Installation and Operation Manual
X-DPT-S-Protocol-4800-eng
Part Number: 541B136AAG
June, 2009

Supplemental Manual for Brooks® 4800 Series S-Protocol over RS485 MFCs/MFMs

BROOKS® INSTRUMENT

Essential Instructions

Read this page before proceeding!

Brooks Instrument designs, manufactures and tests its products to meet many national and international standards. Because these instruments are sophisticated technical products, you must properly install, use and maintain them to ensure they continue to operate within their normal specifications. The following instructions must be adhered to and integrated into your safety program when installing, using and maintaining Brooks Products.

- Read all instructions prior to installing, operating and servicing the product. If this instruction manual is not the correct manual, please see back cover for local sales office contact information. Save this instruction manual for future reference.
- If you do not understand any of the instructions, contact your Brooks Instrument representative for clarification.
- Follow all warnings, cautions and instructions marked on and supplied with the product.
- Inform and educate your personnel in the proper installation, operation and maintenance of the product.
- Install your equipment as specified in the installation instructions of the appropriate instruction manual and per applicable local and national codes. Connect all products to the proper electrical and pressure sources.
- To ensure proper performance, use qualified personnel to install, operate, update, program and maintain the product.
- When replacement parts are required, ensure that qualified people use replacement parts specified by Brooks Instrument. Unauthorized parts and procedures can affect the product's performance and place the safe operation of your process at risk. Look-alike substitutions may result in fire, electrical hazards or improper operation.
- Ensure that all equipment doors are closed and protective covers are in place, except when maintenance is being performed by qualified persons, to prevent electrical shock and personal injury.

ESD (Electrostatic Discharge)

CAUTION

This instrument contains electronic components that are susceptible to damage by static electricity. proper handling procedures must be observed during the removal, installation or other handling of internal circuit boards or devices.

Handling Procedure:

1. Power to unit must be removed.
2. Personnel must be grounded, via a wrist strap or other safe, suitable means before any printed circuit card or other internal device is installed, removed or adjusted.
3. Printed circuit cards must be transported in a conductive container. Boards must not be removed from protective enclosure until immediately before installation. Removed boards must immediately be placed in protective container for transport, storage or return to factory.

Comments

This instrument is not unique in its content of ESD (electrostatic discharge) sensitive components. Most modern electronic designs contain components that utilize metal oxide technology (NMOS, SMOS, etc.). Experience has proven that even small amounts of static electricity can damage or destroy these devices. Damaged components, even though they appear to function properly, exhibit early failure.

Dear Customer,

We appreciate this opportunity to service your flow measurement and control requirements with a Brooks Instrument device. Every day, flow customers all over the world turn to Brooks Instrument for solutions to their gas and liquid low-flow applications. Brooks provides an array of flow measurement and control products for various industries from biopharmaceuticals, oil and gas, fuel cell research and chemicals, to medical devices, analytical instrumentation, semiconductor manufacturing, and more.

The Brooks product you have just received is of the highest quality available, offering superior performance, reliability and value to the user. It is designed with the ever changing process conditions, accuracy requirements and hostile process environments in mind to provide you with a lifetime of dependable service.

We recommend that you read this manual in its entirety. Should you require any additional information concerning Brooks products and services, please contact your local Brooks Sales and Service Office listed on the back cover of this manual or visit www.BrooksInstrument.com

Yours sincerely,
Brooks Instrument

THIS PAGE WAS INTENTIONALLY LEFT BLANK

Table of Contents

Tables

Figures

THIS PAGE WAS INTENTIONALLY LEFT BLANK

1. Introduction

1.1. How to Use This Manual

This instruction manual along with the 4800 Series product manual (X-TMF-4800-MFC-eng) is intended to provide the user with all the information necessary to install, operate and maintain the Brooks 4800 Series RS485 (S-Protocol) digital interface module.

This manual is organized into the following sections:
Section 1 – Introduction
Section 2 – Receipt & Storage of Equipment
Section 3 – Background & Numbering
Section 4 – Installation
Section 5 – Message Protocol Structure
Section 6 – Master Slave Communication
Section 7 – General Transmitter Information
Section 8 – Universal Command Specifications
Section 9 – Common Practice Command Specifications
Section 10 – Transmitter Specific Command Specifications
Section 11 – Transmitter Specific Tables

1.2. Description

The Brooks® S-Protocol is a Digital Communication protocol which provides a reliable, transaction oriented service between a master device, such as a PC, and one or more Brooks® Digital Series Mass Flow Controllers and Meters. The protocol is designed to allow a centralized controller to acquire measurement data from a Mass Flow device and, incase of Mass Flow Controllers, send setpoint values.

The Brooks RS485 on 4800 Series MFCs/MFMs support digital communications as defined by this manual. This protocol is based on the HART® Communication Foundation (HCF) protocol and the Brooks RS485 on 4800 Series MFCs/MFMs support all the Universal Commands and many of the Common Practice commands as defined by the HCF. However, conformance to the HCF specifications is neither claimed nor implied.

The physical layer supported is RS-485 only (see Section 2). The HART Communication Foundation FSK physical layer (Bell-202 modem) is NOT supported and, therefore, the commonly available HART “Hand Held Configurators” are NOT compatible with the RS485 on 4800 Series MFCs/MFMs.

This document is intended to give a user the means to implement the protocol structure into his own control system in order to establish communication between the control system and the RS485 on 4800 Series devices. It does not cover the non-communication functionality of the 4800 Series devices. For this description please refer to Installation and Operation Manual for your specific device.

1.3. Specifications

Electrical Connections

Two (2) 15-pin D-sub connectors, one for connection to the 4800 Series device and one for the remote connection.

Pin No.	Function at remote connector	Function at connector to 4800
1.	Setpoint Signal Ground	Setpoint Signal Ground
2.	Flow Voltage Output	Flow Voltage Output
3.	Alarm Output	Not Used
4.	Flow Current Output	Flow Current Output
5.	Positive Supply Voltage	Positive Supply Voltage
6.	Not Used	Not Used
7.	Setpoint Current Input	Setpoint Current Input
8.	Setpoint Voltage Input	Setpoint Voltage Input
9.	Power Supply Common	Power Supply Common
10.	Flow Signal Ground	Flow Signal Ground
11.	Not Used	Not Used
12.	Valve Override Input	Valve Override Input
13.	Not Used	Not Used
14.	RXD/A-	RXD/A-
15.	TXD/A+	TXD/A+

Materials of Construction
Enclosure: ABS plastic with CU-Ni plating.

Operating Limits:
Temperature: 0 - 50°C
Operating Humidity: 5 to 95% R.H. (ambient)

Refer to the 4800 Series product manual X-TMF-4800-MFC-eng for all other specifications.

1.4. Definition of Terms

Abbreviation	Description
MFC/MFM	Mass Flow Controller/Meter device
MSB	Most Significant Bit
LSB	Least Significant Bit

2. Receipt & Storage

2.1. Receipt of Equipment

This section contains the procedures for the receipt and installation of the instrument. See Section 1 of this manual and the 4800 Series product manual (X-TMF-4800-MFC-eng) for dimensional and connection requirements. Do not attempt to start the system until the instrument has been permanently installed. It is important that the start-up procedures be followed in the exact sequence presented.

When the instrument is received, the outside packing case should be checked for damage incurred during shipment. If the packing case is damaged, the local carrier should be notified at once regarding his liability. A report should be submitted to your nearest Product Service Department.

Brooks Instrument
407 W. Vine Street
P.O. Box 903
Hatfield, PA 19440 USA
Toll Free (888) 554 FLOW (3569)
Tel (215) 362 3700
Fax (215) 362 3745
E-mail: BrooksAm@BrooksInstrument.com
www.BrooksInstrument.com

Brooks Instrument
Neonstraat 3
6718 WX Ede, Netherlands
P.O. Box 428
6710 BK Ede, Netherlands
Tel +31 (0) 318 549 300
Fax +31 (0) 318 549 309
E-mail: BrooksEu@BrooksInstrument.com

Brooks Instrument
1-4-4 Kitasuna Koto-Ku
Tokyo, 136-0073 Japan
Tel +81 (0) 3 5633 7100
Fax +81 (0) 3 5633 7101
Email: BrooksAs@BrooksInstrument.com

Remove the envelope containing the packing list. Carefully remove the instrument from the packing case. Make sure spare parts are not discarded with the packing materials. Inspect for damaged or missing parts.

2.2. Recommended Storage Practice

If intermediate or long-term storage of equipment is required, it is recommended that the equipment be stored in accordance with the following conditions:

a. Within the original shipping container.
b. Stored in a sheltered area, preferably a warm, dry, heated warehouse.
c. Ambient temperature 21°C (70°F) nominal, 32°C (90°F) maximum, 7°C (45°F) minimum.
d. Relative humidity 45% nominal, 60% maximum, 25% minimum.

2.3. Return of Equipment

Prior to returning any instrument to the factory, contact your nearest Brooks location for a Return Materials Authorization Number (RMA#). This can be obtained from one of the following locations:

Brooks Instrument
407 W. Vine Street
P.O. Box 903
Hatfield, PA 19440 USA
Toll Free (888) 554 FLOW (3569)
Tel (215) 362 3700
Fax (215) 362 3745
E-mail: BrooksAm@BrooksInstrument.com
www.BrooksInstrument.com

Brooks Instrument
Neonstraat 3
6718 WX Ede, Netherlands
P.O. Box 428
6710 BK Ede, Netherlands
Tel +31 (0) 318 549 300
Fax +31 (0) 318 549 309
E-mail: BrooksEu@BrooksInstrument.com

Brooks Instrument
1-4-4 Kitasuna Koto-Ku
Tokyo, 136-0073 Japan
Tel +81 (0) 3 5633 7100
Fax +81 (0) 3 5633 7101
Email: BrooksAs@BrooksInstrument.com

Instrument must have been purged in accordance with the following:

⚠WARNING

Before returning the device purge thoroughly with a dry inert gas such as Nitrogen before disconnecting gas connections. Failure to correctly purge the instrument could result in fire, explosion or death. Corrosion or contamination may occur upon exposure to air.

All flow instruments returned to Brooks requires completion of Form RPR003-1, Brooks Instrument Decontamination Statement, along with a Material Safety Data Sheet (MSDS) for the fluid(s) used in the instrument. Failure to provide this information will delay processing by Brooks personnel. Copies of these forms can be downloaded from the Brooks website www.BrooksInstrument.com or are available from any Brooks Instrument location listed above.

2.4. Transit Precautions

To safeguard the instrument against transportation damage, it is recommended to keep the instrument in its factory container until ready for installation.

2.5. Removal from Storage

Upon removal of the instrument from storage, a visual inspection should be conducted to verify its "as-received" condition. If the instrument has been subject to storage conditions in excess of those recommended (See Section 2-3), it should be subjected to a pneumatic pressure test in accordance with applicable vessel codes.

THIS PAGE WAS INTENTIONALLY LEFT BLANK

3. Background & Numbering

3.1. Background

This manual is a supplement to the Brooks 4800 Series installation and operation manual. It is assumed that the owner of this RS485 on 4800 MFC/MFM is thoroughly familiar with the theory and operation of this device. If not, it is recommended that the owner reads the installation and operation manual first before continuing with this supplement.

3.2. Numeric System

Numeric values used throughout this manual will be clearly denoted as to the base numeric system it represents. All hexadecimal numbers (base 16) will be prefixed with a 0x, like 0xA4. All binary numbers (base 2) will be suffixed with a b, like 1001b. All other numbers not annotated this way will be assumed decimal (base 10).

THIS PAGE WAS INTENTIONALLY LEFT BLANK

4. Installation

4.1. Start Up / Assumption

This section assumes the owner of the Digital Series device has a fully operational and trouble-free RS485 communications network with appropriate power supplies (15-24V depending on the 4800 device type).

4.2. Supported Baud Rates

Data communication can be performed at a number of baud rates: 1200, 2400, 4800, 9600, 19.2K, 38.4K baud. A rotary switch is used to select the proper baud rate, the baud rate change is affectuated after a power cycle. The device is shipped with the baud rate set to 19.2K baud, see picture below.

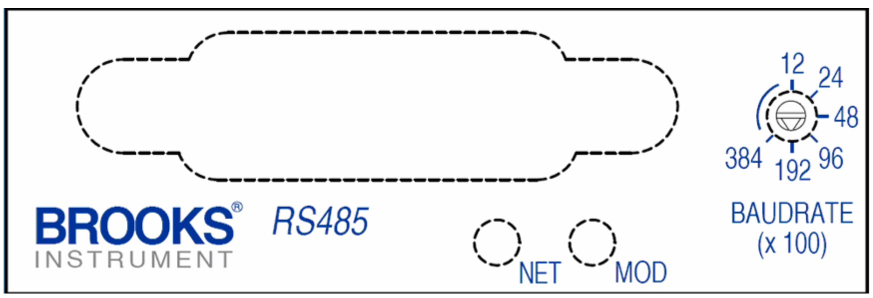


Figure 4-1 RS485 Label on Cover

4.3. Bus and Device LEDs

The device supports a Bus and Device LED to indicate the status of network communication and the device. The Bus LED will indicate the following:

Table 4-1 Bus Led Specification

Flash Code	Description
Off	No Network Connected
Solid Green	Communication Established at least once, resets after power cycle (no periodic check)

The Device LED will indicate the following:

Table 4-2 Device Led Specification

Flash Code	Description
Flashing Red/Green	The device is in the Self-Test/initializing mode
Solid Green	All self-tests/initialisation have passed. No faults have been detected
Flashing Red	A recoverable fault has been detected or the device has been commanded into the Fail/Halt state (Low flow alarm, High flow alarm)
Solid Red	An unrecoverable fault has occurred (MFC communication failure, Internal power supply failure)

4.4. Device Wiring

4.4.1. Electrical Connections

The RS485 on 4800 Series device has 2 15-pin D-sub connectors, one for connection to the 4800 Series device and one for remote connection. See the table below for the pin-outs.

Table 4-3 Pin-outs for D-Sub Connectors

Pin No.	Function at remote connector	Function at connector to 4800
1.	Setpoint Signal Ground	Setpoint Signal Ground
2.	Flow Voltage Output	Flow Voltage Output
3.	Alarm Output	Not Used
4.	Flow Current Output	Flow Current Output
5.	Positive Supply Voltage	Positive Supply Voltage
6.	Not Used	Not Used
7.	Setpoint Current Input	Setpoint Current Input
8.	Setpoint Voltage Input	Setpoint Voltage Input
9.	Power Supply Common	Power Supply Common
10.	Flow Signal Ground	Flow Signal Ground
11.	Not Used	Not Used
12.	Valve Override Input	Valve Override Input
13.	Not Used	Not Used
14.	RXD/A-	RXD/A-
15.	TXD/A+	TXD/A+

4.4.2. Alarm Output

Pin 3 of the remote D-sub connector at the top is a TTL open collector alarm output, type of used transistor is NPN.

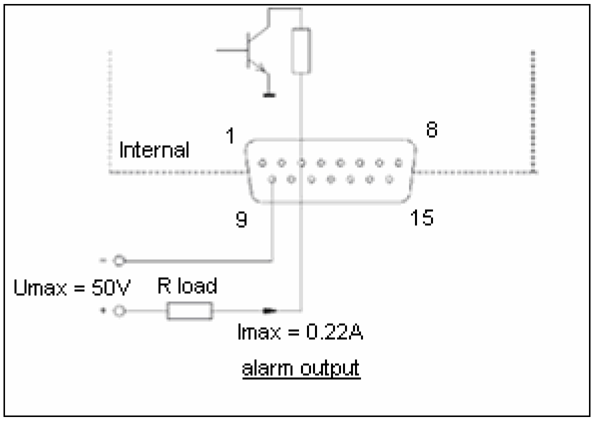


Figure 4-2 Open Collector Alarm Output Connection Example

The maximum parameters for the alarm output are:

Maximum voltage: 50 V
Maximum current: 0.22 A
Maximum power: 0.36 W

4.4.3. Multi Drop

The RS-485 communications interface is a multi drop connection making it possible to connect up to 32 devices to a computer on a single multi drop line as shown Figure 4-3. Most Computers are NOT equipped with RS-485 ports. In order to connect an RS-485 to a computer, one will need an RS-485 to RS-232C converter. Figure 4-3 shows the interconnection diagram of an RS485 on 4800 MFC/MFM via an RS-485 bus and an RS-485 to RS-232C converter to the RS-232 serial port of a typical computer. The RS-485 bus requires two matching resistors of 120Ω, one at the end of the bus and one at the beginning, near the converter. Note the control line from the PC to the converter necessary to control the data direction of the RS-485 buffers. The RTS ("Request To Send") line shown in Figure 4-3 because this line is used to control data direction in many of the commercially available converters. The actual line used depends on the converter selected.

Table 4-4 D-Connector Communication Pins

D-connector pin number	RS-485
Pin #14	A-
Pin #15	A+

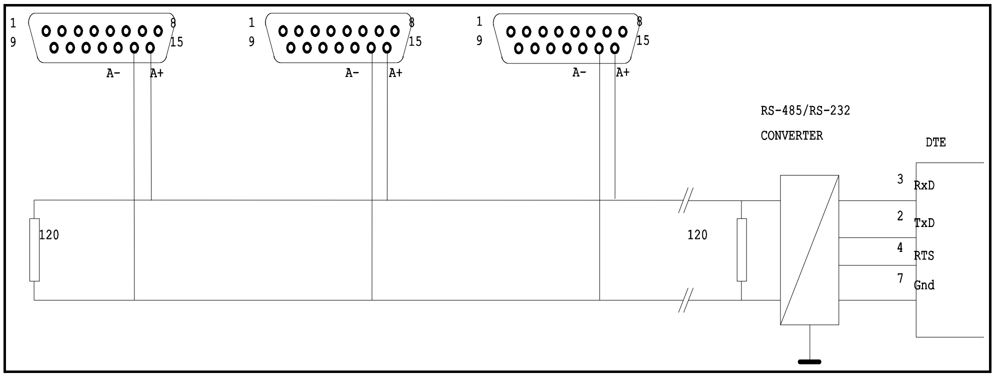


Figure 4-3 RS-485 Multi Drop Interconnection DMFM/C and PC

5. Message Protocol Structure

5.1. Introduction

HART is a "master-slave" protocol: each message transaction is originated by the master (central) station, whereas the slave (field) device only replies when it receives a command message addressed to it. The reply from the slave device will acknowledge that the command has been received and it may contain the data requested by the master.

Brooks RS485 on 4800 Series devices do not guarantee the timing required to support multiple masters communicating simultaneously to slave devices as defined by the HART Communications Foundation.

Brooks RS485 on 4800 Series devices do not support Burst Mode.

5.2. Addressing Concept

HART utilizes two possible addressing modes: short frame addressing and long frame addressing. The short frame addressing uses a one byte address of which the least significant nibble (four bits) is used to indicate the slave address. Because slave address 0 is reserved as a broadcast address, this provides the possibility to attach up to 15 different field devices and one master device on one multi drop bus. The long frame addressing mode uses 5 bytes (40 bits) as an address of which 38 bits are used to indicate the slave device. The slave address is build up from the manufacturer code (1 byte), the device type code (1 byte) and a device identification number (3 bytes). Details on addressing are explained in Section 3.4.4.

5.3. Character Coding

HART messages are coded as a series of 8-bit characters or bytes. These are transmitted serially, using a conventional UART (Universal Asynchronous Receiver/ Transmitter). As in normal RS-232C and other asynchronous communication links, a start bit, a parity bit and a stop bit are added to each byte. These allow the receiving UART to identify the start of each character and to detect bit errors due to electrical noise or other interference. A HART character is build up from:

1 Start bit - 0 bit
8 Data bits
1 Odd parity bit
1 Stop bit - 1 bit

This sequence is summarized the figure below. Since HART is an asynchronous protocol, successive characters may be separated by idle periods (logical 1 level), but the idle period must not exceed 1 character time.

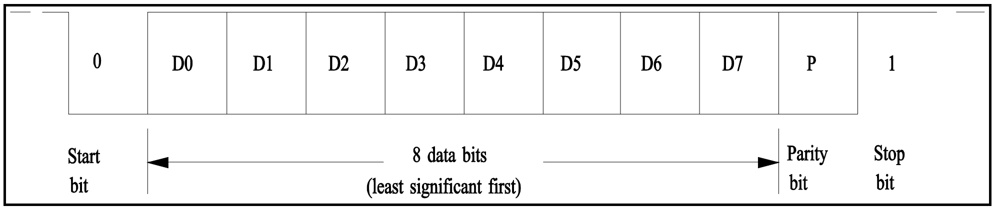


Figure 5-1Single Character Bit Sequence

5.4. Message Format

5.4.1. Message Structure

HART specifies a message structure which is given in the figure below.

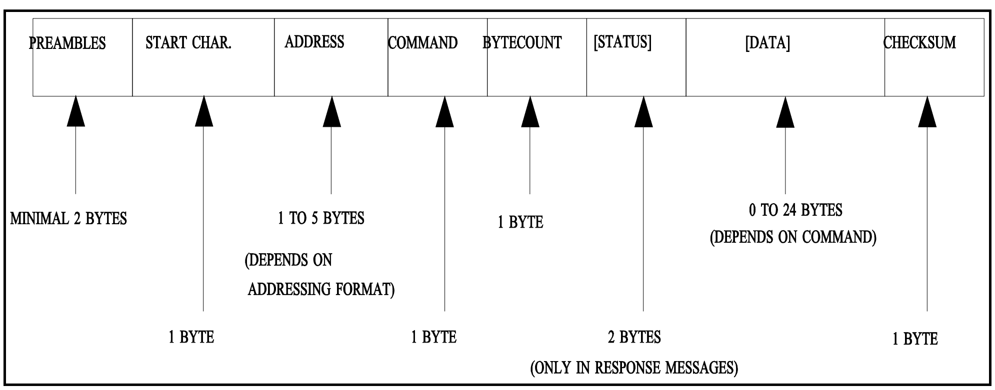


Figure 5-2 HART Message Structure

This structure is used for both the request (master to slave) and the response (slave to master) messages. The status part and the data part are shown in square brackets, because their occurrence in the message depends on the type of message (response or request message) and the command number. The individual items are explained below.

5.4.2. Preamble Characters

Every message, whether from a master or a slave device, is preceded by a specified number of hexadecimal FF characters (databyte with all 1's). These characters, called preamble characters, are used in the message-detect pattern together with the start character. The preamble characters are used to synchronize the field device. The RS485 on 4800 Series devices require at least 2 preamble characters in order to be able to proceed in the message detection with the start of message character. Note that due to potential losses due to RS-232 to RS-485 converters, a master should send a minimum of 5 preamble characters in order to guarantee that slave device receives the required 2 preamble characters.

5.4.3. Start Character

The start character or delimiter is a one byte code used to detect the type of frame (type of message) being transmitted and the type of addressing being used. The most significant bit indicates the addressing mode used: 0 for short frame and 1 for long frame addressing, whereas the three least significant bits indicate the frame type of the message: 010 indicates a Start-Of-Text character and 110 indicates an Acknowledge character. The Start-Of-Text character is used to indicate a message from the master to a slave device whereas the Acknowledge character is used to indicate the response messages from slave devices to the master. The rest of the bits in the character are all zeros. See figure and table below.

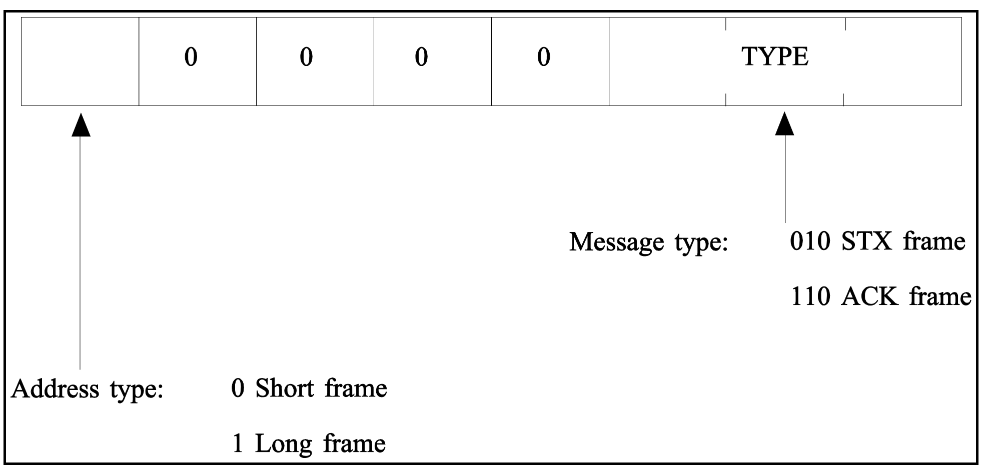


Figure 5-3 Start Character Settings

Table 5-1 Start Character Codings (hexadecimal)

	Short frame	Long frame
Master to slave (STX)	02	82
Slave to master (ACK)	06	86
Address field length	1 byte	5 bytes

5.4.4. Address Characters

The address field contains both the master and the field device addresses for the message. These may be contained in a single byte (short frame format) or in five bytes (long frame format). In either format, the most significant bit is usually the single-bit address of the master device taking part in the message transaction (either sending a command or receiving a reply from a slave device). Since only two masters are allowed only one bit is needed for the master address. This bit will be 1 if it indicates the primary master system, and 0 if it indicates the secondary master system. The rest of the address field is determined by the frame format.

The figure below shows the address character in the short frame format. The 4 least significant bits are the slave address, which can be used as a polling address.

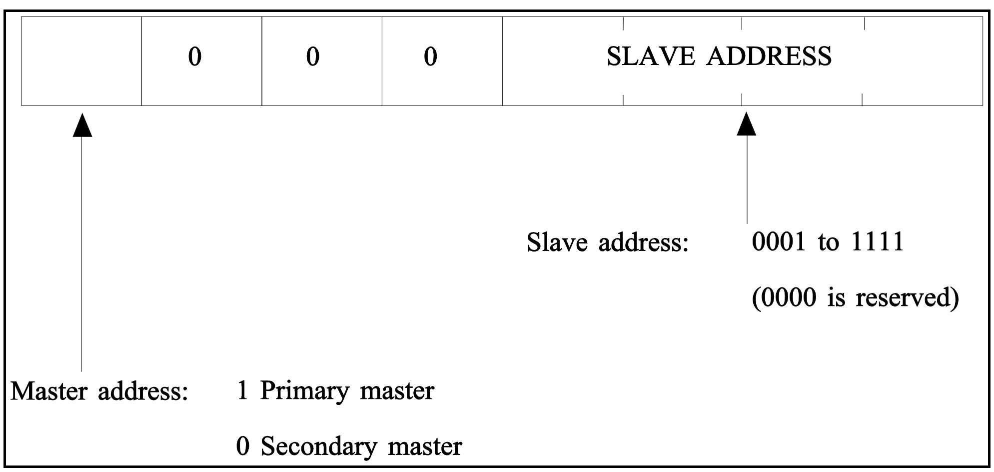


Figure 5-4 Short Frame Address Character

In the long frame format the slave device address is represented by a 38-bit number. The structure of the address is given in the figure below.

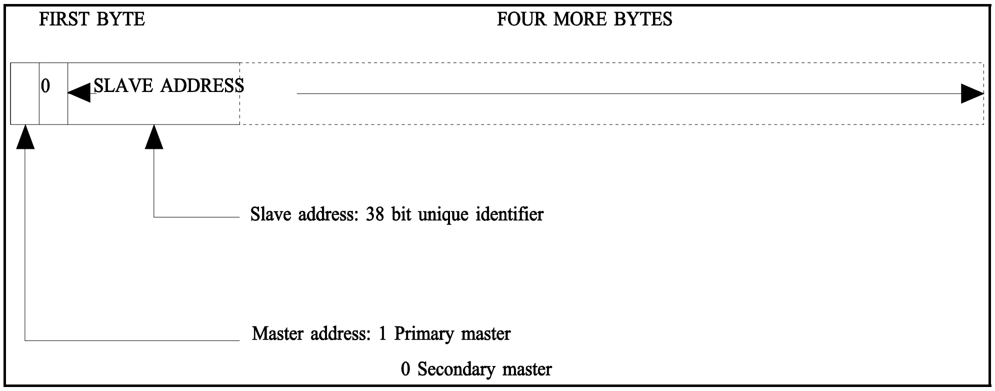


Figure 5-5 Long Frame Address Characters

In the long frame format the slave address part of the five address characters is build up from three sources: The 6 bits of the first byte of the slave address part represent the manufacturers code. In case of devices made by Brooks Instrument this is the number 10 (decimal). The manufacturer number is a number which is stored in the device by the manufacturer and which can not be changed by the user.

The second byte in the address is the device type code. This code indicates the type of the device addressed. The device type code will be 70 for all RS485 on 4800 Series devices. The device type code is a number which is stored in the RS485 on 4800 Series devices by the manufacturer.

The last three bytes form a 24-bit unique identification number. As the name implies, this value must be unique to each RS485 on 4800 Series device on a network. This number is set by the manufacturer and can be changed by using Command #122: Write Device Identification Number (NON-PUBLIC).

A special case occurs when all bits of the slave address part are set to 0. A message with this type of address, called a broadcast address, will be accepted by all slave devices attached to the bus. A slave device will always respond to a message with broadcast address unless the message contains additional information in the data portion of the message that allows the slave device to determine that the message is not addressed to that device. The RS485 on 4800 Series device supports only one such command, Command #11. This type of addressing can be used to address devices of which the manufacturer and the device type codes and the unique identification number are not available to the host system and with which this information can still be retrieved from the unknown device. Command #11 data contains a Tag Name. Only a slave device with the specified Tag Name will respond to Command #11 even if the address in the message is the broadcast address.

5.4.5. Command Character

The command character is a 1 byte unsigned integer in the range from 0 to 255 (decimal), which indicates the action the slave device has to perform. A larger range of commands is theoretically possible by using the expansion code or 254 (decimal) followed by a second byte. This feature however is not implemented by the RS485 on 4800 Series devices. The received command is echoed back by the slave device in its reply to the master.

Three types of commands are available to the user: the 'universal commands', the 'common-practice' commands and the 'transmitter-specific' commands. The universal commands are a number of commands in the range from 0 to 19, which are implemented by all field devices utilizing the HART protocol. Refer to Section 6 for descriptions of all available universal commands. The common-practice commands are a number of commands in the range from 31 to 127, which can be implemented by all devices. These commands perform tasks which are often common to most devices. Refer to Section 7 for descriptions of all implemented common-practice commands. The last category, the transmitter-specific commands are a number of commands, ranging from 128 to 250 which are specific to the type of device. Refer to Section 8 for descriptions of all available transmitter-specific commands. The Commands #251 to #255 are reserved.

5.4.6. Byte Count Character

The byte count character is a 1 byte unsigned integer indicating the number of bytes which will form the remainder of the message. This number includes the two status bytes (only if the message is a response message) and the bytes in the data part. It does NOT include the checksum byte. The byte count character is used by the receiving device to identify the checksum byte and to determine when the message is finished.

5.4.7. Status Characters

Status characters consists of two bytes, which contain bit-coded information about communications errors, command errors, and device status as defined in Table 3.2. Only response messages from the slave device to the master device will contain status characters..

Table 5-2 Status Bytes Coding

	First byte	**Second byte**
Communication errors	Bit 7 1 = Communication error	Bit 7 0
	Bit 6 Parity error (hex C0) Bit 5 Overrun error (hex A0) Bit 4 Framing error (hex 90) Bit 3 Checksum error (hex 88) Bit 2 Reserved (hex 84) Bit 1 Rx buffer overflow (hex 82) Bit 0 Undefined	Bit 6 Bit 5 Bit 4 All 0 Bit 3 Bit 2 Bit 1 Bit 0
Command errors	Bit 7 0 = Command errors	Bit 7 Device malfunction
	Bits 6 to 0 (not bit-mapped): 0 No command specific error 1 Undefined 2 Invalid selection 3 Passed parameter too large 4 Passed parameter too small 5 Incorrect byte count 6 Transmitter specific command error 7 In write-protect mode 8 - 15 Command specific errors 16 Access restricted 32 Device is busy 64 Command not implemented	Bit 6 Configuration changed Bit 5 Cold start Bit 4 More status available. Use Command #48 to get more information Bit 3 Primary variable analog output fixed Bit 2 Primary variable analog output saturated Bit 1 Non-primary variable out of range Bit 0 Primary variable out of range

If the communication failed (i.e. the slave received distorted information) the first byte indicates the receiver error(s) of the slave device. The second byte will then be 0. If communication did not fail, the first byte will give command execution information, whereas the second byte will give information on the status of the device.

5.4.8. Data Characters

For the commands that contain data, the data field may contain up to a maximum of 24 8-bit data bytes. The data can appear in a number of formats:

5.4.9. 8-bit Unsigned Integer Format

This format can be used to transfer codes (e.g unit codes), indexes (e.g analog output numbers) and raw data. If a parameter, represented by an 8-bit unsigned integer in a command data part is not implemented, codes like 250, "Not Used" or 0 will be used.

5.4.10. 24-bit Unsigned Integer Format

This format can be used to transfer large integer data numbers (e.g. the valve values). The command specific errors 8 - 15 are errors which can have a different meaning for different commands. Refer to the Sections 6, 7 and 8 for more information.

5.4.11. IEEE 754 Floating Point Format

This format is based on the IEEE 754 single precision floating point standard:

Table 5-3 IEEE 754 single precision floating point standard

S EEEEEEE	E MMMMMMM	MMMMMMMM	MMMMMMMM
byte #0	byte #1	byte #2	byte #3

Where:

S - Sign of mantissa (1 = negative)
E - Exponent; Biased by 127 in two's complement format
M - Mantissa; 23 least significant bits, fractional portion

The value of a parameter described in the above format can thus be found by:

Value = S 1.M * $2^{(E - 127)}$

This format is also used in most personal computers.

The floating point parameters not used by a device will be filled with 7F A0 00 00 (hexadecimal) or 'Not-A-Number'.

5.4.12. ASCII Data Format

Some of the alphanumeric data passed by the protocol is transmitted to and from the devices in the ASCII format. Refer to any ASCII Code table for the alphanumeric code assignments.

5.4.13. Packed-ASCII (6-bit ASCII) Data Format

Some of the alphanumeric data passed by the protocol is transmitted to and from the devices in the Packed-ASCII format. Packed-ASCII is a subset of ASCII produced by removing the two most significant bits from each ASCII character. This allows four Packed-ASCII to be placed in the space of three ASCII characters. Typically four Packed-ASCII strings are even multiples of three bytes. The figure below illustrates the byte sequence.

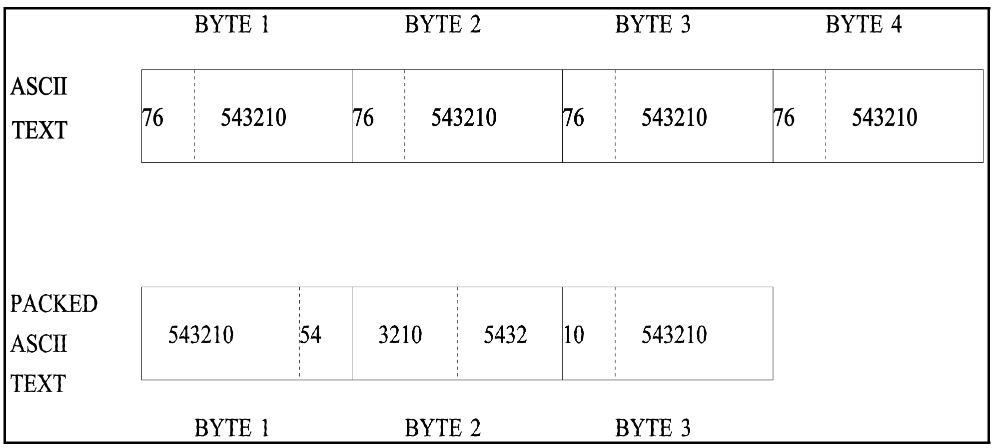


Figure 5-6 Packed-ASCII Construction

Construction of Packed-ASCII:

a. Remove bit #7 and bit #6 from each ASCII character.
b. Pack four 6-bit ASCII bytes into three bytes.

Reconstruction of ASCII characters:

a. Unpack the four 6-bit ASCII characters into four bytes.
b. Place the complement of bit #5 of each unpacked 6-bit ASCII character into bit #6.
c. Set bit #7 of each unpacked ASCII to zero.

Table 5-4 Packed-ASCII Codes

Char	Code	Char	Code	Char	Code	Char	Code
@	00	P	10	(space)	20	0	30
A	01	Q	11	!	21	1	31
B	02	R	12	“	22	2	32
C	03	S	13	#	23	3	33
D	04	T	14	$	24	4	34
E	05	U	15	%	25	5	35
F	06	V	16	&	26	6	36
G	07	W	17	‘	27	7	37
H	08	X	18	(	28	8	38
I	09	Y	19	)	29	9	39
J	0A	Z	1A	*	2A	:	3A
K	0B	[	1B	+	2B	;	3B
L	0C	\	1C	,	2C	<	3C
M	0D	]	1D	-	2D	=	3D
N	0E	^	1E	.	2E	>	3E
O	0F	_	1F	/	2F	?	3F

5.4.14. Checksum Character

The checksum byte contains the 'exclusive-or' ('longitudinal parity') of all the characters preceding it in the message starting with the start character. It provides a further check on transmission integrity, beyond the one provided by the parity check on each individual byte. The exclusive-or of all the message bytes (including the start character, excluding the checksum byte) and the checksum byte itself should read exactly zero.

6. Master Slave Communication

6.1. Introduction

Section 5. Message Protocol Structure defined the S-Protocol message structure in detail. This section will describe how to utilize the S-Protocol message structure to perform master slave communications with a Brooks RS485 device, focusing on RS-485 line handling, establishing communications with a device, error recovery, and timing. Sections 4, 4, and 4 define all S-Protocol commands available in Brooks 4800 Series devices. This section will conclude with examples of typical communications sequences.

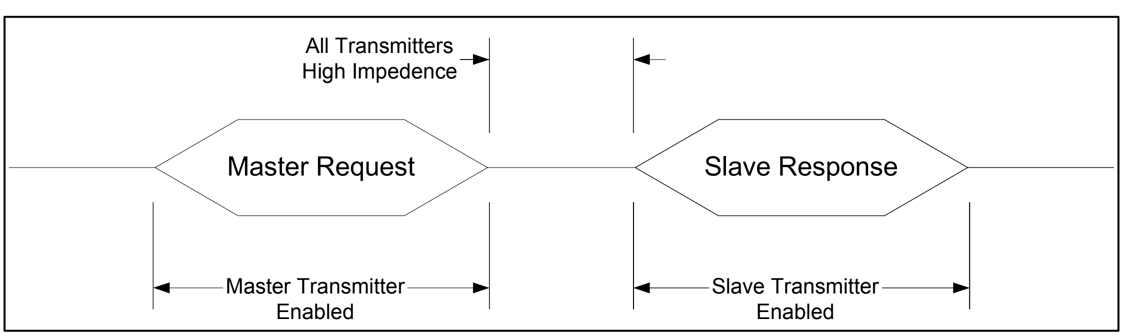
Master devices initiate all communications on a Master/Slave communications network. Master devices are typically a computer of some kind but other devices such as PLCs can also operate as a Master device.

Slaves devices only respond to messages initiated by a Master. Brooks 4800 Series devices are always Slaves on the communications network.

6.2. RS-485 Line Handling

The physical communications layer used by Brooks 4800 Series devices is RS-485. On an RS-485 physical communications layer, all data is transmitted and received using differential signals on a single pair of wires. Since both the Master and the Slave devices use the same pair of wires to transmit their data, care must be taken to ensure that only one device has its transmitter enabled at any point in time.

Figure 8 shows a typical message exchange using RS-485. Notice that the Master's transmitter is enabled only during the Master Request message and the Slave's transmitter is enabled only during the Slave Response message. At all other times, the transmitters on the Master and all Slaves connected to the network must be in their high impendence state and thus the network is "un-driven."



It is the users responsibility to guarantee that the Master's transmitter is enabled only during the Master Request message. Control of the Master's transmitter is dependent upon the hardware used by the Master. If an RS-232 to RS-485 converter is used, the most common control is the RTS signal on the RS-232 interface as shown in Figure 4-3). Refer to the user manual for your hardware to determine the proper control method required in your system.

Figure 6-1 - RS 485 Line Handling

Timing of the enabling/disabling of the transmitter is very important. The transmitter must be enabled before the first bit of the first character is transmitted and must be disabled only after the last bit of the last character is transmitted. Additionally, all transmitters have some finite turn-on/turn-off delays which may be affected by the wire length and wire quality of your network. The S-Protocol message structure attempts to minimize these affects by requiring all messages to have at least 5 preamble characters while only 2 are required for the receiving device to detect a valid message (see Section 5.4.2.). This allows up to 3 lost characters due to turn-on/turn-off delays.

Disabling a transmitter at the proper time is frequently a difficult task. Many UARTS/systems do not provide an indication when the last byte of a message is completely transmitted. It is more likely that an indication is provided when the last byte of a message is starting to be transmitted. Since the last byte of an S-Protocol message is the checksum byte for the message, it is critical that the transmitter remain enabled until the last byte is completely transmitted. One solution is to transmit an extra character at the end of a message (typically 0x00) and then disable the transmitter when the indication is received that the extra character is starting to be transmitted. However, the transmitter cannot be enabled too long after a message is complete. Slave devices will begin transmitting a response as soon as 5 ms after the reception of an error free request message.

High data rates increase the importance of disabling the transmitter quickly. At 19200 baud, one character time is 0.57 ms. Thus, the 3 lost character "cushion" represents only 1.72 ms. While the response of a Brooks 4800 Series device is always at least 5 ms regardless of the data rate, lower data rates provide a longer "cushion" and thus is a possible solution if disabling the transmitter in a timely manner proves difficult. Another solution is to increase the number of preamble characters transmitted by the Master and/or the slave. To increase the number of preambles in a Brooks 4800 Series device, you should use Command #59: Write Number of Response Preambles.

6.3. Establishing Communications with a Device

In order for a Master to establish communications with a Brooks RS485 device, the Master must know the address of the Brooks RS485 device. The S-Protocol supports both Short Frame Addressing and Long Frame Addressing as defined in Section 5.2. .

Short Frame Addressing allows a master to communicate with up to 15 devices. Each device on the network must have a unique Polling Address with a value of 1 – 15. The Polling Address must be set in each device using Command #6: Write Polling Address.

Long Frame Addressing allows a master to communicate with up to 16,777,215 devices on a wide area network (RS-485 has a limit of 32 devices per daisy chain). Each device is pre-programmed with a unique long address derived from the serial number of the 4800 Series device it is mounted on. Using the process described below, the Master can obtain the long address from the device by knowing only the device Tag Name. The Tag Name is derived from the serial number of the base 4800 Series unit where the module is mounted on. The last 8 characters of this serial number is taken as the Tag Name. The Tag Name can be changed by the user using command #18.

The following procedure can be performed online in order to obtain a device's long address:

Send Command #11 (see section 8.6.) using Long Frame Addressing and an address of 0. In the data section of Command #11, use the device's Tag Name to identify the device. Command #11 requires that the Tag Name be transmitted in Packed-ASCII format as defined in section 5.4.13. .

Extract the Manufacturer ID, Manufacturer's Device ID, and Device ID Number from the response and construct the Long Address Frame as shown in Figure 9.

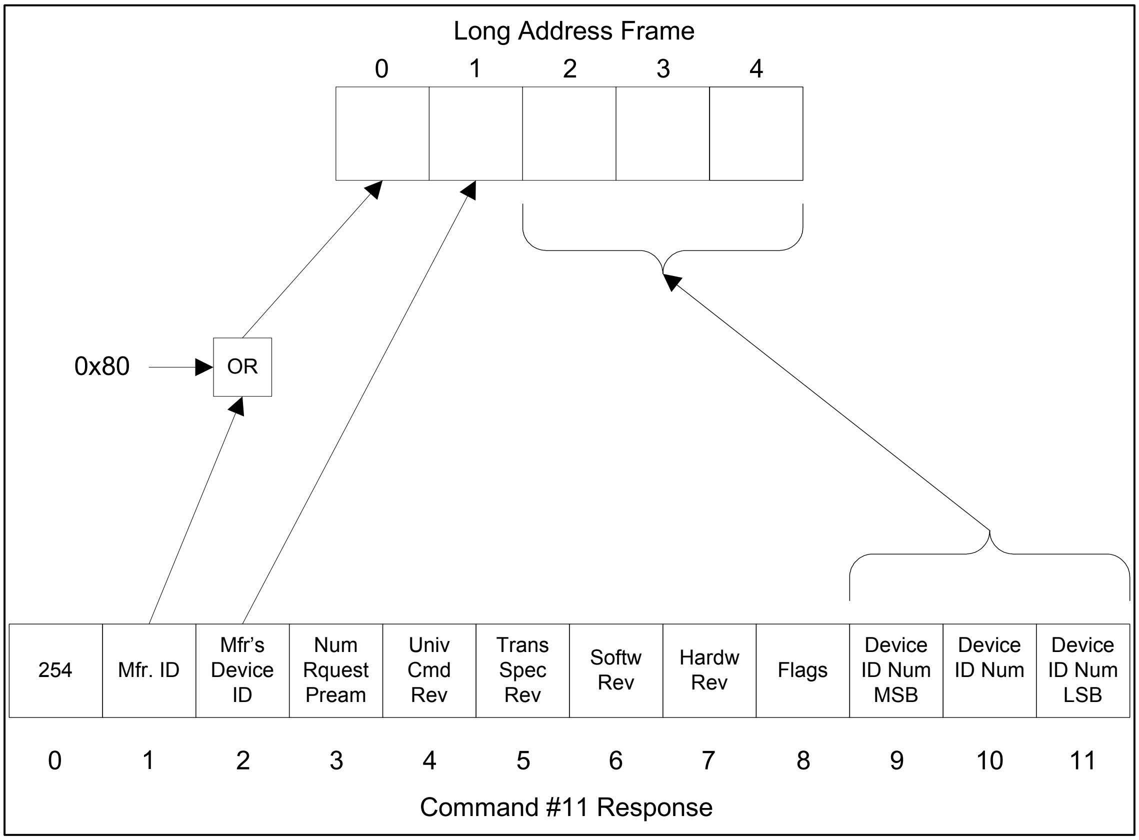


Figure 6-2 Command 11 Response to Long Frame Address

6.3.1. Example of Using Command #11

Command #11 reads the unique identifier from a device whose Tag Name is specified in the Command #11 request from the Master. Tag Names are strings of up to 8 characters which are limited to the reduced ASCII set defined in section 5.4.13. . Table 2 is an example of converting an 8 character Tag Name to 6 bytes in the Packed-ASCII format. In this example, the Tag Name of the device will be "MFC-1234".

Table 6-1 Converting Tag Name to Packed ASCII

	Representation							
Tag Name	MFC-1234							
Characters	M	F	C	-	1	2	3	4
8- bit ASCII (hex)	4D	46	43	2D	31	32	33	34
Bit 7 & 8 removed: 6 bit ASCII (hex)	0D	06	03	2D	31	32	33	34
6 bit ASCII (binary)	001101	000110	000011	101101	110001	110010	110011	110100
Packed (binary)	00110100 0110 0000 11101101 11000111 00101100 11110100							
Packed (hex)	34 60 ED C7 2C F4							

Figure 11 shows the request message for Command #11 sent by the Master to a Brooks RS485 device whose Tag Name is MFC-1234.

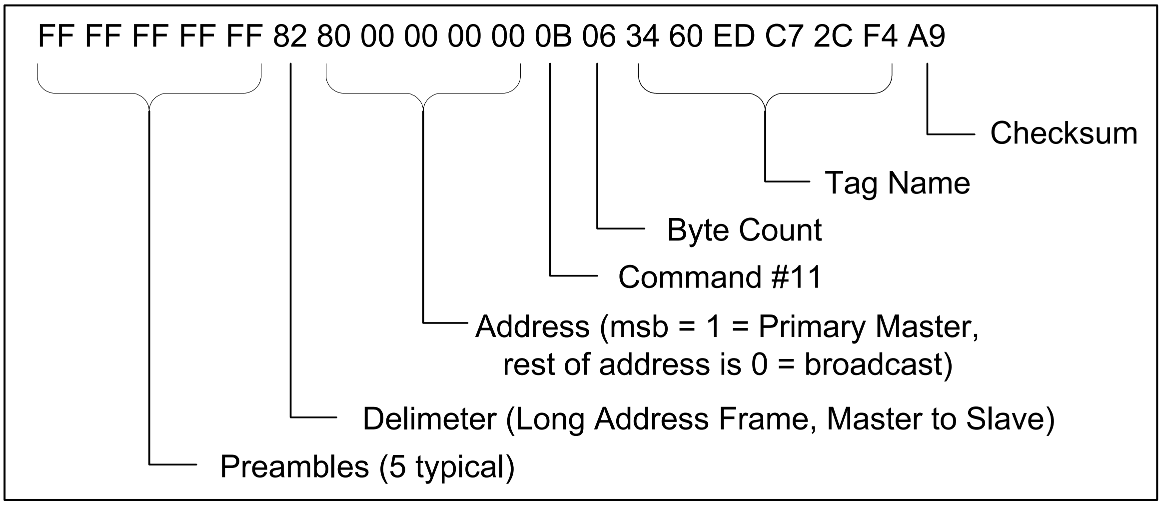


Figure 6-3 Example of Command #11 with Tag Name is MFC-1234

A possible response message from a Brooks RS485 device is shown in Figure 12

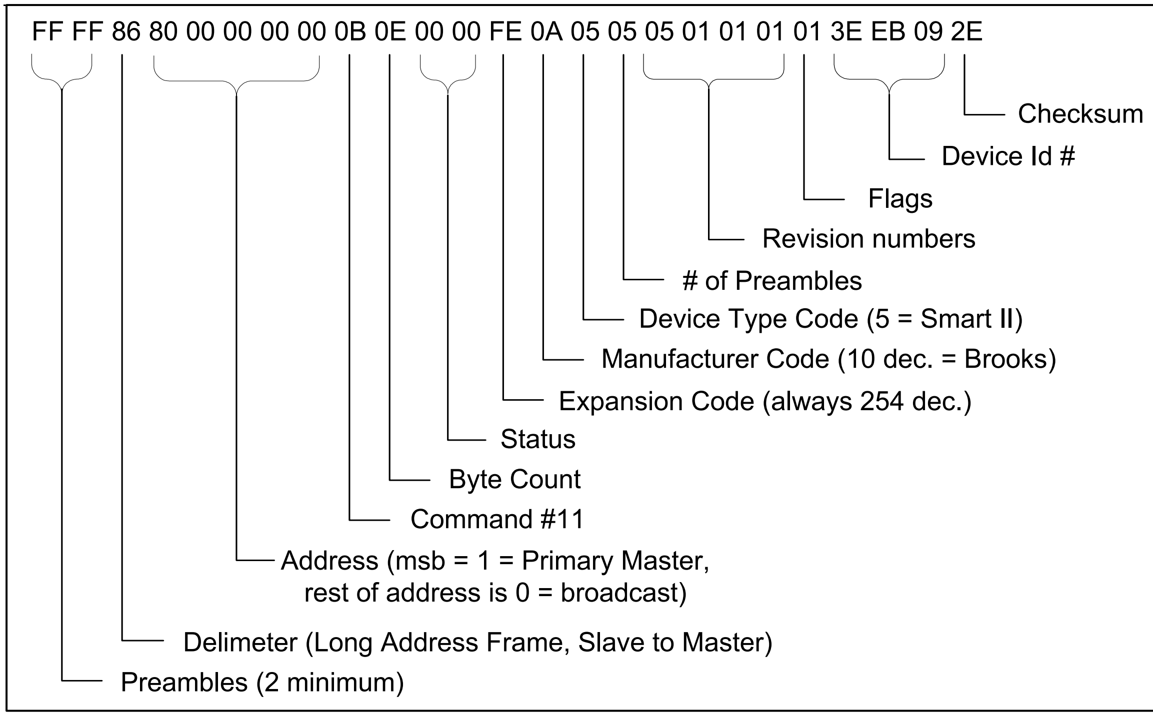


Figure 6-4 Command #11 Response

From the response, the long address can be extracted as shown in Figure 13.

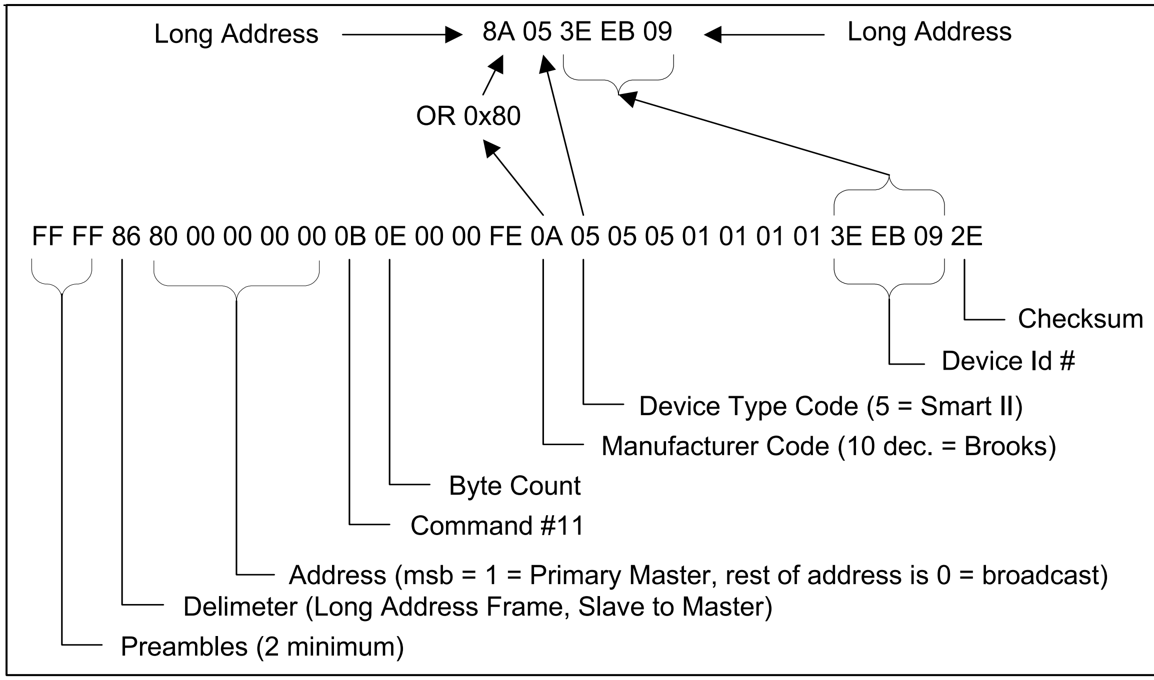


Figure 6-5 Extracting the Long Address

6.4. Alarm Configuration and Monitoring

Brooks RS485 devices monitor for various alarm conditions such as Flow Rate and Diagnostics. To determine which alarm conditions have been detected, use Command #48 (see section 7.3). However, it is not necessary to constantly poll Command #48 to determine when an alarm condition has been detected. All slave response messages contain a 2 byte status. If an alarm condition has been detected, then bit 4 of the second status byte will indicate “More Status Available”. Then Command #48 can be used to determine the alarm condition(s) that has been detected.

To configure which alarm conditions are monitored and reported by the device Refer to Commands 245, 246, 247, and 248 and also Section 11.14.
‘

Additional Device Status and Masking'.

6.5. Error Handling

In all communications networks, communications errors can and will occur. Both the Master and the Slave devices must be able to properly handle errors in order to maintain a operating network. When a Brooks RS485 device detects a communications error, it may respond with an error code or it may not respond at all to the request depending upon the type of error that was detected and where in the message the error was detected. The reasons that a device responds or does not respond after an error is detected is not important. What is important is that the Master handle situation correctly.

There are two basic type of errors defined by the S-Protocol: Communications Errors and Command Response errors. The type of error can be determined by examining the Status Code returned by the slave device (see section 3.4.7). Command Response errors are typically the result of a programming error in the Master and should not normally occur in a mature system. The main focus of this section will be Communication Errors.

Communications Errors are frequently the result of external environment issues, faulty wiring, etc. In a properly designed network, Communications Errors should be rare. A Communications Error can occur in either the Master to Slave Request or the Slave to Master response. If the error occurs in a Master to Slave request, the Slave may respond with an error code or it may not respond at all to the request depending upon the type of error that was detected and where in the message the error was detected. It is the responsibility of the Master device to check all Slave to Master responses for errors including message frame formatting, longitudinal parity, and vertical parity.

Regardless of the type of error and when or where it was detected, the normal way to handle a Communications Error is to simply retry the message. Typically, a master would attempt to retry a message at least twice to allow any external disturbance to clear. In the event that the retries are unsuccessful, then the Master device must handle the situation in a manner consistent with the requirements of the system. Typical responses to such an error are: Taking the device off-line so that the remainder of the network is not affected; Notifying an operator; Triggering a system alarm; etc.

A Master device must allow sufficient time for a Slave to respond before attempting to retry the message. The average response time for a Brooks RS485 device is approximately 7 ms, but it is possible to for the response to be as along as 25 ms. The Master should wait 4 times the maximum response time (100 ms) before retrying the message. As long as communications are infrequent, this retry delay time should not affect system performance.

6.6. Examples

The following 2 examples show the most typical messages used by a Master when communicating to a Brooks RS485 Series device: Reading Flow Rate and Setting the Setpoint. These examples will use the Long Addressing Frame with the long address established in the example in section 4.2.1.

6.6.1. Reading Flow Rate

The flow rate of the device can be read using any of the following commands:

- Command #1 – Read Primary Variable
- Command #2 – Read Primary Variable Current and Percent of Range
- Command #3 – Read Current and All Dynamic Variables

This example will use Command #1 to read the Flow Rate of the device. This command returns the flow rate in the unit of measure as configured in the device. The units can be changed using Command #196, Select Flow Unit.

In the example shown in Figure 14, the device returns a flow of 0.8502 liters/min.

Master to Slave Request

FF FF FF FF FF 82 8A 05 3E EB 09 01 00 D0

Checksum
Byte Count (No data in this request)
Command #1
Long Address
Delimeter (Long Address Frame, Master to Slave)
Preambles (5 typical)

Slave to Master Response

FF FF 86 8A 05 3E EB 09 0B 07 00 10 11 3F 59 A6 B5 AD

Checksum
Flow rate in IEEE 754 Floating Point (0.8502)
Units Code (Liters per Minute)
Status Code (More Status Available)
Byte Count
Command #11
Long Address
Delimeter (Long Address Frame, Slave to Master)
Preambles (2 minimum)

Figure 6-6 Reading Flow Rate Example

6.6.2. Setting the Setpoint

The Setpoint can be controlled via the network using Command #236. In the example shown in Figure 14, the setpoint is set to 85% of full scale.

If Setpoint is controlled via an analog input, then Setpoint can be read using Command #235.

When Command #236 is received by a Brooks 4800 Series device, the Setpoint Source is automatically changed to digital mode. Setpoint source can be changed back to analog by using Command #216 or by cycling power to the device.

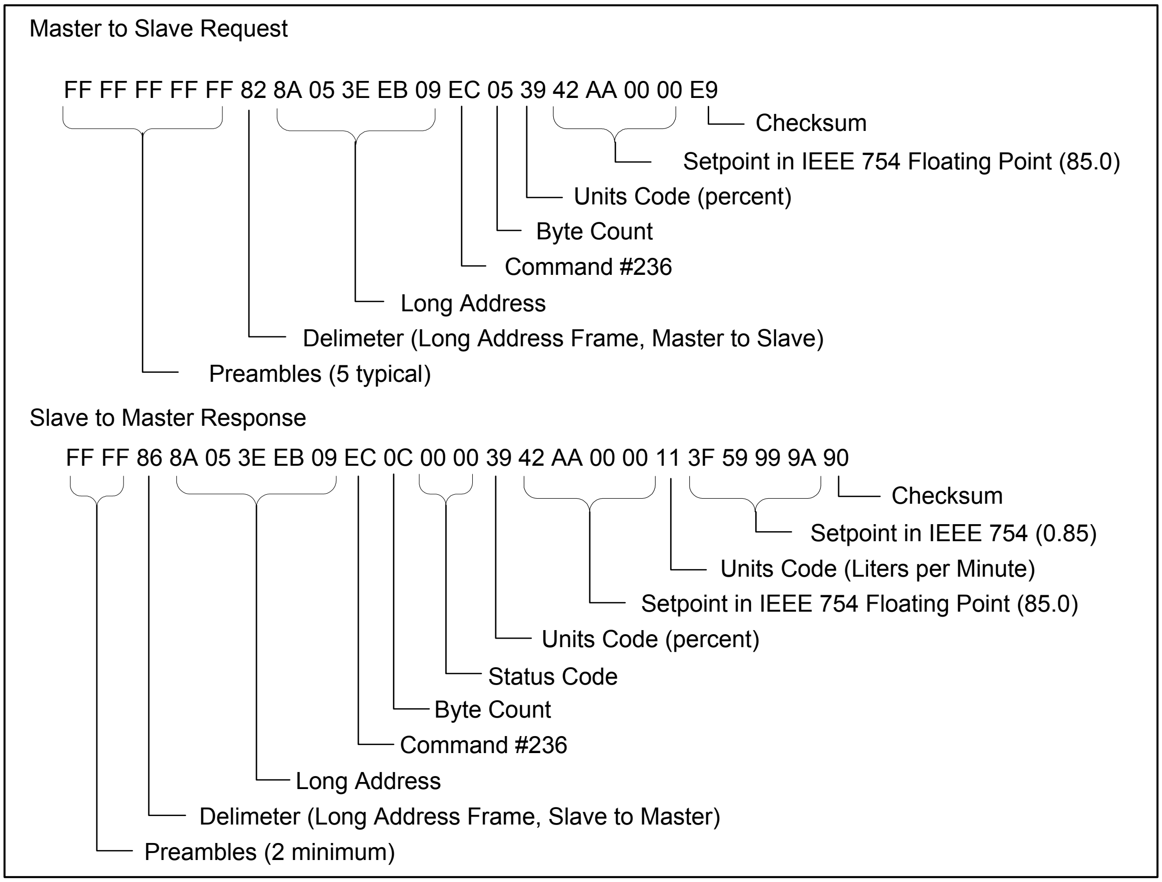


Figure 6-7 Writing Setpoint Example

THIS PAGE WAS INTENTIONALLY LEFT BLANK

7. General Transmitter Information

7.1. Referenced Documents

The following HART documents where referenced in order to implement the protocol.

Data Link Layer Specification Rev.	D8900098	Rev 7.0
Command Summary Information Rev.	D9000035	Rev 7.0
Command-Specific Response Code Defs. Rev.	D8900077	Rev 3.0
Universal Command Specification Rev.	D8900038	Rev 5.1
Common-Practice Command Specification Rev.	D9000037	Rev 7.0
Common Tables Rev.	D9000038	Rev 5.0

7.2. Unit Conversions

7.2.1. Flow Rate Conversions

All flow values involved in the exchange of data during communication are converted to/from the user specified flow units. A list of supported flow units is provided in section 11.3. . The supported flow units can divided into two categories: Mass flow units and Volumetric Flow units. The user can change the flow units to be used for all flow rate conversions with Command #196

The 4800 Series devices internally compute flow in volumetric units at the reference condition as defined at device calibration. In order to compute flow in mass units, the device must know the density of the gas flow through the device. Density is stored in the device at calibration.

Volume flow units are always reported at specific reference conditions. Using Command #196, the user can select reference condition type from 3 options as listed in section 11.3. .

- Normal - reference conditions of 0 °C and 1 atmosphere. (273.15 degrees K/ 101325 Pascals).
- Standard – user specified reference conditions. Use

Command #191: Write Standard Temperature and Pressure, to change these reference conditions.

- Calibration – reference conditions used at calibration.

Reference condition conversions are done using the Boyle-Gay-Lussac law

$$\frac{P_1 \bullet V_1}{T_1} = \frac{P_2 \bullet V_2}{T_2} = Constant \quad (1)$$

where P is pressure, T is temperature and V is volume (per unit of time). The indexes 1 and 2 represent the two different reference conditions. This results in the reference conversion formula

$$V_2 = \frac{(P_1 \bullet T_2)}{(P_2 \bullet T_1)} * V_1 \quad (2)$$

Where applicable the conversion factors are taken from *The handbook of Chemistry and Physics, 60th edition*, R.C. Weast (Ed.), CRC Press Inc., Cleveland, Ohio.

7.2.2. Temperature Conversions

All temperature values involved in the exchange of data during communication are converted to/from the user specified temperature units. A list of supported temperature units is provided in section 11.5. . The user can change the temperature units to be used for all temperature conversions with Command #197.

8. Universal Command Specifications

8.1. Command #0: Read Unique Identifier

Command used to retrieve the expanded device-type codes, revision levels and the device identification number from the specified device. The device type code will always be returned in the expanded three byte format (i.e. "254", manufacturer identification code, manufacturers device type code). The combination of the manufacturer identification code, manufacturer's device type code and device identification code make up the unique identifier for the extended frame format of the data link layer.

Request data bytes:

NONE

Response data bytes:

254	MFR. ID	MFR'S DEVICE TYPE	NUMBER RQUEST PREAM	UNIV. CMD. REV	TRANS. SPEC. REV.	SOFTW. REV.	HARDW. REV	FLAGS	DEVICE ID NUM MSB	DEVICE ID NUM	DEVICE ID NUM
#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11

Data Byte #	***Type***	***Remarks***
0	8-bit unsigned integer	Device type code for "expansion". Contains the code "254" (decimal)
1	8-bit unsigned integer	Manufacturer identification code (Always 10).
2	8-bit unsigned integer	Manufacturers device type code Refer to Section 11.2. 'Device type codes'
3	8-bit unsigned integer	Number of response preamble characters. Required for the request message from master to the slave.
4	8-bit unsigned integer	Universal command revision level implemented by this device
5	8-bit unsigned integer	Transmitter specific command revision level implemented by this device
6	8-bit unsigned integer	Software revision level of the device
7	8-bit unsigned integer	Hardware revision level of the electronics in the device. Format: xxxxx.yyyB x Device hardware revision, 5-bit unsigned integer level 15 is reserved y Physical signaling code, 3-bit unsigned integer refer to Section 11.8. 'Physical Signaling Codes'
8	8-bit unsigned integer	Flags
9-11	24-bit unsigned integer	Device identification number

Command specific response codes:

0	No command-specific errors
1 - 4	Undefined
5	Incorrect byte count
6 - 127	Undefined

8.2. Command #1: Read Primary Variable

Read the primary variable. The primary variable is the flow rate of the device expressed in the selected flow units. The selected flow reference is not incorporated in this command.

Request data bytes:

NONE

Response data bytes:

SEL. FLOW. UNIT	FLOW RATE MSB	FLOW RATE	FLOW RATE	FLOW RATE LSB
#0	#1	#2	#3	#4

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Primary variable unit code Refer to Section 11.3. ‘Flow Rate Unit and Reference Codes’
1-4	32-bit floating point IEEE 754 format	Primary variable flow rate

Command specific response codes:

0	No command-specific errors
1 - 4	Undefined
5	Incorrect byte count
6 – 127	Undefined

8.3. Command #2: Read Primary Variable Current and Percent of Range

Read the primary variable, flow rate as current or voltage and as a percent of the primary variable range. The current/voltage field reports current in mAmps or voltage in volts depending upon the configuration of the output of the device. The current/voltage always matches the analog output of the device including alarm conditions and set values. Percent of range always follows the primary variable, even if the current is in an alarm condition or set to a value. Also, the percent of range is not limited to values between 0% and 100%, but tracks the primary variable to the sensor limits.

Request data bytes:

NONE

Response data bytes:

CURRENT /VOLTAG E MSB	CURRENT /VOLTAG E	CURRENT /VOLTAG E	CURRENT /VOLTAG E LSB	PV % RANGE MSB	PV % RANGE	PV % RANGE	PV % RANGE LSB
#0	#1	#2	#3	#	#5	#6	#7

Data Byte #	Type	Remarks
0-3	32-bit floating point IEEE 754 format	Analog output current or voltage [milliamperes or volts]
4-7	32-bit floating point IEEE 754 format	Primary variable flow rate [% of range]

Command specific response codes:

0	No command-specific errors
1 - 4	Undefined
5	Incorrect byte count
6 – 127	Undefined

8.4. Command #3: Read Current and all Dynamic Variables

Read the current and the dynamic variables. The current/voltage field reports current in mAmps or voltage in volts depending upon the configuration of the output of the device. The current/voltage always matches the analog output current/voltage of the device including alarm conditions and set values. For the 4800 Series devices, the dynamic variable assignments are as follows:

Variable #0: Flow rate (primary variable)
Variable #1: Temperature (secondary variable)

Request data bytes:

NONE

Response data bytes:

CURRENT /VOLTAGE MSB	CURRENT /VOLTAGE	CURRENT /VOLTAGE	CURRENT /VOLTAGE LSB	PRIMARY VAR. UNITS	PRIMARY VAR. MSB	PRIMARY VAR.	PRIMARY VAR.	PRIMARY VAR. LSB
#0	#1	#2	#3	#4	#5	#6	#7	#8

SECOND. VAR. UNITS	SECOND. VAR. MSB	SECOND. VAR.	SECOND. VAR.	SECOND. VAR. LSB
#9	#10	#11	#12	#13

Data Byte #	Type	Remarks
0-3	32-bit floating point IEEE 754 format	Analog output current/voltage [milliamperes or volts]
4	8-bit unsigned integer	Primary variable unit code Refer to Section 11.3. 'Flow Rate Unit and Reference Codes'
5-8	32-bit floating point IEEE 754 format	Primary variable flow rate
9	8-bit unsigned integer	Secondary variable unit code Refer to Section 11.5. 'Temperature Unit Codes'
10-13	32-bit floating point IEEE 754 format	Secondary variable: temperature

Command specific response codes:

0	No command-specific errors
1 - 4	Undefined
5	Incorrect byte count
6 - 127	Undefined

8.5. Command #6: Write Polling Address

This command writes the polling address (the short frame address) to the field device. The address is used to control the analog output and provide a means of device identification in multi drop mode. The analog outputs respond to the applied process only when the polling address is set to 0. When the address assigned to a device is in the range of 1 through 15, the analog outputs will not be active and will not respond to the applied process. While the analog outputs are not active they will be set to their lowest value, being either 0 volt and 0 mA or 1 volt and 4 mA, depending on the selected range. The transmitter status bit #4, analog output fixed is set. If the polling address is set back to 0 the analog outputs will again become active and respond to the applied process.

Request data bytes:

POLLING ADDRESS
#0

Data Byte #	Type	Remarks	
0	8-bit unsigned integer	Polling address	
		0	Analog outputs active
		1-15	Analog outputs not active
		16-255	Undefined

Response data bytes:

POLLING ADDRESS
#0

Data Byte #	Type	Remarks	
0	8-bit unsigned integer	Polling address	
		0	Analog outputs active
		1-15	Analog outputs not active
		16-255	Undefined

Command specific response codes:

0	No command-specific errors
1	Undefined
2	Invalid selection

3 - 4	Undefined
5	Incorrect byte count
6	Undefined
7	In write protect mode
8 - 15	Undefined
16	Access restricted
17 - 127	Undefined

8.6. Command #11: Read Unique Identifier Associated with Tag

This command returns the expanded device-type codes, revision levels and the device identification number of a device containing the requested tag. It will be executed when either the appropriate unique identifier or the broadcast unique identifier, "00000" is received. The unique identifier in the response message of this command always contains the unique identifier received in the request message. This command is unique in that no response is made unless the tag matches that of the device. The device type code will always be returned in the expanded three byte format (that is: "254", manufacturer identification code, manufacturer's device type code).

Request data bytes:

TAG	TAG	TAG	TAG	TAG	TAG
#0	#1	#2	#3	#4	#5

Data Byte #	Type	Remarks
0-5	6 (8-bit) byte packed ASCII	Device tag number

Response data bytes:

254	MFR. ID	MFR'S DEVICE TYPE	NUMBER RQUEST PREAM	UNIV. CMD. REV	TRANS. SPEC. REV.	SOFTW. REV.	HARDW. REV	FLAGS	DEVICE ID NUM MSB	DEVICE ID NUM	DEVICE ID NUM
#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11

Data Byte #	Type	Remarks
0	8-bit unsigned integer	Device type code for "expansion". Contains the code "254" (decimal)
1	8-bit unsigned integer	Manufacturer identification code (Always 10).
2	8-bit unsigned integer	Manufacturers device type code Refer to Section 11.2. 'Device type codes'
3	8-bit unsigned integer	Number of response preamble characters. Required for the request message from master to the slave.
4	8-bit unsigned integer	Universal command revision level implemented by this device
5	8-bit unsigned integer	Transmitter specific command revision level implemented by this device
6	8-bit unsigned integer	Software revision level of the device
7	8-bit unsigned integer	Hardware revision level of the electronics in the device. Format: xxxxx.yyyB x Device hardware revision, 5-bit unsigned integer level 15 is reserved y Physical signaling code, 3-bit unsigned integer refer to Section 11.8. 'Physical Signaling

		Codes’
8	8-bit unsigned integer	Flags
9-11	24-bit unsigned integer	Device identification number

Command specific response codes:

0	No command-specific errors
1 - 4	Undefined
5	Incorrect byte count
6 – 127	Undefined

8.7. Command #12: Read Message

Read the message string contained within the device.

Request data bytes:

NONE

Response data bytes:

MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE
#0	#1	#2	#3	#4	#5	#6	#7	#8	#9

MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE
#9	#10	#11	#12	#13	#15	#16	#17	#18	#19

MESSAGE	MESSAGE	MESSAGE	MESSAGE
#20	#21	#22	#23

Data Byte #	Type	Remarks
0-23	24 (8-bit) byte packed ASCII	Device message

Command specific response codes:

0	No command-specific errors
1 - 4	Undefined
5	Incorrect byte count
6 – 127	Undefined

8.8. Command #13: Read Tag, Descriptor, Date

Read the tag, descriptor and date contained within the device.

Request data bytes:

NONE

Response data bytes:

TAG	TAG	TAG	TAG	TAG	TAG
#0	#1	#2	#3	#4	#5

MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE
#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17

DATE DAY	DATE MONTH	DATE YEAR
#18	#19	#20

Data Byte #	Type	Remarks
0-5	6 (8-bit) byte packed ASCII	Device tag number
6-17	12 (8-bit) byte packed ASCII	Device descriptor
18-20	3 (8-bit) byte packed ASCII	Date. Respectively day, month, year-1900.

Command specific response codes:

0	No command-specific errors
1 - 4	Undefined
5	Incorrect byte count
6 – 127	Undefined

8.9. Command #14: Read Primary Variable Sensor Information

This command is intended to read primary variable sensor information. For the 4800 Series devices these parameters are not applicable.

Request data bytes:

NONE

Response data bytes:

#0	#1	#2	#3	#4	#5	#6	#7
ZERO “0”	ZERO “0”	ZERO “0”	NOT-USED “250”	NAN “7F” (HEX)	NAN “A0” (HEX)	NAN “00” (HEX)	NAN “00” (HEX)

#8	#9	#10	#11	#12	#13	#14	#15
NAN “7F” (HEX)	NAN “A0” (HEX)	NAN “00” (HEX)	NAN “00” (HEX)	NAN “7F” (HEX)	NAN “A0” (HEX)	NAN “00” (HEX)	NAN “00” (HEX)

Data Byte #	Type	Remarks
0-2	24-bit unsigned integer	Sensor serial number. Not implemented for this device, the number consists of all zeros
3	8-bit unsigned integer	Sensor limits/minimum span unit code. Not implemented for this device, the integer returned is the “Not-used” indication or “250” (decimal)
4-7	32-bit floating point IEEE 754 format	Upper sensor limit. Not implemented for this device, the value returned is the “Not-A-Number” indication or the bytes 7F, A0, 00, 00 (hexadecimal).
8-11	32-bit floating point IEEE 754 format	Lower sensor limit. Not implemented for this device, the value returned is the “Not-A-Number” indication or the bytes 7F, A0, 00, 00 (hexadecimal).
12-15	32-bit floating point IEEE 754 format	Minimum span. Not implemented for this device, the value returned is the “Not-A-Number” indication or the bytes 7F, A0, 00, 00 (hexadecimal).

Command specific response codes:

0	No command-specific errors
1 - 4	Undefined
5	Incorrect byte count
6 – 127	Undefined

8.10. Command #15: Read Output Information

This command is intended to read the alarm selection code, transfer function, primary variable/range unit code, upper range value, lower range value, damping value (applied to the sensor, not the output), write protect code and private label distributor. For the 4800 Series devices only the write protect code is implemented. For the first three parameters the code "Not-Used" or "250" (decimal) is returned. For the upper range value, the lower range value and the damping value the "Not-A-Number" code or 7F, A0, 00, 00 (hexadecimal) is returned. For the private label distributor the manufacturer's identification code is returned.

Request data bytes:

NONE

Response data bytes:

#0	#1	#2	#3	#4	#5	#6
ALARM SELECT CODE	TRANSF. FUNCT. CODE	PV/RANGE UNITS CODE	UPPER RANGE MSB	UPPER RANGE	UPPER RANGE	UPPER RANGE MSB

#7	#8	#9	#10	#11	#12	#13	#14	#15	#16
LOWER RANGE MSB	LOWER RANGE	LOWER RANGE	LOWER RANGE MSB	DAMPING VALUE MSB	DAMPING VALUE	DAMPING VALUE	DAMPING VALUE LSB	WRITE PROTECT CODE	PVT LABEL DIST

Data Byte #	***Type***	***Remarks***
0	8-bit unsigned integer	Alarm select code. Not implemented for this device, the integer returned is a “Not-Used” or “250” (decimal)
1	8-bit unsigned integer	Transfer function code. Not implemented for this device, the integer returned is a “Not-Used” or “250” (decimal)
2	8-bit unsigned integer	Primary variable and range unit code. Not implemented for this device, the integer returned is a “Not-Used” or “250” (decimal)
3-6	32-bit floating point IEEE 754 format	Upper range value. Not implemented for this device, the value returned is the “Not-A-Number” indication or the bytes 7F, A0, 00, 00 (hexadecimal)
7-10	32-bit floating point IEEE 754 format	Lower range value. Not implemented for this device, the value returned is the “Not-A-Number” indication or the bytes 7F, A0, 00, 00 (hexadecimal)
11-14	32-bit floating point IEEE 754 format	Damping value. Not implemented for this device, the value returned is the “Not-A-Number” indication or the bytes 7F, A0, 00, 00 (hexadecimal)

15	8-bit unsigned integer	Write protect code. Refer to Section 11.7. 'Write Protect Codes'
16	8-bit unsigned integer	Private label distributor. Not implemented for this device, it defaults to the Manufacturer's identification code

Command specific response codes:

0	No command-specific errors
1 - 4	Undefined
5	Incorrect byte count
6 – 127	Undefined

8.11. Command #16: Read Final Assembly Number

Command used to read the final assembly number associated with the device.

Request data bytes:

NONE

Response data bytes:

FINAL ASS.NUM MSB	FINAL ASS.NUM	FINAL ASS.NUM LSB
#0	#1	#2

Data Byte #	*Type*	*Remarks*
0-2	24-bit unsigned integer	Final assembly number

Command specific response codes:

0	No command-specific errors
1 - 4	Undefined
5	Incorrect byte count
6 – 127	Undefined

8.12. Command #17: Write Message

Write the message into the device.

Request data bytes:

MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE
#0	#1	#2	#3	#4	#5	#6	#7	#8	#9

MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE
#10	#11	#12	#13	#14	#15	#16	#17	#18	#19

MESSAGE	MESSAGE	MESSAGE	MESSAGE
#20	#21	#22	#23

Data Byte #	***Type***	***Remarks***
0-23	24 (8-bit) byte packed ASCII	Device message.

Response data bytes:

MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE
#0	#1	#2	#3	#4	#5	#6	#7	#8	#9

MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE	MESSAGE
#10	#11	#12	#13	#14	#15	#16	#17	#18	#19

MESSAGE	MESSAGE	MESSAGE	MESSAGE
#20	#21	#22	#23

Data	***Type***	***Remarks***

Byte #		
0-23	24 (8-bit) byte packed ASCII	Device message.

Command specific response codes:

0	No command-specific errors
1 - 4	Undefined
5	Incorrect byte count
6	Undefined
7	In write protect mode
8 – 127	Undefined

8.13. Command #18: Write Tag, Descriptor, Date

Write the tag, descriptor and date into the device.

Request data bytes:

TAG	TAG	TAG	TAG	TAG	TAG
#0	#1	#2	#3	#4	#5

DESCR.	DESCR.	DESCR.	DESCR.	DESCR.	DESCR.	DESCR.	DESCR.	DESCR.	DESCR.	DESCR.	DESCR.
#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17

DATE DAY	DATE MONTH	DATE YEAR
#18	#19	#20

Data Byte #	***Type***	***Remarks***
0-5	6 (8-bit) byte packed ASCII	Device tag number.
6-17	12 (8-bit) byte packed ASCII	Device descriptor
18-20	3 (8-bit) unsigned integers	Date. Respectively day, month, year-1900

Response data bytes:

TAG	TAG	TAG	TAG	TAG	TAG
#0	#1	#2	#3	#4	#5

DESCR.	DESCR.	DESCR.	DESCR.	DESCR.	DESCR.	DESCR.	DESCR.	DESCR.	DESCR.	DESCR.	DESCR.
#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17

DATE DAY	DATE MONTH	DATE YEAR
#18	#19	#20

Data Byte #	***Type***	***Remarks***
0-5	6 (8-bit) byte packed ASCII	Device tag number.
6-17	12 (8-bit) byte packed ASCII	Device descriptor
18-20	3 (8-bit) unsigned integers	Date. Respectively day, month, year-1900

Command specific response codes:

0	No command-specific errors
1- 4	Undefined
5	Incorrect byte count
6	Undefined
7	In write protect mode
8 – 127	Undefined

8.14. Command #19: Write Final Assembly Number

Write the final assembly number into the device.

Request data bytes:

#0	#1	#2
FINAL ASS.NUM MSB	FINAL ASS.NUM	FINAL ASS.NUM LSB

Data Byte #	***Type***	***Remarks***
0-2	24-bit unsigned integer	Final assembly number

Response data bytes:

#0	#1	#2
FINAL ASS.NUM MSB	FINAL ASS.NUM	FINAL ASS.NUM LSB

Data Byte #	***Type***	***Remarks***
0-2	24-bit unsigned integer	Final assembly number

Command specific response codes:

0	No command-specific errors
1 - 4	Undefined
5	Incorrect byte count
6	Undefined
7	In write protect mode
8 – 127	Undefined

9. Common Practice Command Specifications

9.1. Command #37: Set Primary Variable Lower Range Value

This command generates a sensor zero action, the same function as pushing the zero button on the analog device. No flow should be applied to the device.

The command will return an error response code 9, "Applied process too high," if flow output is greater than 2% when the command is received.

Request data bytes:

NONE

Response data bytes:

NONE

Command specific response codes:

0	No command-specific errors
1 - 4	Undefined
5	Incorrect byte count
6	Undefined
7	In write protect mode
8	Undefined
9	Applied process too high
10 - 127	Undefined

9.2. Command #38: Reset Configuration Changed Flag

Resets the configuration changed response code, bit #6 of the transmitter status byte. Secondary master devices, address '0' should not issue this command. Primary master devices, address '1', should only issue this command after the configuration changed response code has been detected and acted upon.

Request data bytes:

NONE

Response data bytes:

NONE

Command specific response codes:

0	No command-specific errors
1 - 4	Undefined
5	Incorrect byte count
6	Undefined
7	In write protect mode
8 - 15	Undefined
16	Access restricted
17 - 127	Undefined

9.3. Command #48: Read Additional Transmitter Status

Command used to retrieve additional transmitter status information.

Request data bytes:

NONE

Response data bytes:

ADD. STATUS BYTE#0	ADD. STATUS BYTE#1	ADD. STATUS BYTE#2	ADD. STATUS BYTE#3
#0	#1	#2	#3

Refer to Section 11.14. ‘

Additional Device Status and Masking'.

Command specific response codes:

0	No command-specific errors
1 - 4	Undefined
5	Incorrect byte count
6 – 127	Undefined

9.4. Command #59: Write Number of Response Preambles

Set the minimum number of preambles to be sent by a device before the start of a response packet. This number includes the two preambles contained in the start of message. The value can vary from 2 to 15.

Request data bytes:

NUMBER RESP. PREAM.
#0

Data Byte #	***Type***	***Remarks***
0	8-bit unsigned integer	Number of response preambles to be sent with the response message from slave to master

Response data bytes:

NUMBER RESP. PREAM.
#0

Data Byte #	***Type***	***Remarks***
0	8-bit unsigned integer	Number of response preambles to be sent with the response message from slave to master

Command specific response codes:

0	No command-specific errors
1 - 2	Undefined
3	Passed parameter too large
4	Passed parameter too small
5	Incorrect byte count
6	Undefined
7	In write protect mode
8 - 15	Undefined
16	Access restricted
17 - 127	Undefined

9.5. Command #122: Write Device Identification Number (NON-PUBLIC)

Write the device identification number into the device's memory. The response message will be made using the unique identifier (long frame address) as received in the request message. The device identification number will not be incorporated in the unique identifier until the response message has been sent. The command is a Non-Public one, i.e. execution is protected by a three byte 'password' which has to be sent with the request message. This password should match the device's final assembly number in order to achieve a correct execution of the command. When they do not match, the "Command not implemented" response code will be returned. The command specific response codes will only be returned if the password and the final assembly number have matched.

Request data bytes:

PASSWORD MSB	PASSWORD	PASSWORD LSB	DEVICE ID. # MSB	DEVICE ID. #	DEVICE ID. # LSB
#0	#1	#2	#3	#4	#5

Data Byte #	***Type***	***Remarks***
0-2	24-bit unsigned integer	Password
3-5	24-bit unsigned integer	Device identification number

Response data bytes:

DEVICE ID. # MSB	DEVICE ID. #	DEVICE ID. # LSB
#0	#1	#2

Data Byte #	***Type***	***Remarks***
0-2	24-bit unsigned integer	Device identification number

Command specific response codes:

0	No command-specific errors
1 - 2	Undefined
3	Passed parameter too large
4	Passed parameter too small
5	Incorrect byte count
6	Undefined
7	In write protect mode
8 - 15	Undefined

16	Access restricted
17 - 127	Undefined

THIS PAGE WAS INTENTIONALLY LEFT BLANK

10. Transmitter Specific Command Specifications

10.1. Command #128: Enter/Exit Write Protect Mode (NON-PUBLIC)

This command is implemented to maintain compatibility with other Brooks products, however, it is not required and has no effect. Write Protect mode is not supported by the 4800 Series devices.

Request data bytes:

USER PASS-WORD	USER PASS-WORD	USER PASS-WORD	USER PASS-WORD	USER PASS-WORD	USER PASS-WORD	WRITE PROTECT CODE
#0	#1	#2	#3	#4	#5	#6

Data Byte #	*Type*	*Remarks*
0-5	6 (8-bit) byte packed ASCII	User password
6	8-bit unsigned integer	Write protect code Refer to Section 11.7. 'Write Protect Codes'

Response data bytes:

WRITE PROTECT CODE
#0

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Write protect code (Always returns 0) Refer to Section 11.7. 'Write Protect Codes'

Command specific response codes:

0	No command-specific errors
1 - 4	Undefined
5	Incorrect byte count
6 – 127	Undefined

10.2. Command #129: Clear Address Related Device Data (NON-PUBLIC)

Clear address related device data. The command is a Non-Public one, i.e. execution is protected by a three byte 'password' which has to be sent with the request message. This password should match the device's final assembly number in order to achieve a correct execution of the command. When they do not match, the "Command not implemented" response code will be returned. The command specific response codes will only be returned if the password and the final assembly number have matched. When this action is executed the device will initialize the address related device data and configure factory default settings. This action is needed when the digital interface is mounted onto a different MFM/MFC, address related data is derived from the MFM/MFC serial number.

Request data bytes:

PASSWORD MSB	PASSWORD	PASSWORD LSB
#0	#1	#2

Data Byte #	***Type***	***Remarks***
0-2	24-bit unsigned integer	Password

Response data bytes:
NONE

Command specific response codes:

0	No command-specific errors
1 - 4	Undefined
5	Incorrect byte count
6	Undefined
7	In write protect mode
8-15	Undefined
16	Access restricted
17-127	Undefined

10.3. Command #130: Write Manufacturer's Device Type Code (NON-PUBLIC)

Write the manufacturer's device type code. The command is a Non-Public one, i.e. execution is protected by a three byte 'password' which has to be sent with the request message. This password should match the device's final assembly number in order to achieve a correct execution of the command. When they do not match, the "Command not implemented" response code will be returned. The command specific response codes will only be returned if the password and the final assembly number have matched. When this action is executed the device type code will be set to the value passed with this command. Note that in case of long addressing the address will change.

Request data bytes:

PASSWORD MSB	PASSWORD	PASSWORD LSB	Manufacurer's device type code
#0	#1	#2	#3

Data Byte #	***Type***	***Remarks***
0-2	24-bit unsigned integer	Password
3	8-bit unsigned integer	Manufacturer's device type code

Response data bytes:

Manufacurer's device type code
#0

Data Byte #	***Type***	***Remarks***
0	8-bit unsigned integer	Manufacturer's device type code

Command specific response codes:

0	No command-specific errors
1 - 4	Undefined
5	Incorrect byte count
6	Undefined
7	In write protect mode
8-15	Undefined
16	Access restricted
17-127	Undefined

10.4. Command #150: Read Process Gas Type

Read the type of process gas specified by the gas selection code from the device's memory. The gas can be specified as a string of upper and lower case characters. The gases will in most gases be expressed by their chemical formula

Request data bytes:

GAS SELECT CODE
#0

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Gas selection code. Number between 1 and 10

Response data bytes:

GAS SELECT CODE
#0

PROCES S GAS	PROCES S GAS	PROCES S GAS	PROCES S GAS	PROCES S GAS	PROCES S GAS	PROCES S GAS	PROCES S GAS	PROCES S GAS	PROCES S GAS	PROCES S GAS	PROCES S GAS
#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Gas selection code. Number between 1 and 10
1-12	12 (8-bit) byte ASCII text	Process gas type.

Command specific response codes:

0	No command-specific errors
1	Undefined
2	Invalid selection
3-4	Undefined
5	Incorrect byte count
6 – 127	Undefined

10.5. Command #151 Read Gas Density, Flow Reference and Flow Range

Read the density of the selected gas, the operational flow range and the reference temperature and pressure for the flow range. All numbers are returned with their associated unit codes as entered with command #161. The flow range equals the volume flow in engineering units at 100%. The reference temperature and pressure are the conditions at which the volume flow is specified.

Request data bytes:

GAS SELECT CODE
#0

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Gas selection code. Number between 1 and 10

Response data bytes:

GAS SELECT CODE	DENSITY UNIT CODE	DENSITY MSB	DENSITY	DENSITY	DENSITY LSB
#0	#1	#2	#3	#4	#5

REF TEMP. UNIT	REF TEMP. MSB	REF TEMP.	REF TEMP.	REF TEMP. LSB	REF PRESS. UNIT	REF PRESS. MSB	REF PRESS.	REF PRESS.	REF PRESS. LSB
#6	#7	#8	#9	#10	#11	#12	#13	#14	#15

FLOW UNIT CODE	FLOW RANGE MSB	FLOW RANGE	FLOW RANGE	FLOW RANGE LSB
#16	#17	#18	#19	#20

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Gas selection code. Number between 1 and 10
1	8-bit unsigned integer	Density unit code. Refer to Section 11.4. ‘Density Unit Codes’
2-5	32-bit floating point IEEE 754 format	Process gas density
6	8-bit unsigned integer	Reference temperature unit code Refer to Section 11.5. ‘Temperature Unit Codes’
7-10	32-bit floating point	Reference temperature value.

	IEEE 754 format	
11	8-bit unsigned integer	Reference pressure unit code. Refer to Section 11.6. 'Pressure Unit and Reference Codes'
12-15	32-bit floating point IEEE 754 format	Reference pressure value.
16	8-bit unsigned integer	Flow rate unit code Refer to Section 11.3. 'Flow Rate Unit and Reference Codes'
17-20	32-bit floating point IEEE 754 format	Flow range value.

Command specific response codes:

0	No command-specific errors
1	Undefined
2	Invalid selection
3-4	Undefined
5	Incorrect byte count
6 – 127	Undefined

10.6. Command #190: Read Standard Temperature and Pressure

Read the standard temperature and pressure values from the device's memory. The standard temperature and pressure are reference values which can be set by the user and which are used in the conversion of flow units. They indicate the conditions to which a certain flow rate must be referenced to.

Request data bytes:

NONE

Response data bytes:

STD TEMP UNIT	STD TEMP MSB	STD TEMP	STD TEMP	STD TEMP LSB	STD PRESS UNIT	STD PRESS MSB	STD PRESS	STD PRESS	STD PRESS LSB
#0	#1	#2	#3	#4	#5	#6	#7	#8	#9

Data Byte #	***Type***	***Remarks***
0	8-bit unsigned integer	Standard temperature unit code Refer to Section 11.5. 'Temperature Unit Codes'
1-4	32-bit floating point IEEE 754 format	Standard temperature value.
5	8-bit unsigned integer	Standard pressure unit code. Refer to Section 11.6. 'Pressure Unit and Reference Codes'
6-9	32-bit floating point IEEE 754 format	Standard pressure value.

Command specific response codes:

0	No command-specific errors
1 – 127	Undefined

10.7. Command #191: Write Standard Temperature and Pressure

Write the standard temperature and pressure values into the device's memory. The standard temperature and pressure are reference values which can be set by the user and which are used in the conversion of flow units. They indicate the conditions to which a certain flow rate must be referenced to.

Request data bytes:

STD TEMP UNIT	STD TEMP MSB	STD TEMP	STD TEMP	STD TEMP LSB	STD PRESS UNIT	STD PRESS MSB	STD PRESS	STD PRESS	STD PRESS LSB
#0	#1	#2	#3	#4	#5	#6	#7	#8	#9

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Standard temperature unit code Refer to Section 11.5. 'Temperature Unit Codes'
1-4	32-bit floating point IEEE 754 format	Standard temperature value.
5	8-bit unsigned integer	Standard pressure unit code. Refer to Section 11.6. 'Pressure Unit and Reference Codes'
6-9	32-bit floating point IEEE 754 format	Standard pressure value.

Response data bytes:

STD TEMP UNIT	STD TEMP MSB	STD TEMP	STD TEMP	STD TEMP LSB	STD PRESS UNIT	STD PRESS MSB	STD PRESS	STD PRESS	STD PRESS LSB
#0	#1	#2	#3	#4	#5	#6	#7	#8	#9

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Standard temperature unit code Refer to Section 11.5. 'Temperature Unit Codes'
1-4	32-bit floating point IEEE 754 format	Standard temperature value.
5	8-bit unsigned integer	Standard pressure unit code. Refer to Section 11.6. 'Pressure Unit and Reference Codes'
6-9	32-bit floating point IEEE 754 format	Standard pressure value.

Command specific response codes:

0	No command-specific errors
1	Undefined
2	Invalid selection
3	Parameter too small
4	Parameter too large
5	Incorrect byte count
6	Undefined
7	In write protect mode
8 – 127	Undefined

10.8. Command #193: Read Operational Settings

Read the operational settings from the device. These settings consist of the selected gas number, the selected flow reference condition, the selected flow unit and the selected temperature unit.

Request data bytes:

NONE

Response data bytes:

SEL GAS NUMBER	SEL FLOW REF	SEL FLOW UNIT	SEL TEMP UNIT
#0	#1	#2	#3

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Selected gas number. Number between 1 and 10.
1	8-bit unsigned integer	Selected flow reference. Refer to Section 11.3. ‘Flow Rate Unit and Reference Codes’
2	8-bit unsigned integer	Selected flow unit. Refer to Section 11.3. ‘Flow Rate Unit and Reference Codes’
3	8-bit unsigned integer	Selected temperature unit. Refer to Section 11.5. ‘Temperature Unit Codes’

Command specific response codes:

0	No command-specific errors
1 - 127	Undefined

10.9. Command #195: Select Gas Number

Select a gas number from the calibrated gasses. The selected gas number will be the reference number for the database part containing the calibration data of the selected gas.

Request data bytes:

SEL GAS NUMBER
#0

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Selected gas number. Number between 1 and 10.

Response data bytes:

SEL GAS NUMBER
#0

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Selected gas number. Number between 1 and 10.

Command specific response codes:

0	No command-specific errors
1	Undefined
2	Invalid selection
3 - 4	Undefined
5	Incorrect byte count
6	Undefined
7	In write protect mode
8 - 127	Undefined

10.10. Command #196: Select Flow Unit

Select a flow unit. Selecting a flow unit does not only consist of selecting the flow unit, but also the reference condition. The selected flow unit will be used in the conversion from flow data. Flow data will be made available to the user in the selected flow unit.

Request data bytes:

SEL FLOW REF	SEL FLOW UNIT
#0	#0

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Selected flow reference. Refer to Section 11.3. 'Flow Rate Unit and Reference Codes'
1	8-bit unsigned integer	Selected flow unit. Refer to Section 11.3. 'Flow Rate Unit and Reference Codes'

Response data bytes:

SEL FLOW REF	SEL FLOW UNIT
#0	#0

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Selected flow reference. Refer to Section 11.3. 'Flow Rate Unit and Reference Codes'
1	8-bit unsigned integer	Selected flow unit. Refer to Section 11.3. 'Flow Rate Unit and Reference Codes'

Command specific response codes:

0	No command-specific errors
1	Undefined
2	Invalid selection
3 - 4	Undefined
5	Incorrect byte count
6	Undefined
7	In write protect mode
8 - 127	Undefined

10.11. Command #197: Select Temperature Unit

Select a temperature unit. The selected temperature unit will be used in the conversion of temperature data. Temperature data will be made available to the user in the selected temperature unit.

Request data bytes:

SEL TEMP REF
#0

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Selected temperature unit Refer to Section 11.5. ‘Temperature Unit Codes’

Response data bytes:

SEL TEMP REF
#0

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Selected temperature unit Refer to Section 11.5. ‘Temperature Unit Codes’

Command specific response codes:

0	No command-specific errors
1	Undefined
2	Invalid selection
3 - 4	Undefined
5	Incorrect byte count
6	Undefined
7	In write protect mode
8 - 127	Undefined

10.12. Command #215: Read Setpoint Settings

Read the setpoint related settings from the device. The settings contain the setpoint source indication, i.e. analog 0 - 5 volt/0 - 20 mA, analog 1 - 5 volt/4 - 20 mA or digital, the type of softstart and the softstart ramp.

Request data bytes:

NONE

Response data bytes:

SETP SOURCE CODE	SETP SPAN MSB	SETP SPAN	SETP SPAN	SETP SPAN LSB	SETP OFFSET MSB	SETP OFFSET	SETP OFFSET	SETP OFFSET LSB
#0	#1	#2	#3	#4	#5	#6	#7	#8

SOFT START CODE	S.START RAMP MSB	S.START RAMP	S.START RAMP	S.START RAMP LSB
#9	#10	#11	#12	#13

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Setpoint source selection code. Refer to Section 11.12. 'Setpoint Source Selection Codes'
1-4	32-bit floating point IEEE 754 format	Always returns 1.0
5-8	32-bit floating point IEEE 754 format	Always returns 0.0
9	8-bit unsigned integer	Softstart selection code. Always returns 0.
10-13	32-bit floating point IEEE 754 format	Softstart ramp value. Always returns 0.0

Command specific response codes:

0	No command-specific errors
1 - 127	Undefined

10.13. Command #216: Select Setpoint Source

Select the setpoint source to be used as setpoint input. The setpoint source can be either analog 0 - 5 volt/0 - 20 mA, analog 1 - 5 volt/4 - 20 mA or digital (i.e. through communication). This command allows the user to select between analog setpoint and digital setpoint. The type of analog input/output (Volt or mA) is chosen when the device was ordered, it can not be changed.

Request data bytes:

SETP SOURCE CODE
#0

Data Byte #	***Type***	***Remarks***
0	8-bit unsigned integer	Setpoint source selection code. Refer to Section 11.12. ’Setpoint Source Selection Codes’

Response data bytes:

SETP SOURCE CODE
#0

Data Byte #	***Type***	***Remarks***
0	8-bit unsigned integer	Setpoint source selection code. Refer to Section 11.12. ’Setpoint Source Selection Codes’

Command specific response codes:

0	No command-specific errors
1	Undefined
2	Invalid selection
3 - 4	Undefined
5	Incorrect byte count
6	Undefined
7	In write protect mode
8 - 127	Undefined

10.14. Command #230: Get Valve Override Status

Get the current valve override status from the device. The valve override status can indicate either OFF (No valve override), CLOSE, OPEN, or MANUAL.

Request data bytes:

NONE

Response data bytes:

VALVE OVERR. CODE
#0

Data Byte #	***Type***	***Remarks***
0	8-bit unsigned integer	Valve override code. Refer to Section 11.13. 'Valve Override Codes'

Command specific response codes:

0	No command-specific errors
1 - 127	Undefined

10.15. Command #231: Set Valve Override Status

Set the current valve override status. The valve override can be set to either OFF (No valve override), CLOSE or OPEN. The analog valve override input on the D-Connector of the device will take precedence over the digital command.

Request data bytes:

VALVE OVERR. CODE
#0

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Valve override code. Refer to Section 11.13. 'Valve Override Codes'

Response data bytes:

VALVE OVERR. CODE
#0

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Valve override code. Refer to Section 11.13. 'Valve Override Codes'

Command specific response codes:

0	No command-specific errors
1	Undefined
2	Invalid selection
3 - 127	Undefined

10.16. Command #235: Read Setpoint in % and Selected Units

Read the current setpoint value in percent of full scale and in selected flow units. The setpoint in selected flow units compared to its full scale range should be the equivalent of the setpoint in percent.

Request data bytes:

NONE

Response data bytes:

PERCENT UNIT CODE	SETP PERCENT MSB	SETP PERCENT	SETP PERCENT	SETP PERCENT LSB	SEL. FLOW UNIT	SETP. UNITS MSB	SETP. UNITS	SETP. UNITS	SETP. UNITS LSB
#0	#1	#2	#3	#4	#5	#6	#7	#8	#9

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Setpoint percent unit. Unit always reads 57 (decimal), percent.
1-4	32-bit floating point IEEE 754 format	Setpoint in percent of full scale.
5	8-bit unsigned integer	Selected flow unit. Refer to Section 11.3. ‘Flow Rate Unit and Reference Codes’
6-9	32-bit floating point IEEE 754 format	Setpoint in selected flow unit.

Command specific response codes:

0	No command-specific errors
1 - 127	Undefined

10.17. Command #236: Write Setpoint in % or Selected Units

Write the current setpoint value in percent of full scale or in selected flow units to the device. If the setpoint unit code is set to percent (code 57) the setpoint value is assumed to be in percent. If the setpoint unit code is set to Not Used, the setpoint value is assumed to be in the selected flow unit. The return message is the same as the one of Command #235. The setpoint in selected flow units compared to its full scale range should be the equivalent of the setpoint in percent. When this command is received, the Setpoint Source will be set to digital automatically if not already in digital mode. The Setpoint Source will remain digital mode until the user returns the Setpoint Source to analog mode via Command #216 or until the power to the device is cycled.

Request data bytes:

SETP UNIT CODE	SETP. MSB	SETP.	SETP.	SETP. LSB
#0	#1	#2	#3	#4

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Setpoint unit. Unit reads either 57 (decimal), “Percent” or 0 (decimal) “Not Used”.
1-4	32-bit floating point IEEE 754 format	Setpoint value. In either percent of full scale or in selected flow units.

Response data bytes:

PERCENT UNIT CODE	SETP PERCENT MSB	SETP PERCENT	SETP PERCENT	SETP PERCENT LSB	SEL. FLOW UNIT	SETP. UNITS MSB	SETP. UNITS	SETP. UNITS	SETP. UNITS LSB
#0	#1	#2	#3	#4	#5	#6	#7	#8	#9

Data Byte #	*Type*	*Remarks*
0	8-bit unsigned integer	Setpoint unit. Unit always reads 57 (decimal), percent
1-4	32-bit floating point IEEE 754 format	Setpoint value. In percent of full scale.
5	8-bit unsigned integer	Selected flow unit. Refer to Section 11.3. ‘Flow Rate Unit and Reference Codes’
6-9	32-bit floating point IEEE 754 format	Setpoint in selected flow unit.

Command specific response codes:

0	No command-specific errors
1	Undefined
2	Invalid selection
3	Parameter too small
4	Parameter too large
5	Incorrect byte count
6	Undefined
7	In write protect mode
8 - 127	Undefined

10.18. Command #237: Read Valve Control Value

Read the current valve control value. The valve control value is a dimensionless number in the range from 0 to 4095. It represents the value sent to the D/A-converter used to control the valve.

Request data bytes:

NONE

Response data bytes:

#0	#1	#2
VALVE VALUE MSB	VALVE VALUE	VALVE VALUE LSB

Data Byte #	***Type***	***Remarks***
0-2	24-bit unsigned integer	Valve control value. Dimensionless number between 0 and 4095

Command specific response codes:

0	No command-specific errors
1 - 127	Undefined

10.19. Command #245: Read Alarm Enable Setting

Read the alarm enable settings. These alarm settings can be used to mask specific alarm sources.

Request data bytes:

NONE

Response data bytes:

ALARM-ENABLE BYTE 0	ALARM-ENABLE BYTE 1	ALARM-ENABLE BYTE 2	ALARM-ENABLE BYTE 3
#0	#1	#2	#3

Data Byte #	*Type*	*Remarks*
0	8-bit bit-field	Alarm mask byte 0
1	8-bit bit-field	Alarm mask byte 1
2	8-bit bit-field	Alarm mask byte 2
3	8-bit bit-field	Alarm mask byte 3 Refer to Section 11.14. ‘Additional Device Status and Masking’

Command specific response codes:

0	No command-specific errors
1 – 127	Undefined

10.20. Command #246: Write Alarm Enable Setting

Set the alarm enable settings. These alarm settings can be used to mask specific alarm sources.

Request data bytes:

ALARM-ENABLE BYTE 0	ALARM-ENABLE BYTE 1	ALARM-ENABLE BYTE 2	ALARM-ENABLE BYTE 3
#0	#1	#2	#3

Data Byte #	*Type*	*Remarks*
0	8-bit bit-field	Alarm mask byte 0
1	8-bit bit-field	Alarm mask byte 1
2	8-bit bit-field	Alarm mask byte 2
3	8-bit bit-field	Alarm mask byte 3 Refer to Section 11.14. ‘Additional Device Status and Masking’

Response data bytes:

ALARM-ENABLE BYTE 0	ALARM-ENABLE BYTE 1	ALARM-ENABLE BYTE 2	ALARM-ENABLE BYTE 3
#0	#1	#2	#3

Data Byte #	*Type*	*Remarks*
0	8-bit bit-field	Alarm mask byte 0
1	8-bit bit-field	Alarm mask byte 1
2	8-bit bit-field	Alarm mask byte 2
3	8-bit bit-field	Alarm mask byte 3 Refer to Section 11.14. ‘Additional Device Status and Masking’

Command specific response codes:

0	No command-specific errors
1 – 4	Undefined
5	Too few bytes received
6 – 127	Undefined

10.21. Command #247: Read High/Low Flow Alarm

Read the high/low flow alarm settings as a percent of device full scale. This command can be used to read the actual flow alarm limits.

Request data bytes:

NONE

Response data bytes:

LOW-LIMIT MSB	LOW-LIMIT	LOW-LIMIT	LOW-LIMIT LSB	HIGH-LIMIT MSB	HIGH-LIMIT	HIGH-LIMIT	HIGH-LIMIT LSB
#0	#1	#2	#3	#4	#5	#6	#7

Data Byte #	*Type*	*Remarks*
0-3	832-bit floating point IEEE 754 format	Low-flow alarm limit (Percent of full scale).
4-7	32-bit floating point IEEE 754 format	High-flow alarm limit (Percent of full scale).

Command specific response codes:

0	No command-specific errors
1 – 127	Undefined

10.22. Command #248: Write High/Low Flow Alarm

Set the high/low flow alarm settings in percent of device full scale. This command can be used to configure the flow alarm limits.

Request data bytes:

LOW-LIMIT MSB	LOW-LIMIT	LOW-LIMIT	LOW-LIMIT LSB	HIGH-LIMIT MSB	HIGH-LIMIT	HIGH-LIMIT	HIGH-LIMIT LSB
#0	#1	#2	#3	#4	#5	#6	#7

Data Byte #	***Type***	***Remarks***
0-3	832-bit floating point IEEE 754 format	Low-flow alarm limit (Percent of full scale).
4-7	32-bit floating point IEEE 754 format	High-flow alarm limit (Percent of full scale).

Response data bytes:

LOW-LIMIT MSB	LOW-LIMIT	LOW-LIMIT	LOW-LIMIT LSB	HIGH-LIMIT MSB	HIGH-LIMIT	HIGH-LIMIT	HIGH-LIMIT LSB
#0	#1	#2	#3	#4	#5	#6	#7

Data Byte #	***Type***	***Remarks***
0-3	832-bit floating point IEEE 754 format	Low-flow alarm limit (Percent of full scale).
4-7	32-bit floating point IEEE 754 format	High-flow alarm limit (Percent of full scale).

Command specific response codes:

0	No command-specific errors
1 - 2	Undefined
3	Passed parameter too large
4	Passed parameter too small
5	Too few bytes received
6 – 127	Undefined

THIS PAGE WAS INTENTIONALLY LEFT BLANK

11. Transmitter Specific Tables

11.1. Introduction

This chapter lists all transmitter specific codes as used by the 4800 Series devices. The codes are commonly 8-bit unsigned integers, ranging from 0 to 255. In a number of cases these code tables are subsets of existing "Common Tables" provided by the Rosemount HART communication specification.

11.2. Device type codes

The Device type code for all 4800 Series devices is 70.

11.3. Flow Rate Unit and Reference Codes

The flow rate unit codes are covered by two tables: the table with the reference condition codes and the table with the actual unit codes. The unit codes table is a subset of Table II, Unit codes table.

Table 11-1 Flow rate unit definition

Code	Flow rate unit
0..16	Undefined
17	Litres/minute
18	Undefined
19	Cubic meters/hour
20..23	Undefined
24	Litres/second
25..27	Undefined
28	Cubic meters/second
29..56	Undefined
57	Percent of flow range
58..69	Undefined
70	Grams/second
71	Grams/minute
72	Grams/hour
73	Kilograms/second
74	Kilograms/minute
75	Kilograms/hour
76..79	Undefined
80	Pounds/second
81	Pounds/minute
82	Pounds/hour

83..130	Undefined
131	Cubic meters/minute
132..137	Undefined
138	Liters/hour
139..169	Undefined
170	Millilitres/second
171	Millilitres/minute
172	Millilitres/hour
173..249	Undefined
250..255	Reserved

Table 11-2 Flow Reference Condition

Code	Reference Condition
0	Normal (273.15 Kelvin/1013.33 mBar)
1	Standard (User defined through separate command)
2	Calibration (As defined at calibration)
3..249	Undefined
250..255	Reserved

11.4. Density Unit Codes

The density unit codes table is a subset of Table II, Unit codes table. The density units are always referenced by 273.15 Kelvin and 1013.33 mBar ('normal' conditions).

Table 11-3 Density unit definition

Code	Density unit
0..90	Undefined
91	Grams/cubic centimetre
92	Kilograms/cubic meters
93	Undefined
94	Pounds/cubic feet
95	Undefined
96	Kilograms/litre
97	Grams/Litre
98..249	Undefined
250..255	Reserved

11.5. Temperature Unit Codes

The temperature unit codes table is a subset of Table II, Unit codes table.

Table 11-4 Temperature unit definition

Code	Temperature unit
0..31	Undefined
32	Degrees Celsius
33	Degrees Fahrenheit
34	Undefined
35	Kelvin
36..249	Undefined
250..255	Reserved

11.6. Pressure Unit and Reference Codes

All pressure values can be expressed in the pressure units as given in the table below. In case the unit refers to the inlet and outlet pressure values, the pressure reference is also given. The pressure unit codes table is a subset of Table II, Unit codes table.

Table 11-5 Pressure unit definition

Code	Pressure unit
0..5	Undefined
6	Pounds/square inch
7	Bar
8	Millibar
9	Undefined
10	Kilograms/square centimetre
11	Pascals
12	Kilopascals
13	Torricelli
14	Atmosphere
15..249	Undefined
250..255	Reserved

Table 11-6 Pressure Reference definition

Code	Pressure Reference definition
0	Absolute pressure
1	Effective pressure
2..249	Undefined
250..255	Reserved

11.7. Write Protect Codes

Supported write protect codes.

Table 11-7 Write protect codes

Code	Description
0	Not write protected
2..249	Undefined
250..255	Reserved

11.8. Physical Signaling Codes

The physical signaling codes indicate the physical layer that can be used for communication.

Table 11-8 Physical signaling codes

Code	Physical signaling
0	RS-485
1..249	Undefined
250..255	Reserved

11.9. Transmitter Variable Codes

Definition of the transmitter variable codes.

Table 11-9 Transmitter variable code definition

Code	Transmitter variable
0	Flow rate
1	Temperature
2..249	Undefined
250..255	Reserved

11.10. Flag Assignments

The flag assignments indicate implementation facts of the device.

Table 11-10 Flag assignments

Bit	Indication
#0	Multisensor device
#1	Undefined
#2	Undefined
#3	Undefined
#4	Undefined
#5	Undefined
#6	Undefined
#7	Reserved

11.11. Analog Output Selection Codes

Definition of the analog output selection codes.

Table 11-11 Analog output selection codes

Code	Analog Output
0	Current output
1	Voltage output
2..249	Undefined
250..255	Reserved

11.12. Setpoint Source Selection Codes

The codes define the possible sources for the setpoint signal.

Table 11-12 Setpoint source selection codes

Code	Setpoint source
0	Undefined
1 or 2	Analog Input
3	Communication input
4..249	Undefined
250..255	Reserved

11.13. Valve Override Codes

They are all 'Undefined' for the meter models.

Table 11-13 Valve override codes

Code	Valve override selection
0	Valve override off (normal operation)
1	Valve override open
2	Valve override close
3	Valve override manual (Read Only)
3..249	Undefined
250..255	Reserved

11.14. Additional Device Status and Masking

Table 11-14 Additional device status and masking

Byte #	Bit #	Status bit description (0=no error, 1=specified error ccured)	Device status masking: Mask bit: 0=disabled 1=enabled (*default)	Remarks
0	0	Undefined	0	Zero always
	1	Undefined	0	Zero always
	2	MFC communication failure	1	One always
	3	Undefined	0	Zero always
	4	Sensor zero failed	1	One always
	5	Internal power supply failure	1	One always
	6	Undefined	0	Zero always
	7	Undefined	0	Zero always
1	0	Undefined	0	Zero always
	1	Undefined	0	Zero always
	2	Undefined	0	Zero always
	3	Undefined	0	Zero always
	4	Undefined	0	Zero always
	5	Undefined	0	Zero always
	6	Undefined	0	Zero always
	7	Undefined	0	Zero always
2	0	Low flow alarm	0* / 1	
	1	High flow alarm	0* / 1	
	2	Undefined	0	Zero always
	3	Undefined	0	Zero always
	4	Undefined	0	Zero always
	5	Undefined	0	Zero always
	6	Undefined	0	Zero always
	7	Undefined	0	Zero always
3	0	Undefined	0	Zero always
	1	Undefined	0	Zero always
	2	Undefined	0	Zero always
	3	Undefined	0	Zero always
	4	Undefined	0	Zero always
	5	Undefined	0	Zero always
	6	Undefined	0	Zero always
	7	Undefined	0	Zero always

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 LOCAL AND WORLDWIDE SUPPORT

Brooks Instrument provides sales and service facilities around the world, ensuring quick delivery from local stock, timely repairs and local based sales and service facilities.

Our dedicated flow experts provide consultation and support, assuring successful applications of the Brooks flow measurement and control products.

Calibration facilities are available in local sales and service offices. The primary standard calibration equipment to calibrate our flow products is certified by our local Weights and Measures Authorities and traceable to the relevant international standard.

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 services can be provided under in-situ conditions, and the results will be traceable to the relevant international quality standard.

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,

Americas ☎ +(1) 888 554 FLOW
Europe ☎+31 (0) 318 549 290
Asia ☎ +81 (0) 3 5633 7100

ISO 9001 QUALITY SYSTEM

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

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