

# **SEMI E54.9-0706**

## **SPECIFICATION FOR SENSOR/ACTUATOR NETWORK COMMUNICATION FOR MODBUS/TCP OVER TCP/IP**

This standard was technically approved by the global Information & Control Committee. This edition was approved for publication by the global Audits and Reviews Subcommittee on May 16, 2006. It was available at [www.semi.org](http://www.semi.org) in June 2006 and on CD-ROM in July 2006. Originally published February 2000; previously published March 2003.

### **1 Purpose**

1.1 This standard defines a communication specification based on the Modbus/TCP protocol over a Transmission Control Protocol/Internet Protocol (TCP/IP) network 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 tool.

#### *1.2 Background and Motivation*

1.2.1 Modbus/TCP over TCP/IP is a component level network which provides a simple, inexpensive, and fast means of exchanging data among control level industrial devices (e.g., sensors and actuators) and higher level devices such as controllers. Modbus/TCP over TCP/IP provides:

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

1.2.2 Modbus/TCP specifies a communication model and protocol. The Physical, Data Link, and Network Layer definitions are defined by the network in which the Modbus/TCP protocol is embedded such as TCP/IP Ethernet.

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 Modbus/TCP over a TCP/IP network.

### **2 Scope**

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

2.2 This document specifies the utilization of the Modbus/TCP protocol to present externally visible device structure and behavior, specified in the Common Device Model (CDM) and appropriate Specific Device Models (SDMs), for the Modbus/TCP protocol over a TCP/IP 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 Modicon Modbus Protocol Reference Guide and the Open Modbus/TCP Specification. Together, they describe the Modbus/TCP protocol, the externally visible data structures and behaviors of devices utilizing the Modbus/TCP 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 Modbus/TCP over a TCP/IP network and is a companion document to the Open Modbus/TCP Specification; thus a complete specification of this standard necessarily includes the Modbus/TCP and Modbus protocol 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 Modbus/TCP. In order to avoid document consistency problems, information in the Modbus/TCP standard specifications that relate to this standard is not repeated in this document. This document is limited to describing enhancements or limitations to the Modbus/TCP standard specifications that are imposed by this standard.

## 4 Referenced Standards

### 4.1 SEMI Standards

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

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 In-Situ Particle Monitor Device

SEMI E54.11 — Specification Device Model for Endpoint Device

SEMI E54.18 — Specification for Sensor/Actuator Network Specific Device Model for Vacuum Pump Device

### 4.2 IEEE Standard<sup>1</sup>

IEEE 802.3 — Telecommunication and Information Exchange between System Local and Metropolitan Networks Specific Requirement Part 3: Carrier Sense Multiple Access CSMA/CD Method and Physical Layer Specification, 1998

### 4.3 ISO Standard<sup>2</sup>

ISO 7498 — Basic Reference Model for Open Systems Interconnection

### 4.4 Other Standards

IP RFC 791 — Reference for data transmission<sup>3</sup>

PI-MBUS-300 Rev. E — Modicon Modbus Protocol Reference Guide, March 1993<sup>4</sup>

Specification — Open Modbus/TCP Specification Version 1.0 March 29, 1999<sup>4</sup>

**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. Terminology may be reproduced here which is defined in other SEMI standards.

### 5.2 Abbreviations and Acronyms

5.2.1 CDM — Common Device Model

5.2.2 DM — Device Manager (Object)

5.2.3 IP — Internet Protocol

5.2.4 NCS — Network Communication Standard

5.2.5 OSI — Open Systems Interconnect

5.2.6 OSS — Object Services Standard

5.2.7 SAC — Sensor, Actuator, Controller (Object)

1 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; <http://www.ieee.org>

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

3 <http://src.doc.ic.ac.uk/computing/internet/rfc/rfc791.txt>

4 <http://www.modicon.com/openmbus/standards/standards.htm>

### 5.2.8 *SAN* — Sensor/Actuator Network

### 5.2.9 *SDM* — Specific Device Model

### 5.2.10 *TCP* — Transmission Control Protocol

## 5.3 *Device Component Definitions*

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 SEMI E54.1. Table 1 provides a mapping of fundamental terminology of the CDM document into this document and the Modbus/TCP standard specifications.

NOTE 1: 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**

| <i>CDM Term</i>  | <i>NCS Equivalent</i> | <i>Modbus/TCP Equivalent</i> |
|------------------|-----------------------|------------------------------|
| Device           | =                     | =                            |
| Device Model     | =                     | =                            |
| Object           | =, Class, Instance    | =, Class, Instance           |
| Instance         | =                     | =                            |
| Attribute        | =                     | =                            |
| Behavior         | =                     | =                            |
| Service          | =                     | =                            |
| State Diagram    | =                     | =                            |
| Byte             | =                     | =                            |
| Nibble           | =                     | =                            |
| Character String | =                     | =                            |

## 5.4 *Modbus/TCP Specific Definitions*

5.4.1 *class* — a set of objects that 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 as well as additional attributes and services. Refer to SEMI E39 for further definition.

5.4.2 *master/slave* — communication over a Modbus network, which is referred to as “client/server”, that provides exclusive control of data by a “master” or “host” device acting as a “client”. 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 it’s “slaves” or “servers”. Master/Slave communication provides the typical request/response oriented network communications.

5.4.3 *modbus/TCP* — an open protocol established at The University of Michigan’s Electronics Manufacturing Laboratory as a standard means of interconnection for simple field devices. The Modbus/TCP over TCP/IP standard specifies OSI reference model layers 1, 2, 3, 4 and 7 specifically the physical signaling, the media access/data link protocols, internetworking capability, the transport capability of end-to-end transmission of data, and the application layer.

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

## 6 Communication Protocol High Level Structure

6.1 The Modbus/TCP protocol over TCP/IP is loosely based on a five-layer architecture. These layers constitute a collapsed form of the OSI seven layer architecture, mapping into the physical, data link, network, transport, and application layers of the Reference Model. This section has been formatted to be aligned with the Basic Reference Model for OSI. The high-level protocol architecture is shown in Figure 1.

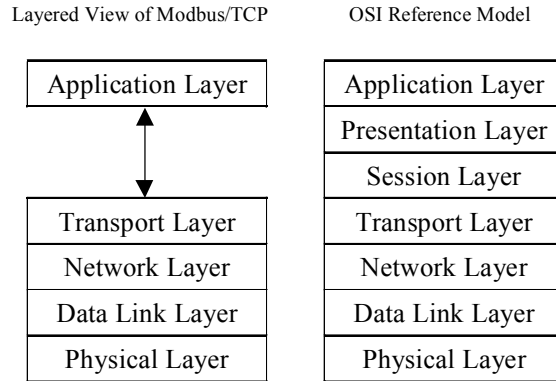


**Figure 1**  
**Layered View of Modbus/TCP Over TCP/IP**

NOTE 2: Figure 1 represents a conceptual view of the communication device 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.

6.1.1 The application layer is specified in the Modicon Modbus Protocol Reference Guide and provides for the definition of Modbus applications as a collection of addressable objects. A subset of these objects may be addressed over the network (as defined by the implementation).

6.1.2 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 Modbus/TCP specification. The recommended and default Modbus/TCP physical layer is IEEE 802.3. If an accepted physical layer specification other than IEEE 802.3 is being used, this must be clearly specified in product literature. 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 of the Modbus/TCP specification. The recommended and default Modbus/TCP data link layer is IEEE 802.3. If an accepted data link layer specification other than IEEE 802.3 is being used, this must be clearly specified in product literature. Data link layer specification includes the media access control mechanism and the logical link control mechanism.

6.4 *Network Layer* — The device shall comply with a network layer specification of the Modbus/TCP specification. This specification is the IP or Internet Protocol as defined in the Modbus/TCP standard specification. The network layer specification includes network routing and internetworking.

6.5 *Transport (Messaging) Layer* — The device shall comply with the Modbus/TCP standard specification for the Transport Layer. This specification is the Transmission Control Protocol as defined in the Modbus/TCP standard specification. The transport layer provides transparent transfer of data between objects in application-entities. Some of the functionality of this layer is implemented in the Application Layer. Specific functions include: object data segmentation/re-assembly (fragmentation) for full message delivery.

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

6.7 *Presentation Layer* — There is no distinct presentation layer. Object addressing and data presentation in Modbus messages are specified as part of the Modbus object definitions and object attribute and service communication protocol. Data byte transmission ordering is defined in IP RFC 791, Appendix B.

6.8 *Application Layer* — The device shall comply with the Modbus/TCP 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 Modbus/TCP standard specifications. In addition, the device shall comply with the object specifications defined in § 7 of this document.

**6.8.1 Object Models** — The Modbus/TCP protocol has been enhanced to provide an object-oriented specification for creating, defining, and addressing objects explicitly, including their attributes and services, and creating, defining, and communicating object attributes in an application dependent format. The device shall comply with the object messaging and object model specifications included in the Modbus/TCP standard specification 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 Modbus/TCP and TCP/IP network management specifications detailed in the Modbus/TCP 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** At this time, the Modbus/TCP standard specifications do not require any specific objects to exist in a Modbus/TCP device in order to be a compliant Modbus/TCP device. The Modbus/TCP standard specifications will be extended to identify and describe objects (i.e., classes) that must exist in devices that are to be interoperable and interchangeable on a Modbus/TCP SEMI compliant SAN network.

**7.1.1** The Common Device Model (CDM) specification identifies two objects (namely the Device Manager (DM) and Sensor Actuator Controller (SAC) objects) that must 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 Modbus/TCP 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. Modbus specific presentation information for these identifiers is given in § 9 of this document.

**Table 2 Required and Optional Object Types**

| Object Name                    | Modbus Class ID/<br>Instance ID <sup>#1</sup> | CDM Tag <sup>#2</sup> | Required by Modbus <sup>#1</sup> | Required by CDM <sup>#2</sup> | Required by NCS |
|--------------------------------|-----------------------------------------------|-----------------------|----------------------------------|-------------------------------|-----------------|
| Device Manager                 | 1/1                                           | DmI0                  | No                               | Yes                           | Yes             |
| Sensor/Actuator/<br>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                                | Ca                    | No                               | No                            | No              |
| Sensor-BI-TH                   | 40/1 through s                                | Sbith                 | No                               | No                            | No              |
| Application Objects            | 129 through x/<br>1 through r                 | <sup>#3</sup>         | No                               | No                            | No              |

<sup>#1</sup> See Modbus specification for further information; values are decimal: “i”, “j”, “k”, “l”, “m”, “n”, “o”, “p”, “q” and “r” 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.

<sup>#2</sup> See CDM specification for further information.

<sup>#3</sup> Application Dependent objects 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 Modbus/TCP specification, is a candidate for a SEMI SDM as well as a Modbus/TCP device definition. Conversely, all SEMI SDM's and Modbus/TCP device definitions specified for operation over a SEMI compliant Modbus/TCP network must be an aggregation of the object types listed in Table 2, and be compliant with both the CDM specification and the Modbus/TCP 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.

NOTE 3: The formats of object attributes and services are detailed in the CDM document; the presentation of object attributes and services to the Modbus/TCP over a TCP/IP network is detailed in the tables contained in the following sub-sections and in the Modbus/TCP 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 Modbus/TCP 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**

| <i>Device Manager Object (DM)</i><br><i>Class ID = 01, Instance ID = 01</i> |                                     |                |
|-----------------------------------------------------------------------------|-------------------------------------|----------------|
| <i>Attributes</i>                                                           |                                     |                |
| <i>ID</i>                                                                   | <i>Attribute Name</i>               | <i>CDM Tag</i> |
| 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           |
| 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          |

| <i>Services</i> |                       |                |
|-----------------|-----------------------|----------------|
| <i>ID</i>       | <i>Service Name</i>   | <i>CDM Tag</i> |
| 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 Modbus/TCP network shall be as indicated in Table 4.

**Table 4 SAC Object Instance Attributes and Services**

| <i>Sensor, Actuator, Controller Object (SAC)</i><br><i>Class ID = 02, Instance ID = 01</i> |                           |                |
|--------------------------------------------------------------------------------------------|---------------------------|----------------|
| <i>Attributes</i>                                                                          |                           |                |
| <i>ID</i>                                                                                  | <i>Attribute Name</i>     | <i>CDM Tag</i> |
| 01                                                                                         | Last Calibration Date     | SacA1          |
| 02                                                                                         | Next Calibration Date     | SacA2          |
| 03                                                                                         | Expiration Timer          | SacA3          |
| 04                                                                                         | Expiration Warning Enable | SacA4          |
| 05                                                                                         | Run Hours                 | SacA5          |
| <i>Services</i>                                                                            |                           |                |
| <i>ID</i>                                                                                  | <i>Service Name</i>       | <i>CDM Tag</i> |
| 01                                                                                         | Reset                     | SacS1          |
| 03                                                                                         | Abort                     | SacS2          |
| 05                                                                                         | Recover                   | SacS3          |
| 07                                                                                         | Get Attribute             | SacA4          |
| 09                                                                                         | Set Attribute             | SacA5          |
| 25                                                                                         | Operate                   | SacA6          |
| 27                                                                                         | Restore Default           | SacA7          |
| 29                                                                                         | Publish Attribute         | SacA8          |

**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 Modbus/TCP network. The presentation of object instance attributes and services shall be as indicated in Table 5.

**Table 5 Assembly Object Instance Attributes and Services**

| <i>ASSEMBLY Object (Asm)</i><br><i>Class ID == 03, Instance ID = 01 through i</i> |                       |                |
|-----------------------------------------------------------------------------------|-----------------------|----------------|
| <i>Attributes</i>                                                                 |                       |                |
| <i>ID</i>                                                                         | <i>Attribute Name</i> | <i>CDM Tag</i> |
| 01                                                                                | Data                  | AsmA1          |
| <i>Services</i>                                                                   |                       |                |
| <i>ID</i>                                                                         | <i>Service Name</i>   | <i>CDM Tag</i> |
| 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**

| <i>Local Link Object (Asm)</i><br><i>Class ID = 04, Instance ID = 01 through j</i> |                              |                |
|------------------------------------------------------------------------------------|------------------------------|----------------|
| <i>Attributes</i>                                                                  |                              |                |
| <i>ID</i>                                                                          | <i>Attribute Name</i>        | <i>CDM Tag</i> |
| 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          |
| <i>Services</i>                                                                    |                              |                |
| <i>ID</i>                                                                          | <i>Service Name</i>          | <i>CDM Tag</i> |
| --                                                                                 | 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**

| <i>Sensor-AI</i><br><i>Class ID = 33, Instance ID = 01 through k</i> |                       |                |
|----------------------------------------------------------------------|-----------------------|----------------|
| <i>Attributes</i>                                                    |                       |                |
| <i>ID</i>                                                            | <i>Attribute Name</i> | <i>CDM Tag</i> |
| 01                                                                   | Name                  | SaiA1          |
| 02                                                                   | Status                | SaiA2          |
| 03                                                                   | Alarm Enable          | SaiA3          |
| 04                                                                   | Warning Enable        | SaiA5          |
| 16                                                                   | Value                 | Sai16          |
| 17                                                                   | ReportInhibitTimer    | Sai17          |

| <i>ID</i>       | <i>Attribute Name</i> | <i>CDM Tag</i> |
|-----------------|-----------------------|----------------|
| 18              | EnableReportRate      | Sai18          |
| 19              | ReportRate            | Sai19          |
| 64              | Offset                | Sai64          |
| 65              | Gain                  | Sai65          |
| 66              | DataType              | Sai66          |
| 67              | DataUnits             | Sai67          |
| 68              | SafeState             | Sai68          |
| 69              | EnableReportDelta     | Sai69          |
| 70              | ReportDelta           | Sai70          |
| 71              | EnableReportROC       | Sai71          |
| 72              | ReportROC             | Sai72          |
| 73              | AlarmTripPointHigh    | Sai73          |
| 74              | AlarmTripPointLow     | Sai74          |
| 75              | AlarmHysteresis       | Sai75          |
| 76              | WarningTripPointHigh  | Sai76          |
| 77              | WarningTripPointLow   | Sai77          |
| 78              | WarningHysteresis     | Sai78          |
| <i>Services</i> |                       |                |
| <i>ID</i>       | <i>Service Name</i>   | <i>SDM Tag</i> |
| 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**

| <i>Sensor-EI</i><br><i>Class ID = 34, Instance = 01 through 1</i> |                       |                |
|-------------------------------------------------------------------|-----------------------|----------------|
| <i>Attributes</i>                                                 |                       |                |
| <i>ID</i>                                                         | <i>Attribute Name</i> | <i>CDM Tag</i> |
| 01                                                                | Name                  | SeiA1          |
| 02                                                                | Status                | SeiA2          |
| 03                                                                | Alarm Enable          | SeiA3          |
| 04                                                                | Warning Enable        | SeiA5          |
| 16                                                                | Value                 | Sei16          |
| 17                                                                | ReportInhibitTimer    | Sei17          |
| 18                                                                | EnableReportRate      | Sei18          |
| 19                                                                | ReportRate            | Sei19          |
| 64                                                                | DebounceControl       | Sei64          |

| <i>ID</i>       | <i>Attribute Name</i> | <i>CDM Tag</i> |
|-----------------|-----------------------|----------------|
| 65              | AlarmStatus           | Sei65          |
| 66              | WarningStatus         | Sei66          |
| <i>Services</i> |                       |                |
| <i>ID</i>       | <i>Service Name</i>   | <i>SDM Tag</i> |
| 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**

| <i>Sensor-BI</i><br><i>Class ID = 35, Instance ID = 01 through m</i> |                       |                |
|----------------------------------------------------------------------|-----------------------|----------------|
| <i>Attributes</i>                                                    |                       |                |
| <i>ID</i>                                                            | <i>Attribute Name</i> | <i>CDM Tag</i> |
| 01                                                                   | Name                  | SbiA1          |
| 02                                                                   | Status                | SbiA2          |
| 03                                                                   | Alarm Enable          | SbiA3          |
| 04                                                                   | Warning Enable        | SbiA5          |
| 16                                                                   | Value                 | Sbi16          |
| 17                                                                   | ReportInhibitTimer    | Sbi17          |
| 18                                                                   | EnableReportRate      | Sbi18          |
| 19                                                                   | ReportRate            | Sbi19          |
| 64                                                                   | DebounceControl       | Sbi64          |
| 65                                                                   | AlarmStatus           | Sbi65          |
| 66                                                                   | WarningStatus         | Sbi66          |
| <i>Services</i>                                                      |                       |                |
| <i>ID</i>                                                            | <i>Service Name</i>   | <i>SDM Tag</i> |
| 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.