling Time	MnsA3
131	Warning Settling Time	MnsA4
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3.6 *Sensor-AI-Counter* — The presentation of the Sensor Analog Input Counter (Sensor-AI-Counter) object instance attributes and services are as indicated in Table 32.

Table 32 Sensor-AI-Counter Object Instance Attributes and Services

Sensor-AI-Counter *Class ID = 140, Instance ID = 01 through 1024*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
128	Reading Valid	CounterA1
129	Full Scale	CounterA2
130	Alarm Settling Time	CounterA3
131	Warning Settling Time	CounterA4
132	Upper Size	CounterA5
133	Lower Size	CounterA6
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3.7 *Assembly-ISPM#1* — The presentation of the Assembly #1 In-Situ Particle Monitor (Assembly-ISPM#1) object instance attributes and services are as indicated in Table 33.

Table 33 Assembly-ISPM#1 Object Instance Attributes and Services

Assembly-ISPM#1 *Class ID = 141, Instance ID = 01*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
01	Data	A1
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3.8 *Assembly-ISPM#2* — The presentation of the Assembly #2 In-Situ Particle Monitor (Assembly-ISPM#2) object instance attributes and services are as indicated in Table 34.

Table 34 Assembly-ISPM#2 Object Instance Attributes and Services

Assembly-ISPM#2 *Class ID = 142, Instance ID = 01*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
01	Data	A1
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3.9 *Assembly-ISPM#3* — The presentation of the Assembly #3 In-Situ Particle Monitor (Assembly-ISPM#3) object instance attributes and services are as indicated in Table 35.

Table 35 Assembly-ISPM#3 Object Instance Attributes and Services

Assembly-ISPM#3 *Class ID = 143, Instance ID = 01*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
01	Data	A1
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3.10 *Assembly-ISPM#4* — The presentation of the Assembly #4 In-Situ Particle Monitor (Assembly-ISPM#4) object instance attributes and services are as indicated in Table 36.

Table 36 Assembly-ISPM#4 Object Instance Attributes and Services

Assembly-ISPM#4 *Class ID = 144, Instance ID = 01*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
01	Data	A1
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--

9.3.11 *Assembly-ISPM#5* — The presentation of the Assembly #5 In-Situ Particle Monitor (Assembly-ISPM#5) object instance attributes and services are as indicated in Table 37.

Table 37 Assembly-ISPM#5 Object Instance Attributes and Services

Assembly-ISPM#5 *Class ID = 145, Instance ID = 01*		
Attributes		
ID	*Attribute Name*	*SDM Tag*
01	Data	A1
Services		
ID	*Service Name*	*SDM Tag*
--	No additional services defined	--