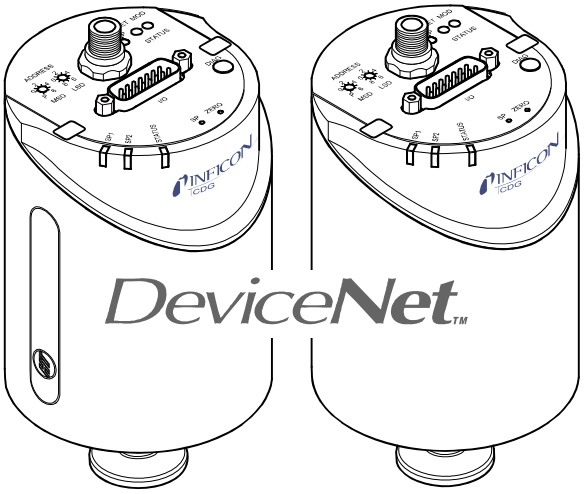
DeviceNet™

for Capacitance Diaphragm Gauge

CDG045D, CDG100D, CDG160D, CDG200D



General Information

Caution

Caution: data transmission errors

Any attempt to simultaneously operate the gauge via the RS232C Serial Interface and a Fieldbus interface (DeviceNet or Profibus) or the diagnostic port may result in incorrect data and data transmission errors.

Therefore, it is inadmissible to simultaneously operate the gauge via the RS232C Serial Interface and DeviceNet, Profibus, or the diagnostic port.

Intended Use

This Communication Protocol contains instructions for operating DeviceNet interfaces (slaves) together with a master.

This manual describes the functionality of DeviceNet for programming purposes. For more information → 🕮 "DeviceNet Specifications" of the Open DeviceNet Vendor Association (ODVA).

For safety information, specifications and operation instructions of the vacuum gauges refer to the appropriate documents (→ 🕮 [1], [2], [3]).

DeviceNet–Interface

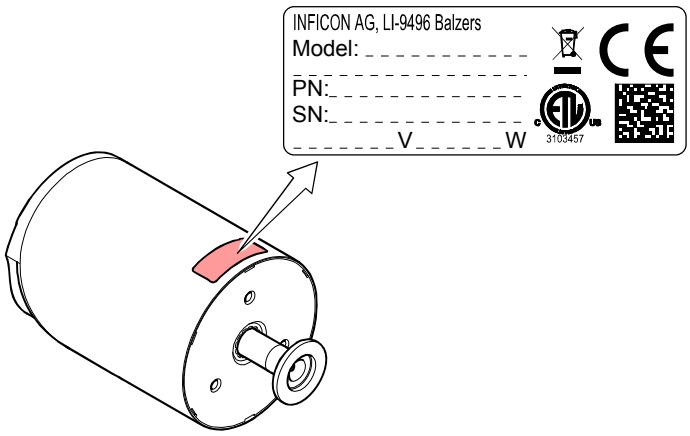
The following description of the DeviceNet interface is compliant to the DeviceNet specification of the Open DeviceNet Vendor Association and to the 🕮 "Interface Guidelines for DeviceNet Devices On Semiconductor Manufacturing Tools".

This manual describes the functionality of a DeviceNet group 2 only slave and supports explicit messaging and the I/O connections change of state/cyclic and polling.

We reserve the right to make technical changes without prior notice.

Product Identification

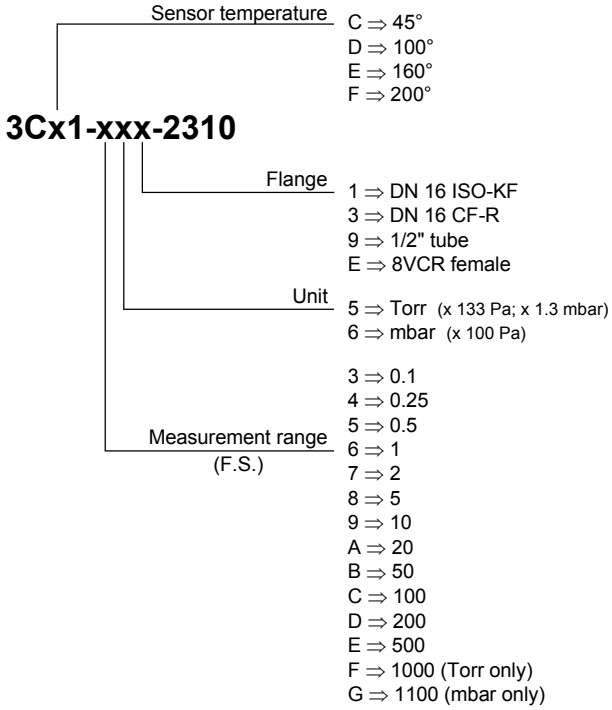
In all communications with INFICON, please specify the information on the product nameplate. For convenient reference copy that information into the space provided below.



Validity

This document applies to products of the temperature controlled CDG045D, CDG100D, CDG160D and CDG200D series with DeviceNet interface.

Part numbers of standard products are indicated below. OEM products have other part numbers and different parameter settings (e.g. factory setting of setpoint) as defined in the corresponding ordering information.



The part number (PN) can be taken from the product nameplate.

If not indicated otherwise in the legends, the illustrations in this document correspond to CDG045D gauges with the DN 16 ISO-KF vacuum connection. They apply to other vacuum connections and to the other gauges by analogy.

We reserve the right to make technical changes without prior notice.

Trademark

DeviceNet™ Open DeviceNet Vendor Association Inc.

Patents

EP 1070239, 1040333

US Patents 6528008, 6591687, 7107855, 7140085

Contents

For cross-references to other documents, the symbol (→ 🕮 [XY]) is used.

1 Technical Data

Further technical data → 🕮 [1], [2], [3].

Supply voltage

CDG045D

Supply voltage at the DeviceNet connector	24 V nominal (11 … 25 V)
Power consumption	
while being heated	≤12 W
at operating temperature	≤10 W

CDG100D, CDG160D, CDG200D

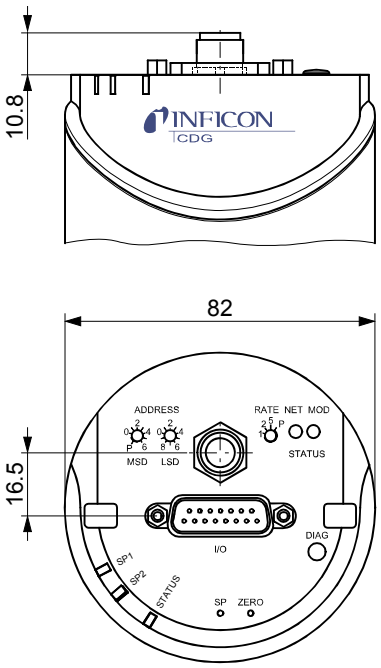
The temperature controlled CDG100D, CDG160D and CDG200D gauges with DeviceNet interface require an additional, separate power supply.

Supply voltage for gauge at D-sub connector	→ 🕮 [2], [3]
Supply voltage for DeviceNet transceiver at micro style connector	24 V nominal (11 … 25 V)
Power consumption	<2 W

DeviceNet interface

Fieldbus name	DeviceNet
Standard applied	→ 🕮 [5]
Communication protocol, data format	→ 🕮 [5]
Interface, physical	CAN bus
Data rate (adjustable via "RATE" switch)	125 kBaud 250 kBaud 500 kBaud (default) "P" (125 kBaud, 250 kBaud, 500 kBaud programmable via DeviceNet
Node address (MAC ID) (Adjustable via "ADDRESS", "MSD", "LSD" switches)	0 … 63_{dec} (default = 63_{dec}) "P" (0 … 63 programmable via DeviceNet)
DeviceNet connector	Micro style, 5 pins, male
Cable	Shielded, special DeviceNet cable, 5 conductors (→ 🗎 7 and 🕮 [4])
Cable length, system wiring	according to DeviceNet specifications (→ 🕮 [4], [5])

Dimensions [mm]



2 Interface Connection

Making a DeviceNet interface cable

For operating the temperature controlled CDGxxxD gauge via DeviceNet, an interface cable conforming to the DeviceNet standard is required.

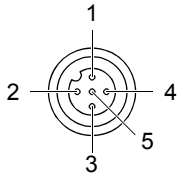
If no such cable is available, make one according to the following indications.

Cable type

A shielded special 5 conductor cable conforming to the DeviceNet standard has to be used (→ 🕮 [4] and [5]).

Procedure

❶ Make the DeviceNet interface cable according to the following indications:

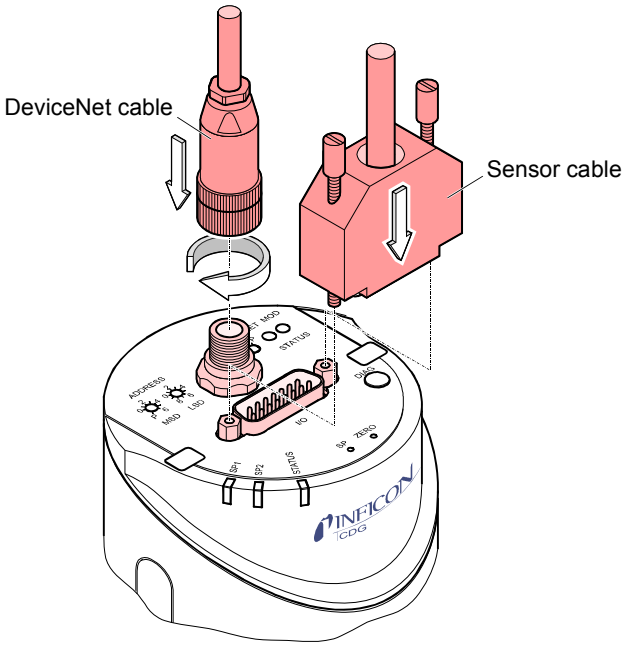


Micro style, 5 pins,
(DeviceNet)
female, soldering side

Pin 1	Drain	
Pin 2	Supply	+24 VDC
Pin 3	Supply common	GND
Pin 4	CAN_H	
Pin 5	CAN_L	

Pin assignment of the D-Sub 15-pins sensor connector according to the respective operating manual (→ 🕮 [1], [2], [3]).

❷ Plug the DeviceNet (and sensor) cable connector into the gauge.



❸ Lock the DeviceNet (and sensor) cable connector.

3 Operation

3.1 Introduction

Via the DeviceNet interface, the following and further data are exchanged in the standardized DeviceNet protocol:

- Pressure reading
- Pressure unit (Torr, mbar, Pa)
- Zero adjustment
- Status and error messages
- Status of the switching functions

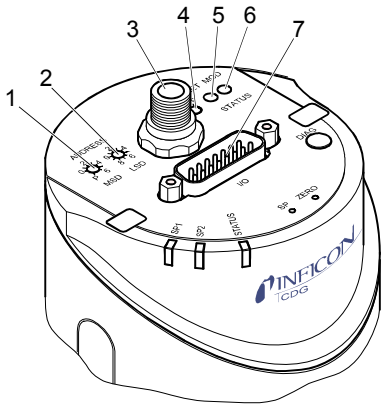
Caution

Caution: data transmission errors

Any attempt to simultaneously operate the gauge via the RS232C Serial Interface and a Fieldbus interface (DeviceNet or Profibus) or the diagnostic port may result in incorrect data and data transmission errors.

Therefore, it is inadmissible to simultaneously operate the gauge via the RS232C Serial Interface and DeviceNet, Profibus, or the diagnostic port.

3.2 Front View



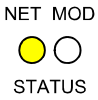
Position	Function
1	Address Switch × 10, decimal
2	Address Switch × 1, decimal
3	DeviceNet connector
4	Data Rate Switch
5	Network Status LED
6	Module Status LED
7	"Sensor cable" connector (Power, analog I/O, RS232C I/O and Relay contacts)

3.3 Indicators and Switches

3.3.1 Module Status LED

Device State	LED State	Description
Power Off	Dark	No power applied to device
Device Self-Test	Flashing green/red	Device is in self-test
Device Operational	Green	Device is operating normally.
Unrecoverable Fault	Red	Device has detected an unrecoverable fault
Recoverable Fault	Flashing red	Device has detected a recoverable fault, e.g. missing DeviceNet power supply

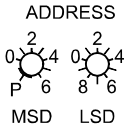
3.3.2 Network Status LED



The network status LED indicates the status of the DeviceNet communication link.

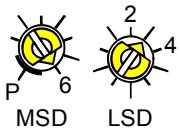
State	LED state	To indicate
Not Powered/not online	Dark	Device is not online. • The device has not completed the Duplicate MAC_ID test yet. • The device may not be powered, look at Module Status LED.
Online, not connected	Flashing green	Device is online but has no connections in the established state. • The device has passed the Dup_MAC_ID test, is online, but has no established connections to other nodes. • The device is not allocated to a master.
Link OK online, connected	Green	The device is online and has connections in the established state. • The device is allocated to a master.
Connection time out	Flashing red	One or more I/O Connections are in the timed out state.
Critical link failure	Red	Failed communication device. The device has detected an error that has rendered it incapable of communicating on the network (Duplicate MAC ID, or Bus–off).

3.3.3 Node Address Switch



During device initialization, the node address switches are read by the device firmware. If the switches specify a valid DeviceNet MAC ID, i.e. a value from 0 … 63, this value is used as the device MAC ID. If the specified MAC ID differs from the value stored in the device's non-volatile RAM (NVRAM), the new MAC ID will be saved in NVRAM. If the switches specify an invalid DeviceNet MAC ID, i.e. a value greater than 63, the current value stored in the device's NVRAM is used as the device MAC ID.

Example: Value of the DeviceNet MAC ID = 63:

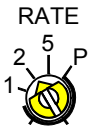


3.3.4 Data Rate Switch

A single, rotary switch is provided for configuring the data rate of the device. The switch has three valid positions for the DeviceNet data rates, 125, 250, and 500 kBaud, where:

1 ≙ 125 kBaud
2 ≙ 250 kBaud
5 ≙ 500 kBaud

Example: Data rate = 250 kBaud:



All remaining switch positions specify software selection of the data rate.

3.4 Abbreviations

Abbr.	Meaning
NV	Nonvolatile; attribute value is maintained through power cycles
V	Volatile
USINT	Unsigned short integer value (Range 0 … 255)
INT	Integer value (Range –32767 … 32768)
UINT	Unsigned integer value (Range 0 … 65635)
UDINT	Unsigned double integer value (Range 0 … 4'294'967'295)
FLOAT	Floating point value (Range according to IEEE 754)
XX_h	Hexadecimal number (Radix = 16)
XX_d	Decimal number (Radix = 10)
XX_b	Binary number (Radix = 2)

4 Object Structure

4.1 Connection Object

Class Code $05_h = 05_d$

The connection class allocates and manages the internal resources associated with both I/O and explicit messaging connections. The specific instance generated by the connection class is referred to as a connection instance or a connection object.

The following Instances of the connection object are supported:

- Instance 1: Explicit messaging
- Instance 2: I/O polling
- Instance 4: I/O change of state/cyclic I/O connection

Please refer to the DeviceNet specification for further information.

4.1.1 Vendor-Specific Object Extension on Instance 2 Poll Connection

For the selection of the active input assembly the following vendor specific attribute can be used.

Vendor specific extension:

Attribute ID	Access rule	NV/ V	Name	DeviceNet Data type	Description of attribute	Semantics
100	Get/Set	NV	Poll produce assembly instance	USINT	Contains the instance number of the assembly used by this connection to send data	1, 2 (default), 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14

This vendor-specific attribute facilitates the configuration of the data assembly, the temperature controlled CDGxxxD will send to the DeviceNet master as poll response. It offers the possibility to select a (predefined) data assembly via a configuration tool + EDS file (→ 🕮 [6]). Attribute 100 allows the user to configure the poll IO data assembly via EDS even if the poll connection of the temperature controlled CDGxxxD is in the established state. The assembly number is stored non volatile. Modifications of the poll produce assembly instance will take effect **only** after a reset of the temperature controlled CDGxxxD.

Thus, definition of the temperature controlled CDGxxxD data assembly can be done in various ways:

1) Standard method (without using attribute 100):
   The DeviceNet master configures the "Produced Connection Path" attribute of the poll connection (connection instance 2) when establishing communication to the temperature controlled CDGxxxD. This requires a master to support expanded explicit messaging capabilities.
2) Directly setting attribute 100 e.g. by a configuration tool
   (e.g. RS networks) + device reset

4.2 Identity Object

Class Code $01_h = 01_d$

4.2.1 Class Attributes

This object provides identification of and general information about the device.

Attribute ID	Access rule	Name	DeviceNet data type	Description of attribute	Semantics of values
1	Get	Revision	UINT	Revision of this object. Note: All class definitions are required to include this class attribute.	The current value assigned to this attribute is one (01). If updates that require an increase in this value are made, then the value of this attribute increases by one.
2	Get	Max instance	UINT	Maximum instance number of an object currently created in this class level of the device.	The largest Instance number of a created object at this class hierarchy level.

4.2.2 Instance Attributes

Attribute ID	Access rule	NV/ V	Name	DeviceNet data type	Description of attribute
1	Get	NV	Vendor ID	UINT	Manufacturer identification Value: 633 (INFICON AG) (default)
2	Get	NV	Device type	UINT	$1C_h$ = Vacuum pressure gauge
3	Get	NV	Product code	UINT	14 (default)
4	Get	NV	Revision	STRUCT	
5	Get	NV	Status	WORD	
6	Get	NV	Serial number	UDINT	
7	Get	NV	Product name	SHORT STRING	CDG045D, CDG100D, CDG160D, CDG200D

The products with different product names only differ in the temperature at which the sensor element is heated up. Different fullscale values of the gauges exist within the same product name. The product code is always 14.

Services

Service Code	Name	Description
5 (05_h)	Reset	0 = last installation, 1 = default installation
14 ($0E_h$)	Get_Attribute_Single	

4.3 S-Device Supervisor Object

Class Code $30_h = 48_d$

4.3.1 Class Attributes

Attribute ID	Access rule	Name	DeviceNet data type	Description of attribute	Semantics of values
1	Get	Revision	UINT	Revision of this object	The current value assigned to this attribute is one (01). If updates that require an increase in this value are made, then the value of this attribute increases by one.
2	Get	Max instance	UINT	Maximum instance number of an object currently created in this class level of the device.	The largest Instance number of a created object at this class hierarchy level.

4.3.2 Instance Attributes

Attribute ID	Access rule	NV/ V	Name	DeviceNet data type	Description of attribute
3	Get	NV	Devicetype	SHORT STRING	ASCII Text, "VG" (vacuum gauge)
4	Get	NV	SEMI standard revision level	SHORT STRING	Specifies the revision level of the SEMI S/A network standard to which the device complies (for example "E54-0997").
5	Get	NV	Manufacturer's name	SHORT STRING	"INFICON AG" (default)
6	Get	NV	Manufacturer's model number	SHORT STRING	ASCII text, manufacturer specified
7	Get	NV	Software revision level	SHORT STRING	ASCII text, for example "010000"
8	Get	NV	Hardware revision level	SHORT STRING	ASCII text, for example "010000"
9	Get	NV	Manufacturer's serial number	SHORT STRING	ASCII text
10	Get	NV	Device configuration	SHORT STRING	Depending on device: Examples: 1000 Torr, 100 C 0.1 Torr, 200 C
11	Get	V	Device status	USINT	→ "Semantics" section below.
12	Get	V	Exception status	BYTE	→ "Semantics" section below.
13	Get	V	**Exception detail alarm**	STRUCT of:	A structure of three structures containing a bit mapped representation of the alarm detail
			Common exception detail	STRUCT of:	
			Size 2	USINT	Number of common detail bytes
			Detail 0	ARRAY of:	→ "Semantics" section below.
			Detail 1	BYTE	→ "Semantics" section below.
			Device exception detail	STRUCT of:	
			Size 2	USINT	Number of device detail bytes
			Detail 0		→ "Semantics" section below.
			Detail 1		→ "Semantics" section below.
			Manufacturer exception detail	STRUCT of:	
			Size 1	USINT	Number of manufacturer detail bytes
			Detail 0	ARRAY of:	Manufacturer Specified

Attribute ID	Access rule	NV/ V	Name	DeviceNet data type	Description of attribute
14	Get	V	**Exception detail warning**	STRUCT of:	A structure of three structures containing a bit mapped representation of the warning detail
			Common exception detail	STRUCT of:	
			Size 2	USINT	Number of common detail bytes
			Detail 0	ARRAY of:	→ "Semantics" section below.
			Detail 1	BYTE	→ "Semantics" section below.
			Device exception detail	STRUCT of:	
			Size 3	USINT	Number of device detail bytes
			Detail 0	BYTE	→ "Semantics" section below.
			Detail 1	BYTE	→ "Semantics" section below.
			Detail 2	BYTE	→ "Semantics" section below.
			Manufacturer exception detail	STRUCT of:	
			Size 1	USINT	Number of manufacturer detail bytes = 1
			Detail 0	ARRAY of:	Manufacturer specified
15	Set	NV	Alarm enable	BOOL	→ "Semantics" section below.
16	Set	NV	Warning enable	BOOL	→ "Semantics" section below.
101	Get	NV	Software revision base unit	SHORT STRING	ASCII text, for example "010000"
102	Get	V	Run hours	UINT	1 = 1 hour

4.3.3 Semantics

Device status

This attribute represents the current state of the device. Its value changes as the state of the device changes. The following values are defined:

Attribute value	State
0	Undefined
1	Self testing
2	Idle
3	Self-test exception
4	Executing
5	Abort
6	Critical fault

In case of a self-test exception restart the device with a reset out of the box (identity object class 1, instance 1, service 5, service target value: 1).

Exception status

A single byte attribute whose value indicates the status of the alarms and warnings for the device. The device supports the expanded mode.

For the *expanded mode*, bit seven of exception status attribute is set to one; exceptions are reported through the communication of this exception status attribute, formatted as specified in the table below. In addition, the exception detail attributes are supported. The exception status bits are determined by a logical "OR" of the related exception detail bits, as indicated.

Exception status bitmap

Bit	Function
0	ALARM / device-common [1)]
1	ALARM / device-specific
2	ALARM / manufacturer-specific
3	reserved, set to 0
4	WARNING / device-common [1)]
5	WARNING / device-specific
6	WARNING / manufacturer-specific
7	1 ≙ Expanded Method

[1)] The alarm or warning is not specific to the device type or device type manufacturer.

Exception detail alarm and exception detail warning

The formats of these two attributes are identical. Therefore, they are described together:

Attributes that relate the detailed status of the alarms or warnings associated with the device. Each attribute is a structure containing three members; these three members, respectively relate the detailed status of exceptions that are common (i.e., not device-specific), device-specific but not manufacturer-specific, and manufacturer-specific. The common and device-specific detail are defined below. A manufacturer-specific detail has a length of 1 byte. A SIZE value of one indicates that one byte detail is defined for the associated exception detail structure.

Each of the three structure members is defined as a structure containing an ordered list (i.e., array) of bytes of length SIZE, and an unsigned integer whose value is SIZE. Each of the bytes in each array has a specific mapping. This mapping is formatted as 8 bits representing 8 independent conditions, whereas a value of 1 indicates that the condition is set (or present), and a value of 0 indicates that the condition is cleared (or not present). Note that if a device does not support an exception detail, the corresponding bit is never set. The bitmaps for alarms and warnings in the corresponding attributes are structured in parallel so that a condition may have either alarm or warning set depending on severity. If a condition inherently cannot be both alarm and warning, then the parallel bit position corresponding to the other state will remain "0".

The existence of an exception detail variable structure is dependent on the value of the Exception Status Attribute; the existence of an exception detail variable structure is only required if bit seven of the Exception Status attribute is set to 1 (indicating Expanded method reporting) and the bit (among bits zero through six) of the Exception Status attribute corresponding to the particular exception type is also set to 1.

Common exception detail

This structure relates exception conditions (i.e. alarms or warnings) which are common to all devices within the *Hierarchy of Semiconductor Equipment Devices*. The Detail element of the structure is an ordered list (i.e., array) of bytes of length [SIZE] which is the value of the structure element Size. For each byte in the Detail field, all bits which are not identified are reserved for future standardization.

The first byte in this attribute is CommonExceptionDetail[0]. Additional exception details, if provided, are named CommonExceptionDetail[1], … CommonExceptionDetail[SIZE]. The specific exception associated with each of the bitmaps is given in the table below. The SIZE for this revision is two (2).

Device exception detail

This structure, similar in form to common exception detail, relates exception conditions which are specific to individual devices on the network and are defined in the following. The detail element of the structure is an ordered list (i.e. array) of bytes of length [size = 2] which is the value of the structure element size.

Manufacturer exception detail

This structure, similar in form to Common Exception Detail, relates exception conditions which are specific to the manufacturers of individual devices on the network. There is one byte manufacturer exception details defined. The Detail element of the structure is an ordered list (i.e. array) of bytes of length [size = 1] which is the value of the structure element size.

Exception detail alarm

Data component	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Common exception detail alarm size	0	0	0	0	0	0	1	0
Common exception detail alarm 0	0	0	0	Data memory	Nonvolatile memory	Code memory	0	0
Common exception detail alarm 1	0	0	0	0	0	0	0	0
Device exception detail alarm size	0	0	0	0	0	0	1	0
Device exception detail alarm 0	S-Analog sensor instance 1, sensor alarm byte 0							
Device exception detail alarm 1	S-Analog sensor instance 1, sensor alarm byte 1							
Manufacturer exception detail alarm size	0	0	0	0	0	0	0	1
Manufacturer exception detail alarm 0	0	0	0	0	0	0	0	SerCom Alarm

Exception detail warning

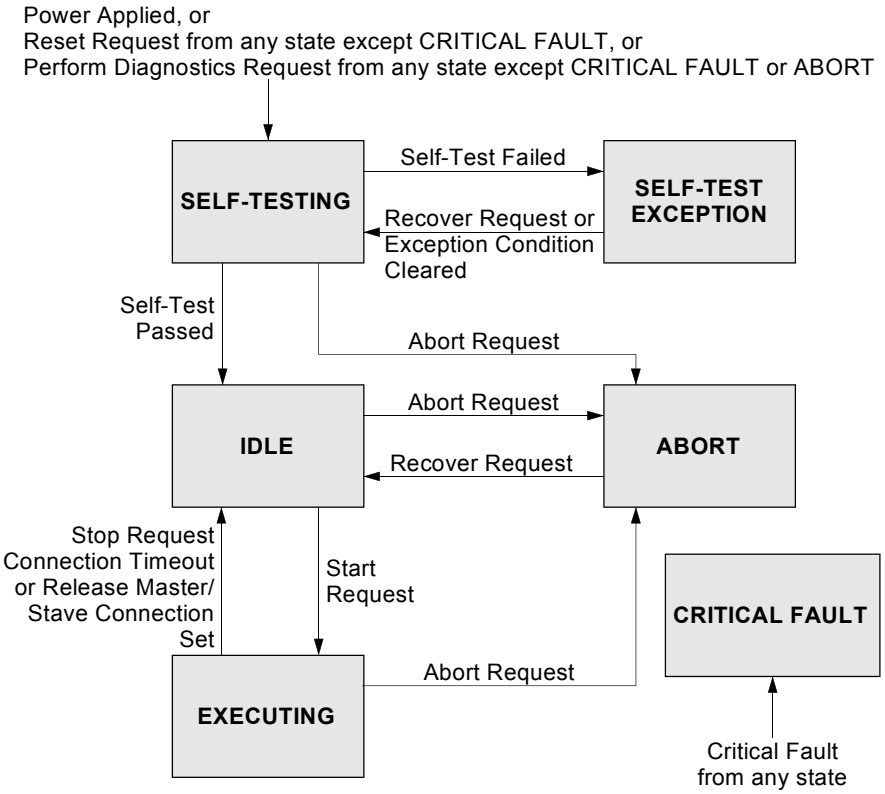
Data component	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Common exception detail warning size	0	0	0	0	0	0	1	0
Common exception detail warning 0	0	0	0	Data memory	Nonvolatile memory	Code memory	0	0
Common exception detail warning 1	0	0	0	0	0	0	0	0
Device exception detail warning size	0	0	0	0	0	0	1	1
Device exception detail warning 0	S-Analog sensor instance 1, status extension							
Device exception detail warning 1	S-Analog sensor instance 1, sensor warning byte 0							
Device exception detail warning 2	S-Analog sensor instance1, sensor warning byte 1							
Manufacturer exception detail warning size	0	0	0	0	0	0	0	1
Manufacturer exception detail warning 0	0	0	0	0	0	0	0	SerCom Warning

Alarm enable and warning enable

These boolean attributes are used to enable (1) or disable (0) the S-Device supervisor object's process of setting exception bits. When disabled, corresponding bits are never set; and, if they were set, disabling clears them. Also, alarm and warning states are not retained; when enabled, bits will be set only if the corresponding condition is true.

The default state for these enable attributes is enabled (1).

4.3.4 S-Device Supervisor Object States



Abort - Used to transition the device application objects to the aborted state. This service request may be (and generally will) originated internally, from application objects.

Recover - Used to transition the device application objects, out of the abort state, to the idle state. This service request may be originated internally, from application objects.

Perform_Diagnostics - Used to instruct the DS object to perform a diagnostic test.

4.3.5 S-Device Supervisor Common Services

Service Code	Service Name	Description of Service
$0E_h$	Get_Attributes_Single	Returns the contents of the specified attribute.
10_h	Set_Attributes_Single	Modifies an attribute value.
05_h	Reset	Resets the device to the **Self-Testing** state.
06_h	Start	Starts the device execution.
07_h	Stop	Moves the device to the **Idle** state.

The device transitions from the IDLE state to the EXECUTING state by a START Request (Service Code 06_h) or by the receipt of the first valid I/O data.

You will not get any valid measurement values from the device until this service has been requested or the I/O-poll message has been received.

4.3.6 S-Device Supervisor Object Specific Services

Service Code	Service Name	Description of Service
$4B_h$	Abort	Moves the device to the **Abort** state
$4C_h$	Recover	Moves the device out of the **Abort** state
$4E_h$	Perform_Diagnostics	Causes the device to perform a set of diagnostic routines

Explanation to Service code $4E_h$

- DS Object Service Parameter dictionary

Parameter	Form	Description
TestID	USINT	Type and possibly detail of diagnostic test to be performed

- TestID parameter

  The following values are defined for the TestID parameter for the Perform_Diagnostics Service Request:

Attribute Value	State
0	Standard

  Type "Standard" is specified if there is only one type of diagnostic defined or if there are more than one including a type standard.

4.4 S-Analog Sensor Object

Class Code $31_h = 49_d$

4.4.1 Class Attributes

Attribute ID	Access rule	Name	DeviceNet data type	Description of attribute	Semantics of values
1	Get	Revision	UINT	Revision of this object Note: All class definitions are required to include this class attribute.	The current value assigned to this attribute is one (01).
2	Get	Max instance	UINT	Maximum instance number of an object currently created in this class level of the device.	The largest Instance number of a created object at this class hierarchy level.
94	Get	Active value	INT or specified by data type	Is used by assemblies to produce this class-level attribute, instead of the value (attribute ID 6) of the S-Analog sensor instances.	
95	Get	Active instance number	UINT	Identifies the object instance that is providing the value which is copied into the active value for all input assemblies and the alarm/warning exception details for the S-Device supervisor object. → Behavior section.	Default = 1
96	Get	Number of gauges	USINT	Identifies the number of gauge instances present in the device.	
99	Get	Subclass	UINT	Identifies a subset of additional class attributes, services and behaviors.	1 ≙ Instance selector

4.4.2 Semantics

Active value

Assemblies or connections may produce this class-level attribute, instead of the value (attribute ID 6) of the active S-Analog sensor instance. The S-Analog sensor class-level attribute active instance number identifies the object instance that is currently active and providing the value to the active value class-level attribute which is, in turn, produced by the input assemblies that have active value as a member.

Active instance number

The device internally modifies this attribute, as required, to identify the S-Analog sensor object instance providing the value member which is copied into the active value for all input assemblies and the alarm/warning exception details for the S-Device supervisor object.

→ Behavior for more information on the mechanism.

Number of gauges

This attribute is used to determine the size of all input assemblies within a node.

→ respective device profile for its usage within a device type.

4.4.3 Instance Attributes

Two S-Analog sensor instances (instance 1 and instance 2) are available. Instance 1 represents the physical sensor reading of the capacitance diaphragm gauge.

Instance 2 represents the physical sensor reading of the atmosphere pressure sensor.

4.4.3.1 Instance Attributes of Instance 1

Following is the instance 1 with the subclass extension of the capacitance manometer (diaphragm gauge) part of the gauge. Many different types of gauges are available. The behavior of all gauges is the same, but the fullscale value is different.

Attribute ID	Access rule	NV/ V	Name	DeviceNet data type	Description of attribute	Semantics of values
3	Set /Cond. → below	NV	Data type	USINT	Determines the data type of *value* and all related attributes as specified in this table.	→ "Semantics" section below. Int ≙ $C3_h$ [default] float ≙ CA_h
4	Set → below	NV	Data units	UINT	Determines the units context of *value* and all related attributes.	→ "Semantics" section below. Supported Values: Counts ≙ 1001_h [default] mbar ≙ 1308_h Torr ≙ 1301_h Pascal ≙ 1309_h
5	Get	V	Reading valid	BOOL	Indicates that the *value* attribute contains a valid value.	0 = invalid 1 = valid (invalid: e.g., not warmed up yet)
6	Get	V	Value	INT or specified by *data type*	Analog input value	The corrected, converted, calibrated final value of the sensor. → "Semantics" section below.
7	Get	V	Status	BYTE	Alarm and warning state of this object instance.	→ "Semantics" section below.
8	Set	NV	Alarm enable	BOOL	Enables the setting of the alarm status bits	0 = disable [default] 1 = enable
9	Set	NV	Warning enable	BOOL	Enables the setting of the warning status bits	0 = disable [default] 1 = enable
10	Get	NV	Full Scale	INT or specified by *data type*	The *value* of full scale for the sensor.	The value of attribute *value* corresponding to the full scale calibrated measurement of the sensor. [default] = maximum allowable value for the *data type*
11	Get	NV	Offset-A data type	USINT	Determines the data type of attribute *offset-A*	[default] = INT ($C3_h$)
12	Set	NV	Offset-A	Specified by attribute 11	An amount added prior to *gain* to derive *value*	This value could be read at the measurement device itself after execution of zero adjust service. 0 = [default]
17	Set	NV	Alarm trip point high		Determines the value above which an alarm condition will occur	[default] = Maximum value for its data type.
18	Set	NV	Alarm trip point low	INT or specified by *data type*	Determines the value below which an alarm condition will occur	[default] = Minimum value for its data type.
19	Set	NV	Alarm hysteresis	INT or specified by *data type*	Determines the amount by which the *value* must recover to clear an alarm condition	[default] = 0

Attribute ID	Access rule	NV/ V	Name	DeviceNet data type	Description of attribute	Semantics of values
21	Set	NV	Warning trip point high	INT or specified by *data type*	Determines the *value* above which a warning condition will occur	[default] = Maximum value for its data type.
22	Set	NV	Warning trip point low	INT or specified by *data type*	Determines the *value* below which a warning condition will occur	[default] = Minimum value for its data type.
23	Set	NV	Warning hysteresis	NT or specified by *data type*	Determines the amount by which the *value* must recover to clear a warning condition	[default] = 0
25	Set	NV	Safe state	USINT	Specifies the behavior for the *value* for states other than execute	→ "Semantics" section below. [default] = 0
26	Set	NV	Safe value	INT or specified by *data type*	The value to be used for safe state = safe value	→ "Semantics" section below. [default] = 0
32	Get	NV	Overrange	INT or specified by *data type*	Specifies the highest valid *value*	The value above which attribute *reading valid* is set to invalid. [default] = maximum allowable value for the *data type*
33	Get	NV	Underrange	INT or specified by *data type*	Specifies the lowest valid *value*	The value below which attribute *reading valid* is set to invalid. [default] = minimum allowable value for the *data type*
93	Get	V	Sensor temperature	Real	The temperature in Celsius at which the sensor has warmed up	
94	Get	V	Sensor warning	Struct of byte	Bit definitions of sensor warnings	0 = [default] → "Semantics" section below
95	Get	V	Sensor alarm	Struct of byte	Bit definitions of sensor alarms	0 = [default] → "Semantics" section below
96	Get	V	Status extension	BYTE	Bit-mapped byte providing additional status bits of an S-Analog sensor instance.	0 = [default] → "Semantics" section below
99	Get	NV	Subclass	UINT	Identifies a subset of additional instance attributes, services and behaviors.	3 = Capacitance Manometer
100	Set	NV	Offset vendor specific	INT or specified by *data type*	Customer specified offset which will be added to the offset-A value from zero adjust service	
101	Set	NV	Filter	USINT		0 = Dynamic [default] 1 = Fast 2 = Slow

4.4.3.2 Semantics

Data type

All Data Type attributes use the enumerated values **integer or float**.

The Data Type value will be set automatically based upon the first valid I/O connection established by the device.

If no established I/O connections exist, which include an attribute from this object, then the *Data Type* attribute is settable provided that the object is in the *Idle State*.

Note: Using data type integer in combination with a pressure unit (mbar, Torr or Pa) will obviously not produce reasonable values below 1.

Data unit

The Data Unit is only settable in the IDLE state.

Value

An S-Analog sensor object instance derives a reading from a physical analog sensor. The reading is converted to the data type and units specified for the *value* attribute.

Status

A bit mapped byte which indicates the Alarm and Warning Exception status of the object instance. The following definition applies:

Bit	Definition
0	High Alarm Exception: 0 = cleared 1 = set
1	Low Alarm Exception: 0 = cleared 1 = set
2	High Warning Exception: 0 = cleared 1 = set
3	Low Warning Exception: 0 = cleared 1 = set
4	Reserved
5	Reserved
6	Reserved
7	Reserved

Safe state

This attribute specifies what value will be held in *value* for states other than executing. The purpose of this mechanism is to allow other devices, who may be using this *value* to transition to (or remain in) a safe state in the event of this device transitioning to a FAULT, IDLE, or ABORT state. The following values are defined:

Attribute value	State
0	Zero
1	Full scale
2	Hold last value
3	Use safe value

Safe value

For safe state set to use safe value, this attribute holds the value to which the *value* attribute will be set for object instance states other than executing.

Sensor alarm

16 Bits are used as sensor faults. Sensor alarm byte 0 is mapped to the device exception detail alarm 0, sensor alarm byte 1 is mapped to the device exception detail alarm 1.

Data component	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Sensor alarm byte 0	0	0	0	0	0	0	0	0
Sensor alarm byte 1	High alarm exception	Low alarm exception	0	0	0	0	Electronics failure	0

Sensor warning

16 Bits are used as sensor warnings. Sensor warning byte 0 is mapped to the device exception detail warning 1, sensor warning byte 1 is mapped to the device exception detail warning 2.

Data component	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Sensor warning byte 0	0	0	0	0	0	0	0	Not at temperature
Sensor warning byte 1	High warning exception	Low warning exception	0	0	0	0	Electronics warning	0

Status extension

8 Bits providing the current sensor alarm state of the instance. They are mapped to device exception detail warning 0.

Data component	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Sensor alarm byte 0	0	0	0	0	0	Underrange exceeded	Overrange exceeded	Reading invalid [1)]

[1)] Logical inversion of reading valid

4.4.3.3 Common Services

The S-Analog Sensor Object provides the following common services:

Service Code	Service Name	Description of Service
$0E_h$	Get_Attribute_Single	Returns the contents of the specified attribute.
10_h	Set_Attribute_Single	Modifies an attribute value.

4.4.3.4 Object–Specific Services

The S-Analog Sensor Object provides the following common services:

Service Code	Service Name	Description of Service
$4B_h$	Zero_Adjust	Used to calibrate the zero pressure value. 2 = zero adjust

The Zero_Adjust service is used to cause the S-Analog sensor object device to modify its *Offset-A* attribute value (instance 1) based upon manufacturer specific algorithms. The target value specified in the service request represents the actual parametric measurement that the physical sensor should be reporting at the time of the request.

There are no state transitions associated with the invocation of this service. It is, therefore, incumbent upon the user to establish the device into the desired configuration prior to, and during, the execution of these services. This will generally involve exposing the sensor to a known environment and treating the values read during execution of the services accordingly.

A success service response indicates that the service was accepted and the application process started.

Zero adjust algorithm

To perform the Zero_Adjust Request pump down to a value 2 decades below the minimum pressure range of the gauge and then start the Zero_Adjust Request Service with a target value of "0".

Zero_Adjust request service data field parameters

Parameter	Data type	Description	Semantics of values
Zero_Adjust parameter or target value	UINT or FLOAT depending on data type	The target value for the zero calibration	The value to which the attribute 6, value, will be set (always 0). If not specified, the default value of zero is used.

4.4.3.5 Behavior

Data type

The following behavior with respect to *Data Type* applies:

The data type value will be set automatically based upon the first valid I/O connection established by the device.

If no established I/O connections exist, which include an attribute from this object, then the *Data Type* attribute is settable provided that the object is in the *Idle State*. The following example demonstrates this behavior:

A device specifies an instance of the S-Analog sensor object as well as two static assembly object instances, both with data attribute components mapped to this object instance. Assembly object instance ID 1 specifies INT data types and assembly object instance ID 2 specifies REAL data types.

After the device is on-line, it is configured with an I/O connection to assembly instance ID 2. When the connection transitions to the *Established State*, this object instance attribute *Data Type* is automatically set with the value for REAL before any data is communicated to, or from, the object instance. Any subsequent attempt to connect to assembly instance ID 1 would then be rejected and result in an INVALID ATTRIBUTE VALUE error with the additional error code indicating the ID of the offending attribute, which in this case would be the connection path.

4.4.3.6 Instance Attributes of Instance 2 / Atmosphere Pressure Sensor

Attribute ID	Access rule	NV/ V	Name	DeviceNet data type	Description of attribute	Semantics of values
3	→ "Semantics"	NV	Data type	USINT	Determines the data type of *value* and all related attributes as specified in this table.	→ "Semantics" section below [default] = INT
4	→ "Semantics"	NV	Data units	USNIT	Determines the units context of *value* and all related attributes.	[default] = Counts
5	Get	V	Reading valid	BOOL	Indicates that the *value* attribute contains a valid value.	0 = invalid 1 = valid (always set)
6	Get	V	Value	INT or REAL	Analog input value.	The corrected, converted, calibrated final value of the sensor. → "Semantics" section
7	Get	V	Status	BYTE	Alarm and warning state of this object instance.	= 0
10	Get	NV	Full scale	INT or REAL	The *value* of full scale for the sensor.	[default] = maximum allowable value for the *data type*
25	Set	NV	Safe state	USINT	Specifies the behavior for the *value* for states other than execute.	→ "Semantics" section below [default] = 0
26	Set	NV	Safe value	INT or specified by *data type*	The Value to be used for safe state = safe value	→ "Semantics" section below [default] = 0
32	Get	NV	Overrange	INT or REAL	Specifies the highest valid *value.*	The value above which attribute *reading valid* is set to invalid. [default] = maximum allowable value for the *data type*
33	Get	NV	Under-range	INT or REAL	Specifies the lowest valid *value.*	The value below which attribute *reading valid* is set to invalid. [default] = minimum allowable value for the *data type*
94	Get	V	Sensor warning	Structure of byte	Bit definitions of sensor warnings.	default = 0 → "Semantics" section below
95	Get	V	Sensor alarm	Structure of byte	Bit definitions of sensor alarms.	default = 0 → "Semantics" section below
96	Get	V	Status extension	Byte	Bit-mapped byte providing additional status bits of an S-Analog sensor instance.	default = 0 → "Semantics" section below
99	Get	V	Subclass	UINT	Identifies a subset of additional instance attributes, services and behaviors.	3 = Capacitance Diaphragm Gauge

4.4.3.7 Semantics of S-Analog Sensor Instance 2

Data type → Instance 1

Data units → Instance 1

Value → Instance 1

Safe state → Instance 1

Safe value → Instance 1

Sensor warning

16 Bits are used as sensor warnings.

Data component	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Sensor warning byte 0	0	0	0	0	0	0	0	0
Sensor warning byte 1	0	0	0	0	0	0	0	0

Sensor alarm

16 Bits are used as sensor faults.

Data component	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Sensor alarm byte 0	0	0	0	0	0	0	0	0
Sensor alarm byte 1	0	0	0	0	0	0	0	0

Status extension

8 Bits providing the current sensor alarm state of the instance

Data component	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Sensor alarm byte 0	0	0	0	0	0	Underrange exceeded = 0	Overrange exceeded = 0	Reading invalid [1] = 0

[1] Logical inversion of reading valid

4.4.3.8 Common Services of S-Analog Sensor Instance 2

The S-Analog sensor object provides the following common services:

Service code	Service name	Description of service
$0E_h$	Get_Attribute_Single	Returns the contents of the specified attribute.
10_h	Set_Attribute_Single	Modifies an attribute value.

4.4.3.9 User Atmosphere Adjust Service

Service Code	Service Name	Description of Service
32_h	User Atmosphere Adjust	Performs an Atmosphere Adjust for the ATM-Sensor

There are no state transitions associated with the invocation of this service. It is, therefore, incumbent upon the user to establish the device into the desired configuration prior to, and during the execution of this service. This will generally involve exposing the sensor to a known environment and treating the values read during execution of the services accordingly.

A success service response indicates that the service was accepted and the application process started.

To perform the User Atmosphere Adjust Request vent the gauge to atmosphere and then start the User Atmosphere Adjust Service without any target value. The pressure value of the atmospheric pressure sensor is then tracked to the value of the CDG.

4.5 Trip Point Object

Class Code $35_h = 53_d$

The Trip Point Object models the action of trip points (two relays) of the CDG (corresponding to physical outputs) (Discrete Output Object). A trip point value, designated as a high or low trip point, is compared to the specified source value. This trip point is intended to be used as a process control indicator.

4.5.1 Class Attributes

Attribute ID	Access rule	Name	DeviceNet data type	Description of attribute
1	Get	Revision	UINT	Revision of this object. Note: All class definitions are required to include this class attribute.
2	Get	Max instance	UINT	Maximum instance number of an object currently created in this class level of the device.

4.5.2 Instance Attributes of Trip Point Objects Instance 1 & 2

Attribute ID	Access rule	NV/ V	Name	DeviceNet data type	Description of attribute	Semantics of values
5	Set	NV	Low trip point	INT or based data type attribute	Defines the value at or below which a trip point condition will occur.	→ "Semantics" section below [default] = 0
6	Get	NV	Low trip enable	BOOL	Enables the low trip point setting.	[default] = 0
7	Get	V	Status	BOOL	State of this object instance.	0 = trip point condition does not exist (unasserted) 1 = trip point condition exists (asserted)
10	Set	NV	Hysteresis	Same as high/low point data type	Determines the amount by which the input must recover to clear a trip point condition.	→ "Semantics" section [default] = 0
12	Get	NV	Destination	Packed EPATH	Specifies the path of the destination attribute whose value will be set by *output.*	Path to attribute 3 of Instance 1 or 2 of DOP Object.
13	Get	V	Output	BOOL	Output of the object the value of which is sent to *destination.*	= Status of the relay
14	Get	NV	Source	Packed EPATH	Specifies the path of the source attribute whose value is retrieved for *Input.*	Path to attribute 6 of instance 1 of S-analog sensor object
15	Get	V	Input	INT or specified by *data type*	Input to the object whose value is retrieved from *source.*	Value of attribute 6 of instance 1 of S-analog sensor object
16	Get	NV	Data units	ENGUNITS	Units of input, trip point.	
17	Get	NV	Data type	USINT	Data type of input.	[default] = INT
101	Set	NV	Percentage of ATM	REAL	Multiplier for ATM control.	If high trip point enabled, this value will be multiplied with pressure value from S-Analog sensor instance 2 (ATM sensor) copied into trip point high
102	Get	V	High trip point	INT or based data type attribute	Defines the value at or above which a trip point condition will occur.	[default] = 0
103	Get	NV	High trip enable	BOOL	Enables the high trip point setting.	[default] = enabled

4.5.3 Semantics

Setpoint function

If "Low Trip Enable" is set, the setpoint function is used (automatically attr. 101 is set to zero and the functionality ATM detection is disabled).

Low trip point is compared to the *input* value to generate a trip point condition.

Status will be set if the input value is at or below the *low trip point.*

If the pressure increases above the low trip point and hysteresis, the *status* will be reset.

ATM detection

This functionality is used to compare the pressure measured by the CDG with the atmospheric pressure.

Attribute 101 is used to define a value "Percentage of Atmosphere". This value will be multiplied with the pressure from the S-Analog sensor instance 2 (ATM sensor). Then the result will be copied into attr. 102 = ATM high trip point. If a value is written to percentage of atmosphere, the low trip enable is set to zero.

Status will be set if the input value is at or above the *high trip point.*

If the pressure decreases below the high trip point - hysteresis, the *status* will be reset.

Hysteresis

The Hysteresis value specifies the amount by which the Input value must transition in order to clear a trip point condition.

The following relationship demonstrates the logic for a Low Trip Point:

For *Status* not set:	If (*Input* ≤ *Trip Point Low*)
	Then set *Status*
For Status set:	If (Input > Trip Point Low +Hysteresis)
	Then clear Status

4.5.4 Common Services

The trip point object provides the following common services:

Service code	Service name	Description of service
$0E_h$	Get_Attribute_Single	Returns the contents of the specified attribute.
10_h	Set_Attribute_Single	Modifies an attribute value.

4.6 Discrete Output Point Object

Class Code $09_h = 09_d$

4.6.1 Class Attributes

Attribute ID	Access rule	Name	DeviceNet data type	Description of attribute
1	Get	Revision	UINT	Revision of this object. Note: All class definitions are required to include this class attribute.
2	Get	Max instance	UINT	Maximum instance number of an object currently created in this class level of the device.

4.6.2 Instance Attributes of Instance 1 & 2

Attribute ID	Access rule	Name	DeviceNet data type	Description of attribute	Semantics of values
3	Get	Value	BOOL	Output point value.	0 = Off 1 = On (activated)

4.6.3 Common Services

The discrete output point object provides the following common services:

Service code	Service name	Description of service
$0E_h$	Get_Attribute_Single	Returns the contents of the specified attribute.

5 I/O Assembly Object

Class Code $04_h = 04_d$

A collection of assembly objects allows the sending of attributes from different application objects in one message (i.e. Polling I/O).

5.1 I/O Assembly Object Instance Data Attribute Format

The *S-Analog Sensor* object definition specifies a behavior that modifies the *Data Type* of certain attributes based upon the first valid I/O connection established to an Assembly Object instance. In order to maintain consistency, this device type will only allow connections to either INT or REAL based Assembly instances. Once a valid connection is established, attempts to configure connections, or otherwise access data, to a different type of Assembly instance will return a RESOURCE UNAVAILABLE error.

The I/O Assembly DATA attribute has the format shown below:

Instance	Type	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	Input	0 - 1	INT Pressure Value							
2 (default)	Input	0	Exception Status							
		1 - 2	INT Pressure Value							
3	Input	0	Exception Status							
		1	0	0	0	0	0	0	Trip Status Inst. 2	Trip Status Inst. 1
		2 - 3	INT Pressure Value							
4	Input	0 - 3	REAL Pressure Value							
5	Input	0	Exception Status							
		1 - 4	REAL Pressure Value							
6	Input	0	Exception Status							
		1	0	0	0	0	0	0	Trip Status Inst. 2	Trip Status Inst. 1
		2 - 5	REAL Pressure Value							
8	Input	0	Exception Status							
9	Input	0 - 1	Active Instance							
		2 - 3	INT Active Pressure Value							
10	Input	0	Exception Status							
		1 - 2	Active Instance							
		3 - 4	INT Active Pressure Value							
11	Input	0	Exception Status							
		1	0	0	0	0	0	0	Trip Status Inst. 2	Trip Status Inst. 1
		2 - 3	Active Instance							
		4 - 5	INT Active Pressure Value							
12	Input	0 - 1	Active Instance							
		2 - 5	REAL Active Pressure Value							
13	Input	0	Exception Status							
		1 - 2	Active Instance							
		3 - 6	REAL Active Pressure Value							
14	Input	0	Exception Status							
		1	0	0	0	0	0	0	Trip Status Inst. 2	Trip Status Inst. 1
		2 - 3	Active Instance							
		4 - 7	REAL Active Pressure Value							

Appendix

A: Conversion of a Floating Number According to IEEE 754

General

Number received: AA BB CC DD_h (4-Byte, floating format)

Legend:
- XX_h Hexadecimal number (Radix = 16)
- XX_d Decimal number (Radix = 10)
- XX_b Binary number (Radix = 2)

1. Reverse the sequence of the HEX words

DD_h	CC_h	BB_h	AA_h

2. Separate into bytes

SEEE $EEEE_b$	EMMM $MMMM_b$	MMMM $MMMM_b$	MMMM $MMMM_b$

Sign | 8-Bit exponent | 23-Bit mantissa

S	EEEE $EEEE_b$	MMM MMMM MMMM MMMM MMMM $MMMM_b$

3. Calculate

Sign =	Exponent =	Mantissa =
-1^S	XYZ_d	$RSTUV_d$

$$1+\left(\frac{\text{MMM MMMM MMMM MMMM MMMM MMMM}_b}{2^{23}}\right) = 1+\left(\frac{\text{NNNNN}_d}{8388608_d}\right)$$

Converted number:

$$\text{Sign} \times 2^{(\text{Exponent}-127)} \times \text{Mantissa}$$

Example

Number received

$00\ 00\ CA\ 42_h$ (4-Byte, floating format)

1. Convert sequence of the HEX words

42_h	CA_h	00_h	00_h

2. Separate into bytes

$0100\ 0010_b$	$1100\ 1010_b$	$0000\ 0000_b$	$0000\ 0000_b$

Sign | 8-Bit exponent | 23-Bit mantissa

0	$1000\ 0101_b$	$100\ 1010\ 0000\ 0000\ 0000\ 0000_b$

3. Calculate

⇩	⇩	⇩
-1^0		$1+\left(\frac{100\ 1010\ 0000\ 0000\ 0000\ 0000_b}{2^{23}}\right)$ $= 1+\left(\frac{4849664_d}{8388608_d}\right)$
⇩	⇩	⇩
Sign =	Exponent =	Mantissa =
1	**133**	**1.578125**

Converted number

$$1 \times 2^{(133-127)} \times 1.578125 = 101$$

B: Typical Start-Up Procedure

The start up of a device is divided into the steps:

- Allocation process
- Setting of the EPR attribute
- Choice of the input and output assemblies

Allocation process

Send an allocation string as defined in the DeviceNet specification to the device you want to allocate.

Set the bits in the allocation choice byte to 1 for these connections you want to use.

Example

Master MAC ID: 0
Allocation choice: Explicit, poll, bit strobe, COS
Slave address: 2

⇒ Allocation string: 416 00 4B 03 01 57 00
Slave's explicit/unconnected response message: 413 00 CB 00

Within the first allocation message the explicit connection has to be established.

Setting of the EPR attribute (expected packet rate)

After the allocation, the device activates an INACTIVITY WATCHDOG TIMER. This timer has to be set for every single connection (connection object, attribute 9) which is allocated in the allocation choice byte. This attribute defaults to 2500 (2500 milliseconds) within explicit messaging connections, and to zero within an I/O connection. If the INACTIVITY WATCHDOG TIMER expires, the established connection will be released. With every message the device receives, this timer is reloaded with the value specified in the according connection object and so it doesn't expire. The value zero deactivates the INACTIVITY WATCHDOG TIMER.

So in this step the inactivity watchdog timer has to be set. In testing mode you could use the value 0 to deactivate the inactivity watchdog timer.

In the following you see the strings for setting the EPR attribute (addresses as specified above):

ID	Message body	
414	00 10 05 01 09 00 00	set EPR of the explicit connection to zero
414	00 10 05 02 09 00 00	set EPR of the poll connection to zero
414	00 10 05 03 09 00 00	set EPR of the bit strobe connection to zero
414	00 10 05 04 09 00 00	set EPR of the COS/cyclic connection to zero

The responses of the slave are:

ID	Message body	
413	00 90 00 00	set EPR of the explicit connection to zero

Choice of the input and output assemblies

You can specify which of the several in a device predefined input/output assemblies should be used for **every single** connection.

Reading or setting of the input/output assemblies is possible only if the corresponding connection (polling, change of state, bit strobe) has been allocated in the allocation message.

Reading the configured assemblies

If you want to read the number of the chosen assembly, you must read the attributes 14 and 16 in the corresponding Instance of the connection object (object ID 5).

For reading this value, the connection has to be established. The EPR attribute may be set.

Instance 2: Polling

Instance 3: Bit strobe

Instance 4: Change of state/cyclic

Setting of assemblies

If you want to set the number of the chosen assembly, you must set the attributes 14 and 16 in the corresponding instance of the connection object.

For setting this value, the connection has to be allocated, but the EPR attribute has not to be set to any value.

Examples

Read a configured assembly (addresses as specified above)

Get single request:

ID	Message body	
414	00 0E 05 02 0E	Get produced connection path (request for input assembly by master).

Get single response:

ID	Message body	
413	00 8E 20 04 24 05 30 03	Response from slave

The addressing format of the attribute values differ from the normal mode. A connection path attribute that specifies class 4, instance 5, and attribute ID 3 is illustrated below:

	Class #4
20	04

	Instance #5
24	05

	Attribute #3
30	03

The instance defines the assembly you want to use. This format has to be used by the master in the request and is used by the slave in the response.

Set the input assembly 04 for a poll connection (addresses as specified above)

Set single request:

ID	Message body	
414	80 00 10 05 02 10 20 04	First fragment
414	80 81 24 04 30 03	Second fragment

Because the message body is greater than 8 bytes, the fragmented protocol has to be used.

Set single response:

ID	Message body	
413	80 C0 00	Response on first fragment
413	80 C1 00	Response on second fragment

C: Literature

🕮 [1] www.inficon.com
Operating Manual
CDG045D
tina51e1
INFICON AG, LI–9496 Balzers, Liechtenstein

🕮 [2] www.inficon.com
Operating Manual
CDG100D
tina52e1
INFICON AG, LI–9496 Balzers, Liechtenstein

🕮 [3] www.inficon.com
Operating Manual
CDG160D, CDG200D
tina53e1
INFICON AG, LI–9496 Balzers, Liechtenstein

🕮 [4] www.odva.org
Open DeviceNet Vendor Association, Inc.
DeviceNet™ Specifications

🕮 [5] Common Industrial Protocol (CIP™) Ed. 3.5 and DeviceNet™ Adaption of CIP Ed. 1.6 (Open DeviceNet Vendor Association)

🕮 [6] www.inficon.com
Product descriptions and downloads
INFICON AG, LI–9496 Balzers, Liechtenstein

Original: English tira54e1 (2009-01)
tira54e1

INFICON
LI–9496 Balzers
Liechtenstein
Tel +423 / 388 3111
Fax +423 / 388 3700
reachus@inficon.com

www.inficon.com