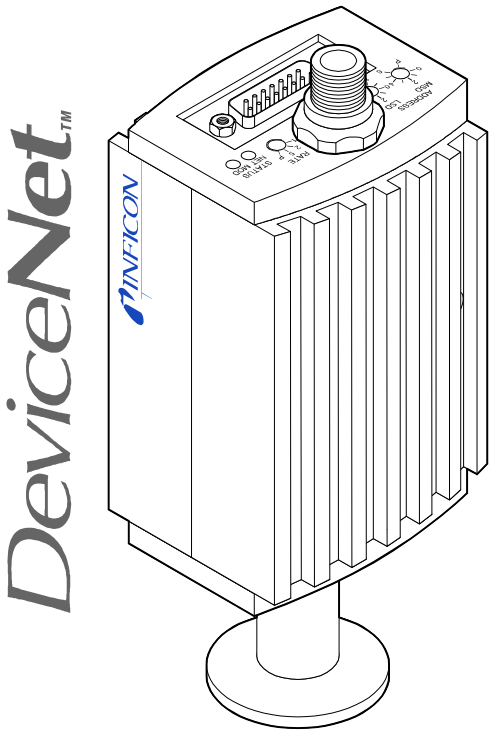
DeviceNet™

Interface for Bayard-Alpert / Pirani Gauge
and High Pressure / Pirani Gauge

BPG400-SD
HPG400-SD



Intended Use of this Document

This Communication protocol contains instructions for operating the vacuum gauges BPG400-SD and HPG400-SD (featuring DeviceNet interfaces) as slaves together with a DeviceNet master.

This manual describes the functionality of DeviceNet for programming purposes. For more information refer to the "DeviceNet specifications" of the Open DeviceNet Vendor Association (ODVA) (→ 🕮 [2]) and the corresponding european standard (→ 🕮 [3]).

For specifications and operation instructions of the vacuum gauges refer to the appropriate documents:

BPG400-SD → 🕮 [1], [4], [5], [6]
HPG400-SD → 🕮 [1], [7], [8], [9]

DeviceNet Interface

The following description of the DeviceNet Interface is compliant to the DeviceNet specification of the Open DeviceNet Vendor Association.

This manual describes the functionality of a DeviceNet Group 2 Only Slave and supports Explicit Messaging and the I/O Polling.

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

Abbreviations

Abbr.	Meaning	
NV	Nonvolatile; attribute value is maintained through power cycles	
V	Volatile	
INT	Integer value	(Range –32767 … 32768)
UINT	Unsigned integer value	(Range 0 … 65635)
USINT	Unsigned character value	(Range 0 … 255)
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)

Trademarks

DeviceNet™ Open DeviceNet Vendor Association Inc.

Contents

For cross-references within this document, the symbol (→ 🗎 XY) is used, for cross-references to further documents listed under literature, the symbol (→ 🕮 [Z]).

1 Starting-Up of the Slave

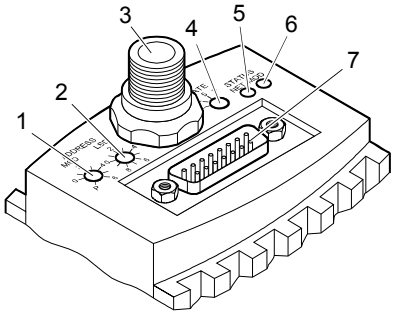
1.1 Power Supply Requirements

The BPG400-SD / HPG400-SD has to be powered with two voltages:

1.) 24 Volt DC, 18 W at the 15 pole Sub-D connector for the gauge itself;

2.) 24 Volt DC nominal, <2 W range 11 … 25 V) at the DeviceNet micro style connector for the DeviceNet transceiver.

1.2 Front View of the BPG400-SD and HPG400-SD

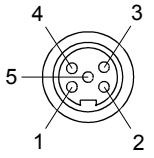


Position	Function
1	Address Switch × 10, decimal
2	Address Switch × 1, decimal
3	DeviceNet connector
4	Data Rate Switch
5	Network Status LED
6	Module Status LED
7	"Sensor cable" connector (Power, analog I/O, RS232C I/O and Relay contacts)

1.3 Connectors on the Device

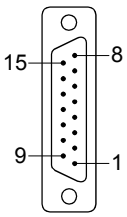
The BPG400-SD / HPG400-SD uses a "Sealed Micro-Style Connector" for the DeviceNet connection. The DeviceNet part of the gauge is powered via the DeviceNet connector.

Pin Assignment of the Sealed Micro-Style Connector



Pin	Function
1	Drain
2	V+ nominal 24 V (range 11 … 25 V)
3	V–
4	CAN_H
5	CAN_L

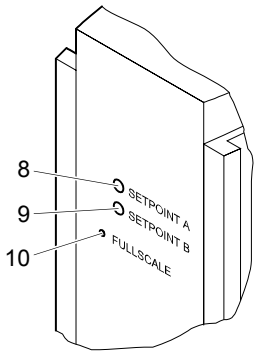
Pin Assignment of the 15-pin D-Sub connector



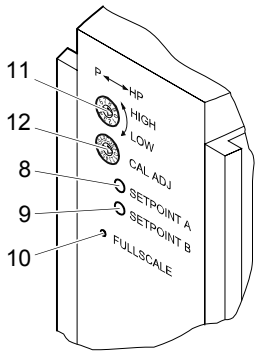
Pin	Function
1	Setpoint A, Relay Common
2	Pressure Output (Signal Output 0 … +10 V)
3	Threshold Setpoint A (Output 0 … +10 V)
4	Setpoint A, Relay, n.o. contact
5	Supply sensor electronics common
6	Threshold Setpoint B (Output 0 … +10 V)
7	Degas Input , High Active (only BPG400-SD)
8	Supply sensor electronics +24 Volt
9	Setpoint B, Relay Common
10	Gauge identification
11	Setpoint B, Relay, n.o. contact
12	Signal common GND
13	RS232, TxD
14	RS232, RxD
15	Housing, shielding, GND

1.4 Side View of the BPG400-SD and HPG400-SD

BPG400-SD



HPG400-SD



Position	Function
8	Potentiometer for Setpoint A threshold
9	Potentiometer for Setpoint B threshold
10	Fullscale adjustment push button
11	Emission On threshold adjustment (marked P↔HP, only HPG400-SD)
12	Hot cathode sensor calibration adjustment (only HPG400-SD)

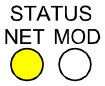
1.5 Indicators and Switches

1.5.1 Module Status LED

STATUS
NET MOD

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. As stated previously, all module level faults are considered as unrecoverable faults.

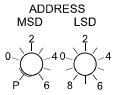
1.5.2 Network Status LED



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

For this 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.
On–line, 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).

1.5.3 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.

1.5.4 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 the position 1 is equal to 125 kbaud, 2 equal 250 kbaud and 5 equal 500 kbaud. All remaining switch positions specify software selection of the data rate.

1.5.5 Setpoint

The device has two setpoint relays (Setpoint A and Setpoint B). The setpoints of these relays are adjustable only by two potentiometers (marked SETPOINT A and SETPOINT B).

The relay contacts are available at the 15-pin D-Sub connector.

2 Object Structure

2.1 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

Please refer to the DeviceNet specification for further information.

2.1.1 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	Requirement in Implementation	Access Rule	NV/ V	Name	DeviceNet Data Type	Description of Attribute	Semantics
100	Optional	Get/Set	NV	Poll Produce Assembly Instance	USINT	Contains the Instance number of the assembly used by this connection to send data	1, 2, 4, 5, 8, 9, 10, 12, 13

This vendor-specific attribute facilitates the configuration of the data assembly, sent by the BPG400-SD / HPG400-SD to the DeviceNet Master as Poll response. It offers the possibility to select a (predefined) data assembly via a configuration tool + EDS file → 🕮 [1]. Attribute 100 allows the user to configure the Poll I/O Data Assembly via EDS even when the Poll Connection of the BPG400-SD / HPG400-SD 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 BPG400-SD / HPG400-SD.

Thus, definition of the BPG400-SD / HPG400-SD data assembly can be done in two ways:

1) Standard methode (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 BPG400-SD / HPG400-SD. 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.

2.2 Identity Object

Class Code $01_h = 01_d$

2.2.1 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.

2.2.2 Instance Attributes

Attribute ID	Access Rule	NV/ V	Name	DeviceNet Data Type	Description of Attribute
1	Get	NV	Vendor ID	UINT 79 02	Manufacturer identification Value: 633 (INFICON)
2	Get	NV	Device Type	UINT	$1C_h$ = Vacuum pressure gauge
3	Get	NV	Product Code	UINT	for example: 9
4	Get	NV	Revision	STRUCT	
5	Get	NV	Status	WORD	
6	Get	NV	Serial Number	UDINT	
7	Get	NV	Product Name	SHORT STRING	for example: BPG400-SD

Services

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

2.3 S-Device Supervisor Object

Class Code $30_h = 48_d$

2.3.1 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.

2.3.2 Instance Attributes

Attribute ID	Access Rule	NV/ V	Name	DeviceNet Data Type	Description of Attribute
3	Get	NV	DeviceType	SHORT STRING	ASCII Text, "CG" combination 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"
6	Get	NV	Manufacturer's Model Number	SHORT STRING	for example: BPG400-SD
7	Get	NV	Software Revision Level	SHORT STRING	ASCII Text, for example *"xxxyyy"* *xxx ≙ version of the measuring print* *yyy ≙ version of the DeviceNet print*
8	Get	NV	Hardware Revision Level	SHORT STRING	ASCII text, for example *"1.001"*
9	Get	NV	Manufacturer's Serial Number	SHORT STRING	ASCII text
10	Get	NV	Device Configuration	SHORT STRING	Depending on device configuration (→ Appendix B)
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 4	USINT	Number of device detail bytes
			Detail 0	ARRAY of:	S-Analog Sensor Instance 1 **(Pirani)** Sensor alarm byte 0
			Detail 1	BYTE	S-Analog Sensor Instance 1 **(Pirani)** Sensor alarm byte 0
			Detail 2		S-Analog Sensor Instance 2 **(Hot cathode)** Sensor Alarm byte 0
			Detail 3		S-Analog Sensor Instance 2 **(Hot cathode)** Sensor alarm byte 1
			Manufacturer Exception Detail	STRUCT of:	
			Size 1	USINT	Number of Manufacturer detail bytes
			Detail 0	ARRAY of:	Serial communication alarm

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 6	USINT	Number of Device Detail Bytes
			Detail 0	ARRAY of:	S-Analog Sensor Instance 1 **(Pirani)** Status Extension
			Detail 1	BYTE	S-Analog Sensor Instance 1 **(Pirani)** Sensor Warning Byte 0
			Detail 2	BYTE	S-Analog Sensor Instance 1 **(Pirani)** Sensor Warning Byte 1
			Detail 3		S-Analog Sensor Instance 2 **(hot cathode)** Status Extension
			Detail 4		S-Analog Sensor Instance 2 **(hot cathode)** Sensor Warning Byte 0
			Detail 5		S-Analog Sensor Instance 2 **(hot cathode)** Sensor Warning Byte 1
			Manufacturer Exception Detail	STRUCT of:	
			Size 1	USINT	Number of Manufacturer Detail Bytes
			Detail 0	Byte	serial common warning
15	Set	NV	Alarm Enable	BOOL	→ "Semantics" section below.
16	Set	NV	Warning Enable	BOOL	→ "Semantics" section below.
101	Get	NV	Sensitivity	USINT	→ "Semantics" section below. **only HPG400-SD**
102	Get	NV	Emission On Threshold	INT or REAL	→ "Semantics" section below. **only HPG400-SD**

2.3.2.1 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 Bit Map

Bit	Function
0	ALARM/device-common (The alarm or warning is not specific to the device type or device type manufacturer.)
1	ALARM/device-specific
2	ALARM/manufacturer-specific
3	reserved, set to 0
4	WARNING/device-common
5	WARNING/device-specific
6	WARNING/manufacturer-specific
7	1 ≙ Expanded Method

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".

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=2] 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.

Two bytes Common Exception Detail are provided: Common Exception Detail[0] and Common Exception Detail[1]. The specific exception associated with each of the bitmaps is given in the table below. The SIZE for this revision is two (2).

Common Exception Detail Attribute Values

Bit	Common Exception Detail [0]	Common Exception Detail [1]
0	0	0
1	0	0
2	EPROM exception	0
3	EEPROM exception	power supply input voltage
4	RAM exception	0
5	reserved	0
6	0	0
7	0	0

Common Exception Detail Format Summary

Data Component	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Common Exception Detail Size	0	0	0	0	0	0	1	0
Common Exception Detail 0	0	0	0	Data Memory	Nonvolatile Memory	Code Memory	0	0
Common Exception Detail 1	0	0	0	0	PS Input Voltage	0	0	0

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 = 4 for Alarms and SIZE = 6 for Warning] 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.

Device Exception Detail Alarms and Manufacturer Exception Detail Alarms Format

Data Component	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Device Exception Detail Alarm Size	0	0	0	0	0	1	0	0
Device Exception Detail Alarm 0 **Pirani**	0	0	0	0	0	0	0	0
Device Exception Detail Alarm 1 **Pirani**	0	0	0	0	0	0	Electronics Failure	0
Device Exception Detail Alarm 2 **Hot cathode**	0	0	0	0	0	0	0	0
Device Exception Detail Alarm 3 **Hot cathode**	0	0	0	0	0	0	Electronics Failure	0
Manufacturer Exception Detail Alarm Size	0	0	0	0	0	0	0	1
Manufacturer Exception Detail Alarm	0	0	0	0	0	0	0	serial comm.

Exception Detail Warning

Data Component	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Device Exception Detail Warning Size	0	0	0	0	0	1	1	0
Device Exception Detail Warning 0 **Pirani**	0	0	0	0	0	Underrange Exceeded	Overrange Exceeded	Reading Invalid *)
Device Exception Detail Warning 1 **Pirani**	0	0	0	0	0	0	0	0
Device Exception Detail Warning 2 **Pirani**	0	0	0	0	0	0	Electronics Warning	0
Device Exception Detail Warning 3 **Hot cathode**	0	0	0	0	0	Underrange Exceeded	Overrange Exceeded	Reading Invalid *)
Device Exception Detail Warning 4 **Hot cathode**	0	0	0	0	0	0	0	0
Device Exception Detail Warning 5 **Hot cathode**	0	0	0	0	Pressure too high for degas	0	0	0
Manufacturer Exception Detail Warning Size	0	0	0	0	0	0	0	1
Manufacturer Exception Detail Warning	0	0	0	0	0	0	0	serial comm. Warning

*) Logical inversion of *Reading Valid*.

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).

Sensitivity
(Only HPG400-SD)

This attribute contains the position of the hot cathode sensor calibration switch (→ 🗎 6).

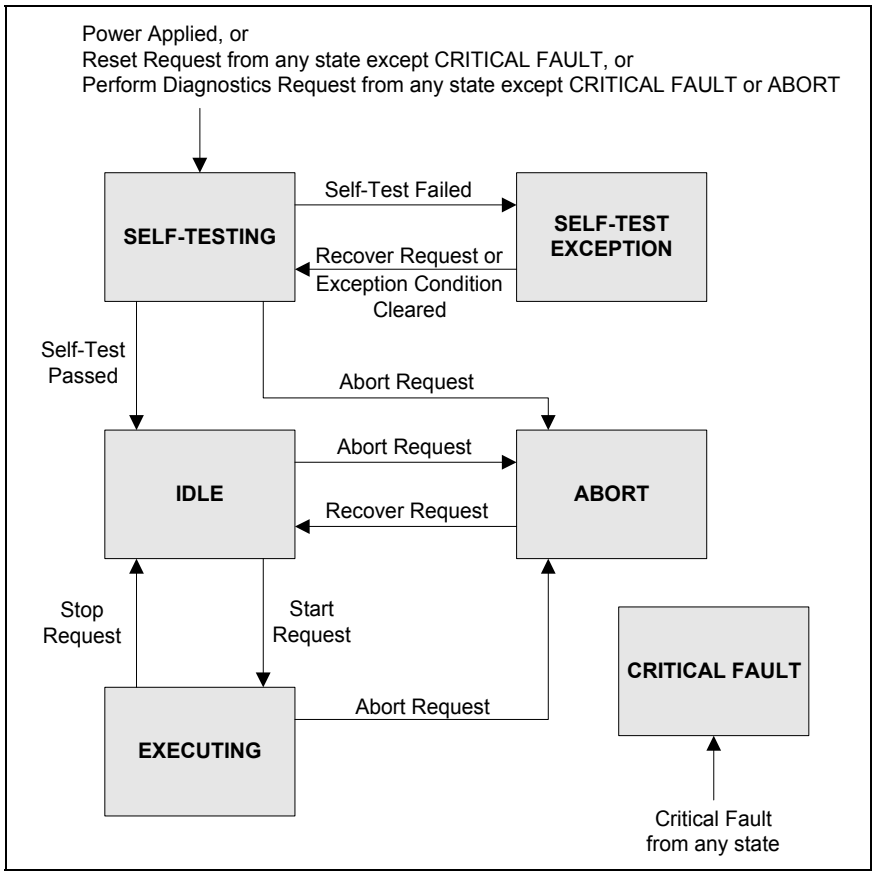
Emission On Threshold
(Only HPG400-SD)

This attribute contains the position of the Emission On threshold switch (→ 🗎 6).

Depending on the value of the data type attribute, this attribute is INT or REAL value.

Position of the "Emission On" Switch (P↔HP)	Emission on threshold [mBar]	Emission on threshold [Torr]	Emission on threshold [Pa]	Emission on threshold [Counts]
0 or 1	1	0.75006168	100	28333.2625
2 or 3	0.5	3.75031×10^{-1}	50	28132.57634
4 or 5	0.2	1.50012×10^{-1}	20	27867.2837
6 or 7	0.1	7.50062×10^{-2}	10	27666.5975
8 or 9	0.05	3.75031×10^{-2}	5	27465.9113

2.3.3 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.

2.3.4 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.

2.3.5 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
$4D_h$	Perform_Diagnostics	Causes the device to perform a set of diagnostic routines

Explanation to Service code $4D_h$

- 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.

2.4 S-Analog Sensor Object

Class Code $31_h = 49_d$

2.4.1 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	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.	2
99	Get	Subclass	UINT	Identifies a subset of additional class attributes, services and behaviors.	1 ≙ Instance Selector

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.

The Active Instance Number will be modified based upon the Active Value in order that the best gauge, corresponding to a given S-Analog Sensor instance, will be active for the given measurement range.

Number of Gauges

This attribute is used to determine the size of all Input Assemblies within a node.

2.4.2 Instance Attributes

Three S-Analog Sensor Instances (Instance 1, Instance 2, Instance 21 and Instance 22) are available. Instance 1 represents the physical sensor reading of the heat transfer vacuum gauge (pressure), Instance 2 represents the physical sensor reading of the hot cathode ion gauge (pressure).

Instance 21 and Instance 22 represent the value of the Setpoint A and Setpoint B.

2.4.2.1 Instance Attributes of Instance 1 Pirani Instance

Following is the Instance 1 with the subclass extension of **the heat transfer vacuum gauge (pirani gauge)** part of the BPG400-SD. This instance is used to provide control and status information for the Pirani gauge part of the BPG400-SD.

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.	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	Always zero, because Alarm and Warning Trip Points are not implemented
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*
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*

Attribute ID	Access Rule	NV/ V	Name	DeviceNet Data Type	Description of Attribute	Semantics of Values
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*
94	Get	V	Sensor Warning	Struct of Byte	Bit definitions of Sensor Warnings	0 = [default] → "Semantics" section
95	Get	V	Sensor Alarm	Struct of Byte	Bit definitions of Sensor Alarms	0 = [default] → "Semantics" section
96	Get	V	Status Extension	BYTE	Bit-mapped byte providing additional status bits	Bit description: 0 Reading Invalid (Logical Inversion of *Reading Valid)* 1 Overrange Exceeded 2 Underrange Exceeded
99	Get	NV	Subclass	UINT	Defines a subset of additional attrib-utes, services and behaviors.	02 = Heat Transfer Vacuum Gauge

2.4.2.2 Semantics of S-Analog Sensor Instance 1

Data Type

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

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.

Using Counts and INT the following conversion has to be used:

BPG400-SD:

$$\text{Counts} = [\log_{10}(\text{pressure}) + k] \times 2000$$

where:

$k_{mbar} = 12.5$

$k_{Torr} = 12.624903$

$k_{Pa} = 10.5$

HPG400-SD:

Pirani gauge measuring range 27000 < counts < 30333

$$\text{Counts} = [\log_{10}(\text{pressure}) + k] \times 666.665$$

where:

$k_{mbar} = 42.5$

$k_{Torr} = 42.624903$

$k_{Pa} = 40.5$

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. Bits 8 … 16 are mapped to the Exception Detail Alarm 1, Bits 0 … 7 are mapped to the Device Exception Detail Alarm 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	0	0	0
Sensor Alarm Byte 1	0	0	0	0	0	0	Electronics Failure	0

Sensor Warning

16 Bits are used as Sensor Warnings. Bits 8 … 16 are mapped to the Exception Detail Warning 2, Bits 0 … 7 are mapped to the Device Exception Detail Warning 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	0	0	0	0	0	0	Electronics Warning	0

2.4.2.3 Instance Attributes of Instance 2 Hot Cathode Ion Gauge

Following is the Instance 2 with the subclass extension of the hot cathode ion gauge part of the BPG400-SD / HPG400-SD. This instance is used to provide control and status information for the hot cathode ion gauge part of the BPG400-SD / HPG400-SD.

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 [default] = INT
4	→ "Semantics"	NV	Data Units	ENGUNITS	Determines the Units context of *Value* and all related attributes.	→ "Semantics" section [default] = Counts
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* if supported	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	→ "Semantics" section
10	Get	NV	Full Scale	INT or specified by *Data Type* if supported	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*
25	Set	NV	Safe State	USINT	Specifies the behavior for the *Value* for states other than Execute	→ "Semantics" section [default] = 0
26	Set	NV	Safe Value	INT or specified by *Data Type* if supported	The Value to be used for Safe State = Safe Value	→ "Semantics" section [default] = 0
32	Get	NV	Overrange	INT or specified by *Data Type* if supported	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* if supported	Specifies the lowest valid *Value*	The value below which attribute *Reading Valid* is set to invalid. [default] = minimum allowable value for the *Data Type*
88	Get	V	Degas Status	BOOL	Indicates current degas state	0 ≙ OFF 1 ≙ ON
91	Get	V	Emission Current	REAL	Indicates setting level of emission current in amps	
93	Get	V	Emission Status	BOOL	Indicates whether the emission is turned ON or OFF	0 ≙ OFF 1 ≙ ON
94	Get	V	Sensor Warning	Structure of Byte		default ≙ 0 → "Semantics"
95	Get	V	Sensor Alarm	Structure of Byte		default ≙ 0 → "Semantics"

Attribute ID	Access Rule	NV/ V	Name	DeviceNet Data Type	Description of Attribute	Semantics of Values
96	Get	V	Status Extension	BYTE	Bit mapped byte providing additional status bits	Bit description: 0 Reading Invalid (Logical Inversion of *Reading Valid)* 1 Overrange Exceeded 2 Underrange Exceeded
99	Get	NV	Subclass	UINT	Identifies the subset of additional attributes, services and behaviors for hot cathode ion gauges	5 ≙ Hot Cathode Ion Gauge

2.4.2.4 Semantics of S-Analog Sensor Instance 2

Data Type

→ Instance 1

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.

Using Counts and INT the following conversion has to be used:

BPG400-SD:

$$\text{Counts} = [\log_{10}(\text{pressure}) + k] \times 2000$$

where:

$k_{mbar} = 12.5$

$k_{Torr} = 12.624903$

$k_{Pa} = 10.5$

HPG400-SD, Hot cathode measuring range 8333 < counts < 24333:

$$\text{Counts} = [\log_{10}(\text{pressure}) + k] \times 2666.665$$

where:

$k_{mbar} = 9.125$

$k_{Torr} = 9.249903$

$k_{Pa} = 7.125$

Safe State

→ Instance 1

Safe Value

→ Instance 1

Sensor Alarm

16 Bits are used as sensor faults . Bit 8 … Bit 16 are mapped to the Exception Detail Alarm 3, Bit 0 … Bit 7 are mapped to the Exception Detail Alarm 2.

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	Electronics Failure	0

Sensor Warning

16 Bits are used as sensor warnings. Bit 8 … Bit 16 are mapped to the Exception Detail Warning 5, Bit 0 … Bit 7 are mapped to the Exception Detail Warning 4.

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	Pressure too high for degas (Only BPG400-SD).	0	0	0

The bit "Pressure too high for degas" will be set if the pressure is above 7.2×10^{-6} mbar when a degas service is requested. The bit will be reset when the pressure is below 7.2×10^{-6} mbar (Only BPG400-SD).

2.4.2.5 Instance Attributes of Instance 21 / Setpoint A (Instance 22 / Setpoint B)

Attribute ID	Access Rule	NV/ V	Name	DeviceNet Data Type	Description of Attribute	Semantics of Values
3	Set / Conditional: → Instance 1	NV	Data Type	USINT	Determines the Data Type of *Value* and all related attributes as specified in this table.	→ "Semantics" section below. int ≙ C3 hex [default] float ≙ CA hex
4	Get	NV	Data Units	UINT	Determines the Units context of *Value* and all related attributes.	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
6	Get	V	Value	INT or specified by *Data Type*	The value of the Setpoint A (Setpoint B) relay.	
7	Get	V	Status	BYTE	Alarm and Warning State of this object instance	→ "Semantics" section below. Behavior as Trip Point LOW

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

Set Points

If the pressure (attribute 6, instance 1 and 2) decreases below the Set Point value (attribute 6, instance 21 and 22) an alarm or warning exception condition will be generated. The hysteresis is set to 10%.

For example: A SET Point value of 100 will result in an exception condition being set when the Value is below 100 and cleared when the *Value* increases above 110.

The setpoints can only be read by DeviceNet. It is not possible to set the values or to influence the state of the relays by DeviceNet. Even if the device is not allocated, the relay status is set corresponding to the voltage adjusted by the two setpoint potentiometers.

Only in case of a microcontroller RESET (for example reset on Identity Object), the relays will be reset too.

The setpoint is available in the actual pressure unit and data type. If the pressure unit "Counts" is set, use the following formula for the conversion from Counts to pressure:

BPG400-SD:

$$p_{mbar} = 10^{Counts / 2000 - k}$$

where:

$k_{mbar} = 12.5$

$k_{Torr} = 12.624903$

$k_{Pa} = 10.5$

The relation between setpoint and voltage is:

$$p_{mbar} = 10^{(U - 7.75) / 0.75 + c}$$

where:

$c_{mbar} = 0$

$c_{Torr} = -0.125$

$c_{Pa} = 2$

HPG400-SD:

$$p_{mbar} = 10^{9 \times Counts / 1024 - 6}$$

The relation between setpoint and voltage is:

$$p_{mbar} = 10^{9 \times U / 10 - 6}$$

The setpoints are only activated, if the pressure is below 100 mbar, therefore a setpoint above 100 mbar can not be realised.

2.4.3 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.

2.4.4 Object-Specific Services on Instance 1 / Pirani

Service Code	Service Name	Description of Service
$4C_h$	Full Scale Adjust	Performs a Full Scale Adjust for the Pirani

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.

Full Scale adjust algorithm

To perform the Full Scale Adjust Request vent the gauge to atmosphere and then start the Full Scale Adjust Service with a target value as defined below .

Full Scale Adjust Request Service Data Field Parameters

BPG400-SD:

Parameter	Data Type	The target value for the full scale calibration			
Data Unit		Counts	mbar *)	Pascal *)	Torr *)
Target Value	INT	0x7918	0x03E8	0x7FFF	0x02ED
	REAL	0x46F23000	0x447A0000	0x47C35000	0x443B83F3

HPG400-SD:

Parameter	Data Type	The target value for the full scale calibration			
Data Unit		Counts	mbar *)	Pascal *)	Torr *)
Target Value	INT	0x767D	0x03E8	0x7FFF	0x02ED
	REAL	0x46ECFA00	0x447A0000	0x47C35000	0x443B83F3

*) INFICON recommends to use mbar, Torr or Pascal only with the data type REAL.

Special limitations apply for INT:

The full scale value of the gauge is 1000 mbar ≙ 10000 Pascal. But the max. pressure of the gauge can not be described using INT and Pascal. The target value 0x7FFF (the maximum INT value) allows to perform a Full Scale Adjust Service for the unit Pascal.

For the data type INT and pressure values below 1 (in the actual pressure unit) you always get a pressure of zero.

2.4.5 Object-Specific Services on Instance 2 / Hot Cathode Ion Gauge (Only BPG400-SD)

Service Code	Service Name	Description of Service
61_h	Set Degas State	Activates/deactivates degas mode according to the parameter Degas State. Degas mode may be terminated either automatically by device timeout (3 min) or remotely by this service.

Set Degas State Request Service Data Field parameters

Parameter	Data Type	Description	Semantics of Values
Degas State	BOOL		0 ≙ switches Degas OFF 1 ≙ switches Degas ON

2.4.6 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 online, 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.

3 I/O Assembly Object

Class Code 04_h

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

3.1 I/O Assembly Instances

The following table identifies the I/O assembly instances supported by the gauge device.

Number	Type	Name
1	Input	Pressure Value (Active Instance)
2	Input	Exception Status and INT Pressure Value (Active Instance)
4	Input	REAL Pressure Value (Active Instance)
5	Input	Exception Status and REAL Pressure Value (Active Instance)
8	Input	Exception Status
9	Input	Active Instance, Active Pressure Value
10	Input	Exception Status and Active Instance and INT Active Pressure Value
12	Input	Active Instance REAL Active Pressure Value
13	Input	Exception Status Active Instance REAL Active Pressure Value

3.2 I/O Assembly Object Instance Data Attribute Format

In order to maintain consistency, this device type will only allow connections to either INT or REAL based Assembly instances (→ Data Type definition 🗎 19). Once a valid connection is established, attempts to configure connections to a different type of Assembly instance will return an error.

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

Instance	Type	Byte	Bit 0 … 7
1	Input	0	INT Pressure Value (low byte)
		1	
2	Input	0	Exception Status; Class 48, Instance 1, Attribute 12
		1	INT Pressure Value (low byte)
		2	Class 49, Active Value
4	Input	0	REAL Pressure Value (low byte)
		1	
		2	Class 49, Active Instance Value
		3	
5	Input	0	Exception Status Class 48, Instance 1, Attribute 12
		1	REAL Pressure Value (low byte)
		2	Class 49, Active Instance Value
		3	
		4	
8	Input	0	Exception Status ; Class 48, Instance 1, Attribute 12
9		0	Active Instance
		1	
		2	INT Active Pressure Value
		3	
10	Input	0	Exception Status
		1	Active Instance
		2	
		3	INT Active Pressure Value
		4	
12	Input	0	Active Instance
		1	
		2	REAL Pressure Value
		3	
		4	
		5	
13	Input	0	Exception Status
		1	Active Instance
		2	
		3	REAL Pressure Value
		4	
		5	
		6	

Appendix

A: Range of Values

Integer	int	–32767 … 32768
Unsigned integer	uint	0 … 65535
Float	float	according IEEE 754

B: Specific Codes

Manufacturer product code

9 = BPG400-SD

10 = HPG400-SD

C: 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

⇩	⇩	⇩
		$1+\left(\frac{MMM\ MMMM\ MMMM\ MMMM\ MMMM\ MMMM_b}{2^{23}}\right)$
		$=1+\left(\frac{NNNNN_d}{8388608_d}\right)$
		⇩
Sign = -1^S	Exponent = XYZ_d	Mantissa = $RSTUV_d$

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

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

D: 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 of the principal allocation process.

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

Allocated instances may not be valid for the BPG400-SD

⇒ 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.

The I/O connections bit strobe and COS/Cyclic are not supported by the BPG400-SD. Appendix D describes only the general allocation procedure for all devices (group 2 slave only).

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 ms) 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, therefore it normally doesn't expire. The value zero deactivates the INACTIVITY WATCHDOG TIMER.

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 input/output assemblies predefined in a device 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 have to set the attributes 14 and 16 in the corresponding instance of the connection object.

To set 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 differs 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 longer 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

E: Literature

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

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

[3] European Standard for DeviceNet EN 50325

[4] www.inficon.com
Operating Instructions
BPG400, BPG400-SD, BPG400-SP, BPG400-SR
tina03e1
INFICON AG, LI–9496 Balzers, Liechtenstein

[5] www.inficon.com
Instruction Sheet
BPG400, BPG400-SD, BPG400-SP, BPG400-SR
tima03e1
INFICON AG, LI–9496 Balzers, Liechtenstein

[6] www.inficon.com
Instruction Sheet
BPG400-SD, BPG400-SP, BPG400-SR
tima36e1
INFICON AG, LI–9496 Balzers, Liechtenstein

[7] www.inficon.com
Operating Instructions
HPG400, HPG400-SD, HPG400-SP
tima31e1
INFICON AG, LI–9496 Balzers, Liechtenstein

[8] www.inficon.com
Instruction Sheet
HPG400, HPG400-SD, HPG400-SP
tima31e1
INFICON AG, LI–9496 Balzers, Liechtenstein

[9] www.inficon.com
Instruction Sheet
HPG400-SD, HPG400-SP
tima32e1
INFICON AG, LI–9496 Balzers, Liechtenstein

Notes

Notes

Original: English

tirra03e1-b

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

www.inficon.com