Installation and Operation Manual
X-DPT-DeviceNet-GF100-Series-MFC-eng
Part Number: 541B184AAG
March, 2013

DeviceNet™ Supplemental Manual for GF100 Series Mass Flow Controllers and Meters

Brooks® GF135 Series and GF125 Series

Dear Customer,

We recommend that you read this manual in its entirety as this will enable efficient and proper use of the DeviceNet MFCs. Should you require any additional information concerning the DeviceNet MFCs, please feel free to contact your local Brooks Sales and Service Office; see back cover for contact information, or visit us on the web at www.BrooksInstrument.com. We appreciate this opportunity to service your fluid measurement and control requirements, and trust that we will be able to provide you with further assistance in future.

Yours sincerely,

Brooks Instrument

Contents

THIS PAGE WAS INTENTIONALLY LEFT BLANK

1. Introduction

This document and the DeviceNet Statement of Compliance (SOC) from the Open DeviceNet Vendors Association (ODVA) provide a complete interoperability specification for the GF135 Digital Mass Flow Controller device from Brooks. This device is defined by the ODVA specification in the Device Profiles chapter, section entitled “Enhance Mass Flow Controller, Type: 27_{hex}”. Information contained in this document was derived from the following sources:

- DeviceNet Specification Enhancements for the S-Device Supervisor Objects:
  - S-Device Supervisor Object (DSE-93-01) [1]
  - S-Analog Sensor Object (DSE-93-02) [2]
  - S-Analog Actuator Object (DSE-93-03) [3]
  - S-Single Stage Controller Object (DSE-93-04) [4]
  - S-Gas Calibrator Object (DSE-93-05) [5]
- ODVA Mass Flow Controller Device Profile (DSE 93-06) [6]
- ODVA DeviceNet Specifications Version 2.0 [7]
- ODVA Enhanced Mass Flow Controller Device Profile (Edition 3.4, CIP Spec.)
- AMAT (various docs)

This device also complies with the ODVA Semiconductor SIG Interface Guidelines for DeviceNet Devices on Semiconductor Manufacturing Tools.

The GF100 Series Digital Mass Flow Controller (hereafter referred to as GF100 Series) supports the following DeviceNet objects: Identity, DeviceNet, Connection, and Assembly. In addition, support is also provided for the S-Device Supervisor, S-Analog Sensor, S-Analog Actuator, S-Single Stage Controller, and S-Gas Calibration objects. Supported objects are summarized in the following table.

Object Class	Subclass		Optional/Required	# of Instances
	Class	Inst		
Identity	-	-	Required	1
Message Router	-	-	Required	1
DeviceNet	-	-	Required	1
Connection	-	-	Required (note 1)	at least 1 I/O Polled and 1 Explicit
Acknowledge Handler Object	-	-	Conditional (note 2)	1
Assembly	-	-	Required	at least 1 Input and 1 Output
S-Device Supervisor	-	-	Required	1
S-Gas Calibration	-	01	Optional (Supported)	0 or More
S-Analog Sensor	-	01	Required (note 3)	3
S-Analog Actuator	-	-	Conditional (note 4) (Supported)	1
S-Single Stage Controller	-	-	Conditional (note 4) (Supported)	1

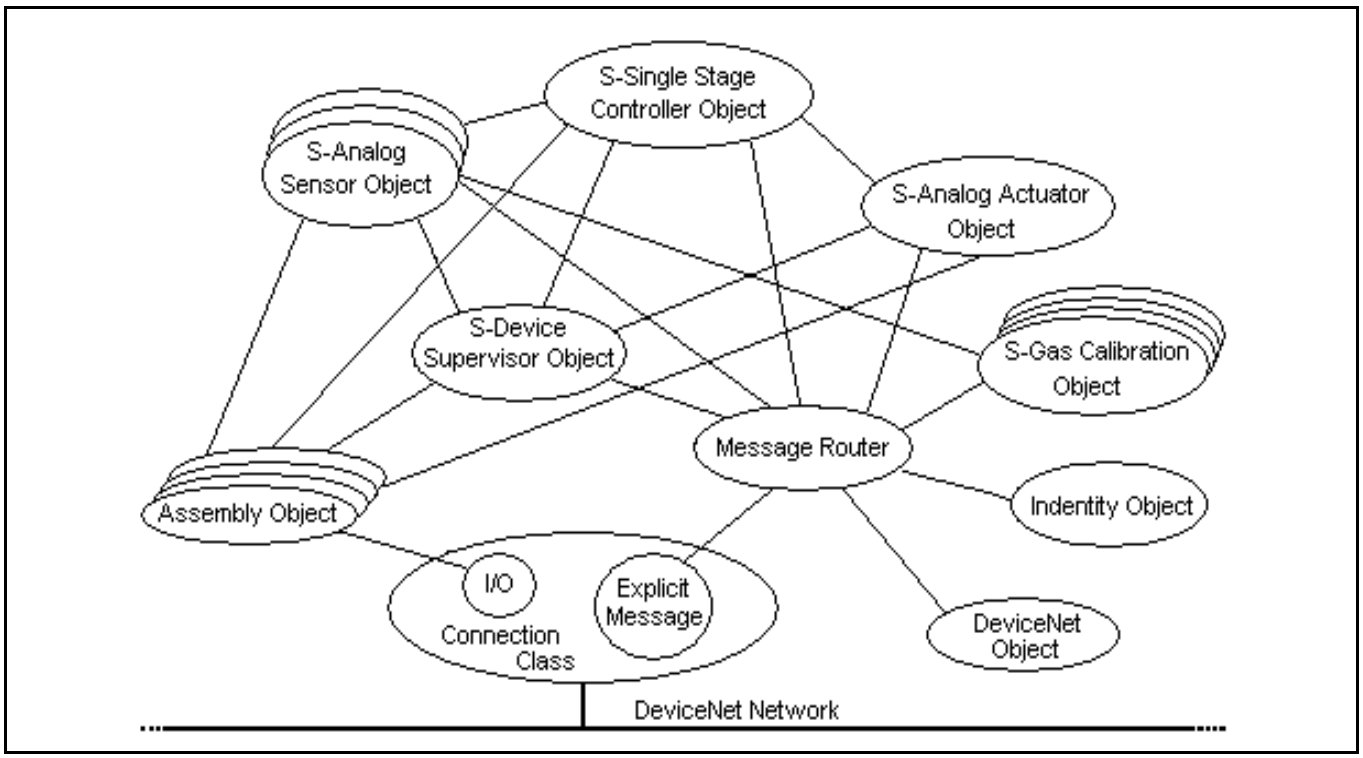
Notes:

1. *GF100 Series supports one I/O Polled and one Explicit Connection*
2. *Required for Change-of-State/Cyclic I/O connection support. (Not supported).*
3. *The GF100 Series supports three instances of the S-Analog Sensor object; instance 1 for flow, instance 2 for pressure and instance 3 for temperature. This conforms to the Enhanced MFC profile, device type = 27_{hex}.*
4. *Required for a Mass Flow Controller, a device that contains a Valve and a Controller. Not supported in a Mass Flow Meter Device (an MFC without a Valve or a Controller).*

1.1. Device Profile –Enhanced Mass Flow Controller Device (Type 0x27)

A Mass Flow Controller is a device that measures and controls the mass flow rate of gas or liquid. The MFC contains three principle components: a mass flow rate sensor, a metering valve, and a closed-loop controller. The sensor can consist of a variety of types, including thermal or pressure-based. Flow can be regulated by a variety of actuator types, including solenoid, voice coil, or piezoelectric transducer. The closed-loop controller accepts a setpoint from the host and controls the flow to that setpoint. Control is accomplished by monitoring the flow and adjusting the valve position to reduce the error between the setpoint flow value and actual flow value.

1.2. Object Model for Mass Flow Controller



Object Model for the MFC Device

1.3. How Objects Affect Behavior

Object	Effect on behavior
Identity	Supports the Reset service. Upon receipt of a *Reset* Service Request of any *Type*, the Identity Object sends a *Reset* Service Request to the S-Device Supervisor.
Message Router	No effect
DeviceNet	Configures port attributes (node address, data rate, and BOI)
Connection Class	Contains the number of logical ports into or out of the device
Acknowledge Handler	Used to manage the reception of I/O message acknowledgements. (Not used or required in the GF100 Series).
Assembly	Defines input/output and configuration data format
S-Device Supervisor	Supports the Stop, Start, Reset, Abort, Recover and Perform_Diagnostic services for ALL Application Objects in the device and consolidates the Exception Conditions and Application Objects' Status. This object behaves differently from the Identity Object in that the S-Device Supervisor object provides a single point of access to the Application Objects only; it does not effect the DeviceNet specific objects (i.e., Identity, DeviceNet, Connection, etc.).
S-Gas Calibration	Modifies the correction algorithm of the S-Analog Sensor object which includes the selection mechanism to enable an S-Gas Calibration object instance.
S-Analog Sensor	Feeds the process variable to the Single Stage Controller object
S-Single Stage Controller	Feeds the control variable to the Analog Actuator object
S-Analog Actuator	Operates the Flow Control Valve of the device

2. Identity Object (Class 0x01)

The Identity Object provides general information about the identity of a device. This object is summarized in the following tables.

2.1. Instance Attributes

Attribute ID	Need in implementation	Access Rule	Name	DeviceNet Data Type	Description of Attribute
1	Required	Get	Vendor ID	UINT	ODVA Assigned Vendor Number = 41 (0x29)
2	Required	Get	Device Type	UINT	ODVA Assigned Device Number = 39 (0x27)
3	Required	Get	Product Code	UINT	Brooks Assigned Product Number = 724
4	Required	Get	Revision	STRUCT of:	Product Revision
			Major Rev	USINT	(byte)
			Minor Rev	USINT	(byte)
5	Required	Get	Status	WORD	DeviceNet Status
6	Required	Get	Serial Number	UDINT	DeviceNet Device Serial Number
7	Required	Get	Product Name	SHORT STRING	“GF100” (1-32 characters)

2.2. Common Services

Service Code	Need in Implementation		Service Name	Description of Service
	Class	Instance		
0Ehex 14dec	Conditional	Required	Get_Attributes_Single	Returns the contents of the specified attribute.
05hex 05dec	n/a	Required	Reset	Resets the device to the **Self-Testing** state.

RESET Request Service Data Field Parameters

Parameter	Required	Data Type	Description	Semantics of Values
Type	Required	USINT	Type of Reset	0 = Power Cycle type [default if parameter omitted] 1 = Out-of-Box type

3. DeviceNet Object (Class 0x03)

The DeviceNet Object maintains configuration and status of physical attachments to DeviceNet. It also allocates and releases connection instances associated with the Predefined Master/Slave Connection Set.

3.1. Instance Attributes

Attribute ID	Need in implementation	Access Rule	Name	DeviceNet Data Type	Description of Attribute
1	Required	Set	MAC ID	USINT (byte)	Values 0-63 or "P" (Programmable MAC ID) See "3.3. Semantics."
2	Required	Set	Baud Rate	USINT (byte)	Values 0,1,2 or "P" See "3.3. Semantics."
3	Required	Set	BOI	USINT (byte)	Bus Off Interrupt
4	Required	Set	Bus-off Counter	USINT (byte)	Number of times CAN chip went to bus off state
5	Required	Get	Allocation Information	USINT (byte)	Indicates whether or not the Predefined Master/Slave Connection Set has been allocated
6	Conditional (supported)	Get	MAC ID switch changed (note 1)	BOOL	Indicates the Node ID switches have changed since last power-up or reset. 0=no change, 1=change
7	Conditional (supported)	Get	Baud Rate switch changed (note 1)	BOOL	Indicates the baud rate switch has changed since last power-up or reset. 0=no change, 1=change
8	Conditional (supported)	Get	MAC ID switch value	USINT (byte)	Actual value of the Node address switches, (0-99)
9	Conditional (supported)	Get	Baud rate switch value	USINT (byte)	Actual value of the baud rate switch, (0-9)

Note 1: When either one of these two attributes are true (=1), then the module LED will flash red to indicate the status. See "3.4. Module Status LED" for more information.

3.2. Common Services

Service Code	Need in Implementation		Service Name	Description of Service
	Class	Instance		
0Ehex 14dec	Conditional	Required	Get_Attributes_Single	Returns the contents of the specified attribute.
10hex 16dec	n/a	Required	Set_Attributes_Single	Sets the attribute to the specified value
4Bhex 75dec	n/a	Required	Allocate_Master/Slave_Connection_Set	This is the Service utilized to perform the allocation of the Predefined Master/Slave Connection Set.
4Chex 76dec	n/a	Required	Release_Master/Slave_Connection_Set	This service is used to deallocate the Predefined Master/Slave Connection Set within a Slave

3.3. Semantics

The Mac ID and Baud Rate are switch selectable. Baud Rate will be 125K, 250K, or 500K baud if the switch is set to 1,2,5 respectively. The Mac Id switch sets the unit's DeviceNet address to 0-63, according to the switch settings. Both switches may be placed in the "P" position, which selects "programmable" Mac Id or Baud Rate. If the switch is placed in the "P" position, the Mac Id or Baud Rate will assume the last valid value.

Mac ID and Baud Rate attributes are software settable ONLY when the switches are in the "P" position. Behavior related to the Mac ID and the Baud Rate attributes conforms to the requirements defined in the *Open DeviceNet Vendor Association Semiconductor Special Interest Group (SIG) Interface Guidelines Conformance Test Procedure* (Section 5.6).

3.4. Module Status LED

The module status LED indicates the status of the DFC Module.

Module Status	LED State	Description
Power Off	Off	No Power applied to device
Device Self-test	Flashing Green-Red	Device is in Self-test. The Module LED will flash Green for 250mSec, followed by RED for 250mSec. If the device passes the self-test, LED will stay Green
Device Operational	**Green**	**Device is operating normally.**
Recoverable Fault	Flashing Red	The Node (MAC ID) address or baud rate switches have changed since the last power-up/reset.
Unrecoverable Fault	**Red**	**Device has detected an unrecoverable fault.**

3.5. NET Status LED

The Network status LED indicates the status of the DFC DeviceNet Connection.

Network Status	LED State	Description
Power Off	Off	No Power applied or device is the only node on the network.
On-line Not Connected	Flashing Green	Device is Operating normally. It is on-line, but no connections have been established to the Device.
Device Operational	**Green**	**Device is operating normally.**
Connection Timeout	Flashing Red	One or more connections have timed out.
Unrecoverable Fault	**Red**	**Device cannot communicate on the network. Duplicate MacId or Bus-off condition**

4. Connection Object (Class 0x05)

The Connection Class allocates and manages internal resources associated with both I/O and Explicit Messaging connections. The Explicit and I/O Connection Objects manage the communication aspects associated with a particular application to application network relationships. The GF100 Series supports both the Explicit and Polled or I/O Connections.

4.1. Instance Attributes (Explicit Connection, instance 1

Attribute ID	Need in implementation	Access Rule	Name	DeviceNet Data Type	Description of Attribute
1	Required	Get	State	USINT (byte)	State of the object
2	Required	Get	Instance Type	USINT (byte)	Indicates either I/O or Messaging Connection
3	Required	Get	Transport Class Trigger	Byte	Defines behavior of the Connection
4	Required	Get	Produced Connection ID	UINT	Placed in CAN Identifier Field when connection transmits
5	Required	Get	Consumed Connection ID	UINT	CAN Identifier Field value that denotes message to be received
6	Required	Get	Initial Comm. Characteristics	Byte	Defines the Message Group(s) across which productions and consumptions associated with this Connection occur
7	Required	Get	Produced Connection Size	UINT	Maximum number of bytes transmitted across this connection
8	Required	Get	Consumed Connection Size	UINT	Maximum number of bytes transmitted across this connection
9	Required	Set	Expected Packet Rate	UINT	Defines timing associated with this Connection
12	Required	Get	Watchdog time-out Action	USINT (byte)	Defines how to handle Inactivity/Watchdog timeouts
13	Required	Get	Produced Path Length	UINT	Number of bytes in the produced_connection_path length

Attribute ID	Need in implementation	Access Rule	Name	DeviceNet Data Type	Description of Attribute
14	Required	Get	Produced Connection Path	Array of USINT	Specifies the Application Object(s) whose data is to be produced by this Connection Obj.
15	Required	Get	Consumed Path Length	UINT	Number of bytes in the consumed_connection_path attr.
16	Required	Get	Consumed Connection Path	Array of USINT	Specifies the Application Objs that are to receive data consumed by this Connection Obj.
17	Required	Get	Production Inhibit time	UINT	Defines minimum time between new data production. This attribute is required for I/O Client Connections.

4.2. Instance Attributes (Polled Connection, instance 2)

Attribute ID	Need in implementation	Access Rule	Name	DeviceNet Data Type	Description of Attribute
1	Required	Get	State	USINT (byte)	State of the object
2	Required	Get	Instance Type	USINT (byte)	Indicates either I/O or Messaging Connection
3	Required	Get	Transport Class Trigger	Byte	Defines behavior of the Connection, (server, class 2)
4	Required	Get	Produced Connection ID	UINT	Placed in CAN Identifier Field when connection transmits
5	Required	Get	Consumed Connection ID	UINT	CAN Identifier Field value that denotes message to be received
6	Required	Get	Initial Comm. Characteristics	Byte	Defines the Message Group(s) across which productions and consumptions associated with this Connection occur
7	Required	Get	Produced Connection Size	UINT	Maximum number of bytes transmitted across this connection

Attribute ID	Need in implementation	Access Rule	Name	DeviceNet Data Type	Description of Attribute
8	Required	Get	Consumed Connection Size	UINT	Maximum number of bytes transmitted across this connection
9	Required	Set	Expected Packet Rate	UINT	Defines timing associated with this Connection
12	Required	Get	Watchdog time-out Action	USINT (byte)	Defines how to handle Inactivity/Watchdog timeouts
13	Required	Get	Produced Path Length	UINT	Number of bytes in the produced_connection_path length
14	Required	Set *	Produced Connection Path	Array of USINT	Specifies the Application Object(s) whose data is to be produced by this Connection Obj.
15	Required	Get	Consumed Path Length	UINT	Number of bytes in the consumed_connection_path attr.
16	Required	Set*	Consumed Connection Path	Array of USINT	Specifies the Application Objs that are to receive data consumed by this Connection Obj.
17	Required	Get	Production Inhibit time	UINT	Defines minimum time between new data production. This attribute is required for I/O Client Connections.

** Produced and Consumed Connection Path attributes are settable ONLY when the I/O connection is in the "Configuring" State. These attributes must reference consistent data types at the time the I/O connection transitions to the Established State. See MFC Device Profile, Version J for more information regarding consistent data types.*

4.3. Common Services

Service Code	Need in Implementation		Service Name	Description of Service
	Class	Instance		
0Ehex 14dec	Conditional	Required	Get_Attributes_Single	Returns the contents of the specified attribute.
10hex 16dec	n/a	Required	Set_Attributes_Single	Sets the attribute to the specified value
05hex 05dec	n/a	Optional (supported)	Reset	Dependent on watchdog timeout action.

5. Assembly Object (Class 0x04)

The Assembly Object groups attributes of multiple objects into a single block of data, which can be produced and consumed over an I/O connection. Various combinations of S-Device Supervisor Attributes are grouped together to form the assemblies supported by the GF100 Series. Both the MFC and EMFC device profiles do NOT allow “mixed” integer and real assemblies to be allowed at the same time. That is, it is not allowed to produce an integer assemble and consume a floating-point assembly over a polled connection. See the EMFC Device Profile in the ODVA DeviceNet specification for more detail.

5.1. Instance Attributes

Number	Required	Supported	Type	# bytes	Name
1	N	Y	Input	2	Flow
2	Y (default)	Y	Input	3	Status and Flow
3	N	Y	Input	5	Status, Flow and Valve
4	N	Y	Input	5	Status, Flow, and Setpoint
5	N	Y	Input	7	Status, Flow, Setpoint and Valve
6	Y	Y	Input	8	Status, Flow, Setpoint, Override and Valve
7	Y (default)	Y	Output	2	Setpoint
8	Y	Y	Output	3	Override and Setpoint
9	N	Y	Input	1	Status
10	-	N	-	-	(assembly not used)
11	-	N	-	-	(assembly not used)
12	-	N	-	-	(assembly not used)
13	N	Y	Input	4	FP Flow
14	Y	Y	Input	5	Status, FP Flow
15	N	Y	Input	9	Status, FP Flow and FP Valve
16	N	Y	Input	9	Status, FP Flow, and FP Setpoint
17	N	Y	Input	13	Status, FP Flow, FP Setpoint and FP Valve
18	Y	Y	Input	14	Status, FP Flow, FP Setpoint, Override and FP Valve
19	Y	Y	Output	4	FP Setpoint
20	Y	Y	Output	5	Override and FP Setpoint

Number	Required	Supported	Type	# bytes	Name
21	Y	Y	Input	7	Status, Flow, Pressure, Temperature
22	Y	Y	Input	9	Status, Flow, Valve, Pressure, Temperature
23	Y	Y	Input	13	Status, FP Flow, FP Pressure, and FP Temperature

The number of bytes indicates how many data bytes are produced or consumed for each assembly. The “FP” abbreviation is for Floating Point, or real data. Each real data value will consist of 4 bytes of IEEE 754 single precision data.

5.2. Common Services

Service Code	Need in Implementation		Service Name	Description of Service
	Class	Instance		
0Ehex 14dec	Conditional	Required	Get_Attributes_Single	Returns the contents of the specified attribute.

5.3. Object Instances

Producing Object Instances must be one of the following: 1, 2, 3, 4, 5, 6, 9, 13, 14, 15, 16, 17, 18, 21, 22 or 23. These instances send data to the master. Consuming Object Instances must be one of the following: 7, 8, 19, or 20. These instances receive data from the master. As mentioned before, both the Produced and Consumed Paths must reference either integer OR real assemblies. The following section details each assembly and its data type. The “FP” designation will indicate a real, floating point value. Otherwise, the data will be an integer or, in the case of the “status” byte, a bit-mapped value.

5.4. I/O Assembly Object Instance Data Attribute Format

The manufacturer of a Mass Flow Controller Device must specify which Assembly instances are supported by the device. The GF100 Series supports the following assemblies.

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

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	0	Flow (low byte)							
	1	Flow (high byte)							
2	0	Status							
	1	Flow (low byte)							
	2	Flow (high byte)							
3	0	Status							
	1	Flow (low byte)							
	2	Flow (high byte)							
	3	Valve (low byte)							
	4	Valve (high byte)							
4	0	Status							
	1	Flow (low byte)							
	2	Flow (high byte)							
	3	Setpoint (low byte)							
	4	Setpoint (high byte)							
5	0	Status							
	1	Flow (low byte)							
	2	Flow (high byte)							
	3	Setpoint (low byte)							
	4	Setpoint (high byte)							
	5	Valve (low byte)							
	6	Valve (high byte)							

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
6	0	Status							
	1	Flow (low byte)							
	2	Flow (high byte)							
	3	Setpoint (low byte)							
	4	Setpoint (high byte)							
	5	Override							
	6	Valve (low byte)							
	7	Valve (high byte)							
7	0	Setpoint (low byte)							
	1	Setpoint (high byte)							
8	0	Override							
	1	Setpoint (low byte)							
	2	Setpoint (high byte)							
9	0	Status							
10	-	The assembly instances							
11	-	10, 11 and 12							
12	-	are not used in the EMFC implementation.							
13	0	FP Flow (low byte)							
	1	FP Flow							
	2	FP Flow							
	3	FP Flow (high byte)							
14	0	Status							
	1	FP Flow (low byte)							
	2	FP Flow							
	3	FP Flow							
	4	FP Flow (high byte)							

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
15	0	Status							
	1	FP Flow (low byte)							
	2	FP Flow							
	3	FP Flow							
	4	FP Flow (high byte)							
	5	FP Valve (low byte)							
	6	FP Valve							
	7	FP Valve							
	8	FP Valve (high byte)							
16	0	Status							
	1	FP Flow (low byte)							
	2	FP Flow							
	3	FP Flow							
	4	Flow (high byte)							
	5	FP Setpoint (low byte)							
	6	FP Setpoint							
	7	FP Setpoint							
	8	FP Setpoint (high byte)							

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
17	0	Status							
	1	FP Flow (low byte)							
	2	FP Flow							
	3	FP Flow							
	4	FP Flow (high byte)							
	5	FP Setpoint (low byte)							
	6	FP Setpoint							
	7	FP Setpoint							
	8	FP Setpoint (high byte)							
	9	FP Valve (low byte)							
	10	FP Valve							
	11	FP Valve							
	12	FP Valve (high byte)							
18	0	Status							
	1	FP Flow (low byte)							
	2	FP Flow							
	3	FP Flow							
	4	FP Flow (high byte)							
	5	FP Setpoint (low byte)							
	6	FP Setpoint							
	7	FP Setpoint							
	8	FP Setpoint (high byte)							
	9	Override							
	10	FP Valve (low byte)							
	11	FP Valve							
	12	FP Valve							
	13	FP Valve (high byte)							

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
19	0	FP Setpoint (low byte)							
	1	FP Setpoint							
	2	FP Setpoint							
	3	FP Setpoint (high byte)							
20	0	Override							
	1	FP Setpoint (low byte)							
	2	FP Setpoint							
	3	FP Setpoint							
	4	FP Setpoint (high byte)							
21	0	Status							
	1	Flow (low byte)							
	2	Flow (high byte)							
	3	Pressure (low byte)							
	4	Pressure (high byte)							
	5	Temperature (low byte)							
	6	Temperature (high byte)							
22	0	Status							
	1	Flow (low byte)							
	2	Flow (high byte)							
	3	Valve (low byte)							
	4	Valve (high byte)							
	5	Pressure (low byte)							
	6	Pressure (high byte)							
	7	Temperature (low byte)							
	8	Temperature (high byte)							

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
23	0	Status							
	1	FP Flow (low byte)							
	2	FP Flow							
	3	FP Flow							
	4	FP Flow (high byte)							
	5	FP Pressure (low byte)							
	6	FP Pressure							
	7	FP Pressure							
	8	FP Pressure (high byte)							
	9	FP Temperature (low byte)							
	10	FP Temperature							
	11	FP Temperature							
	12	FP Temperature (high byte)							

6. S-Device Supervisor Object (Class 0x30)

This object models the interface, functions and behavior associated with the management of application objects for devices within the "*Hierarchy of Semiconductor Equipment Devices*". Throughout this DeviceNet Standard, objects belonging to this hierarchy are identified as such by a naming convention that includes a prefix of "S-" in the object class name. This "*Hierarchy of Semiconductor Equipment Devices*" is completely defined in this object definition such that all objects belonging to this hierarchy require the existence of an S-Device Supervisor object to manage its functions and behaviors.

The S-Device Supervisor object centralizes application object state definitions and related status information, exception status indications (alarms and warnings), and defines a behavior model which is assumed by objects identified as belonging to the *Hierarchy of Semiconductor Equipment Devices*. If a reset is requested of the S-Device Supervisor object instance, it will reset this object instance as well as all of its associated application objects.

Similarly, the Identity object provides an interface to the S-Device Supervisor object. A reset request to the Identity object (of any type) causes a reset request to the S-Device Supervisor object. Further relationships are specified in the Behavior section below.

Additionally, some device attributes are defined which are required in order to specify device models such that they are compliant with the SEMI S/A Network Standard *, from which the *Hierarchy of Semiconductor Equipment Devices* is derived. Objects defined to exist within the *Hierarchy of Semiconductor Equipment Devices* are done so in order to simplify the management and description of object behavior while insuring compliance with the SEMI Standard.

NOTE: By association with this object, the Start, Stop, Reset, Abort, Recover and Perform_Diagnostic Services are inherently supported by all objects within the *Hierarchy of Semiconductor Equipment Devices*. These services are not accessible over the network for the associated object instances.

* Semiconductor Equipment and Materials International, Mountain View CA, Standard E54: *Sensor/Actuator Network Common Device Model.*

6.1. S-Device Supervisor Class Attributes

The Object Class Attribute ID 1-7 are reserved. See DeviceNet Volume II, Section 5-4.1. for more specification detail on these attributes.

Attribute ID	Need in Implementation	Access Rule	Name	DeviceNet Data Type	Description of Attribute
1 thru 7	These class attributes are either optional or conditional and are described in chapter 5 of this specification.				
97 & 98	Reserved by DeviceNet				
99	Conditional *	Get	Subclass	UINT	Identifies a subset of additional attributes, services and behaviors. The subclasses for this object are specified at the end of this object specification section.

** If the value of Subclass is 00 which identifies "no subclass", then this attribute is OPTIONAL in implementation, otherwise, this attribute is REQUIRED.*

6.2. S-Device Supervisor Instance Attributes (Object/Class 0x30)

DeviceNet reserves Attribute ID 100-199 (64_{hex}-$C7_{hex}$) for Vendor Defined Attributes. See Volume II, Section 7 for more information on Object Definitions.

Note: All required attributes are supported. Optional attributes are indicated as (Supported) or (Not Supported).

Attr ID	Need in Implementation	Access Rule	NV*	Name	DeviceNet Data Type	Description of Attribute
1	Optional (Not Supported)	Get	NV	Number of Attributes	USINT (byte)	Number of Attributes supported by the object instance
2	Optional (Not Supported)	Get	NV	Attribute List	Array of USINT (bytes)	List of attributes supported by the object instance
3	Required	Get	NV	Device Type	SHORT STRING	ASCII Text, Max. 8 Characters, See “6.3. Semantics.”
4	Required	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 this revision, this attribute must be: “E54-0997”
5	Required	Get	NV	Manufacturer’s Name	SHORT STRING	ASCII Text, Max. 20 Characters. See “6.3. Semantics.”
6	Required	Get	NV	Manufacturer’s Model Number	SHORT STRING	ASCII Text, Max. 20 Characters, Manufacturer Specified.
7	Required	Get	NV	Software Revision Level	SHORT STRING	ASCII Text, Max. 6 Characters. See “6.3. Semantics.”
8	Required	Get	NV	Hardware Revision Level	SHORT STRING	ASCII Text, Max. 6 Characters, see “Semantics” section
9	Optional (Supported)	Get	NV	Manufacturer’s Serial Number	SHORT STRING	ASCII Text, Max. 30 Characters, Manufacturer Specified. See “6.3. Semantics.”
10	Optional (Supported)	Get	NV	Device Configuration	SHORT STRING	ASCII Text, Max. 50 Characters, Manufacturer Specified. Optional additional information about the device configuration.
11	Required	Get	V	Device Status	USINT (byte)	See “6.3. Semantics.”
12	Required	Get	V	Exception Status	BYTE	See “6.3. Semantics”

Attr ID	Need in Implementation	Access Rule	NV*	Name	DeviceNet Data Type	Description of Attribute
13	Conditional based on Exception Status Bit 7 (Supported)	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	USINT (byte)	Number of Common Detail Bytes (size = 2)
				Detail	ARRAY of:	See “6.3. Semantics.”
				Detail n	BYTE	See “6.3. Semantics.”
				Device Exception Detail	STRUCT of:	
				Size	USINT (byte)	Number of Device Detail Bytes (size = 2)
				Detail	ARRAY of:	See Device Profile
				Detail n	BYTE	See Device Profile
				Manufacturer Exception Detail	STRUCT of:	
				Size	USINT (byte)	Number of Manufacturer Detail Bytes (size = 1)
				Detail n	BYTE	Manufacturer Specified

Note: The Enhanced (Next Generation) MFC Device Profile specifies two bytes of Common Detail, two bytes of Device Exception Detail, and one byte of Manufacturer Specified Detail. See “6.3. Semantics” for more information.

Attr ID	Need in implementation	Access Rule	NV*	Name	DeviceNet Data Type	Description of Attribute
14	Conditional based on Exception Status Bit 7 (Supported)	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	USINT (byte)	Number of Common Detail Bytes (size = 2)
				Detail	ARRAY of:	See “6.3. Semantics.”
				Detail n	BYTE	See “6.3. Semantics.”
				Device Exception Detail	STRUCT of:	
				Size	USINT (byte)	Number of Device Detail Bytes (size = 2)
				Detail	ARRAY of:	See Device Profile
				Detail n	BYTE	See Device Profile
				Manufacturer Exception Detail	STRUCT of:	
				Size	USINT (byte)	Number of Manufacturer Detail Bytes (size = 1)
				Detail n	BYTE	Manufacturer Specified

Note: The Enhanced (Next Generation) MFC Device Profile specifies two bytes of Common Detail, two bytes of Device Exception Detail, and one byte of Manufacturer Specified Detail. See “6.3. Semantics” for more information.

Attr ID	Need in implementation	Access Rule	NV*	Name	DeviceNet Data Type	Description of Attribute
15	Required	Set	NV	Alarm Enable	BOOL	See “6.3. Semantics.”
16	Required	Set	NV	Warning Enable	BOOL	See “6.3. Semantics.”
17	Optional (Not Supported)	Set	**	Time	DATE_AND_ TIME	The value of the device’s internal real-time clock. See “6.3. Semantics.”
18	Optional (Not Supported)	Get	NV	** Clock Behavior	USINT (byte)	0 = [default] clock always resets during power cycle 1 = clock value is stored in non-volatile memory at power down 2 = clock is battery-backed and runs without device power. 3-255 - not defined
19	Optional (Not Supported)	Get	NV	Last Maintenance Date	DATE	The date on which the device was last serviced.
20	Optional (Not Supported)	Get	NV	Next Scheduled Maintenance Date	DATE	The date on which it is recommended that the device next be serviced.
21	Optional (Not Supported)	Get	NV	Scheduled Maintenance Expiration Timer	INT	See “6.3. Semantics.”
22	Conditional – Required if Calibration Expiration is supported (Not Supported)	Set	NV	Scheduled Maintenance Expiration Warning Enable	BOOL	See “6.3. Semantics.”
23	Optional (Not Supported)	Get	NV	Run Hours	UDINT	An indication of the number of hours that the device has had power applied. It has a resolution of 1 hour. This value shall be maintained in nonvolatile memory.

Attr ID	Need in implementation	Access Rule	NV*	Name	DeviceNet Data Type	Description of Attribute
97-98	Reserved by DeviceNet					
99	Conditional ** (Supported)	Get	NV	Subclass	UINT	Identifies a subset of additional attributes, services and behaviors. The subclasses for this object are specified at the end of this object specification section.

** NV = Nonvolatile; attribute value is maintained through power cycles; V = Volatile*
*** If the value of Subclass is 00 which identifies "no subclass", then this attribute is OPTIONAL in implementation, otherwise, this attribute is REQUIRED.*

6.3. Semantics

Device Type

The Device Type attribute identifies the Specific Device Model to which the device is modeled within the Hierarchy of Semiconductor Equipment Devices. The value of this string is specified in the SEMI standard suite referenced in the introduction section of this object definition and is represented for reference in the applicable device profile where used.

Manufacturer's Name

The Manufacturer's Name attribute identifies the manufacturer of the device. It is the responsibility of the manufacturer to insure that this ASCII coded text string is sufficiently long to insure uniqueness among manufacturers.

The Device Manufacturer attribute is not guaranteed, by specification, to be unique. Therefore, it is not a substitute for the corresponding attribute of the Identity Object and should not be used for identification purposes.

Software Revision Level

This is an ASCII coded text string representing the revision of the software corresponding to the specific device identified by the Identity object and the S-Device Supervisor object.

Hardware Revision Level

This is an ASCII coded text string representing the revision of the hardware, which is identified by the Identity object and the S-Device Supervisor object. The manufacturer of the device must control this revision such that modifications to the device hardware may be tracked.

Manufacturer's Serial Number

This attribute is a string representation of the manufacturer's serial number of the device, formatted to fit the appropriate manufacturing tracking systems. This is not the same as the Identity Object's serial number, which is used to uniquely identify the device in the network environment.

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
7-50	Reserved by DeviceNet
51-99	Device Specific (None Used)
100-255	Vendor Specific (None Used)

Exception Status

A single byte attribute whose value indicates that the status of the alarms and warnings for the device. This indication may be provided in one of two methods: Basic or Expanded.

For the *Basic Method*, bit seven of the Exception Status attribute is set to zero; all exceptions are reported exclusively through communication of this Exception Status attribute. The format of bits zero through six in this mode is device specific; the format may be further specified in an appropriate device profile specification; if it is not specified, then the format of bits zero through six is equivalent to that specified for the expanded method.

For the *Expanded Method*, 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 7 set to 1
Bit	**Function**
0	ALARM/device-common*
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

** 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 here:

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 detail is defined below. The device-specific detail is defined in the appropriate Device Profile. The manufacturer defines the manufacturer-specific detail. A SIZE value of zero indicates that no 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, which represents 8 independent conditions. 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 the Expanded method reporting. Bits 0-6 of the Exception Status attribute correspond to the particular exception type.

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 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). The criteria details for each exception condition are outside the scope of this document. If a device does not support an exception detail, the corresponding bit is never set.

Common Exception Detail Attribute Values

Bit	Common Exception Detail [0]*
0	internal diagnostic exception
1	Microprocessor exception
2	EPROM exception
3	EEPROM exception (**)
4	RAM exception
5	Reserved by DeviceNet
6	Internal real-time exception
7	Reserved by DeviceNet

*** Exception Supported*

Bit	Common Exception Detail [1]*
0	power supply overcurrent
1	reserved power supply
2	power supply output voltage (**)
3	power supply input voltage
4	scheduled maintenance due
5	notify manufacturer
6	reset exception
7	reserved by DeviceNet

*** Exception Supported*

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 their respective device profiles. 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 a detailed description of this attribute, consult the appropriate specific device profile.

Note: The MFC profile has been updated and device exception detail has been modified to accommodate the extra instances in the S-Analog Sensor Object.

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 and are defined by them in their product documentation. 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 a detailed description of this attribute, consult the appropriate specific device manufacturer documentation.

Exception Detail Format Summary

Data Component	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MFC Device Exception Detail Size	0	0	0	0	0	0	1	0
MFC Device Exception Detail Byte 0	Reserved 0	Reserved 0	Valve High S-Analog Actuator	Valve Low S-Analog Actuator	Flow Control S-Single Stage Controller	Flow High S-Analog Sensor	Flow Low S-Analog Sensor	Not Reading Valid* S-Analog Sensor
MFC Device Exception Detail Byte 1	Reserved 0	Reserved 0	Not Reading Valid S-Analog Sensor temperature	Not Reading Valid S-Analog Sensor pressure	Gas Temp High S-Analog Sensor Instance 3	Gas Temp Low S-Analog Sensor Instance 3	Pressure High S-Analog Sensor Instance 2	Pressure Low S-Analog Sensor Instance 2
Manufacturer Exception Detail Size	0	0	0	0	0	0	0	1
Manufacturer Exception Detail	0	0	0	0	0	0	0	0

** Only used in the Warning Exception Detail, this bit is always = 0 in the Alarm Exception Detail.*

- **Valve High** indicates that the Actuator current has exceeded the upper alarm or warning limit.
- **Valve Low** condition never occurs, because low valve current is not an alarm or warning condition.
- **Flow Control** indicates that the closed-loop control system is not able to control the flow within the desired specification. The GF100 Series only supports the alarm condition.
- **Flow High** indicates that the sensor resistance has exceeded the upper alarm or warning limit.
- **Flow Low** indicates that the sensor resistance has fallen below the lower alarm or warning limit.

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).

Time

This optional attribute represents the value of the time and date as maintained by the device's realtime clock with a resolution of one millisecond.

The default value for the Time attribute is zero (0), corresponding to 12:00AM, January 1, 1972, as specified by DeviceNet Volume I, Appendix J.

Scheduled Maintenance Expiration Timer

This attribute, with a resolution of one hour, is used to cause a warning, which indicates that a device calibration is due. A S-Device Supervisor timer decrements this attribute once per hour while power is applied. When the attribute is no longer positive and the Scheduled Maintenance Expiration Warning Enable attribute is set to enabled, a Scheduled Maintenance Expiration Warning condition is generated. This causes the Scheduled Maintenance Due Warning bit to be set.

The attribute will not wrap; when the attribute reaches its most negative value, it no longer decrements. The attribute will continue to decrement irrespective of the state of the Scheduled Maintenance Expiration Warning Enable attribute. The value shall be maintained in nonvolatile memory.

Scheduled Maintenance Expiration Warning Enable

This Boolean attribute is used to enable (1) or disable (0) the S-Device Supervisor object's process of setting the Scheduled Maintenance Due Exception bit. When disabled, the corresponding bit is never set; and, if it was set, disabling clears it. When enabled, the bit will be set only if the corresponding condition is true.

The default-state for this Enable attribute is enabled (1).

6.4. S-Device Supervisor Common Services (Object/Class 0x30)

Service Code	Need in Implementation		Service Name	Description of Service
	Class	Instance		
0Ehex 14dec	Conditional	Required	Get_Attributes_Single	Returns the contents of the specified attribute.
10hex 16dec	n/a	Required	Set_Attributes_Single	Modifies an attribute value.
5	n/a	Required	Reset	Resets the device to the Self-Testing state.
6	n/a	Required	Start	Starts the device execution by moving the device to the Executing state. Equivalent to SEMI S/A Network Execute Service
7	n/a	Optional Supported	Stop	Moves the device to the Idle state

See the DeviceNet Communication Model and Protocol for definitions of these common services.

6.5. S-Device Supervisor Object-Specific Services

Service Code	Need in Implementation		Service Name	Description of Service
	Class	Instance		
4Bhex 75dec	n/a	Required	Abort	Moves the device to the Abort state
4Chex 76dec	n/a	Required	Recover	Moves the device out of the Abort state
4Ehex 78dec	n/a	Required	Perform_Diagnostics	Causes the device to perform a set of diagnostic routines

DS Object Service Parameter Dictionary

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

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

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

Perform_Diagnostics — Used to instruct the S-Device Supervisor object to perform a diagnostic test. A diagnostic test is either of type *common* or *device-dependent*. *Common* diagnostic tests include RAM, EPROM, non-volatile memory, and communications. *Common* diagnostic tests are implementation-specific. All detail of *device-dependent* diagnostics is outside the scope of this document.

TestID Parameter

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

Attribute Value	State
0	Standard
1-63	Reserved
64-127	Device Specific (defined in Device Profile)
128-255	Manufacturer Specific (defined by manufacturer)

Type “Standard” is specified if there is only one type of diagnostic defined or if there is more than one including a type standard. Additional diagnostic types may be defined in the device profile or by the manufacturer.

THIS PAGE WAS INTENTIONALLY LEFT BLANK

7. S-Analog Sensor Object (Class 0x31)

The S-Analog Sensor Object models the acquisition of a reading from a physical sensor in a device. Associated with an analog sensor is a reading that has been acquired and corrected with an offset and a gain coefficient, optionally, settable in the object. Additional correction algorithms may be specified by other objects identified in the device profile or as extensions specified by the manufacturer.

The GF100 Series supports three instances of the S-Analog Sensor Object. Instance 1 is associated with the flow sensor. Instance 2 is associated with the pressure sensor. Instance 3 is associated with the temperature sensor.

This object is a member of the *Hierarchy of Semiconductor Equipment Devices*. The S-Device Supervisor Object manages the behavior of the S-Analog Sensor Object. See Section 6 of this document.

7.1. S-Analog Sensor Class Attributes

The Object Class Attribute ID 1-7 are reserved. See DeviceNet Volume II, Section 5-4.1. for more specification detail on these attributes.

Attribute ID	Need in Implementation	Access Rule	Name	DeviceNet Data Type	Description of Attribute
1 thru 7	These class attributes are either optional or conditional and are described in chapter 5 of this specification.				
97 & 98	Reserved by DeviceNet				
99	Conditional * (Supported)	Get	Subclass	UINT	Identifies a subset of additional attributes, services and behaviors. The subclasses for this object are specified at the end of this object specification section.

** If the value of Subclass is 00, which identifies "no subclass", then this attribute is OPTIONAL in implementation, otherwise, this attribute is REQUIRED.*

7.2. S-Analog Sensor Instance Attributes (Object/Class 0x31)

Certain minimal implementations may support any optional “Set” attributes as “Get Only” and still be compliant with this object specification. All required attributes must be supported as specified. Not all attributes are supported for all instances.

Attr ID	Need in Implementation	Access Rule	NV*	Name	DeviceNet Data Type	Description of Attribute	Semantics of Values
1	Optional (Not Supported)	Get	NV	Number of Attributes	USINT (byte)	Number of attributes supported	The number of attributes supported by this object instance
2	Optional (Not Supported)	Get	NV	Attribute List	ARRAY OF USINT (bytes)	List of attributes supported by this object instance	List of attributes supported by this object instance
3	Optional (Supported)	See Semantics Set [1]	NV	Data Type	USINT (byte)	Determines the Data Type of *Value* and all related attributes as specified in this table.	See “7.3. Semantics.” [default] = INT INT and Real supported
4	Optional (Supported)	See Semantics Set [1]	NV	Data Units	UINT	Determines the Units context of *Value* and all related attributes.	See “7.3. Semantics.” [default] = Counts Counts or SCCM supported
5	Required	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	Required	Get	V	Value	INT or specified by *Data Type* if supported	Analog input value	The corrected, converted, calibrated final value of the sensor. Range is one of: **0-6000H (0 – 100%)** **0-7FFFH (0-100%)** See “7.3. Semantics.”
7	Required	Get	V	Status	BYTE	Alarm/Warning State of this object instance	See “7.3. Semantics.”
8	Optional (Supported)	Set	NV	Alarm Enable	BOOL	Enables the setting of the Alarm Status Bits	0 = disable [default] 1 = enable
9	Optional (Supported)	Set	NV	Warning Enable	BOOL	Enables the setting of the Warning Status Bits	0 = disable [default] 1 = enable

Attr ID	Need in Implementation	Access Rule	NV*	Name	DeviceNet Data Type	Description of Attribute	Semantics of Values
10	Optional (Supported)	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* See “7.3. Semantics.”
11	Optional (Not Supported)	Get	NV	Offset-A Data Type	USINT (byte)	Determines the Data Type of attribute *Offset-A*	See “7.3. Semantics.” [default] = INT
12	Optional (Not Supported)	Set	NV	Offset-A	INT or specified by *Offset-A Data Type* if supported	An amount added prior to *Gain* to derive *Value*	See “7.3. Semantics.” 0 = [default]
13	Required if Attribute “Gain” is other than REAL (Not Supported)	Get	NV	Gain Data Type	USINT (byte)	Determines the Data Type of attribute *Gain*	See “7.3. Semantics.” [default] = REAL
14	Optional (Not Supported)	Set	NV	Gain	REAL or specified by *Gain Data Type* if supported	An amount scaled to derive *Value*	See “7.3. Semantics.” 1.0 = [default]
15	Required if Attribute “Gain” is other than REAL (Not Supported)	Get	NV	Unity Gain Reference	REAL or specified by *Gain Data Type* if supported	Specifies the value of the *Gain* attribute equivalent to a gain of 1.0	Used for normalizing the *Gain* attribute. [default] = 1.0 e.g., for an UINT type Gain, a Unity Gain Reference may be 10000, allowing a gain of 0.0001 to 6.5535.
16	Optional (Not Supported)	Set	NV	Offset-B	INT or specified by *Data Type* if supported	An amount added to derive *Value*	See “7.3. Semantics.” 0 = [default]

Attr ID	Need in Implementation	Access Rule	NV*	Name	DeviceNet Data Type	Description of Attribute	Semantics of Values
17	Optional (Not Supported)	Set	NV	Alarm Trip Point High	INT or specified by *Data Type* if supported	Determines the Value above which an Alarm Condition will occur	See “7.3. Semantics.” [default] = Maximum value for its data type.
18	Optional (Not Supported)	Set	NV	Alarm Trip Point Low	INT or specified by *Data Type* if supported	Determines the Value below which an Alarm Condition will occur	See “7.3. Semantics.” [default] = Minimum value for its data type.
19	Optional (Not Supported)	Set	NV	Alarm Hysteresis	INT or specified by *Data Type* if supported	Determines the amount by which the *Value* must recover to clear an Alarm Condition	See “7.3. Semantics.” [default] = 0
20	Optional (Not Supported)	Set	NV	Alarm Settling Time	UINT	Determines the time that the *Value* must exceed the Trip Point before the exception condition is generated.	Time in milliseconds See “7.3. Semantics.” [default] = 0
21	Optional (Not Supported)	Set	NV	Warning Trip Point High	INT or specified by *Data Type* if supported	Determines the *Value* above which a Warning Condition will occur	See “7.3. Semantics.” [default] = Maximum value for its data type.
22	Optional (Not Supported)	Set	NV	Warning Trip Point Low	INT or specified by *Data Type* if supported	Determines the *Value* below which a Warning Condition will occur	See “7.3. Semantics.” [default] = Minimum value for its data type.
23	Optional (Not Supported)	Set	NV	Warning Hysteresis	INT or specified by *Data Type* if supported	Determines the amount by which the *Value* must recover to clear a Warning Condition	See “7.3. Semantics.” [default] = 0

Attr ID	Need in Implementation	Access Rule	NV*	Name	DeviceNet Data Type	Description of Attribute	Semantics of Values
24	Optional (Not Supported)	Set	NV	Warning Settling Time	UINT	Determines the time that the *Value* must exceed the Trip Point before the exception condition is generated.	Time in milliseconds See “7.3. Semantics.” [default] = 0
25	Optional (Not Supported)	Set	NV	Safe State	USINT (byte)	Specifies the behavior for the *Value* for states other than Execute	See “7.3. Semantics.” [default] = 0
26	Optional (Not Supported)	Set	NV	Safe Value	INT or specified by *Data Type* if supported	The Value to be used for Safe State = Safe Value	See “7.3. Semantics.” [default] = 0
27	Optional (Not Supported)	Set	NV	Autozero Enable	BOOL	Enables the Autozero	See “7.3. Semantics.” 0 = disable [default] 1 = enable
28	Optional (Supported)	Get	V	Autozero Status	BOOL	Indicates the status of the automatic nulling	See “7.3. Semantics.” [default] = 0
29	Optional (Not Supported)	Set	NV	Autorange Enable	BOOL	Enables the automatic range switching	See “7.3. Semantics.” 0 = disable [default] 1 = enable
30	Optional (Not Supported)	Get	V	Range Multiplier	REAL	Indicates the current range multiplier	See “7.3. Semantics.” [default] = 1.0
31	Optional (Not Supported)	Set	NV	Averaging Time	UINT	Specifies the time over which analog samples are averaged.	Time in Milliseconds of a moving-window average. 0 = disable averaging [default] Values less than the sample rate of the device also disable averaging.

Attr ID	Need in Implementation	Access Rule	NV*	Name	DeviceNet Data Type	Description of Attribute	Semantics of Values
32	Optional (Not Supported)	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	Optional (Not Supported)	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*
34	Optional (Not Supported)	Set	NV	Produce Trigger Delta	INT or specified by *Data Type* if supported	The amount by which *Value* must change before a Change of State Production is triggered	0 = Disabled [default] See “7.3. Semantics.”
35	Conditional [2] (Supported)	Set	NV	Gas Calibration Object Instance	UINT	Indicates which Gas Calibration object instance is active for this object	0 = Disabled [default] See “7.3. Semantics.”
97-98	Reserved by DeviceNet						
99	Conditional [3] (Supported)	Get	NV	Subclass	UINT	Identifies a subset of additional attributes, services and behaviors. The subclasses for this object are specified at the end of this object specification section.	0 = No subclass n = subclass as defined herein
110	Optional (Supported)	Get	NV	Full Scale	Struct: real, uint	Full scale amount (real) and data units (uint)	Default = 0, 0.

[1] Data Type and Data Units are ONLY settable under certain conditions (see “7.3. Semantics”).

[2] Attribute is settable; however, it should only be set while in the Idle state (see “7.3. Semantics”).

[3] If the value of Subclass is 00, then this attribute is OPTIONAL in implementation, otherwise, this attribute is REQUIRED.

7.3. Semantics

Data Type

All Data Type attributes, including *Data Type, Offset-A Data Type* and *Gain Data Type*, use the enumerated values specified in DeviceNet Vol. I, Appendix J-6.1.

The *Data Type* attribute is settable only in the *Idle State* and only if no attribute belonging to the object instance is the endpoint of an I/O connection in the *Established State*.

The *Data Type* attribute may change automatically based upon established I/O connections. See “7.6. Behavior” for more information on this mechanism.

Data Units

Specifies the context of *Value* and related attributes (such as, offset and trip points) for this object instance. See Appendix K for a list of values. A request to set attribute to an unsupported value will return an error response.

The *Data Units* attribute is settable only in the *Idle State*.

Value, Offset (A and B) and Gain

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. The ***Offset-A, Offset-B*** and ***Gain*** attributes are applied to the sensor reading as specified by the following formula:

Value = Gain • (Sensor Reading + Offset-A) + Offset-B

Typically, the *Offset-A* or *Offset-B* attributes are modified by the Zero-Adjust service and the *Gain* attribute is modified by the Gain_Adjust services; particularly, when the device utilizes a non-linear conversion algorithm. However, support of these services is not required.

See “7.6. Behavior”.

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

Trip Points, Hysteresis and Settling Time

Trip Point High is the level above which the *Value* attribute will cause an Alarm or Warning exception condition.

Trip Point Low is the level below which the *Value* attribute will cause an Alarm or Warning exception condition.

A Hysteresis value specifies the amount by which the *Value* attribute must transition in order to clear an Alarm or Warning condition. For example: A Trip Point High value of 100 and a hysteresis value of 2 will result in an exception condition being set when the *Value* is above 100 and cleared when the *Value* drops below 98. Similarly, A Trip Point Low value of 100 and a hysteresis value of 2 will result in an exception condition being set when the *Value* is below 100 and cleared when the *Value* increases above 102.

The Settling Time determines the amount of time that the *Value* attribute must exceed the Trip Point before the exception condition is generated. The Settling Time also applies to the clearing of the condition.

Safe State

This attribute specifies what value will be held in *Value* for states other than Executing. See the S-Device Supervisor object definition in Section 6 for a description of object states. The purpose of this mechanism is to allow other devices, that 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
4-50	Reserved
51-99	Device Specific
100-255	Vendor Specific

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.

Autozero Enable and Autozero Status

When the autozero is enabled, the device will automatically invoke a Zero_Adjust service request (no parameter) contingent upon a set of conditions specified by the manufacturer. These conditions may be determined by the value of an attribute (e.g., setpoint) or some other mechanism defined by the manufacturer. See Zero_Adjust service.

The GF100 Series uses the Autozero Status attribute to convey the status of the Zero-Adjust Service operation. If the device receives an explicit message from the host to perform a Zero-Adjust Service, the GF100 Series will perform the service and set the Autozero Status to 1 for the duration of the service. After the Zero-Adjust service has completed, the Autozero Status will be set to zero. The MFC Device Profile appears to indicate that the Autozero Status attribute is only to be used for an internally triggered Zero-Adjust Service; however, the GF100 Series uses the Autozero Status to convey the status of the Zero-Adjust Service, no matter how the service was triggered.

Autorange Enable and Range Multiplier

When the autorange is enabled, the device will automatically switch full scale range based on a set of conditions specified by the manufacturer. The Range Multiplier indicates the range scale.

An example of how Autorange may work is: when the *Value* is less than 9% with a *Range Multiplier* of 1.0, the *Range Multiplier* switches to 10.0 (the *Value* then reads 90% of the 10X range). When the *Value* then reaches 100% with a *Range Multiplier* of 10.0, the *Range Multiplier* returns to 1.0 (the *Value* then reads 10% of the 1X range).

Produce Trigger Delta

This attribute is used in conjunction with the "Change of State" production trigger type. Upon transition of the associated connection object instance (any Change of State connection pointing to the S-Analog Sensor object *Value* attribute) to the established state, a production is immediately triggered and this reported *Value* is stored internally for the determination of the next production trigger. When the *Value* changes by an amount of at least the *Produce Trigger Delta* (i.e., the *Value* as compared to the internally stored previously produced Value), a new production is triggered, and this reported *Value* becomes the new internally stored Value for the determination of the next production trigger.

Gas Calibration Object Instance

This attribute is used to select an instance of the S-Gas Calibration object. The selected S-Gas Calibration object instance provides the data with which an S-Analog Sensor object instance enacts the appropriate calibration algorithm for a given gas type.

A Set_Attribute_Single request, specifying a value not supported, will return an “invalid attribute value” error response. A list of acceptable values for this attribute is derived from a class level service request to the S-Gas Calibration object.

Conditionally Required: If a device profile specifies an S-Gas Calibration object relationship for an S-Analog Sensor object instance, then this attribute is required.

See the S-Gas Calibration object definition for more information.

Caution: Care should be taken when changing the gas instance. The MFC profile allows the user to change the gas instance at any time; however, the attribute should only be changed when the device is Idle. Unpredictable results may occur if the gas instance is changed while the MFC is in the Execute State.

7.4. S-Analog Sensor Common Services

The S-Analog Sensor Object provides the following Common Services:

Service Code	Need in Implementation		Service Name	Description of Service
	Class	Instance		
0Ehex 14dec	Conditional *	Required	Get_Attribute_Single	Returns the contents of the specified attribute.
10hex 16dec	n/a	Required	Set_Attribute_Single	Modifies an attribute value.
*The Get_Attribute_Single service is REQUIRED if any attributes are implemented.				

See the DeviceNet Communication Model and Protocol for definitions of these common services.

7.5. S-Analog Sensor Object–Specific Services

Service Code	Need in Implementation		Service Name	Description of Service
	Class	Instance		
4Bhex 75dec	N/a	Optional (Supported)	Zero_Adjust	Causes the device to modify attribute *Offset-A* and/or *Offset-B* such that attribute *Value* equals the Target Value sent with the request.
4Chex 76dec	N/a	Optional (Not Supported)	Gain_Adjust	Causes the device to modify attribute *Gain*, such that attribute *Value*, equals the Target Value sent with the request.

The Zero_Adjust and Gain_Adjust services are used to cause the S-Analog Sensor Object device to modify its *Offset-A* and/or *Offset-B* and *Gain* attribute values 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 these services. 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.

7.5.1. Zero_Adjust Request Service Data Field Parameters

Parameter	Required	Data Type	Description	Semantics of Values
Target Value	Optional (Supported)	Specified by the value of attribute *Data Type*	The target value for the zero calibration	The value to which the *Value* attribute will be set. If not specified, the default value of zero is used.

7.5.2. Gain_Adjust Request Service Data Field Parameters

Parameter	Required	Data Type	Description	Semantics of Values
Target Value	Required	Specified by the value of attribute *Data Type*	The target value for the gain calibration	The value to which the *Value* attribute will be set.

Note: Support of the Zero Adjust Service - target Value must be zero. To invoke Zero Adjust, the user should put the MFC in a steady-state condition with zero flow, prior to sending the Service.

ADD DETAILS FOR PRESSURE ZERO!

If Data Type is Integer:

ServiceCode=4BH, Class=31H, Instance=1, Data or Target Value = (00 00).

If Data Type is Real:

ServiceCode=4BH, Class=31H, Instance=1, Data or Target Value = (00 00 00 00).

7.6. Behavior

The S-Device Supervisor Object manages the behavior of the S-Analog Sensor Object. See Section 6 of this document.

An S-Analog Sensor object instance acquires a reading from a physical sensor, as identified by the application of the object, and applies an algorithm to modify the reading into the appropriate *Data Type* and *Data Units*. Optionally, additional corrective algorithms are applied to further correct for various calibration effects. These additional algorithms are specified in other objects, as identified in the device profile, or as extensions, specified by the manufacturer.

All Full Scale, Trip Point, Overrange and Underrange calculations, as specified above, utilize the *Value* attribute.

Data Type

If the implementation of this object specifies more than one valid Data Type value, in the device profile or by vendor, then 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. This configuration will then remain in effect for this object instance, even after all I/O connections are lost. For devices that support only one Data Type, this behavior is not supported.

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 profile 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.

GF100 Series Implementation

Data Type values supported are Integer (0xC3) and Real (0xCA). Data Units supported are Counts (0x1001) and SCCM (0x1400). Both Data Type and Data Units attributes are settable. The supported combinations of Data Type and Data Units on Intelli*Flow*™ are Integer-Counts (default), Real-SCCM, Integer-SCCM, and Real-Counts. The full-scale range for indicated flow is determined by the full-scale attribute (31H, 1,10).

7.7. S-Analog Sensor Object Instance Subclass 01

The following specification applies to a subclass of this object for application in Mass Flow Controller devices.

7.7.1. Subclass 01 Instance Attributes

The following Instance Attributes are specified for this object subclass.

Attr ID	Need in Implementation	Access Rule	NV*	Name	DeviceNet Data Type	Description of Attribute	Semantics of Values
95	Optional (Supported)	Get	NV	Flow Totalizer	ULINT	Total gas flowed through the device since this value was last set to zero	Units are Standard CCs. See “7.7.3. Subclass 01 Behavior.” Default = 0.
96	Optional (Not Supported)	Set	NV	Flow Hours	UDINT	Total time device has been powered and flowing gas since this value was last set to zero	Resolution is one hour See “7.7.3. Subclass 01 Behavior.” Default = 0.

** NV = Nonvolatile; attribute value is maintained through power cycles; V = Volatile*

7.7.2. Subclass 01 Services

There are no additions or restrictions to the Object Services for this object subclass.

7.7.3. Subclass 01 Behavior

Flow Totalizer and Flow Hours Process

The factory configured out-of-box values for the Flow Totalizer and Flow Hours attributes are both zero. The attributes are only modifiable with *set_attribute_single* service requests; they are not altered by the *Reset* service, including power-cycle, of either the Identity or the S-Device Supervisor objects.

The Flow Totalizer attribute is incremented, at a rate of once every cubic centimeter of gas flow, by the S-Analog Sensor object instance to reflect the amount of gas that has flowed through the device. Upon reaching its maximum value, the Flow Totalizer value is no longer incremented and remains at its maximum value.

The Flow Hours attribute is incremented, at a rate of once every hour, by the S-Analog Sensor object instance to reflect the amount of time that gas has flowed through the device. This condition is determined by the *Value* attribute being greater than 0.5% of full scale. Upon reaching its maximum value, the Flow Hours value is no longer incremented and remains at its maximum value.

THIS PAGE WAS INTENTIONALLY LEFT BLANK

8. S-Analog Actuator Object (Class 0x32)

The S-Analog Actuator Object models the interface to a physical actuator in a device. Associated with an analog actuator is a value, which is corrected with an offset and a gain coefficient, optionally settable in the object before it is output to the physical actuator. Manufacturers may specify additional correction algorithms as extensions to this object.

Additionally, the S-Analog Actuator Object provides two sets of trip-point definitions. The behavior associated with these trip points is described in sections below.

This object is a member of the *Hierarchy of Semiconductor Equipment Devices*. The S-Device Supervisor manages the behavior of the S-Analog Actuator Object. See Section 6 of this document.

8.1. S-Analog Actuator Class Attributes

The Object Class Attribute ID 1-7 are reserved. See DeviceNet Volume II, Section 5-4.1. for more specification detail on these attributes.

Attribute ID	Need in implementation	Access Rule	Name	DeviceNet Data Type	Description of Attribute
1 thru 7	These class attributes are either optional or conditional and are described in chapter 5 of this specification.				
97 & 98	Reserved by DeviceNet				
99	Conditional * (Supported)	Get	Subclass	UINT	Identifies a subset of additional attributes, services and behaviors. The subclasses for this object are specified at the end of this object specification section.

** If the value of Subclass is 00, which identifies "no subclass", then this attribute is OPTIONAL in implementation, otherwise, this attribute is REQUIRED.*

8.2. S-Analog Actuator Instance Attributes

Certain minimal implementations may support any optional “Set” attributes as “Get” only and still be compliant with this object specification. All required attributes must be supported as specified.

Attr ID	Need in Implementation	Access Rule	NV*	Name	DeviceNet Data Type	Description of Attribute	Semantics of Values
1	Optional (Not Supported)	Get	NV	Number of Attributes	USINT (byte)	Number of supported attributes	The number of attributes supported by this object instance
2	Optional (Not Supported)	Get	NV	Attribute List	ARRAY OF USINT (byte)	List of supported attribute	List of attributes supported by this object instance
3	Optional (Supported)	See “8.3. Semantics” Set [1]	NV	Data Type	USINT (byte)	Determines the Data Type of *Value* and all related attributes as specified in this table.	See “8.3. Semantics.” [default] = INT INT or Real supported
4	Optional (Supported)	See “8.3. Semantics” Set [1]	NV	Data Units	UINT	Determines the context of *Value*	See “8.3. Semantics.” [default] = Counts Counts or Percent supported
5	Required	Set	V	Override	USINT (byte)	Specifies an override for the physical actuator. For values other than zero (normal control), the *Value* attribute is ignored.	0 = normal [default] See “8.3. Semantics.”n
6	Required	Set	V	Value	INT or specified by *Data Type* if supported	Analog output value	The uncorrected value. see Semantics section [default] = 0
7	Required	Get	V	Status	BYTE	Alarm and Warning State of this object instance	See “8.3. Semantics.” [default] = 0
8	Optional (Supported)	Set	NV	Alarm Enable	BOOL	Enables the setting of the Alarm Bit	0 = disable [default] 1 = enable
9	Optional (Supported)	Set	NV	Warning Enable	BOOL	Enables the setting of the Warning Bit	0 = disable [default] 1 = enable

Attr ID	Need in Implementation	Access Rule	NV*	Name	DeviceNet Data Type	Description of Attribute	Semantics of Values
10	Optional (Not Supported)	Set	NV	Offset	INT or specified by *Data Type* if supported	An amount to be added to *Value* prior to the application of gain	See “8.3. Semantics.” 0 = [default]
11	Optional (Not Supported)	Set	NV	Bias	INT or specified by *Data Type* if supported	An amount to be added to *Value* prior to the application of gain	See “8.3. Semantics.” 0 = [default]
12	Required if Attribute “Gain” is other than REAL (Not Supported)	Get	NV	Gain Data Type	USINT (byte)	Determines the Data Type of attribute *Gain*	See “8.3. Semantics.” [default] = REAL
13	Optional (Not Supported)	Set	NV	Gain	REAL or specified by *Gain Data Type* if supported	An amount by which *Value* is scaled prior to driving the physical actuator	See “8.3. Semantics.” 1.0 = [default]
14	Required if Attribute 12 is other than REAL (Not Supported)	Get	NV	Unity Gain Reference	REAL or specified by *Gain Data Type* if supported	Specifies the value of the *Gain* attribute equivalent to a gain of 1.0	Used for normalizing the *Gain* attribute. See “8.3. Semantics.” [default] = 1.0
15	Optional (Not Supported)	Set	NV	Alarm Trip Point High	INT or specified by *Data Type* if supported	Determines the Value above which an Alarm Condition will occur	See “8.3. Semantics.” [default] = Maximum value for its data type.
16	Optional (Not Supported)	Set	NV	Alarm Trip Point Low	INT or specified by *Data Type* if supported	Determines the Value below which an Alarm Condition will occur	See “8.3. Semantics.” [default] = Minimum value for its data type.
17	Optional (Not Supported)	Set	NV	Alarm Hysteresis	INT or specified by *Data Type* if supported	Determines the amount by which the Value must recover to clear an Alarm Condition	See “8.3. Semantics.” [default] = 0
18	Optional (Not Supported)	Set	NV	Warning Trip Point High	INT or specified by *Data Type* if supported	Determines the Value above which a Warning Condition will occur	See “8.3. Semantics.” [default] = Maximum value for its data type.

Attr ID	Need in Implementation	Access Rule	NV*	Name	DeviceNet Data Type	Description of Attribute	Semantics of Values
19	Optional (Not Supported)	Set	NV	Warning Trip Point Low	INT or specified by *Data Type* if supported	Determines the Value below which a Warning Condition will occur	See “8.3. Semantics.” [default] = Minimum value for its data type.
20	Optional (Not Supported)	Set	NV	Warning Hysteresis	INT or specified by *Data Type* if supported	Determines the amount by which the Value must recover to clear a Warning Condition	See “8.3. Semantics.” [default] = 0
21	Optional (Supported)	Set	NV	Safe State	USINT (byte)	Specifies the behavior of the physical actuator for states other than Execute	See “8.3. Semantics.” 0 = [default]
22	Optional (Not Supported)	Set	NV	Safe Value	INT or specified by *Data Type* if supported	The Value to be used for Safe State = Safe Value	See “8.3. Semantics.” 0 = [default]
97-98	Reserved by DeviceNet						
99	Conditional [2] (Supported)	Get	NV	Subclass	UINT	Identifies a subset of additional attributes, services and behaviors. The subclasses for this object are specified at the end of this object specification section.	0 = No subclass n = subclass as defined herein

NV = Nonvolatile; value is maintained through power cycle. V = Volatile

[1] Data Type and Data Units Attribute are settable ONLY under certain conditions (see “8.3. Semantics.”)

[2] If the value of Subclass is 00, then this attribute is OPTIONAL in implementation, otherwise, this attribute is REQUIRED.

8.3. Semantics

Data Type

All Data Type attributes, including *Data Type* and *Gain Data Type*, use the enumerated values specified in DeviceNet Vol. I, Appendix J-6.1.

The *Data Type* attribute is settable only in the *Idle State* and only if no attribute belonging to the object instance is the endpoint of an I/O connection in the *Established State*.

The *Data Type* attribute may change automatically based upon established I/O connections. See “8.6. S-Analog Actuator Behavior” for more information on this mechanism.

Data Units

Specifies the context of *Value* and related attributes (such as, offset and trip points) for this object instance. See Appendix K for a list of values. A request to set attribute to an unsupported value will return an error response.

The *Data Units* attribute is settable only in the *Idle State*.

Value, Offset, Gain, Bias and Unity Gain Reference

The *Offset, Gain* and *Bias* attributes are applied to the *Value* attribute to derive the actual signal, which drives the physical actuator. The gain is normalized using the *Unity Gain Reference* attribute value. (e.g., for an UINT type *Gain*, a *Unity Gain Reference* value may be 10000, allowing an effective gain of 0.0001 to 6.5535.)

The following formula applies:

$$\text{physical actuator drive signal} = \textbf{\textit{Gain}}_{N} \bullet (\textbf{\textit{Value}} + \textbf{\textit{Offset}}) + \textbf{\textit{Bias}}$$

where:

$$\textbf{\textit{Gain}}_{N} = \textbf{\textit{Gain}}/\textbf{\textit{Unity Gain Reference}}$$

There may be additional nonlinear conversions applied to the drive signal as specified by the manufacturer.

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

Trip Points and Hysteresis

Trip Point High is the level above which the *Value* attribute will cause an Alarm or Warning exception condition.

Trip Point Low is the level below which the *Value* attribute will cause an Alarm or Warning exception condition.

A *Hysteresis* value specifies the amount by which the *Value* attribute must transition in order to clear an Alarm or Warning condition.

For example: A *Trip Point High* value of 90 and a *Hysteresis* value of 2 will result in an exception condition being set when the *Value* is above 90 and cleared when the *Value* drops below 88. Similarly, A *Trip Point Low* value of 90 and a *Hysteresis* value of 2 will result in an exception condition being set when the *Value* is below 90 and cleared when the *Value* increases above 92.

Override

This attribute is used to override the function of the *Value* attribute in driving the physical actuator. The primary application of this feature is in devices where the object instance is being driven by another object such as an S-Single Stage Controller object instance.

The *Safe State* attribute provides a mechanism for override depending upon object state and will take precedents over this. That is, if an object instance implements the *Safe State* attribute and related behavior, then this *Override* attribute and related behavior will only function in the Executing State.

Attribute Value	State
0	Normal (Supported)
1	Off/Closed (Supported)
2	On/Open (Supported)
3	Hold
4	Safe State
5-63	Reserved
64-127	Device Specific
128-255	Vendor Specific

Safe State

This attribute specifies the behavior of the drive to the physical actuator for states other than Executing. See the S-Device Supervisor object definition in Section 6-48 for a description of object states. The following values are defined:

Attribute Value	State
0	Zero/Off/Closed
1	Full Scale/On/Open
2	Hold Last Value
3	Use Safe Value
4-63	Reserved
64-127	Device Specific
128-255	Vendor Specific

Intelli*Flow* supports the Safe State Attribute as a Get Only attribute that returns a value of zero or “closed”.

Safe Value

For *Safe State* set to “Use Safe Value”, this attribute holds the value to which the actuator will be driven for object instance states other than Executing. Specifically, this attribute value will become the value of the *Value* attribute. Therefore, the correction formula specified above applies.

8.4. S-Analog Actuator Common Services

The S-Analog Actuator Object provides the following Common Services:

Service Code	Need in Implementation		Service Name	Description of Service
	Class	Instance		
0Ehex 14dec	Conditional*	Required	Get_Attribute_Single	Returns the contents of the specified attribute.
10hex 16dec	n/a	Required	Set_Attribute_Single	Modifies an attribute value.
*The Get_Attribute_Single service is REQUIRED if any attributes are implemented.				

See the DeviceNet Communication Model and Protocol for definitions of these common services.

8.5. S-Analog Actuator Object–Specific Services

The S-Analog Actuator Object provides no Object–Specific services.

8.6. S-Analog Actuator Behavior

The S-Device Supervisor Object manages the behavior of the S-Analog Actuator Object. See Section 6.

An S-Analog Actuator object instance modifies the *Value* by applying the formula specified above with the associated attribute values. *Value* is specified as *Data Type* and *Data Units*. Optionally, additional corrective algorithms are applied to further correct for various calibration effects. These additional algorithms are specified in other objects, as identified in the device profile, or as extensions, specified by the manufacturer.

All Trip Point calculations, as specified above, utilize the *Value* attribute before the application of *Offset* and *Gain*.

Data Type

If the implementation of this object specifies more than one valid Data Type value, in the device profile or by vendor, then 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. This configuration will then remain in effect for this object instance, even after all I/O connections are lost. For devices that support only one Data Type, this behavior is not supported.

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 profile specifies an instance of the S-Analog Actuator 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.

GF100 Series Implementation

Data Type values supported are Integer (0xC3) and Real (0xCA). Data Units supported are Counts (0x1001) and Percent (0x1007). Data Type and Data Units attributes are settable. The supported combinations of Data Type and Data Units on Intelliflow™ are Integer-Counts (default), Real-Percent, Integer-Percent, and Real-Counts. The Real-Percent values range from 0.0 to 100.0, where the value represents percent of full-scale that the actuator is being driven. Integer-Percent value range is 0 to 100 integer. Integer-Counts value range is 0 to 0x7FFF. Typical count range is 0 to 24576, (0x6000).

THIS PAGE WAS INTENTIONALLY LEFT BLANK

9. S-Single Stage Controller Object (Class 0x33)

The S-Single Stage Controller Object models a closed-loop control system within a device. Associated with a single stage controller is a Process Variable, a Setpoint and a Control Variable. As normally described by *classic control theory*, a closed-loop controller will drive the Control Variable in order to affect the value of the Process Variable such that it is made to equal the Setpoint. See the Semantics section, below, for more information regarding these variable definitions. Manufacturers may specify additional correction algorithms as extensions to this object.

This object is a member of the *Hierarchy of Semiconductor Equipment Devices*. The S-Device Supervisor Object manages the behavior of the S-Single Stage Controller Object. See Section 6.

9.1. S-Single Stage Controller Class Attributes

The Object Class Attribute ID 1-7 are reserved. See DeviceNet Volume II, Section 5-4.1. for more specification detail on these attributes.

Attribute ID	Need in implementation	Access Rule	Name	DeviceNet Data Type	Description of Attribute
1 thru 7	These class attributes are either optional or conditional and are described in chapter 5 of this specification.				
97 & 98	Reserved by DeviceNet				
99	Conditional * (Supported)	Get	Subclass	UINT	Identifies a subset of additional attributes, services and behaviors. The subclasses for this object are specified at the end of this object specification section.

** If the value of Subclass is 00, which identifies "no subclass", then this attribute is OPTIONAL in implementation, otherwise, this attribute is REQUIRED.*

9.2. S-Single Stage Controller Instance Attributes

Certain minimal implementations may support any optional “Set” attributes as “Get” only and still be compliant with this object specification. All required attributes must be supported as specified.

Attr ID	Need in Implementation	Access Rule	NV*	Name	DeviceNet Data Type	Description of Attribute	Semantics of Values
1	Optional (Not Supported)	Get	NV	Number of Attributes	USINT (byte)	Number of supported attributes	Number of attributes supported in this object instance
2	Optional (Not Supported)	Get	NV	Attribute List	ARRAY OF USINT (byte)	Attribute List	List of attributes supported in this object instance
3	Optional (Supported)	See “9.3. Semantics.” Set [2]	NV	Data Type	USINT (byte)	Determines the Data Type of *Setpoint*, *Process Variable* and related attributes	See “9.3. Semantics.” [default] = INT INT and Real supported
4	Optional (Supported)	See “9.3. Semantics” Set[2]	NV	Data Units	UINT	Determines the context of the Process related variables such as Setpoint and Process Variable	See Appendix K. [default] = Counts Counts and SCCM supported
5	Optional (Not Supported)	Set	NV	Control Mode	USINT (byte)	Specifies the operational mode of the controller	See “9.3. Semantics.” [default] = Normal (0)
6	Required	Set	V	Setpoint	INT or specified by *Data Type* if supported	The setpoint to which the process variable will be controlled	See “9.3. Semantics.” See ”9.6. Behavior.” 0 = [default] Range is one of: 0-6000H (0 – 100%) 0-7FFFH (0-100%)
7	Conditional [2] (Not Supported)	Set	V	Process Variable	INT or specified by *Data Type* if supported	The measured process parameter	The device profile must specify the data connection for this attribute. It may be internally linked to a sensor. See Semantics section. 0 = [default]

Attr ID	Need in Implementation	Access Rule	NV*	Name	DeviceNet Data Type	Description of Attribute	Semantics of Values
8	Optional (Not Supported)	Get	NV	CV Data Type	USINT (byte)	Determines the Data Type of *Control Variable*	See “9.3. Semantics.” [default] = INT
9	Conditional ** (Not Supported?)	Get	V	Control Variable	INT or specified by CV *Data Type* if supported	The drive signal output of this object. The algorithm by which this attribute is calculated is manufacturer specific.	The device profile must specify the data connection for this attribute. It may be internally linked to an actuator. [default] = 0 See “9.3. Semantics.”
10	Required	Get	V	Status	BYTE	Alarm and Warning State of this object instance	See “9.3. Semantics.” [default] = 0
11	Optional (Supported)	Set	NV	Alarm Enable	BOOL	Enables the setting of the Alarm Status Bit	0 = disable [default] 1 = enable
12	Optional (Supported)	Set	NV	Warning Enable	BOOL	Enables the setting of the Warning Status Bit	0 = disable [default] 1 = enable
13	Optional (Not Supported)	Set	NV	Alarm Settling Time	UINT	Number of Milliseconds allowed for the control-loop to settle to within the error band	See “9.6. Behavior.” [default] = 0
14	Optional (Not Supported)	Set	NV	Alarm Error Band	INT or specified by *Data Type* if supported	The amount by which the *Setpoint* must equal the *Process Variable*	See “9.6. Behavior.” [default] = 0
15	Optional (Not Supported)	Set	NV	Warning Settling Time	UINT	Number of Milliseconds allowed for the control-loop to settle to within the Error Band	See “9.6. Behavior.” [default] = 0
16	Optional (Not Supported)	Set	NV	Warning Error Band	INT or specified by *Data Type* if supported	The amount by which the *Setpoint* must equal the *Process Variable*	See “9.6. Behavior.” [default] = 0

Attr ID	Need in Implementation	Access Rule	NV*	Name	DeviceNet Data Type	Description of Attribute	Semantics of Values
17	Optional (Not Supported)	Set	NV	Safe State	USINT (byte)	Specifies the Control Variable behavior for states other than Execute	See “9.3. Semantics.” 0 = [default]
18	Optional (Not Supported)	Set	NV	Safe Value	INT or specified by *Data Type* if supported	The value to be used for Safe State = Safe Value	See “9.3. Semantics.” 0 = [default]
19	Optional (Supported)	Set	NV	Ramp Rate	UDINT (4-bytes)	Time in Milliseconds to reach Setpoint	0 = Disabled [default] x = value in milliseconds Where: 0 < x < 7FFF DeviceNet specifies 4 bytes of data, but only 2 are used. See “9.6. Behavior.”
97-98	Reserved by DeviceNet						
99	Conditional [3] (Supported)	Get	NV	Subclass	UINT	Identifies a subset of additional attributes, services and behaviors. The subclasses for this object are specified at the end of this object specification section.	0 = No subclass n = subclass as defined herein

NV = Nonvolatile; value is retained through power cycle.

[1] Data Type is settable ONLY under certain conditions (see semantics)

[2] The Process Variable is only optional if this device includes an internal sensor. Otherwise, the Process Variable is required. Similarly, The Control Variable is only optional if this device includes an internal actuator. Otherwise, the Control Variable is required.

[3] If the value of Subclass is 00, then this attribute is OPTIONAL in implementation, otherwise, this attribute is REQUIRED.

9.3. Semantics

Data Type

All Data Type attributes, including *Data Type* and *CV Data Type*, use the enumerated values specified in DeviceNet Vol. I, Appendix J-6.1.

The *Data Type* attribute is settable only in the *Idle State* and only if no attribute belonging to the object instance is the endpoint of an I/O connection in the *Established State*.

The *Data Type* attribute may change automatically based upon established I/O connections. See “9.6. Behavior” for more information on this mechanism.

Data Units

Specifies the context of *Setpoint and Process Variable* and related attributes (such as, offset and trip points) for this object instance. See Appendix K for a list of values. A request to set attribute to an unsupported value will return an error response.

The *Data Units* attribute is settable only in the *Idle State*.

In applications where this object is used in a relationship with an S-Analog Sensor object, this attribute may be specified as Get only, by the device profile or the vendor, where the value mirrors that of the S-Analog Sensor object *Data Units* attribute.

Setpoint, Process Variable and Control Variable

These three attributes compose the primary aspects of basic closed-loop control. The *Process Variable* is the measured parameter of the process or system being controlled. The *Setpoint* is the desired value for the measured parameter. By affecting the value of the *Control Variable*, the closed-loop controller drives the process or system to the desired state of:

Process Variable = Setpoint

The *Control Variable* is, therefore, connected to the process or system in such a way that it affects the value of the *Process Variable*. Examples of *Control Variable/Process Variable* combinations include: heater/temperature; valve/flow; or regulator/pressure.

Status

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

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

Control Mode

This attribute is used to override the value of the *Control Variable* attribute. Further, it may cause the object to modify the internal control algorithm such that a smooth, or “bumpless” transitions occurs upon activating control to setpoint.

The *Safe State* attribute provides a mechanism for override depending upon object state and will take precedents over this. That is, if an object instance implements the *Safe State* attribute and related behavior, then this *Override* attribute and related behavior will only function in the Executing State.

Attribute Value	State
0	Normal
1	Zero/Off/Closed
2	Full/On/Open
3	Hold
4	Safe State
5-63	reserved
64-127	Device Specific (specified by device profile)
128-255	Vendor Specific

Safe State

This attribute specifies what value will be held in the *Control Variable* attribute for states other than Executing. See the S-Device Supervisor object definition in Section 6-48. for a description of object states. The following values are defined:

Attribute Value	State
0	Zero/Off
1	Full Scale/On
2	Hold Last Value
3	Use Safe Value
4-63	Reserved
64-127	Device Specific (specified by device profile)
128-255	Vendor Specific

Safe Value

For Safe State set to Use Safe Value, this attribute holds the value to which the Control Variable attribute will be set for object instance states other than Executing.

Ramp Rate

The ramp rate is limited to values 0 through 32,767. See “9.6. Behavior.”

9.4. S-Single Stage Controller Common Services

Service Code	Need in Implementation		Service Name	Description of Service
	Class	Instance		
0Ehex 14dec	Conditional*	Required	Get_Attribute_Single	Returns the contents of the specified attribute.
10hex 16dec	n/a	Required	Set_Attribute_Single	Modifies an attribute value.
*The Get_Attribute_Single service is REQUIRED if any attributes are implemented.				

See the DeviceNet Communication Model and Protocol for definitions of these common services.

9.5. S-Single Stage Controller Object–Specific Services

The S-Single Stage Controller Object provides no Object–Specific services.

9.6. Behavior

The S-Device Supervisor Object manages the behavior of the S-Single Stage Controller Object. See Section 6. Additionally, this object exhibits the following behavior:

Alarm and Warning Exception Conditions

While in the Executing State as defined by the S-Device Supervisor Object: Immediately upon detecting that the Setpoint does not equal the Process Variable by an amount plus-or-minus the associated (alarm or warning) Error Band, a timer is started. This internal timer is incremented as long as the above condition exists. If the timer exceeds the amount indicated by the associated (alarm or warning) Settling Time and the associated (alarm or warning) Exception Enable is set, then the appropriate (alarm or warning) Exception Condition is set. Note that two internal timers are required in order to support both Alarm and Warning Exception reporting.

This behavior is modified for Ramp Rate values not equal to zero. In such cases, the timer is not enabled until after the expiration of the Ramp Time.

Ramp Rate

For Ramp Rate values other than zero, the S-Single Stage Controller Object internally modifies the Setpoint value in such a way that the Process Variable is “ramped” to its final value. An example follows. A Ramp Rate of 1000 is set and a new Setpoint is sent to the MFC. The setpoint feed to the controller will be internally (transparently) modified, in whatever time increments the object is able to sustain, in order to affect a smooth transition over one second from the old Setpoint to the new Setpoint, finally reaching the new Setpoint at the one second mark.

Note: The GF100 Series supports Ramp Rates from 1000ms to 32767ms. Rates greater than 7FFFh will return an error. Rates below 1000ms will be accepted; however, the ramping algorithm will not be invoked.

Data Type

If the implementation of this object specifies more than one valid Data Type value, in the device profile or by vendor, then 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. This configuration will then remain in effect for this object instance even, after all I/O connections are lost. For devices that support only one Data Type, this behavior is not supported.

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 profile specifies an instance of the S-Single Stage Controller 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.

GF100 Series implementation

Data Type values supported are Integer (0xC3) and Real (0xCA). Data Units supported are Counts (0x1001) and SCCM (0x1400). Data Type and Data Units attributes are settable. The supported combinations of Data Type and Data Units on the GF100 Series are Integer-Counts (default), Real-SCCM, Integer-SCCM, and Real-Counts. The full-scale range for Integer-Counts is either 0x6000 or 0x7FFF, depending on the configuration. The MFC Device Profile specifies that the full-scale range for the setpoint is 0x7FFF; however, the default GF100 Series configuration supports a full-scale setpoint range of 0 to 0x6000.

Control

The application of this object is further specified in the applicable device profile; primarily, the interfaces and object relationships are defined. Generally, the *Process Variable* attribute is restricted to "Get Only" access and an internal connection is defined to another object. Similarly, the *Control Variable* is generally not supported due to internal connections.

When in the EXECUTING state, this object is running an application process designed to cause the *Process Variable* to be driven to the value of the *Setpoint*. In any state other than EXECUTING, the application process is stopped and the *Safe State* is activated for the output of the object.

Any fault detected by the object application process causes the object to transition to the appropriate state as defined by the managing S-Device Supervisor object.

10. S-Gas Calibration Object (Class 0x34)

An S-Gas Calibration Object affects the behavior of an associated S-Analog Sensor object instance; a device profile will show a relationship between these two objects where an S-Gas Calibration Object is used. The S-Analog Sensor object uses a selection attribute as the gas type selection mechanism. The S-Gas Calibration Object provides the data with which a device enacts the appropriate calibration algorithm for a given gas type. Each S-Gas Calibration Object Instance contains a set of attribute values for one particular calibration set; each identified by the Gas Standard Number.

The S-Gas Calibration class level object provides a service for retrieving a list of all valid object instances. The service response includes a list of elements. Each element includes the Instance ID, Gas Standard Number and the valid S-Analog Sensor object instance ID for which the instance is valid.

There may be more than one instance with the same Gas Standard Number. These instances may be differentiated by Full Scale, Gas Symbol, Additional Scaler and/or other parametric distinctions, including valid S-Analog Sensor object instance ID. The distinctions may, or may not, be evident in the Get_All_Instances service response, depending upon what the distinction is.

S-Gas Calibration Objects most often utilize the region of Manufacturer Specified Attributes (ID > 100) for specific calibration parameters.

This object is a member of the *Hierarchy of Semiconductor Equipment Devices*. As such, its behavior is managed by the Device Supervisor Object. See Section 6.

The S-Gas Calibration object makes use of a list of Standard Gas Type Numbers. This list is described in publication:

> SEMI E52-95 “Practice for Referencing Gases Used in Digital Mass Flow Controllers”, Semiconductor Equipment and Materials International (SEMI), Mountain View, CA 94043-4080.

Note: It is implied that the reference above is to the latest revision as specified by SEMI.

10.1. S-Gas Calibration Class Attributes

The Object Class Attribute ID 1-7 are reserved. See DeviceNet Volume II, Section 5-4.1. for more specification detail on these attributes.

Attribute ID	Need in Implementation	Access Rule	Name	DeviceNet Data Type	Description of Attribute
1 thru 7	These class attributes are either optional or conditional and are described in chapter 5 of this specification.				
97 & 98	Reserved by DeviceNet				
99	Conditional * (Supported)	Get	Subclass	UINT	Identifies a subset of additional attributes, services and behaviors. The subclasses for this object are specified at the end of this object specification section.
170	Optional (Supported)	Get	Bin ID	INT	Specifies the Device Bin number. This value is vendor specific. **

** If the value of Subclass is 00, which identifies "no subclass", then this attribute is OPTIONAL in implementation, otherwise, this attribute is REQUIRED.*

*** This class attribute is required for certain customer applications, such as the “multi-gas, multi-range” application.*

10.2. S-Gas Calibration Instance Attributes

Certain minimal implementations may support any optional “Set” attributes as “Get” only and still be compliant with this object specification. All required attributes must be supported as specified. The GF100 Series supports 9 instances of the S-Gas Calibration Object.

Attr ID	Need in Implementation	Access Rule	NV*	Name	DeviceNet Data Type	Description of Attribute	Semantics of Values
1	Optional (Not Supported)	Get	NV	Number of Attributes	USINT	Number of attributes supported	Number of attributes supported in this object instance
2	Optional (Not Supported)	Get	NV	Attribute List	ARRAY OF USINT	List of attributes supported by this object instance	List of attributes supported in this object instance
3	Required	Get	NV	Gas Standard Number	UINT	Gas Type Number	[default] = 0 (no gas type specified) See “10.3. Semantics.”
4	Required	Get	NV	Valid Sensor Instance	UINT	S-Analog Sensor object instance ID for which this object instance is valid	0 = No Valid Sensor n = Instance ID See “10.3. Semantics.” [default] = 0
5	Optional (Supported)	Set	NV	Gas Symbol	SHORT STRING	Gas Type Name	See “10.3. Semantics.” [default] = null
6	Optional (Supported)	Get	NV	Full Scale	STRUCT of:	Full Scale of the device using this object instance	See “10.3. Semantics.” [default] = 0, 0
					REAL	Amount	The amount of measured parameter corresponding to full scale.
					UINT	Units	The units for the above. See Data Units Appendix K.

Attr ID	Need in Implementation	Access Rule	NV*	Name	DeviceNet Data Type	Description of Attribute	Semantics of Values
7	Optional (Not Supported)	Set	NV	Additional Scaler	REAL	Additional Correction Factor	In addition to the correction algorithm, this amount is multiplied to the reading. Generally used for Gas Correction for a gas other than the type identified for the object instance by attribute 3. (E.g., scale a nitrogen object instance to measure argon). Default = 1.0
8	Optional (Supported)	Get	NV	Calibration Date	DATE	Date of Calibration	The date this object instance was last calibrated [default] = 0
9	Optional (Supported)	Get	NV	Calibration Gas Number	UINT	Calibration Gas	The gas number of the gas used to calibrate this object instance. [default] = 0
10	Optional (Not Supported)	Get	NV	Gas Correction Factor	REAL	Gas Correction Factor For devices that support simple correction factors (as opposed to algorithms) for gas selection.	[default] = 1.0
97-98	Reserved by DeviceNet						
99	Conditional ** (Supported)	Get	NV	Subclass	UINT	Identifies a subset of additional attributes, services and behaviors. The subclasses for this object are specified at the end of this object specification section.	0 = No subclass n = subclass as defined herein

** NV = Nonvolatile; value is maintained through power cycle.*

*** If the value of Subclass is 00, then this attribute is OPTIONAL in implementation, otherwise, this attribute is REQUIRED.*

10.3. Semantics

Gas Standard Number

Used to identify a gas standard number, for which the object instance is currently calibrated. See Instance Application Example below.

The actual coding of the values are described in the following publication:

See “1. Introduction” for reference to the SEMI publication: *Practice for Referencing Gases Used in Digital Mass Flow Controllers*.

Since the actual attributes, and their context, for the parameterization of object instances for particular gas types is beyond the scope of this standard (i.e., vendor specific) the Access Rule for this attribute has been specified as Get. Vendors may choose to specify an Access Rule of Set for this attribute.

Valid Sensor Instances

This attribute specifies the S-Analog Sensor object instance for which the S-Gas Calibration object instance is valid. An S-Gas Calibration object instance will be valid for zero or one S-Analog Sensor object instances.

Gas Symbol

This optional attribute is a string-coded representation of the name of the gas for which the object instance has been configured. It is coded as a user defined text symbol or it is coded as defined in the above referenced SEMI publication.

This attribute may indicate a different gas from the one, which has been specified by the Gas Standard Number. See Instance Application Example below.

Full Scale

This optional attribute identifies the amount of measured parameter (e.g., Mass Flow) corresponding to the Full Scale of the associated S-Analog Sensor object. A primary purpose for this attribute is to allow for simple S-Analog Sensor object implementations where the Value is reported in raw units; this attribute allows a mapping to engineering units.

For example, the Full Scale for a S-Gas Calibration object may be 100 SCCM, while the Full Scale for the associated S-Analog Sensor object may be 0x6000 counts (i.e., S-Analog Sensor object Data Type = INT and Data Units = Counts).

Instance Application Example

The following is an example to demonstrate the usage of Gas Calibration object instances and their attributes:

A device has been supplied with three gas calibration object instances: nitrogen (13)*, helium (1)* and argon (4)*. The user wishes to use the device for silane (39)* and knows that a correction factor of 0.60 will properly convert a nitrogen calibration for this application. The object instance for nitrogen would be selected and the Additional Scaler attribute for this instance would be set to 0.60. To identify this modification, the Gas Symbol may be set to read “silane”, “SiH4”, or “39”.

** (Gas Standard Number)*

10.4. S-Gas Calibration Common Services

Service Code	Need in Implementation		Service Name	Description of Service
	Class	Instance		
0Ehex 14dec	Required	Required	Get_Attribute_Single	Returns the contents of the specified attribute.
10hex 16dec	Required	Required	Set_Attribute_Single	Modifies an attribute value.

See the DeviceNet Communication Model and Protocol for definitions of these common services.

10.5. S-Gas Calibration Object–Specific Services

Service Code	Need in Implementation		Service Name	Description of Service
	Class	Instance		
4Bhex 75dec	Required	n/a	Get_All_Instances	Requests a list of all available object instances with their respective gas numbers

If a gas instance is changed or added, the device must be reset before performing the “get_all_instances” service.

Success Response Service Data Field Parameters

Parameter	Required	Data Type	Description	Semantics of Values
Size of List	Required	UINT	Specifies the number of elements in the Array	Number of gas calibrations in the list
List of Gas Calibrations	Required if Size > 0	ARRAY of	Supported List	The list of gas calibrations
		STRUCT of	Supported Gas Type	
		UINT	S-Gas Calibration Object Instance ID	[34-n-4], where n is the instance value 1 – 9.
		UINT	Gas Standard Number	[34-n-3], where n is the instance value 1 – 9.
		UINT	Valid Sensor Instance	Always =1 for the GF100 Series.

On the GF100 Series: Gas instance 6 is considered the “test” gas instance. There are a total of 6 gas instances, (sometimes referred to as “gas pages”) available.

10.6. S-Gas Calibration Object Behavior

The behavior of this object is managed by the Device Supervisor Object, defined in Section 6-48.5.

10.7. S-Gas Calibration Object Instance Subclass 01

The following specification applies to a subclass of this object for application in Mass Flow Controller devices.

10.7.1. Subclass01 Instance Attributes

Attribute ID	Need in Implementation	Access Rule	Name	DeviceNet Data Type	Description of Attribute	Semantics of Values
95	Optional (Supported)	Get	Calibration Pressure	REAL	The gas pressure in KiloPascal	The Standard Pressure with respect to the calibration conditions. Default = 101.32, (14.7 PSIA).
96	Optional (Not Supported)	Get	Calibration Temperature	REAL	The Gas Temperature in Degrees C	The Standard Temperature with respect to the calibration conditions. Default = 0.0

10.7.2. Subclass 01 Instance Services

There are no additions or restrictions to the Object Services for this object subclass.

10.7.3. Subclass 01 Behavior

There are no additions or restrictions to the Behavior for this object subclass.

11. Advanced Diagnostics (GF135 Only)

11.1. New Communication Protocol Overview

11.1.1. Objectives and Problem Statement

Through the years, issues found in the field on Brooks MFCs have been very difficult to troubleshoot due to lack of information given to failure analysis teams. Issues that are random in occurrence and are specific to the field setup are the most difficult to reproduce therefore troubleshooting takes longer.

This new communication capability will enable the device to perform a series of self validation at regular interval and report its status to the tool software. Some of the self validation will require at least knowing the state of certain part of the tool over which the device typically had neither control nor access. The capabilities of the GF135 will enable short interval control of the upstream isolation valve in a manner consistent with good safety practices.

Those capabilities do not exist in any protocol currently in use in the field.

11.1.2. Protocol Description

The communication protocol shall be implemented over RS485 physical layer.

The protocol shall be able to handle the following type of communications between the device and the tool:

- Tool request for Commissioning status
- Tool request for valve leak status, ROD measurements

These are the minimum requirements to support advanced diagnostics.

11.2. Specific Requirements

11.2.1. Introduction

The following requirements are a subset of Brooks' new communication protocol definition.

11.2.2. Advanced Protocol

The advanced communication protocol is more fully defined in GF135-SRS-011.

11.2.3. New Attributes

Note: Those attributes are valid as of the date of writing. Attribute ID are subject to change and new attributes will likely be added by the time this document is finalized.

New attributes for the advanced diagnostic:

Attribute	Access	Class	Instance	Att. ID	Values
isolation valve status	READ ONLY	177	1	3	0: Upstream open 1: Upstream closed
ROD delay	Read/Write	177	1	20	Delay before first ROD measurement after a setpoint change (seconds). Minimum = default = 4 sec.
ROD interval	Read/Write	177	1	21	Interval between ROD measurement when setpoint is constant (seconds). Minimum = 5 sec. Default = 10 sec.
ROD enable Default flag	Read/Write	177	1	55	0: ROD is disabled 1: ROD is enabled Non Volatile. Copied to Attribute 62 at power up. Default = Enabled

Attribute	Access	Class	Instance	Att. ID	Values
ROD Error Status	READ ONLY	177	1	56	0: ROD Error (attribute 11) is not valid 1: ROD Error (attribute 11) is valid
Valve Leak status	READ ONLY	177	1	57	0: Valve Leak Meas. is not valid 1: Valve Leak Meas. is valid
Valve Leak value	READ ONLY	177	1	59	Float, fraction of configured range
ROD Current Setpoint	READ ONLY	177	1	60	Current setpoint at which the ROD is being measured (see attribute 11) (fraction of configured range)
ROD error	READ ONLY	177	1	61	ROD measured flow change (from baseline) in % SP for the current setpoint. Same as ROD error N.
ROD enable flag	Read/Write	177	1	62	0: ROD is disabled 1: ROD is enabled Volatile
Commissioning status	Read/Write	103	n	132	0: Not done or failed 1: Commissioning was successful
ROD 0 Status	READ ONLY	177	1	70	0: ROD Error 0 is not valid 1: ROD Error 0 is valid
ROD 1 Status	READ ONLY	177	1	71	0: ROD Error 1 is not valid 1: ROD Error 1 is valid
ROD 2 Status	READ ONLY	177	1	72	0: ROD Error 2 is not valid 1: ROD Error 2 is valid
ROD 3 Status	READ ONLY	177	1	73	0: ROD Error 3 is not valid 1: ROD Error 3 is valid
ROD 4 Status	READ ONLY	177	1	74	0: ROD Error 4 is not valid 1: ROD Error 4 is valid
ROD 5 Status	READ ONLY	177	1	75	0: ROD Error 5 is not valid 1: ROD Error 5 is valid
ROD 6 Status	READ ONLY	177	1	76	0: ROD Error 6 is not valid 1: ROD Error 6 is valid
ROD 7 Status	READ ONLY	177	1	77	0: ROD Error 7 is not valid 1: ROD Error 7 is valid
ROD 8 Status	READ ONLY	177	1	78	0: ROD Error 8 is not valid 1: ROD Error 8 is valid
ROD 9 Status	READ ONLY	177	1	79	0: ROD Error 9 is not valid 1: ROD Error 9 is valid
ROD 10 Status	READ ONLY	177	1	80	0: ROD Error 10 is not valid 1: ROD Error 10 is valid

Attribute	Access	Class	Instance	Att. ID	Values
ROD 11 Status	READ ONLY	177	1	81	0: ROD Error 11 is not valid 1: ROD Error 11 is valid
ROD 12 Status	READ ONLY	177	1	82	0: ROD Error12 is not valid 1: ROD Error 12 is valid
ROD 13 Status	READ ONLY	177	1	83	0: ROD Error 13 is not valid 1: ROD Error 13 is valid
ROD 14 Status	READ ONLY	177	1	84	0: ROD Error 14 is not valid 1: ROD Error 14 is valid
ROD 15 Status	READ ONLY	177	1	85	0: ROD Error 15 is not valid 1: ROD Error 15 is valid
ROD 16 Status	READ ONLY	177	1	86	0: ROD Error 16 is not valid 1: ROD Error 16 is valid
ROD 17 Status	READ ONLY	177	1	87	0: ROD Error 17 is not valid 1: ROD Error 17 is valid
ROD 18 Status	READ ONLY	177	1	88	0: ROD Error 18 is not valid 1: ROD Error 18 is valid
ROD 19 Status	READ ONLY	177	1	89	0: ROD Error 19 is not valid 1: ROD Error 19 is valid
ROD Error 0	READ ONLY	177	1	90	Float, Avg. error for setpoint 0-5%
ROD Error 1	READ ONLY	177	1	91	Float, Avg. error for setpoint 5-10%
ROD Error 2	READ ONLY	177	1	92	Float, Avg. error for setpoint 10-15%
ROD Error 3	READ ONLY	177	1	93	Float, Avg. error for setpoint 15-20%
ROD Error 4	READ ONLY	177	1	94	Float, Avg. error for setpoint 20-25%
ROD Error 5	READ ONLY	177	1	95	Float, Avg. error for setpoint 25-30%
ROD Error 6	READ ONLY	177	1	96	Float, Avg. error for setpoint 30-35%
ROD Error 7	READ ONLY	177	1	97	Float, Avg. error for setpoint 35-40%
ROD Error 8	READ ONLY	177	1	98	Float, Avg. error for setpoint 40-45%
ROD Error 9	READ ONLY	177	1	99	Float, Avg. error for setpoint 45-50%
ROD Error 10	READ ONLY	177	1	100	Float, Avg. error for setpoint 50-55%
ROD Error 11	READ ONLY	177	1	101	Float, Avg. error for setpoint 55-60%
ROD Error 12	READ ONLY	177	1	102	Float, Avg. error for setpoint 60-65%
ROD Error 13	READ ONLY	177	1	103	Float, Avg. error for setpoint 65-70%
ROD Error 14	READ ONLY	177	1	104	Float, Avg. error for setpoint 70-75%
ROD Error 15	READ ONLY	177	1	105	Float, Avg. error for setpoint 75-80%
ROD Error 16	READ ONLY	177	1	106	Float, Avg. error for setpoint 80-85%

Attribute	Access	Class	Instance	Att. ID	Values
ROD Error 17	READ ONLY	177	1	107	Float, Avg. error for setpoint 85-90%
ROD Error 18	READ ONLY	177	1	108	Float, Avg. error for setpoint 90-95%
ROD Error 19	READ ONLY	177	1	109	Float, Avg. error for setpoint 95-100%

The following set of attributes is required for the data logging and trending:

Attribute	Access	Class	Instance	Att. ID	Values
Time Synchronization	WRITE ONLY	0x43 (67)	1	TBD	Time and date information Not implemented
Data Request	READ ONLY	0x41 (65)	1	TBD	Trending information Not implemented

11.3. Tool – Device Communication

The following items need to be addressed with the customer to define the new advanced diagnostic protocol:

Tool to provide status (e.g. isolation valve status upstream and downstream)

Some operations require knowing the status of the isolation valves on both sides of the device. The tool shall provide this information by writing to the isolation valve status attribute whenever the status changes.

Note: At this time, this is not implemented on the customer tool software.

Request from the device to the tool to modify the tool status (e.g. Operating the isolation valves at the request of the MFC)

Some operations require a specific state of the isolation valves on both sides of the device. The tool shall read the isolation valve request attribute on a schedule TBD and satisfy the request of the MFC.

Note: At this time, this is not implemented on the customer tool software.

Time synchronization

In order to provide accurate data logging for the trending diagnostic, the tool will need to provide accurate time synchronization. The MFC is capable of running a real time clock, so only occasional synchronization is required (at least once per power on cycle)

Note: At this time, this is not implemented on the customer tool software.

Request from the tool for some trending information

Data can be logged at regular interval and time stamped using the time information

Data can be retrieved by the customer to provide long term trend (e.g. zero information logged once per week)

Note: At this time, this is only supported via the diagnostic port.

12. References

[1] S-Device Supervisor Object. Open DeviceNet Venders Association (ODVA) DeviceNet Specification Enhancement 93-01. Version J. 1/27/1999.

[2] S-Analog Sensor Object. Open DeviceNet Venders Association (ODVA) DeviceNet Specification Enhancement 93-02. Version J. 1/27/1999.

[3] S-Analog Actuator Object. Open DeviceNet Venders Association (ODVA) DeviceNet Specification Enhancement 93-03. Version J. 1/27/1999.

[4] S-Single Stage Controller Object. Open DeviceNet Venders Association (ODVA) DeviceNet Specification Enhancement 93-04. Version J. 1/27/1999.

[5] S-Gas Calibration Object. Open DeviceNet Venders Association (ODVA) DeviceNet Specification Enhancement 93-05. Version J. 1/27/1999.

[6]Mass-Flow Controller Device Profile. Open DeviceNet Venders Association (ODVA) DeviceNet Specification Enhancement 93-06. Version J. 1/27/1999.

[7]Open DeviceNet Venders Association (ODVA) DeviceNet Specification, Volume 1 and 2. Version 2.0. 12/2/1998.

THIS PAGE WAS INTENTIONALLY LEFT BLANK

THIS PAGE WAS INTENTIONALLY LEFT BLANK

Brooks DeviceNet PCs/PMs

LIMITED WARRANTY

Seller warrants that the Goods manufactured by Seller will be free from defects in materials or workmanship under normal use and service and that the Software will execute the programming instructions provided by Seller until the expiration of the earlier of twelve (12) months from the date of initial installation or eighteen (18) months from the date of shipment by Seller. Products purchased by Seller from a third party for resale to Buyer ("Resale Products") shall carry only the warranty extended by the original manufacturer.
All replacements or repairs necessitated by inadequate preventive maintenance, or by normal wear and usage, or by fault of Buyer, or by unsuitable power sources or by attack or deterioration under unsuitable environmental conditions, or by abuse, accident, alteration, misuse, improper installation, modification, repair, storage or handling, or any other cause not the fault of Seller are not covered by this limited warranty, and shall be at Buyer's expense.
Goods repaired and parts replaced during the warranty period shall be in warranty for the remainder of the original warranty period or ninety (90) days, whichever is longer. This limited warranty is the only warranty made by Seller and can be amended only in a writing signed by an authorized representative of Seller.

BROOKS SERVICE AND SUPPORT

Brooks is committed to assuring all of our customers receive the ideal flow solution for their application, along with outstanding service and support to back it up. We operate first class repair facilities located around the world to provide rapid response and support. Each location utilizes primary standard calibration equipment to ensure accuracy and reliability for repairs and recalibration and is certified by our local Weights and Measures Authorities and traceable to the relevant International Standards.

Visit www.BrooksInstrument.com to locate the service location nearest to you.

START-UP SERVICE AND IN-SITU CALIBRATION

Brooks Instrument can provide start-up service prior to operation when required.
For some process applications, where ISO-9001 Quality Certification is important, it is mandatory to verify and/or (re)calibrate the products periodically. In many cases this service can be provided under in-situ conditions, and the results will be traceable to the relevant international quality standards.

CUSTOMER SEMINARS AND TRAINING

Brooks Instrument can provide customer seminars and dedicated training to engineers, end users and maintenance persons.

Please contact your nearest sales representative for more details.

HELP DESK

In case you need technical assistance:

USA	☎	888 275 8946	Korea	☎	+82 31 708 2521
Netherlands	☎	+31 (0) 318 549290	Taiwan	☎	+886 3 5590 988
Germany	☎	+49 351 215 2040	China	☎	+86 21 5079 8828
Japan	☎	+81 3 5633 7100	Singapore	☎	+6297 9741

Due to Brooks Instrument's commitment to continuous improvement of our products, all specifications are subject to change without notice.

TRADEMARKS

Brooks . Brooks Instrument, LLC
DeviceNet Open DeviceNet Vendors Association, Inc.

Brooks Instrument
407 West Vine Street
P.O. Box 903
Hatfield, PA 19440-0903 USA
T (215) 362 3700
F (215) 362 3745
E-Mail BrooksAm@BrooksInstrument.com
www.BrooksInstrument.com

Brooks Instrument
Neonstraat 3
6718 WX Ede, Netherlands
T +31 (0) 318 549 300
F +31 (0) 318 549 309
E-Mail BrooksEu@BrooksInstrument.com

Brooks Instrument
1-4-4 Kitasuna Koto-Ku
Tokyo, 136-0073 Japan
T +81 (0) 3 5633 7100
F +81 (0) 3 5633 7101
E-Mail BrooksAs@BrooksInstrument.com