HART Protocol Specification

Field Communications

HART® COMMUNICATION FOUNDATION

Command Summary Specification

HCF_SPEC-99, Revision 8.0

Release Date: 18 April, 2001

Release Date: 18 April, 2001

Document Distribution / Maintenance Control / Document Approval
To obtain information concerning document distribution control, maintenance control and document approval, please contact the HART Communication Foundation at the address shown below.

Trademark Information
HART® is a registered trademark of the HART Communication Foundation, Austin, Texas, USA. Any use of the term HART hereafter in this document, or in any document referenced by this document, implies the registered trademark. All other trademarks used in this or referenced documents are trademarks of their respective companies. For more information contact the HCF Staff at the address below.

Attention: Foundation Director
HART Communication Foundation
9390 Research Boulevard
Suite I-350
Austin, TX 78759, USA

Voice: (512) 794-0369
FAX: (512) 794-3904

http://www.hartcomm.org

Table of Contents

Table of Figures

Index to Tables

Preface

This preface is included for informational purposes only.

The *Command Summary Specification* has long provided the basis for the HART Application Layer. Often, in the past, requirements that did not fit well in other Application Layer documents could be found in the *Command Summary Specification*. The assembly of diverse topics in this specification negatively impacted organization and readability. As a result, some implementations have overlooked requirements of this document.

With the introduction of HART 6, the *Command Summary Specification* has been significantly updated. In fact, while the many of the requirements can be found in the previous revision of the specification, revision 8.0 is virtually a new specification. The *Command Summary Specification* now truly provides a foundation for the entire HART Protocol Application Layer.

This document contains the following additions and functional enhancements as approved by the HART Communication Foundation Membership:

The Protocol now supports a classification mechanism that allows access to significantly more information about Device Variables and Dynamic Variables. Classification is based on the type of process connection (for example, pressure, temperature, actuators, flow, etc.) and provides for Unit Code expansion, Process Data Quality and Device Families (see Section 8 for details).

All cyclical process data now consist of a floating point value, an implicit or explicit engineering unit code and a Device Variable Status byte (see Section 8.4). Of the eight status bits in a Device Variable Status byte, the definition of 5 bits are assigned in this specification and the remaining 3 bits are determined by the Device Variable Family (see Command 54). For example, Command 9 returns Device Variable Status bytes for each Dynamic or Device Variable it returns. The classification of the Dynamic Variables is returned in a new Universal Command (i.e., Command 8).

A Long Tag is now supported by the Protocol. As a result, a new data type, ISO Latin-1, was also introduced. A section on ISO Latin-1 was added to Section 5.

Previous versions of the Protocol supported 256 Commands and used only the Data Link Layer's Command Number field. HART 6 supports a new Extended Command Number. Extended Command Numbers are indicated by 31 (0x1F) in the Command Number field. Extended Command Numbers are 16 bits long and used to support a new Device Family class of commands. Device Family Commands enable HART-compatible hosts to setup and commission devices without requiring the use of Device Descriptions, adding another level of interoperability to the HART Protocol.

The Protocol now supports Sub-Devices as well. These are devices connected to a HART compatible network via a Bridge Device or I/O System. Sub-Devices can be identified by a

master. Although Sub-Devices are not directly connected to the network, they communicate the same as any other HART compatible device.

In addition to functional changes, Revision 8.0 includes several new sections: Preface, Introduction, Scope, References, Definitions, and Symbols/Abbreviations. The additional sections and a new format improve precision and consistency of the Specifications.

The organization of Revision 8.0 differs from Revision 7.1 in the following ways:

Section 5 specifies all data types supported by the Protocol, and now includes figures and tables for clarification. The new ISO Latin-1 strings, floating point numbers, signed and unsigned integers, a date format and two types of lookup tables are described in detail within the section.

Adherence to field device revision rules have long been a specification requirement. Once buried in the "Implementation Notes" section, they were moved to an independent section and updated to reflect HART 6 requirements.

Command Number Partitions, Command Requirements, and Command Status Bytes were consolidated into Section 7. These were formerly found scattered throughout Sections 2.1, 3 and 6.2 of Revision 7.1 of the *Command Summary Specification*.

The Application Layer is responsible for the content of the Command and Data fields. Requirements for the Data Field Format section were shared between the *Data Link Layer* and *Command Summary Specifications* under HART Rev. 5. Section 7.2 was added to provide a thorough discussion of these requirements in one location. The Data Field section also specifies the use of extended commands and shows the format of Data fields for all possible legal permutations of the Data field format.

Understanding the Protocol's use of Dynamic Variables and Device Variables is critical as they are used throughout the Application Layer. Section 8 was significantly clarified and figures added to ensure the proper use of these terms and to ensure Application Layer compatibility of every HART device with these requirements.

Requirements for Section 9, Field Device Identification, were clarified and expanded from Sections 6.1, and 6.6 through 6.9 of the previous version.

Network Management (Section 10) was added, including information formerly spread across several documents. Existing requirements were updated to reflect provisions of HART Rev. 6. Identification of field devices, Delayed Response Mechanism, Sub-Devices are some of the requirements added or clarified.

Host Conformance Class was moved to Section 11 and requirements clarified.

Introduction

The Application Layer is the topmost layer in the Open System Interconnect (OSI) model for communications protocols. HART is a master/slave protocol loosely organized around the ISO/OSI 7-layer model (see Figure 1). The HART Application Layer defines device commands, responses, data types and status reporting. Conventions in HART, such as how to trim the loop current, are also part of the Application Layer.

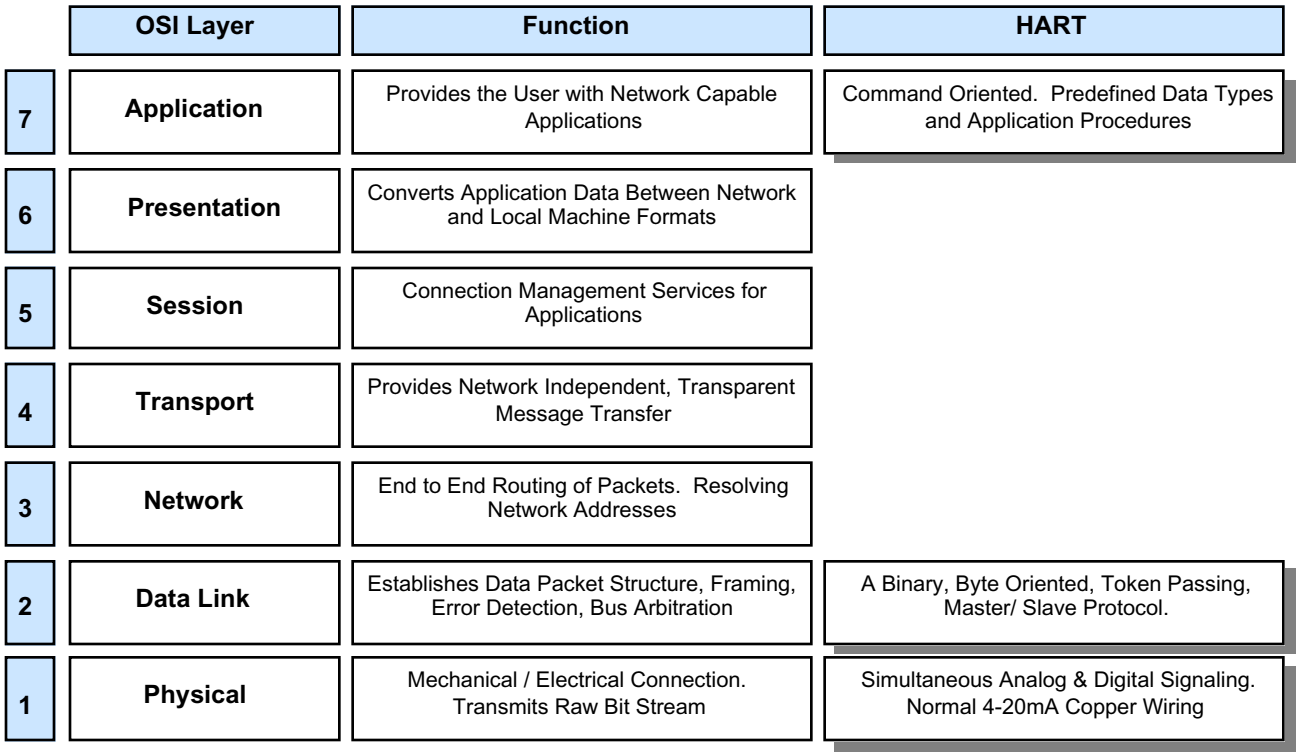

Figure 1. OSI 7-Layer Model

The *Command Summary Specification* establishes the core requirements for implementation of the HART Protocol Application Layer by master and slave devices. In addition, procedures for utilizing application layer functions in HART networks are defined.

The *Command Summary* defines the data types that are allowed to be communicated using the Protocol. Several numeric data types are allowed including single and double precision floating point; integers and unsigned integers. Text strings may be transmitted using Packed-ASCII or ISO Latin-1. Look-up tables can be used to associate fixed meaning to an unsigned integer (see the *Common Tables Specification* for example usage of these data types). Any combination of the specified data types can be mixed in a HART command.

The *Command Summary* specifies rules governing all HART commands and allocates commands numbers (e.g., as Universal or Common Practice). The allowed command behaviors are defined. Commands may read from or write to a field device and there are commands that require a field

device to perform a designated action or function. The command requirements found in this specification are adhered to in the *Universal*, *Common Practice*, and *Device Family Command Specifications*. In addition, Device-Specific commands developed by manufacturers must comply to the requirements in this specification.

The OSI model includes a network layer that is responsible for resolving device addresses and routing messages. While the HART Protocol does not explicitly support this layer, functions are included that address network management functions. This specification identifies the data items and procedures masters must use to resolve message addressing and routing. In addition, support requirements for network configurations (like multi-drop and slave burst mode operation) are specified.

This specification defines requirements for slaves to provide continuous operational feedback to hosts. This feedback includes device health information, command execution reporting and (for data link layer use) communication error detection. The *Command Response Code Specification* builds on the requirements for command execution reporting and provides detailed information about individual Response Codes.

Finally, this specification includes requirements identifying the level of application layer support found in field devices and host applications. Slave devices are identified using their Manufacturer ID, Device Type, and Device Revision. This specification includes rules that govern the values returned for the Device Type and Device Revision when the field device's application layer support is changed or enhanced. Host applications are identified by the level of application layer support they offer. This allows end users to understand the capabilities they can expect from a host.

This key document in the HART Specifications specifies requirements that serve as the foundation for all other HART Application Layer specifications.

1. SCOPE

This specification provides the core Application Layer requirements for HART-compatible devices. The Application Layer builds on the requirements specified for the Data Link Layer. Figure 2 shows the fields found in a HART message. While the Data Link Layer is responsible for the error-free transmission of messages and uses the Delimiter and Byte Count fields to frame the message, it does not interpret message content. The Application Layer is responsible for message content, including the definition of commands and the interpretation of data. The Command, Byte Count and Data fields constitute the substance of a HART message frame. These fields are the focus of the Application Layer.

Figure 2. HART Frame Format

This document specifies:

- The allowable formats of data transmitted via the Protocol;
- Revision rules for all field devices;
- The allocation of Commands Numbers for use by Universal, Common Practice, Device-Specific and Device Family commands;
- The organization of the Data field for differing combinations of command numbers, master requests, slave responses, and error conditions;
- The requirements for the construction of any HART command;
- The Command Status Bytes required to be returned with all commands responses;
- The use of Dynamic and Field Device Variables; and
- Procedures used by masters to identify field devices and manage HART networks

These requirements provide the basis for all HART Application Layer Specifications.

2. REFERENCES

2.1 HART Field Communications Protocol Specifications

HART Field Communications Protocol Specification. HCF_SPEC-12

Data Link Layer Specification. HCF_SPEC-81

Universal Command Specification. HCF_SPEC-127

Common Practice Command Specification. HCF_SPEC-151

Device Families Command Specification. HCF_SPEC-160

Common Tables Specification. HCF_SPEC-183

Command Response Code Specification. HCF_SPEC-307

2.2 Related HART Documents

The HART Protocol Specifications frequently reference the manufacturers' device-specific document. Device-specific documents are developed and controlled by the respective manufacturer and should follow the requirements of the following HART Communication Foundation document:

Field Device Specification Guide. HCF_LIT-18

3. DEFINITIONS

Definitions for terms can be found in *HART Field Communications Protocol Specification* (HCF_SPEC-12). Terms used throughout the *Command Summary Specification* include: ASCII, Data Link Layer, Delayed Response, Delayed Response Mechanism, Device Variable, Busy, DR_CONFLICT, DR_DEAD, DR_INITIATE, DR_RUNNING, Dynamic Variable, Device Type, Device Revision, Fixed Current Mode, Floating Point, ISO Latin-1, Master, Multi-drop, Not-A-Number, Packed ASCII, Preamble, Request Data Bytes, Response Data Bytes, Response Message, Slave, Slave Time-Out, Time Constant, Units Code.

4. SYMBOLS/ABBREVIATIONS

ADC	**A**nalog-to-**D**igital **C**onverter
DAC	**D**igital-to-**A**nalog **C**onverter.
DAQ	**D**ata **Ac**quisition. This refers to a devices specific ADC or DAC
DR	**D**elayed **R**esponse
DRM	**D**elayed **R**esponse **M**echanism
HCF	**H**ART **C**ommunication **F**oundation
LSB	**L**east **S**ignificant **B**yte. The LSB is always the last byte transmitted over a HART data link
MSB	**M**ost **S**ignificant **B**yte. The MSB is always the first byte transmitted over a HART data link.
NAN	**N**ot-**a**-**N**umber.
RC	**R**esponse **C**odes
UART	**U**niversal **A**synchronous **R**eceiver **T**ransmitter

5. DATA TYPES

This section defines the data types supported by the HART Protocol. Devices claiming compatibility with the HART Protocol may only transmit these data types across a HART network:

- Strings using ISO Latin-1 or Packed ASCII Formats
- Date
- Single and Double Precision Floating-Point
- Signed Integers
- Unsigned Integers
- Table Based Data using Enumerations or Bit Fields

All multi-byte data must be transmitted sequentially starting from the most significant byte (MSB) and ending with the least significant byte (LSB). The length of all data items is fixed. Therefore, the length of a data item may not vary from one bus transaction to another.

Unsigned integers, enumerations and bit fields may be packed together. Various lengths of unsigned integers, enumerations and bit fields may be combined, provided the data (with null bits added as necessary) is always passed in multiples of eight bits.

5.1 String Formats

Two string formats are supported: ISO Latin-1 and Packed ASCII. Strings are always fixed in length.

5.1.1 Packed ASCII

Packed ASCII is a HART-specific 6-bit character code representing a subset of the ASCII character code set (see Table 1). Produced by compressing four Packed ASCII characters into three 8-bit bytes, Packed ASCII strings must be a multiple of 4 characters (3 bytes) and must be padded out to the end of the data item with space characters. For example, 4 space characters at the end of a string would appear as the 3 bytes: 0x82, 0x08, and 0x20.

Uninitialized Packed ASCII strings must be set to the question mark (?) character. Four question mark characters appear as the bytes: 0xFF, 0xFF, and 0xFF.

Table 1. Packed ASCII Character Set

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
2	SP	!	"	#	$	%	&	’	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?

Note: Most significant hexadecimal digit top to bottom; least significant left to right.

Table 2. Packed ASCII Characters versus 8-Bit Bytes

Packed ASCII Character	**0**	**1**	**2**	**3**
Packed ASCII Bit	5 0	5 0	5 0	5 0

Bit Position	7 0	7 0	7 0
Byte Number	0	1	2

Note: A field device does not need to unpack a Packed ASCII string unless the field device can display the string.

Construction of Packed ASCII characters:
Constructing a Packed ASCII string is a simple matter of discarding the most significant two bits from each character and compressing the result:

1. Truncate Bit #6 and #7 of each ASCII character.
2. Pack four, 6 bit-ASCII characters into three bytes.
3. Repeat until the entire string is processed.

This algorithm can be implemented by masking and shifting four 6-bit characters into a 24 bit register then moving the three bytes into the Packed ASCII string.

Reconstruction of ASCII characters:
Unpacking Packed ASCII strings requires flipping some bits in addition to uncompressing the string itself. To unpack a Packed ASCII string:

1. Unpack the four, 6-bit ASCII characters.
2. For each character, place the complement of Bit 5 into Bit 6.
3. For each character, reset Bit 7 to zero.
4. Repeat until the entire string is processed.

This algorithm can be implemented by loading three bytes into a 24-bit register and shifting the four 6-bit characters into the string. Parse the resulting character to flip bit 6 as needed.

5.1.2 ISO Latin-1

ISO Latin-1 (ISO 8859-1) strings consist of one character per byte (see Table 3). ISO Latin-1 strings must be padded out to the end of the data item with zeros (0x00). One zero (0x00) at the end of a string ISO Latin-1 data item is not sufficient to meet this requirement. A valid ISO Latin-1 character is allowed in the final byte of an ISO Latin-1 data item. Compared strings (e.g., in Universal Command 21, Read Unique Identifier Associated With Long Tag) are sensitive to character case.

Uninitialized ISO Latin-1 strings must be set to the question mark (?) character.

Table 3. ISO Latin-1 Characters

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8																
9																
A		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	¯	®	¯
B	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Note: 1. Most significant hexadecimal digit top to bottom; least significant left to right.
2. SP indicates a space character.
3. NBSP indicates a non-breaking space character.
4. SHY indicates a soft hyphen

5.2 Dates

In the Protocol, dates are represented by three 8-bit binary unsigned integers representing, respectively, the day, month, and year minus 1900. Date is transmitted day first followed by the month and year bytes. This allows the representation of any date between 1 January 1900 and 31 December 2155.

5.3 Floating-Point Formats

IEEE-754 (IEC 559) compatible single and double precision floating-point formats are supported by the Protocol. In addition, all devices must support single precision floating-point numbers. Floating-point numbers consist of three parts: a sign bit, the exponent, and the fractional portion of the mantissa. The following summarizes the IEEE 754 floating-point formats. Detailed implementation information is beyond the scope of this specification.

S Sign of the mantissa (1 = negative)

E The biased exponent. Subtracting the bias results in a 2's complement integer.

F The fractional portion of the mantissa. Since the mantissa is between 1.0 and 2.0 the integer portion of the mantissa is always 1. The integer portion is not included in IEEE 754 single and double precision formats.

The sign and MSB of the exponent is transmitted first followed by the balance of the exponent and the MSB-LSB of the fraction. Furthermore, any floating-point number communicated via the Protocol must have either explicitly or implicitly an associated Engineering Unit Code (see *Common Tables Specification*). Figure 3 shows the floating-point formats supported.

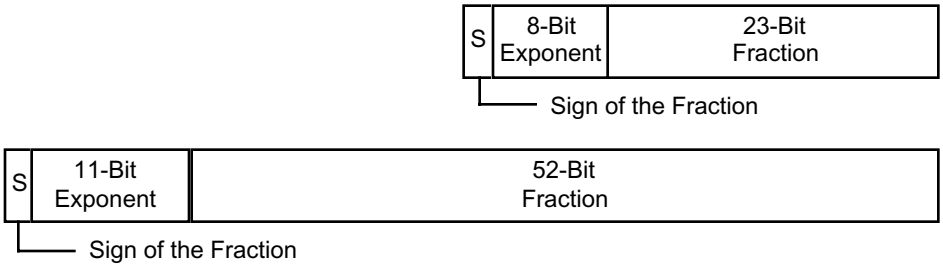

Figure 3. Single and Double Precision Floating-Point Formats

There are four specially defined values: positive infinity (+), negative infinity (-), Not-a-Number (NaN), and denormalized numbers. When the Command Specification permits, a floating-point parameter not used by a device must be filled with a specific NaN: 0x7F, 0xA0, 0x00, 0x00.

Table 4. Summary of Floating-Point Number Properties

	Single Precision	Double Precision
Number of Bytes	4	8
Range	3.4E +/- 38 (7 digits)	1.7E +/- 308 (15 digits)
Bias of Exponent	127	1023
Conversion to Real Number	$(-1)^S * 2^{(E-127)} * 1.F$	$(-1)^S * 2^{(E-1023)} * 1.F$
Normalized	0 < Exponent < 255	0 < Exponent < 2047
Denormalized	Exponent = 0 Nonzero Fraction	Exponent = 0 Nonzero Fraction
Zero	Exponent = 0 Fraction = 0	Exponent = 0 Fraction = 0
Infinity	Exponent = 255 Fraction = 0	Exponent = 2047 Fraction = 0
NaNs	Exponent = 255 Nonzero Fraction	Exponent = 2047 Nonzero Fraction

5.3.1 Normalized Numbers

Normalized numbers (Figure 4) represent all real values between the minimum and maximum allowed for that floating-point format. These values can be used directly in calculations or for values to or from a HART field device.

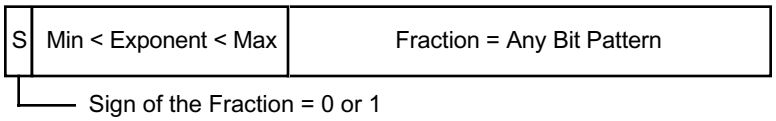

Figure 4. Normalized Numbers

5.3.2 Denormalized Numbers

The IEEE Standard allows for a smooth transition when an underflow occurs. In other words, zero can be gradually approached from nonzero by incrementally loosing precision. A denormalized number always has a zero valued exponent; the integer portion of the mantissa equal to zero; and a nonzero fraction. Denormalized number may be positive or negative.

Note: Some HART compatible devices may not support denormalized numbers.

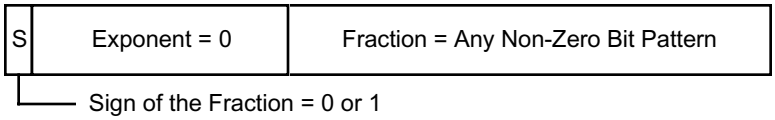

Figure 5. Denormalized Numbers

5.3.3 Zeros

A zero always has: a zero valued exponent; the integer portion of the mantissa is zero; and the fraction is zero. Zeros may be positive or negative (i.e., +0.0 or -0.0).

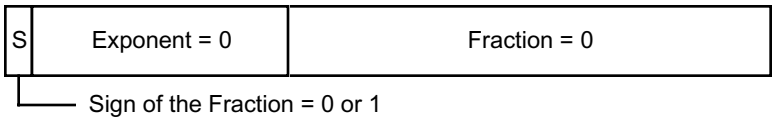

Figure 6. Zeros

5.3.4 Infinities

Infinities indicate a positive or negative overflow condition. An infinity always has: the exponent set to its maximum value; the integer portion of the mantissa is zero; and the fraction is zero.

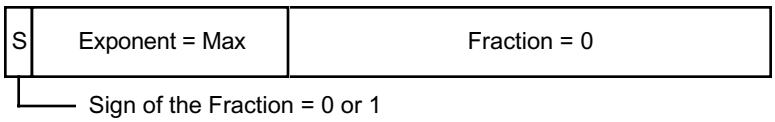

Figure 7. Infinities

5.3.5 Not-a-Numbers

NaN indicates that the number cannot be interpreted. The IEEE Standard allows for both signaling and non-signaling NaNs. A NaN always has the exponent set to its maximum value; the integer portion of the mantissa equal to zero, and a nonzero fraction. The most significant bit of the fraction is set to indicate a non-signaling NaN. Non-Signaling NaNs are generated as the result of a calculation that has no mathematical interpretation and signaling NaNs allow for implementation specific data definitions. Signaling NaNs must not be created as the result of a calculation. For example, a single, specific signaling NaN (0x7F, 0xA0, 0x00, 0x00) is allowed in some Command Specifications to indicate when the field device does not support certain data values.

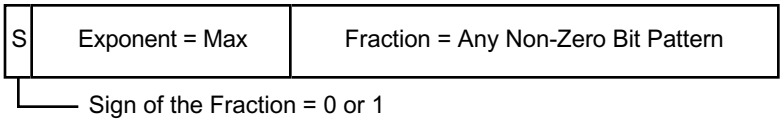

Figure 8. NaNs

5.4 Unsigned Integer Format

This data type is a binary representation of a positive whole number. Unsigned integers must not have engineering units explicitly or implicitly associated with their value. While the length of an individual data item is fixed, unsigned integers range from 1 to 24 bits or more in length. Unsigned integers shall be transmitted MSB first.

Note: The maximum size of an unsigned integer supported by Host Applications varies. Caution should be exercised when using long unsigned integers in field device designs

Table 5. Unsigned Integer to Decimal Conversion

15 Bit 8				7 Bit 0			
15 12		11 8		7 4		3 0	
4th Hex Digit		3rd Hex Digit		2nd Hex Digit		1st Hex Digit	
Hex	Decimal	Hex	Decimal	Hex	Decimal	Hex	Decimal
0	0	0	0	0	0	0	0
1	4,096	1	256	1	16	1	1
2	8,192	2	512	2	32	2	2
3	12,288	3	768	3	48	3	3
4	16,384	4	1024	4	64	4	4
5	20,480	5	1280	5	80	5	5
6	24,576	6	1536	6	96	6	6
7	28,672	7	1792	7	112	7	7
8	32,768	8	2048	8	128	8	8
9	36,864	9	2304	9	144	9	9
A	40,960	A	2560	A	160	A	10
B	45,056	B	2816	B	176	B	11
C	49,152	C	3072	C	192	C	12
D	53,248	D	3328	D	208	D	13
E	57,344	E	3584	E	224	E	14
F	61,440	F	3840	F	240	F	15

Note: The decimal number is calculated by adding the values of each hex digit together.

5.5 Signed Integer Format

This data type is a 2's compliment binary representation of a whole number. Signed integers must not have engineering units explicitly or implicitly associated with their value. Since the most significant bit being set indicates the signed integer is negative, this bit must be extended when a signed integer is extended (e.g., when moving a 1 byte signed integer into a 2-byte signed integer for internal calculations). Signed integers must always be passed in multiples of eight bits. Signed integers shall be transmitted MSB first.

Note: The maximum size of an signed integer supported by Host Applications varies. Caution should be exercised when using long signed integers in field device designs.

Table 6. Signed Integer to Decimal Conversion

Positive Values

15 Bit 0							
15 12		11 8		7 4		3 0	
4th Hex Digit		3rd Hex Digit		2nd Hex Digit		1st Hex Digit	
0	0	0	0	0	0	0	0
1	4096	1	256	1	16	1	1
2	8192	2	512	2	32	2	2
3	12288	3	768	3	48	3	3
4	16384	4	1024	4	64	4	4
5	20480	5	1280	5	80	5	5
6	24576	6	1536	6	96	6	6
7	28672	7	1792	7	112	7	7
8		8	2048	8	128	8	8
9		9	2304	9	144	9	9
A		A	2560	A	160	A	10
B		B	2816	B	176	B	11
C		C	3072	C	192	C	12
D		D	3328	D	208	D	13
E		E	3584	E	224	E	14
F		F	3840	F	240	F	15

Negative Values

15 Bit 0							
15 12		11 8		7 4		3 0	
4th Hex Digit		3rd Hex Digit		2nd Hex Digit		1st Hex Digit	
0		0	-3840	0	-128	0	-16
1		1	-3584	1	-240	1	-15
2		2	-3328	2	-224	2	-14
3		3	-3072	3	-208	3	-13
4		4	-2816	4	-192	4	-12
5		5	-2560	5	-176	5	-11
6		6	-2304	6	-160	6	-10
7		7	-2048	7	-144	7	-9
8	-28672	8	-1792	8	-112	8	-8
9	-24576	9	-1536	9	-96	9	-7
A	-20480	A	-1280	A	-80	A	-6
B	-16384	B	-1024	B	-64	B	-5
C	-12288	C	-768	C	-48	C	-4
D	-8192	D	-512	D	-32	D	-3
E	-4096	E	-256	E	-16	E	-2
F	0	F	0	F	0	F	-1

Note: The decimal number is calculated by adding the values of each hex digit together. If the most significant bit is set then the number is negative and the right hand table should be used.

5.6 Lookup Table Formats

The Protocol uses lookup tables to associate specific definitions to numeric codes or to define the interpretation of an individual bit when it is set. When a field device uses tables, the numeric values have specific definitions that allow all hosts to consistently utilize that data. Tables are coded using one of the following:

- **Enumerations** communicated as an unsigned integer with each value having a specific, unique, unambiguous definition. The enumeration provides the index to an entry in a lookup table containing the corresponding definition. A single byte allows a data item to depict one of 256 possible terms

- **Bit Fields** communicated as part of an unsigned integer with each bit having a specific, unique, unambiguous meaning when that bit is set. Each bit in the unsigned integer is listed as a separate entry in a lookup table. A single byte can depict up to 8 terms, one definition for each bit set.

The *Common Tables Specification* contains a collection of standard lookup tables (e.g., Engineering Units, Manufacturer Identification Code) used throughout the Protocol. These tables must be used whenever possible.

When Common Tables are referenced, the tables or subsets of the tables must be used exactly as specified. For tables not covered by the *Common Tables Specification*, refer to the associated device-specific tables. Device-specific tables must be included in the device-specific document. All tables must comply with the following revision requirements:

1. When a table is created, all possible values need not be defined. All undefined codes in the table are reserved. Reserved values must not be used, returned or communicated by any device.
2. A term or definition may be added to a table only by replacing a reserved code and performing a major revision to the table. This will result in an increase in the increment of the major revision level of the associated document.
3. Codes may be added to a table. Once defined, however, the term and its corresponding code must not be changed and may not be deleted.

5.6.1 Enumerations

Data items that take on a single meaning from a list or table are encoded as enumerations and communicated as an unsigned integer. When an enumerated table is created, the largest code is reserved. Single byte (i.e., 8-bit) enumeration-based lookup tables are defined using the following format:

Table 7. Single Byte Enumerated Table Format

Code	Definition
0	Specified definition 0
1	Specified definition 1
n	Specified definition n
n+1	Reserved
n+2	Reserved
249	Reserved
250	"Not Used"
251	"None"
252	"Unknown"
253	"Special"
254	"Expansion"
255	Reserved

The definitions for codes 250—255 in single byte enumeration-based look-up tables are mandatory if the enumerations are allowed (i.e. the codes are not reserved). Codes need not be allocated sequentially. Any intervening unused codes shall be "Reserved". When designated by the Command Specification, a table-based data item unused by a device is coded 250 ("Not Used").

5.6.2 Bit Fields

Communication of information encoded as single-bit data (such as status and diagnostic information) may be communicated using bit fields. Individual bits are packed together, and reserved bits are added as necessary to communicate a whole number multiple of eight bits. As a result, only a whole number of bytes is communicated.

A lookup table is used to define the meaning of each bit set in the bit field. In other words, each bit shall be either true (set to a 1) or false (set to a 0). The entry in the lookup table shall define the meaning of the bit when that bit is true (set to a 1). Single-byte (8-bit) tables for bit fields must use the following format:

Table 8. Single-Byte Bit Field Table Format

Bit Mask	Specified Definition
0x01	For Bit 0 (LSB)
0x02	For Bit 1
0x04	For Bit 2
0x08	For Bit 3
0x10	For Bit 4
0x80	For Bit 7 (MSB)

Codes need not be allocated sequentially. Any unused codes shall be "Reserved". Multi-byte bit fields are transmitted MSB first.

6. FIELD DEVICE REVISION RULES

Compatibility between devices is a primary objective of the Protocol. The Field Device Revision Rules in this section allow enhancement and modification of HART compatible field devices while ensuring that compatibility is maintained. These rules ensure that a field device may be replaced with a new version of the same Device Type without disrupting system operation. Furthermore, hosts must provide the same functionality for a new Device Revision as available for the previous supported Device Revision without requiring a software upgrade or change.

All devices must adhere to the following revision rules:

1. All devices must adhere to a valid revision of the *HART Field Communications Protocol Specification*.
2. Device support of Command 48 must adhere to the requirements in the *Common Practice Command Specification*. Use of Extended Device Status, Operating Mode, Analog Channel Saturated, and Analog Channel Fixed must be compliant.

   Note: HART 5 Field Devices that do not adhere to Protocol requirements for Operating Mode 1, Operating Mode 2, Analog Channel Saturated, and Analog Channel Fixed must correct their implementation and change their Device Type number. For example, Operating Mode 1 and Operating Mode 2 are reserved (i.e., always zero) in HART 5 compliant Field Devices.

3. At no time may a command be deleted from any device. Once a command is supported by any Device Revision, all subsequent revisions must support the command.
4. The meaning of any data item in a command supported by the field device must never be changed. Commands may not be modified in any way to produce results different than would be obtained in a previous Device Revision. For example, any supported command issued to a Device Revision 1 field device must have exactly the effect when issued to Device Revision 2 field device.
5. Data items may be added to the end of any command. Any addition will result in an increment of the major revision level of the associated document or specification (i.e. a major Device Revision).
6. Entries may not be deleted from enumerated tables, bit fields or Response Codes. Furthermore, the meaning of an enumeration, bit field or Response Code may not be changed. Any addition will result in an increment of the major revision level of the associated document or specification (i.e. a major Device Revision).
7. The code number for a Device Variable must not change. The addition of a new Device Variable shall result in an increment of the Device Revision and the major revision level of the manufacturer's device-specific document (i.e. a major Device Revision).
8. The Device Family of a Device Variable must remain fixed. The Device Family may be changed from 250, "Not Used" by adding the appropriate Device Family Commands, an increment of the

Device Revision, and the major revision level of the manufacturer's device-specific document (i.e. a major Device Revision).

9. The classification of a Device Variable must remain fixed. The classification may be changed from 0, "Unassigned" by supporting the appropriate unit codes; an increment of the Device Revision; and the major revision level of the manufacturer's device-specific document (i.e. this change is a major Device Revision).
10. Field Devices may upgrade from HART 4 without changing the Device Type by following these rules:
    - All requirements of the HART Field Communications Protocol Specification being targeted by the Device Revision must be adhered to.
    - Commands 4 and 5 must be deleted.
    - All commands, except Command 0, must be implemented only in the Long Frame Format. Command 0 must be implemented in both Short and Long Frame Format.
    - A separate Configuration Changed bit in the Device Status byte must be provided for each master (i.e. one for the Primary and another for the Secondary master). When received, Command 38 must reset only the bit corresponding to the master issuing the command. As required in HART 5, these bits must be non-volatile.
    - A separate Cold Start bit in the Device Status byte must be provided for each master (i.e. one for the Primary and another for the Secondary master).
    - If Command 48 is implemented, the device must return at least bytes 0—7 of the Response Data sub-field.
11. Field Devices may upgrade from HART 5 without changing the Device Type by following these rules:
    - All requirements of the *HART Field Communications Protocol Specification* being targeted by the Device Revision must be adhered to.
    - A separate Configuration Changed bit in the Device Status byte must be provided for each master (i.e. one for the Primary and another for the Secondary master). When received, Command 38 must reset only the bit corresponding to the master issuing the command. As required in HART 5, these bits must be non-volatile.
    - A separate Cold Start bit in the Device Status byte must be provided for each master (i.e. one for the Primary and another for the Secondary master).
    - If Command 48 is implemented, the device must return at least bytes 0—7 of the Response Data sub-field.

When device changes do not comply with these revision rules, a new Device Type number must be assigned for the device. To avoid customer confusion, a visible change to the product (such as a new product name or packaging) should accompany the assignment of the new Device Type number.

7. APPLICATION LAYER INTERFACE

The HART Protocol provides specifications to facilitate two-way communication between field devices and host applications. The Protocol's Application Layer is command based. In other words, commands from master or slave devices are the basis for HART communication and the command number, embedded in the communication, determines the content of the message.

Figure 9 shows the HART Frame Format. This section defines the contents and format of the Command, Byte Count and Data fields. In other words, this section defines how host applications and slave devices interface to the HART Application Layer to accomplish two-way communication. This section includes:

- The allocation of command numbers between Universal, Common Practice, Device Family and Device-Specific functions;
- Defining the format of the Data field based upon command number, master requests and slave responses, and the type of status information contained in the Data field;
- Specifying common requirements for all HART commands; and
- Defining the contents of the Command Status Bytes.

Figure 9. HART Frame Format

The Command Number indicates the specification of a unique, unambiguous packet of Data with a fixed Byte Count. These command number specifications are classified in Section 7.1.

7.1 Command Number Partitions

This section allocates command numbers between Universal, Common Practice, Device Family and Device-Specific functions. The Protocol supports single byte command numbers (0—255) and two byte extended command numbers (256—65,535). Commands 0—255 use the Command field of the message to indicate the command number. For commands 256—65,535 the Command field contains 31 (0x1F) and the two byte extended command number is found in the Data field. The command set is divided into the following classes (see Table 9):

- **Universal Commands.** A collection of commands that must be supported by all HART compatible devices. All Universal Commands must be implemented exactly as specified (see Universal Command Specification).

- **Common Practice Commands.** A collection of commands applicable to a wide range of devices. Common Practice Commands should be supported by Devices whenever possible (see Common Practice Command Specification). If a device uses a Common Practice Command, the command must be implemented exactly as specified.

- **Non-Public.** A special set of commands (122 through 126) intended for factory-only use during the construction of a field device. These commands should not be used when servicing a device in the field.

- **Device Family Commands.** Collections of commands that allow the setup and parameterization of field devices without requiring using device specific commands or special device-specific drivers (see *Device Family Command Specification*). Device Variables are classified into families based on the type of process connection they support (e.g., pressure, temperature, valve/ actuators, flow, etc.). The Device Variable is the target of the Device Family Commands.

- **Device-Specific Commands.** Commands defined by the manufacturer according to the need of the field device. These commands are controlled by the manufacturer of the field device (see the manufacturer's device-specific document).

Reserved command numbers shall not be used in any device.

Table 9. HART Command Number Partitions

Command No.	Type	Message Field	
		Command	Data
0—30	Universal		See *Universal Command Specification*.
31	Expansion Flag	31 (0x1F)	If no data bytes are present, then a field device must answer "too few bytes received". A 31 in the Command Field indicates an extended command number and requires a minimum of two bytes in the Data field .
32—121	Common Practice		See *Common Practice Command Specification*.
122—126	Non-Public		Intended for factory use only during the construction of the field device.
127	Reserved		
128—253	Device-Specific		See the manufacturer's device-specific document.
254—1023	Reserved		
1024—33,791	Device Family	31 (0x1F)	See *Device Families Command Specification*
33,792—65,535	Reserved		

7.2 Data Field Format

This section discusses the contents and organization of the Data field. The Data field supports two-way communication between devices in the following ways:

- When communication is successful, the Data field contains the information communicated between devices.
- For extended commands, the Data field includes the two byte extended command number allowing devices to determine unambiguously the information content of the message.
- For all slave responses the Data field contains at least the two Command Status Bytes providing continuous feedback to host applications.
- When an error is detected in the host communication, the slave response does not contain the information indicated by the command number. This gives host applications clear indication that the error prevented successful execution of the command.

To meet the above objectives the structure of the Data field varies between eight different Data field formats. Figure 10 can be used to determine the format of a Data field for any HART message. There are two master request formats (normal and extended command) and five possible slave response formats. Slaves generally echo the master request except when there is an error. When an error is detected, the slave response is truncated returning only the error information. For slave devices not supporting extended command numbers further simplification can reduce the data field formats to four: Master Request, Normal Slave Response, Communications Error Slave Response and Command Error Slave Response.

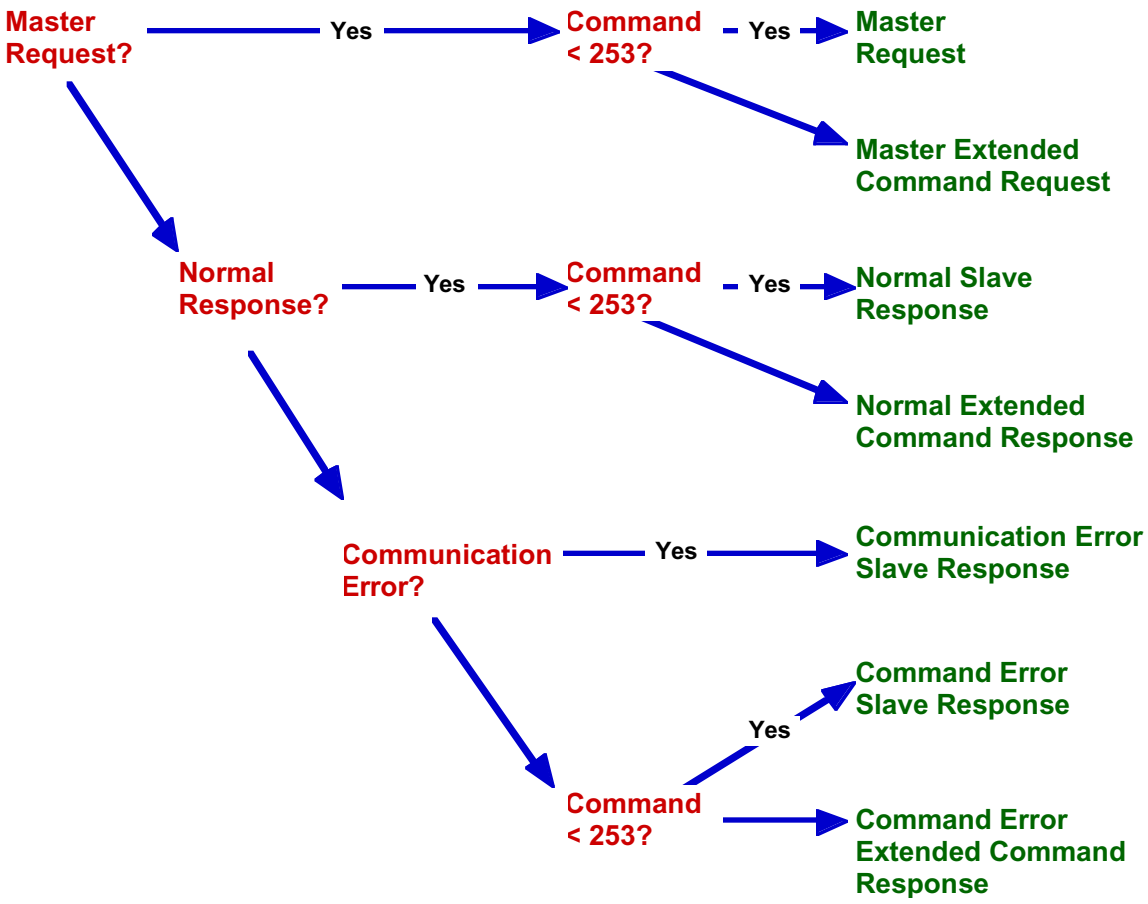

Figure 10. Determining Data Field Format

These eight Data field formats consist of differing combinations of the following six sub-fields:

- **Request Data Bytes.** This sub-field contains the data items communicated from the host to the field device. The content and length of this field is defined in the specification for the issued command. For any command the bytes in this field are always numbered starting at zero, although the first byte in the Request Data may not be the first byte in the Data field of the message.

- **Response Data Bytes.** This sub-field contains the data items communicated from the field device to the host. The content and length of this field is defined by the Command Specification. The bytes in this field are always numbered starting at zero, although the first byte in the Response Data sub-field is never the first byte in the Data field of the message.

- **Extended Command Numbers.** This is a 16-bit command number used to extend the number of HART commands to 65,536. Commands 256—65,535 may not be used for device-specific commands (see the *Device Families Command Specification*).

- **Communication Status.** This byte is multiplexed with the Response Code byte and indicates field device detection of a communication error. A communication error is always indicated by a one (1) in the most significant bit of this byte. When the field device does not detect a communication error, the Response Code is returned in the response message.

- **Response Code.** This byte is multiplexed with the communication status byte and indicates the status of the field device's execution of the host request. The most significant bit of the Response Code is always zero (0).

- **Device Status.** The Device Status sub-field provides a high level indication of the field device health and status. Field Device Status is returned in every slave response to provide continuous host feedback regarding field device health and operation.

These six sub-fields are not included in the construction of every message. The following sections describe message construction in application layer communication based upon the function being performed.

7.2.1 Requests With Single Byte Command Numbers

Figure 11 shows the format of the application layer fields that must be used for a master request containing a single byte command number. The complete command number is contained in the one byte Command field and the Request Data Bytes start at byte zero (0) of the Data Field.

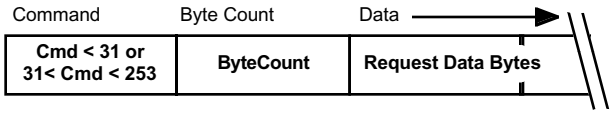

Figure 11. Master Request

All slave response messages must contain two status bytes (see Figure 12). Normally, the first status byte contains the Response Code providing information about the field device's execution of the command. See Section 7.2.3 for response formats when the field device detects an error.

The second status byte contains Field Device Status information. The Response Data Byte field follows these two bytes provided, an error was not encountered in the master request.

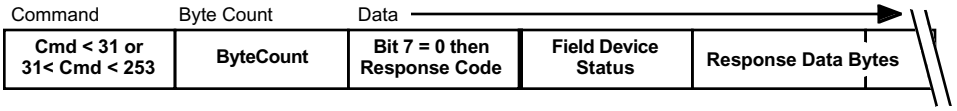

Figure 12. Normal Slave Response

7.2.2 Request With Extended Command Numbers

Figure 13 shows the format of the application layer fields that must be used for a master request containing an extended command number. When the Command field is 31 (0x1F), the first two bytes of the Data field must contain the Extended Command sub-field. These two bytes must be followed by the Request Data Byte field as defined in the command specification. A device receiving a message with a Byte Count less then 2 and 31 in the Command field must return the single-byte command error Response Code 5, "Too Few Data Bytes" (see Figure 16).

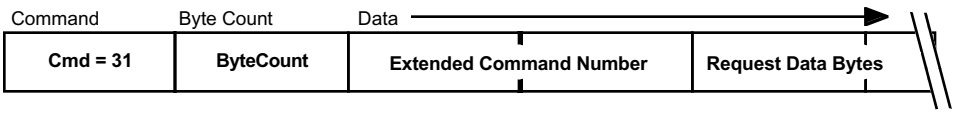

Figure 13. Master Extended Command Request

All slave response messages to extended commands must contain both the two status bytes plus the two byte extended command number (see Figure 14). The two status bytes are the same as for

single-byte command numbers. A device receiving a message with a 16bit extended command number less than 512 must return the command error Response Code 20, "Invalid Extended Command Number". Section 7.2.3 characterizes response formats when the field device detects an error.

The Response Data Byte field defined in the command specification must follow these four bytes provided an error was not encountered in the master request.

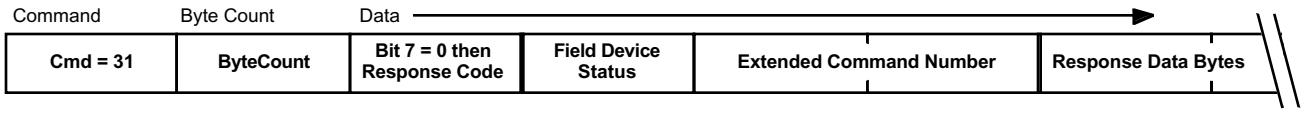

Figure 14. Normal Extended Command Response

7.2.3 Error Responses

Figure 15, Figure 16, and Figure 17 show the format of the response message when a field device detects an error in a the master request. The Response Data Bytes must not be returned when the field device detects an error in the master request.

A communication error has priority and is indicated when the most significant bit of the first byte of the Data field is set. See the *Data Link Layer Specification* for more information about slave device operation when a communication error is detected. Any slave response indicating a communication error must use the format shown in Figure 15.

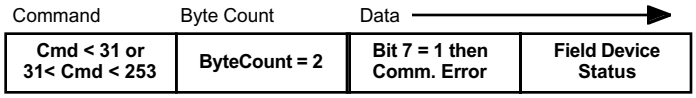

Figure 15. Slave Response with Communication Error

If there are no communication errors, the slave device must return the Command Response Code in the first byte of the Data field. Figure 16 shows the format of the slave response to a master request containing a single byte command number when the Response Code sub-field indicates an error. Response Codes must be less then 127 (0x7F), and the most significant bit of the first byte of the Data field must be cleared (see the *Command Response Code Specification*).

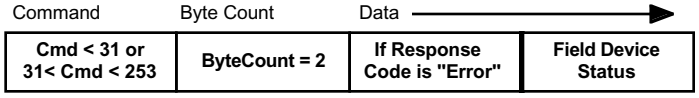

Figure 16. Slave Response for Single Byte Command with Command Error

Field Device Status must be returned in every slave response to provide continuous host feedback regarding field device health and operation.

For extended commands, the two-byte command number must be included in error response from the field device. The Response Code and the Field Device Status sub-fields are always returned first to maintain backward compatibility. Figure 17 shows the format of a slave error response to an extended command.

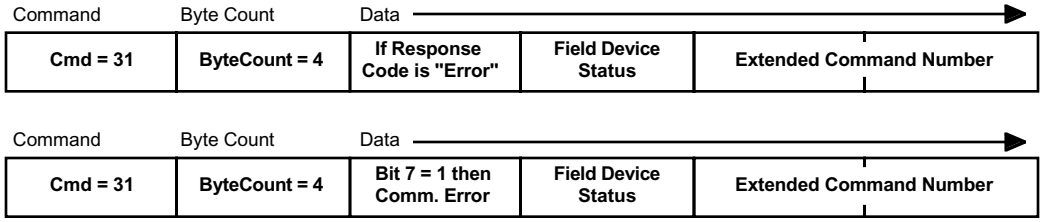

Figure 17. Extended Command Response with Command Error

7.3 Command Requirements

The HART Application Layer is command based. To insure consistent implementations all HART commands must meet the requirements in this section. These requirements are applicable to both commands in the HART Protocol Specifications and device-specific commands developed by individual manufacturers of HART compatible devices.

7.3.1 Autonomous & Asynchronous Requirements

HART commands must be designed to be autonomous and allow stateless operation of the device's application layer. In other words, neither a slave nor a master can be required to remember prior communications in order to understand the current message. The response to be supplied by a slave device must be clearly and uniquely specified by the corresponding master request alone.

7.3.2 Command Operations

A HART command must perform one and only one of the following functions:

- **Read** data from a field device. READ commands contain no data in the Request Data Byte field other than that required by the field device to return the desired data items. In other words, the only data item that may be in the request is an index referencing an array entry in a field device (e.g., the device variable number). READ commands should have several related data items in the Response Data Byte field in order to minimize host upload times. A READ command must not change the operation of the field device in any way or change any data item stored in the field device.
- **Write** data to a field device. WRITE commands shall contain the same data items in the Request and Response Data Byte fields. Furthermore, the field device must return in the Response Data Byte sub-field, the actual value of the data item as used by the field device in the same Engineering Units as sent with the data item. Any data item that can be written by a host must be contained in a READ command to allow the configuration of the field device to be determined or saved by the host. Index data items and unit codes may not be stored or changed in a field device as the result of a WRITE commands.

  Unless otherwise stated in the Command Specification, all data transmitted to a field device using a WRITE commands must be retained through a Device Reset, Self Test or removal of power from the field device.
- **Command** the device to perform some action. As a result, a COMMAND command may or may not have data items in the Request or Response Data Byte fields. COMMAND commands may affect the operation or configuration of the field device, or data items in the field device.

  Unless otherwise stated in the Command Specification, all device operations or configurations affected by a COMMAND command shall be retained through a Device Reset, Self Test, or removal of power from the field device.

7.3.3 Indexed Commands

Commands may contain indices allowing access to arrays or tables of data stored in a field device. The index is represented as an unsigned integer. This allows single command access to an array of data. The number and the type of data items must be the same and must occur in the same sequential order in the Request and Response Data Byte field for all values assumed by the index. In other words, an indexed command must define a rigid packet of information and, for each index value, the structure of this packet must be identical.

There must be only one set of Command-Specific Response Codes for all values of the index.

7.3.4 Multi-Transaction Commands

Multi-transaction commands allow a sub command number to be placed in the Request and Response Data field to increase the number of device-specific commands. Multi-transaction commands should only be used when a field device exhausts the allowed set of device-specific commands. Furthermore, all transactions must perform the same READ, WRITE or COMMAND operation (see Section 7.3.2).

Unlike indexed commands, the number and the type of data items in the Request and Response Data Byte field can vary by transaction number. As a result, the command specification must include a separate Command-Specific Response Code specification for each transaction.

Both the Request and Response Data field must include the transaction number to meet the Protocol requirements for autonomous operation.

7.4 Command Status Bytes

This section defines the Command Status Bytes that provide host application feedback on slave device command execution. All slave response messages must return two Command Status bytes in the first two bytes of the Data field. The first byte is multiplexed and contains either the Communication Status (most significant bit is set) or the Response Code (most significant bit is reset). The second byte of a slave response message always contains Field Device Status.

- **Communication Status** is returned if a communication error is detected by the field device.
- If there are no communication errors then the **Response Code** gives the result of the executed command.
- The **Device Status** represents the current state of the slave.

The Response Data Bytes must not be returned if a communications or command error is reported in the Command Status Bytes.

7.4.1 Communication Status

This byte is multiplexed with the Response Code byte and indicates field device detection of a communication error. A communication error is always indicated when the most significant bit is set to one. The Communication Status is defined as bit field table (see Table 10). The Communication Status is only returned when a communication error is detected.

Table 10. Communication Status

Bit Mask	Definition
0x80	**1** - This bit must always be set to indicate a communication error
0x40	**Vertical Parity Error** —The parity of one or more of the bytes received by the device was not odd.
0x20	**Overrun Error** — At least one byte of data in the receive buffer of the UART was overwritten before it was read (i.e., the slave did not process incoming byte fast enough).
0x10	**Framing Error** —The Stop Bit of one or more bytes received by the device was not detected by the UART (i.e. a mark or 1 was not detected when a Stop Bit should have occurred)
0x08	**Longitudinal Parity Error** — The Longitudinal Parity calculated by the device did not match the Check Byte at the end of the message.
0x04	**Reserved**, set to zero
0x02	**Buffer Overflow** — The message was too long for the receive buffer of the device.
0x01	**Reserved**, set to zero

7.4.2 Response Code

When no communications errors are detected, the first byte in the Data field contains the Response Code. The Response Codes contain a command completion report indicating the status of the command's execution by the field device. Response Codes provide a Notification, Warning or Error indication to the host (see Table 11)

Table 11. Response Code Classification

Response Code Class	Definition
Notification	Command executed properly. The Response Code equals zero (0) and the Response Data Bytes are returned.
Warning	Command executed with the deviation as described in response (e.g., a value was set to its nearest legal value). The Response Data Bytes are returned.
Error	Command execution was not properly completed and the Response Code indicates the reason (e.g., the device is in Write Protect mode). While the Extended Command number is included (if appropriate) in the slave response, the Response Data Bytes are NOT returned.

The most significant bit of the Response Code is always set to zero. As a result, the Response Code is encoded as a 7-bit enumeration (i.e., as an enumeration between 0 and 127). In addition to the above classifications, some enumeration must always use the same definition while others may have one of several possible definitions.

Single-definition Response Codes have the same meaning independent of the command that uses them. Multiple-definition Response Codes have several meanings. However, all Response Codes have a single meaning for a given command. The only legal Response Codes for a command are documented in the Command Specification found in the Protocol Specifications or, for device-specific commands, the manufacturer's device-specific documentation. Reserved Response Codes may not be used by any device.

Device-Specific Commands must use a single-definition Response Code wherever possible. Multiple-definition Response Codes may be recycled and used in device-specific commands. In other words, a manufacturer may use multiple-definition Response Codes as needed to return command completion information for their device-specific commands. For multiple-definition Response Codes used in this manner, the meaning of the Response Code must be defined in the manufacture's device-specific documentation.

For more information on Response Codes and their use, see the *Command Response Code Specification*.

7.4.3 Field Device Status

The Field Device Status is contained in the second data byte in a Slave-to-Master frame as a bit field table. The second data byte indicates the current operating status of the field device as a whole and is not associated with the completion of any command. Unlike the requirements of the HART Protocol prior to revision 6.0, this byte must be meaningful even if a communication error is reported in the first byte. Table 12 lists the meaning of bit masks for the Device Status Byte.

Note: Since this byte contains no unused bit, the Extended Device Status Byte has been added to Identity Commands and commands providing cyclical process data.

Table 12. Device Status

Bit Mask	Definition
0x80	**Device Malfunction** — The device detected a serious error or failure that compromises device operation.
0x40	**Configuration Changed** —An operation was performed that changed the device's configuration.
0x20	**Cold Start** — A power failure or Device Reset has occurred.
0x10	**More Status Available** —More status information is available via Command 48, Read Additional Status Information.
0x08	**Loop Current Fixed** —The Loop Current is being held at a fixed value and is not responding to process variations.
0x04	**Loop Current Saturated** —The Loop Current has reached its upper (or lower) endpoint limit and cannot increase (or decrease) any further.
0x02	**Non-Primary Variable Out of Limits** —A Device Variable not mapped to the PV is beyond its operating limits.
0x01	**Primary Variable Out of Limits** — The PV is beyond its operating limit.

Note: Voltage Mode Field Devices must use "Loop Current Fixed" and "Loop Current Saturated" to indicate the signaling voltage is fixed or saturated (see Section 8.1.1).

A **Device Malfunction** must be indicated when an error that prevents proper operation of the field device is detected (for example, at least one of the Device Variables is incorrect). Hosts should consider this an alarm condition and refrain from using the device for process control applications. Devices supporting Command 48 must provide diagnostic information in the Command 48 Response Data Byte field. The Device Malfunction bit mask must remain reset until a device completes its power-on self tests. If possible, the device must answer any requested HART command even if a Device Malfunction is detected.

A field device must contain a **Configuration Changed** bit for both the primary and the secondary master. Both bits are set when any configuration item in a field device is modified. These bits are used along with the **Configuration Change Counter** (see Command 0) to allow host to monitor the configuration of the field device. The Configuration Change Counter is a 16 bit counter that must be incremented once for every command received that changes the devices configuration. The value of the Configuration Changed bits Configuration and Change Counter must be maintained even if power is removed from the device or a Device Reset is performed.

A **Cold Start** bit is maintained by the field device for each master as well. Both Cold Start bits are set when the field device is powered up and after a Device Reset. The first command from a primary or secondary master automatically resets the corresponding Cold Start bit.

When **More Status Available** is set, the measurements may still be correct and suitable for use by control systems. More Status Available merely indicates that Command 48 contains diagnostic information that is useful to the host. Since setting More Status Available will cause most hosts to issue Command 48, thus decreasing available communication bandwidth, field device designers must consider which diagnostics must set this bit.

When the Loop Current is directly (e.g., using Command 40) or indirectly (e.g., using Command 79) fixed and not responding to process changes, the **Loop Current Fixed** bit must be set. Setting this bit indicates to hosts that the Loop Current should not be used for analog signaling of the PV between the control system and the field device.

Note: A field device may be in multi-drop with the Loop Current still responsive to changing process conditions (see Universal Command 6)

The Loop Current varies from 4—20mA. The Upper and Lower Range Values set the 4 and 20mA points (see Commands 15 and 35). Most field devices allow the loop current to exceed these endpoints by some small amount. Once these electrical limits established by the field device are exceeded, the **Loop Current Saturated** bit must be set. For transmitters, their output becomes saturated. For actuators, their input is over or under range. Figure 18 shows the Loop Current operation as it reaches saturation. The alarm level set by the device (e.g., when the **Device Malfunction** bit is set) must be sufficiently different from the Loop Current Saturated levels to allow differentiation by devices monitoring the Loop Current.

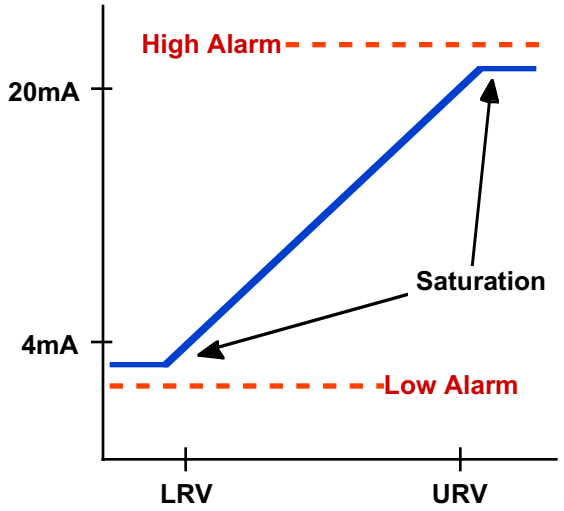

Figure 18. Loop Current Saturated versus Alarm Levels

The Percent Range, PV and the corresponding Device Variable must return valid readings beyond the Upper and Lower Range Values. The reading must be accurate accurately across the range defined by the Upper and Lower Transducer Limits (See Command 14). When the Device Variable (input or output) mapped to the PV reaches its Upper or Lower Transducer Limit, the **PV Out of Limits** bit must be set. For a device variable not mapped to the PV, the **Non-PV Out of Limits** bit must be used for this purpose instead. Limits may be electrical (e.g., in a transmitter) or mechanical (e.g., actuator at its travel limit as indicated by a limit switch).

8. DYNAMIC AND DEVICE VARIABLES

The HART Protocol is designed to support smart field device technology and the 4—20mALoop Current (see HART commands 1—3). Commands 1 and 2 return the Primary Variable (PV), Loop Current and Percent Range. Command 3 adds to these the Secondary (SV), Tertiary (TV) and Quaternary (QV) Variables. Collectively these are called the Dynamic Variables:

> **Dynamic Variables** consist of a Device Variable and an Analog Channel that, when combined, establish a communication link between the field device and the host application (e.g. the control system). All HART field devices may contain Primary, Secondary, Tertiary, and Quaternary Variables that are mapped to the first 4 analog channels in a field device. (see Figure 19). The first analog channel must support the Loop Current and HART communication. The SV, TV, and QV may or may not be supported and, furthermore, may not have an associated Analog Channel.

In addition, the Protocol supports Device Variables for use in more sophisticated smart field devices and multi-variable field devices. Furthermore, multi-variable field devices can configure which Device Variable to connect to the current loop.

Device Variables[1] are uniquely defined data items within a field device providing process-related information. A Device Variable's value changes as its connected process varies. A code number is assigned to each Device Variable and this assignment must never be changed for a given Device Type. The number of Device Variables is returned in Identity Commands. Device Variables should be numbered consecutively starting from zero (0).

Each Device Variable represents a direct or indirect connection between the process and the field device. An indirect connection infers the process state or level via calculations using direct process values from other Device Variables. In any case, a field device has a number of Device Variables (see Figure 19). Up to four of these Device Variables are mapped to the Dynamic Variables using the appropriate Device Variable number (see Command 50 and Command 51).

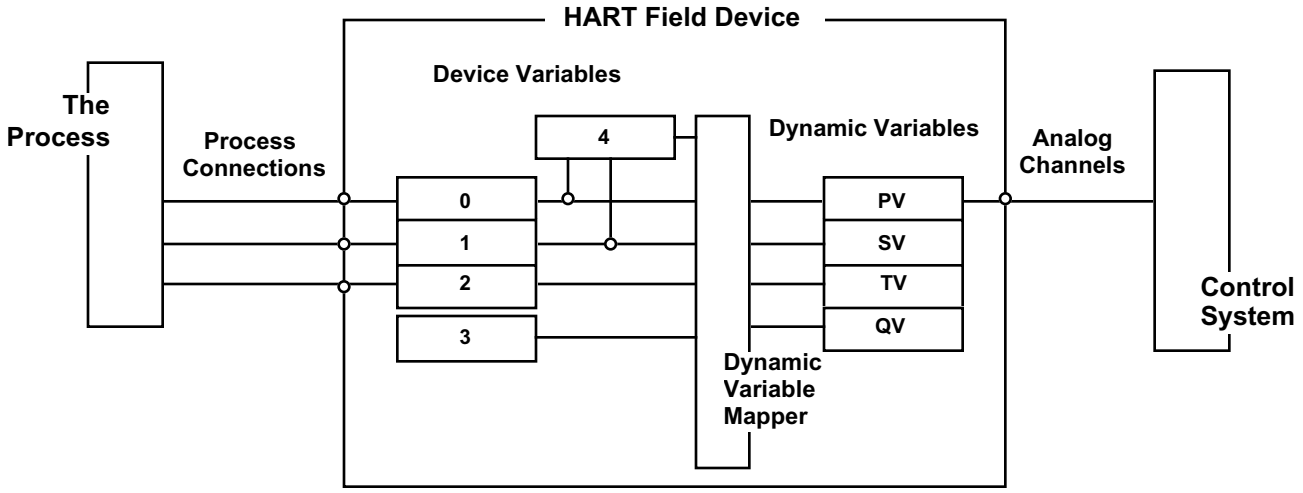

Figure 19. Device Variables and Dynamic Variables

The Dynamic Variables allow access to process-related data via HART Universal Commands and all HART compatible devices have Dynamic Variables. However, Device Variables are accessed via Common Practice commands. This means that even though all HART compatible devices have Device Variables, the device developer may choose to not expose the underlying Device Variables. As a result, a simple device may support only the Dynamic Variables and not implement the Common Practice Commands that access Device Variables.

[1] HART Rev. 5 referred to Device Variables as "transmitter variables". HART Rev. 6 uses the term "device variables" in recognition of actuators and non-transmitter type devices supporting the HART Protocol.

8.1 Primary Variable (PV)

The term "Primary Variable" is widely used in Command Specifications. Commands referring to the Primary Variable include Commands 1, 2, 3, 9, 34-37, 40, 43-47, 49. In all cases, Primary Variable refers to properties of the Device Variable and Analog Channel associated with the Loop Current. In effect, the Primary Variable Commands allow convenient access to all configuration properties and process-related values connected with the Loop Current. In addition, this coupling of the Loop Current to the Primary Variable allows the Loop Current to serve as a communication channel that transmits the Primary Variable's value from the field device to the measurement or control system.

The Primary Variable's properties can be segregated into two domains (see Figure 20).

- **Transducer**. This domain characterizes the connection between the field device and the process. Properties found in this domain include: Upper and Lower Transducer Limit, Transducer Serial Number, Transducer Trim Points, and Damping Factor.
- **Analog Channel.** This domain is responsible for conversion between the digital value and engineering units found in the Device Variable domain and the milliamp signal found on the current loop. There are two sub-domains encapsulated by any Analog Channel:
  - **Range Conversion**. This domain is responsible for converting between the Percent Range and the Primary Variable's value. Properties found in this domain include: Upper and Lower Range Values and the Transfer Function.
  - The **DAQ** domain converts between the physical milliamp value that can be measured on the loop and the Loop Current value returned in Command 2 and Command 3. Properties found in this domain include: the Alarm Code, the DAQ's Zero and Gain, and Loop Current Damping.

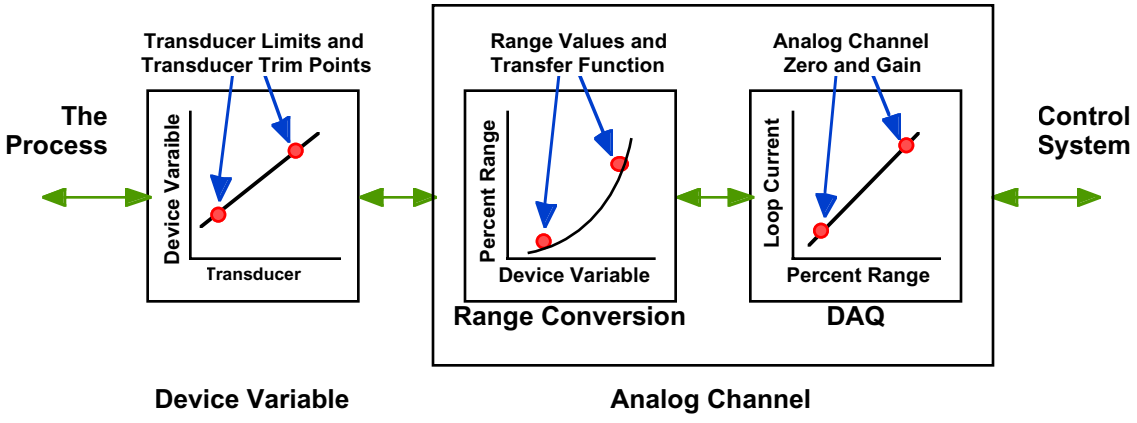

Figure 20. Domains Accessed Using PV

8.1.1 Voltage Mode Devices

A small number of HART compatible Field Devices support voltage signaling (e.g., 1-5 VDC) rather then Loop Current signaling (see Common Table 10, Physical Signaling Codes and the *FSK Physical Layer Specification*). Despite this, the phrase "Loop Current" is always used in the Protocol Specifications to refer to the value of the first analog channel in a Field Device which, in turn, is communicating the PV.

Voltage Mode Field Devices returns data in engineering units of "Volts DC" everywhere the Loop Current value is used (e.g., Commands 2, 3, 40, 45, 46)

Host applications must recognize Voltage Mode Field Devices when they are encountered. These devices are identified by the Physical Signaling Code found in Identity Commands.

8.2 Device Variable Classification

Device Variables can be classified by the function performed (see Common Table 21). The Device Variable Classification property can be read with Command 54, Read Device Variable Information. Once defined, the classification of Device Variable must not change. The Device Variable Classification indicates the type of process connection the Device Variable characterizes and the Engineering Units that may be supported by the Device Variable (See the *Common Tables Specification*). Any Device Variable that does not support a classification must set its classification to zero ("0") indicating that the base unit code table must be used by the Host.

Since field devices are not required to support the Device Variable Common Practice Commands (e.g., Command 54), Universal Command 8, Read Dynamic Variable Classifications, can be used to determine the classification of the Dynamic Variables returned in Commands 1, 2, and 3.

8.3 Device Families

The *Device Families Command Specification* provides collections of Command Specifications based on the Device Variable Family (see Common Table 20). These collections of commands allow the setup and parameterization of field devices without requiring special device-specific commands. Command 54 indicates the Device Family supported by a Device Variable (any Device Variable that does not support a family returns 250, "Not Used"). The Device Variable Family code indicates:

- Applicable Device Family commands (see the *Device Families Specification*); and
- Meaning of the Device Variable Status byte (see Section 8.4).

Note: A Field Device must not set any of the Device Family specific status bits or the "More Device Variable Status Available" unless that Device Variable supports a Device Family.

The *Device Families Command Specification* include a separate sub-specification document for each Device Family defined. These Device Family Specifications are developed for specific types of process connection or for specific process related functions. Each Device Family:

- Has a narrowly defined specified scope.
- Defines the properties for configuration or commissioning of the device variable.
- Classifies the device variable properties as optional or mandatory.
- Groups the properties into READ commands to optimize upload speeds.
- Defines WRITE commands for all properties in the collection.
- Specifies recommended Standard Operating Procedures (SOPs) for managing the device variable.

Device Variables that support the corresponding Device Family must support all mandatory commands and properties for that Device Family.

Since there are a limited number of HART commands and a large number of potential device families, Device Family commands all use Extended Command Numbers (see Section 7.2.2). Furthermore, Device Family command numbers are two bytes long with the Device Variable Classification in the most significant byte of the extended command number. 256 commands are allowed per device family.

8.4 Device Variable Status

All cyclical process data (i.e., Device Variables and Dynamic Variables) include a Device Variable Status byte (see Figure 21). The most significant two bits (i.e., bits 6 and 7) of every Device Variable Status byte returns the overall status of the Device or Dynamic Variable value. The next two bits indicate whether the Device Variable value is limited (i.e., not responding to the process). These four bits provide useful status about the Device Variable's value. For example, if the Process Data Status is "Manual / Fixed" and the Limit Status is "Not Limited" then the value is being manually controlled. Limit status would not be "Constant" because the value could be changed at any time by the user.

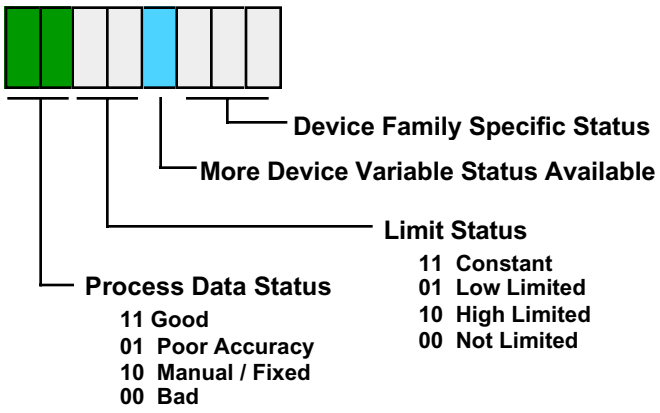

Figure 21. Device Variable Status Byte Format

In addition, the content of the lower 4 bits depend on the Device Variable Family. Each Device Family can have its own Device Family-specific status defining the least significant bits. If set, Bit 3 indicates that additional Device Family-specific status is available via the appropriate Device Family Command (see the *Device Families Command Specification*).

Note: If the Field Device ever sets any of the least significant 4 bits, then the Field Device must support Command 54 and the appropriate Device Family.

9. FIELD DEVICE IDENTIFICATION

All field devices must provide identification information to master devices upon request. The identifying data allows a master to, for example, address the field device or to ascertain the set of commands supported by the field device. The following data items are used for field device identification:

Table 13. Field Device Identifying Data

Property	Definition
Manufacturer ID	Indicates the company that produced the device. Manufacturer IDs are allocated by the HART Communication Foundation. Only the designated manufacturer may use this ID.
Device Type	Indicates the manufacturer's type of the device (i.e., the product name). The Device Type indicates the set of commands and data items supported by the Field Device. See Section 9.2 and 9.3 for more information. It is the manufacturer's responsibility to allocate the Device Type numbers for its products and ensure the Device Type numbers are unique.
Device ID	A number unique to a particular field device. This number must be different for every device produced with a given Manufacturer ID and Device Type.
Device Revision	A whole number indicating the revision level of command and data item set for this manufacturer's device type. Additions may be added to the commands and data items for a Device Type (see Sections 9.2 and 9.3).
Software Revision	An unsigned integer indicating the revision level of the firmware in the field device. An increment of this revision number must occur for every released version of the field device's firmware.
Hardware Revision	An unsigned integer indicating the major revision level of the hardware in the field device.
Universal Command Revision	A whole number indicating the HART major revision level supported by the field device.

Property	Definition
Private Label Distributor	Indicates the company that sells or distributes the device. This number must be an entry in Common Table 8, Manufacturer Identification Codes. This code is set to the primary manufacturer of the device whenever the device is not private labeled (i.e., labeled and marketed by someone other then the manufacturer).
Tag	An 8-character label assigned by the end user based on the location and use of the field device. The Tag supports only the Packed ASCII character set (see Section 5.1.1).
Long Tag	A 32-character label assigned by the end user based on the location and use of the field device. The Long Tag supports ISO Latin-1 characters.

Groups of these data items are used for several purposes. Table 14 shows different identification activities and the data items used in each.

Table 14. Application of Identifying Data

Data Items	Purpose
Tag; Long Tag; Private Label Distributor; Device Type	User Identification of the Field Device
Manufacturer ID; Device Type; Device ID (see the *Data Link Layer Specification*)	Link Layer Addressing of the Field Device. These three data, when combined, identify a unique field device. In other words, no two devices ever manufactured may have the same combination of these three data.
Manufacturer ID; Device Type; Device Revision	Field Device Command and Data Item Set Identification
Device Revision; Software Revision; Hardware Revision; Universal Command Revision	Field Device Revision Information

9.1 User Identification of the Field Device

When providing the end user with field device identification information the host must display:

- The Private Label Distributor and Device Type; and
- The Tag, the Long Tag, or both.

The Private Label Distributor and Device Type allows the user to know which product is being communicated with. Since the Protocol allows for private labeling of devices, the Manufacturer ID should not be displayed. This allows one manufacturer to sell devices manufactured by another manufacturer without the name of the primary manufacturer appearing.

The Tag or Long Tag allows the end user to know where the field device is installed in the process.

9.2 Field Device Revisions

Identity Commands provide the following revision information about a field device:

- Device Revision
- Software Revision
- Hardware Revision
- Universal Command Revision

Each of these revision numbers returns an unsigned integer (i.e., a whole number) and provides a different function. The Universal Command revision indicates the major revision of the Universal Command Specification and, as a result, the major revision of the Protocol as well. This allows the host to determine the minimum command set supported by a field device and infers operation of the Data Link Layer, Response Codes, and other Protocol related features.

The Hardware and Software revisions track the configuration of the field device. The Hardware revision indicates the major revision level of the electronics of the field device. Minor, non-functional hardware changes are not indicated. However, an increment of the software revision is necessary for all changes to the field device's firmware. These two revision numbers allow identification of the field device for technical support operations.

The Device revision indicates the revision level of the device's Application Layer and indicates the commands and data items available via the Protocol. The major revision level of the manufacturer's device-specific document must match this number. An increment of this revision number is necessary whenever a new command or data item is added to the field device. In addition, the rules governing the use of Device Type and Device Revision in Section 9.3 and Section 6 must be adhered to.

9.3 Identifying the Field Device's Command Set

To identify the commands and data items supported by a field device, a host recognizes three data items returned by the Identity Commands:

- Manufacturer ID
- Device Type
- Device Revision

In effect, the Manufacturer ID and the Device Type indicate a specific product, and the Device Revision indicates the version of the command set for that product. Together these data items define the field device in relationship to the HART Protocol. Since a HART host is only concerned with this relationship, several devices that are packaged differently may use the same firmware.

A complete list of commands, Device-Specific Command Definitions, and Common-Practice Command Definitions must be documented in the manufacturers Device-Specific documentation.

10. NETWORK MANAGEMENT

The Protocol does not explicitly support a Network Layer. However, the Protocol does have features typically found in the ISO Network Layer. This section describes the mechanisms and procedures for masters to locate and address slave devices and to accomplish proper message routing. In particular this section includes:

- A description of the Identity Commands that allow communications with field device to be established.
- The definition of Sub-Devices that allow communications to HART-compatible devices via intermediate bridging or I/O devices. Commands are provided to allow the identification of connected devices and the routing of messages to them.
- Procedures that are specified that allow masters to identify field devices and initiate communications with them.
- Specifications for device support of multi-drop networks and the routing of messages across multi-drop networks.
- Requirements for support of Burst Mode Operation and references to the commands used by a master to manage Burst Mode Operation.

10.1 Identity Commands

Identity Commands are used by a master to identify a field device and begin a communication session with the field device. There are five Identity Commands:

- Command 0 Read Unique Identifier
- Command 11 Read Unique Identifier Associated With Tag
- Command 21 Read Unique Identifier Associated With Long Tag
- Command 73 Find Device
- Command 75 Poll Sub-Device

All of these commands return the same Response Data Bytes. These commands return the data items necessary to:

- Generate the long frame address,
- Route commands to the appropriate field device, and
- Determine the command set and data items supported by the field device.

Command 0 is unique in that it is the only command that still supports short frame messages. Command 0 can be use the polling address to automatically identify devices on the loop. Commands 11, 21, and 73 all use long frame messages containing the Broadcast Address. The Broadcast Address has all zeros for the field device address and uses Request Data Bytes (Command 11, 21) or a mechanical means (Command 73) to identify the field device. Command 75 allows sub-devices connected to an I/O system to be identified (see Section 10.2).

Once basic addressing and command set identification is complete, the host should use Commands 13 (Tag), 15 (Private Label Distributor), and 20 (Long Tag) to complete the identification of the field device.

10.2 Sub-Devices and I/O Systems

The Protocol allows communication to multiple devices via an intermediate Bridging Device or I/O System. The intermediate devices are identified by setting Protocol_Bridge_Device (bit 2) in the Flags byte of Identity Commands. The Protocol allows the devices connected to an I/O System to be identified by a master using:

- Command 74 Read I/O System Capabilities
- Command 75 Poll Sub-Device

Figure 22 shows the I/O System architecture supported by the Protocol. I/O systems appear to HART network like any other field device. However, an I/O system must have one or more I/O Cards. Each I/O card in turn must contain one or more sub-network channels and each of these channels must support a minimum of 1 or more sub-devices. Command 74 allows the master to read the maximum values for each of these data items (I/O Cards, Channels, Sub-Devices). This information, returned in Command 74, allows the master to poll all combinations of I/O Cards, Channels and sub-device polling addresses to identify connected sub-devices.

Note: This is an Application Layer view of the I/O System and its sub-devices. Many physical and electrical configurations are possible and the I/O system components may not be readily apparent to the naked eye.

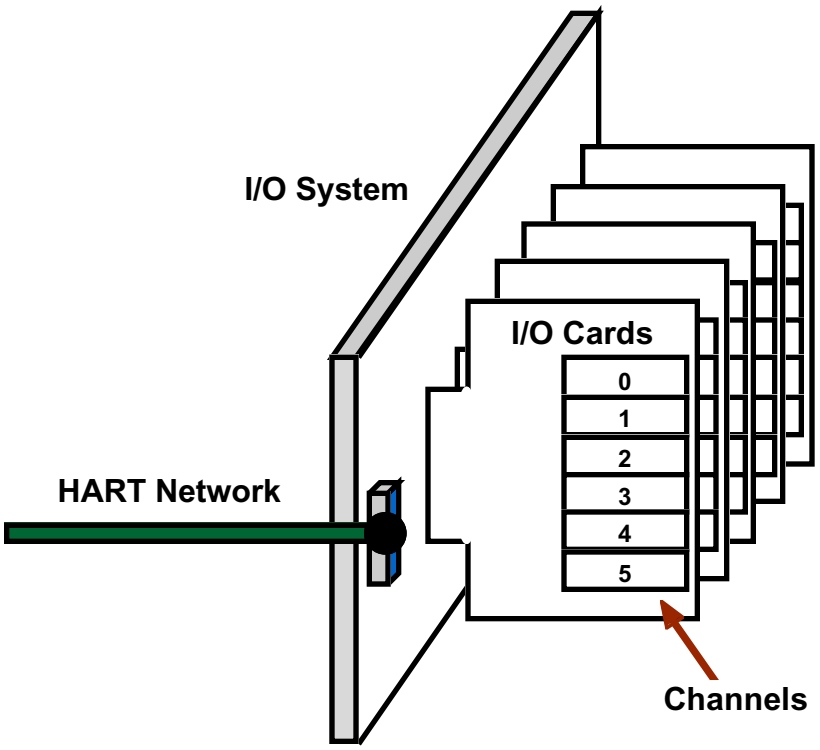

Figure 22. Bridge or I/O System Components

Once identified, the sub-device's Unique Identifier is used for further communication to the sub-device. The I/O System must automatically route all commands addressed to the connected sub-device. In other words, after the sub-device is identified all further commands and responses appear, at the network level, to be communicating directly to a HART compatible field device.

Note: If the I/O system utilizes a HART Physical Layer at the channel level all HART Protocol Requirements must be met (e.g., multi-drop and burst mode)

10.3 Establishing Communication with a Field Device

To establish connection with the field device, the master must determine the long frame address of the field device. This long frame address is used by all HART commands (see the *Data Link Layer Specification*). The long frame address can be determined using one of the following procedures:

- Poll using Command 0;
- Poll for Sub-Devices Using Command 75;
- Poll by Tag using Command 11;
- Poll by Long Tag using Command 21;
- Mechanically Identify the field device using Command 73; and
- Manual Entry of the Manufacturer ID, Device Type, and Device ID.

For each of these techniques, if an asynchronous Physical Layer (e.g., FSK or RS-485) is used then until the connection is established with the slave, the master should use a large number of preamble characters (e.g., 20). Once connected to the field device, the Identity Command indicates the minimum number of preambles the master must use.

Masters must support all procedures in this section.

10.3.1 Using Polling Addresses

A master can poll the loop using all legal short frame addresses from Command 0 to identify the devices connected to the loop. Command 0 is unique because master request messages are the same for all versions of the Protocol. Using the identity information the master can determine the Protocol version supported by the field device and assemble the long frame address for the slave device. The basic procedure is:

12. The Master sends Command 0 using short frame format.
13. If a slave with the requested short frame (polling) address is connected to the loop, the slave must answer. Since there are several polling addresses possible, polling all addresses may take a few minutes (perhaps longer if retries are attempted). As a result the master should allow the user to limit the range of Polling Addresses when using this field device identification algorithm.
14. Steps 1 and 2 are repeated until all the devices on the loop are identified.

This procedure requires all field devices on a loop to be set to a unique Polling Address prior to commissioning the loop.

10.3.2 Polling for Sub-Devices

When a Bridge Device or I/O System is detected on the network, a master may use Command 75 to identify the connected sub-devices. Using the identity information, the master can determine the

Protocol version supported by the field device and assemble the long frame address for the slave device. The basic procedure is:

1. The Master identifies a Bridge Device or I/O System (i.e., bit 2, Protocol_Bridge_Device, in the Flags byte of Identity Commands is set).
2. The Master issues Command 74 to determine how many sub-devices may be connected.
3. The Master issues Command 75 to a polling address on an I/O Card Channel.
4. If a sub-device with the requested polling address is connected, it must answer.
5. Steps 3 and 4 are repeated until all the sub-devices are identified.

This procedure requires all sub-devices connected to an I/O Card Channel be set to a unique Polling Address prior to commissioning. Since the number of possible sub-devices is potentially very large, the I/O System must provide information to minimize the polling time required by the host. This is achieved primarily by setting the data item in Command 74 appropriately. In addition, I/O systems are encouraged to implement Command 75 Response Code 9, "No Sub-Device Found" to expedite sub-device identification and minimize master polling time.

10.3.3 Polling by Tag or Long Tag

All HART field devices support both an 8 character Tag, and a 32 Character Long Tag. Tags have been used in process plants to identify field devices long before smart field devices were ever possible. HART allows a field device to be identified using the Tag (via Command 11) or Long Tag (via Command 21). In both cases the request message is transmitted using the Broadcast Address with the Data field containing the string indicating which field device is to answer. The basic procedure is:

1. Allow the user to enter the Tag or Long Tag to be located.
2. The Master sends the appropriate command using long frame format containing the Broadcast Address.
3. If a slave with the Tag or Long Tag is connected to the loop, it must answer.
4. Steps 1 through 3 are repeated until all the devices on the loop are identified.

This procedure requires all field devices on a loop to be set to the correct Tag and Long Tag prior to commissioning the loop.

Note: Broadcast addresses are addressed to all field devices (see *Data Link Layer Specification*). As a result, Broadcast addresses must be routed through I/O Systems and Bridge Devices to all sub-devices.

10.3.4 Mechanical Identification of the Field Device

If the field device supports Command 73, the field device can be mechanically identified. Field devices implementing this command only respond when physically/mechanically signaled to do so. For example, the technician presses a special button or combination of buttons that indicate the slave is to answer this command. The basic procedure is:

1. The user arms the field device (e.g., by pressing a button).
2. The master then sends the Command 73 to identify the field device
3. The field device answers the command once and disarms itself.
4. Steps 1 through 3 are repeated until all the devices on the loop are identified.

In addition to establishing the connection to the field device, this procedure allows the user to verify the device is installed in the correct plant location and on the correct wire pair.

10.3.5 Manual Entry of the Manufacturer ID, Device Type, and Device ID

Field device identification using the above procedures requires little or no information to be supplied by the user. In addition to the above procedures, the communications connection can also be established by the manual entry of the data necessary to construct the long frame address. With the long frame address known, Command 0 cant be sent (with the long frame address) to complete the identification of the field device.

Since this requires manual entry of data not generally available to the user, masters should not rely on this procedure as its primary means of establishing a connection to the field device.

10.4 Multi-drop Networks

A Multi-drop network connects multiple field devices and up to two masters across a common pair of wires or other medium. All devices shall support multi-drop operation. Masters must establish communications using one of the procedures in Section 10.2 and use long frame commands to communicate to the desired field device. See Command 6 and the *Data Link Layer Specification*.

Independent of the Protocol, the interconnecting medium may place additional constraints on the number of field devices allowed on one network. For example, the *FSK Physical Layer Specification* assumes 17 (two Masters and 15 Slaves) devices when defining the maximum noise allowed to be generated by a device.

10.5 Burst Mode Operation

Burst Mode allows a field device to continuously transmit cyclical process data to HART compatible hosts (see Commands 105, 107—109). Only one slave in a HART network may be in burst mode. The burst mode commands are Common Practice and, as a result, implementation of Burst Mode in a Slave device is optional. A Field Device must implement all 4 commands as set when it supports Burst-Mode.

Masters must support Burst Mode for successful bus arbitration (see the *Data Link Layer Specification*). Masters are strongly recommended to use Burst Mode for cyclical data acquisition and control. The following commands control Burst Mode operation:

- **Command 109** is used to place the field device into and out of Burst Mode.
- **Command 108** selects the response message that the field device is to send. Field devices must support at least Commands 1, 2, 3 and 9 if Burst Mode is implemented.
- **Command 107** must be used to select the Device Variables returned when Command 9 is burst.
- **Command 105** allows the Burst Mode configuration of the Field Device to be read.

The device must retain Burst Mode Settings through a Device Reset, Self Test or the power being removed and reapplied.

10.6 Delayed Slave Responses

The Delayed Response Mechanism (DRM) enables the slave to indicate to a master that it received the request but is not able to formulate a reply in the time allowed by the Data Link Layer. This mechanism provides an informative and flexible convention for slave devices needing additional time to perform relatively infrequent operations like self diagnostics, calibration or configuration. Unlike the Response Code BUSY, the master knows that the slave has understood the command and is still communicating.

DRM implementation in a slave is optional and limited to WRITE commands only. In other words, if the slave can ensure that a response can be generated within the link layer timing, then DRM does not need to be implemented. All slave devices supporting the DRM must adhere to the following rules:

1. Only new WRITE commands may use delayed response. All Commands must explicitly state in their specification whether the DRM Response Codes may be used.
2. To maintain compatibility with HART 5 and earlier hosts, any device specific commands supported in a HART 5 Field Device implementation must not use the DRM. In other words, all commands existing prior to HART 6 must continue to begin the slave response within the slave time out (see the *Data Link Layer Specification*).
3. If a command is received that would normally initiate a DR and no buffers are available then the slave will respond with a Busy.
4. A field device must always respond to an Identity Command even if a DR is in progress. Furthermore, field devices should always successfully execute READ commands even if a DR is in progress.
5. If a DR is being processed that will prevent a slave from responding to a master request then the slave must answer with a DR_CONFLICT. For example, DR_CONFLICT is returned if a master tries to initiate a transducer trim while one is already in process.
6. Once DR processing is complete, the slave must answer READ commands even if the master has not fetched the result of the DR.
7. Bridging devices (e.g., multiplexers and I/O systems) may use the DRM on all commands since they generally communicate with the host at higher bit rates then supported by the FSK Physical Layer. Bridging devices should support one DR buffer for each HART communication channel supported.

Slave devices must use the Response Codes listed in Table 15 to synchronize DR operation with master devices. Any Command Specification not listing these Response Codes may not use the DRM with the single exception listed in Rule 7 above.

Table 15. DRM Related Response Codes

Mnemonic	Value	Description
Busy	32	Indicates the device is busy performing an indivisible operation. For Commands supporting DRM, Busy indicates that no more DR buffers are available.
DR_INITIATE	33	Indicates the start of a DR. The slave device needs additional time to process the command.
DR_RUNNING	34	Indicates that the slave device is still processing the command.
DR_DEAD	35	Used by HART multiplexers to indicate that no response was received from the slave device.
DR_CONFLICT	36	Indicates that the command cannot be processed because it is in conflict with DR currently being processed.

All HART compatible masters must support DRM. When a master encounters a DR_INITIATE the exact command (including the data) must be reissued to retrieve the results. The master need not reissue the command immediately, in the interim he can service other slaves.

10.6.1 Normal DR Operation

A simple DR transaction is shown in Figure 23. The master sends a command that the slave cannot immediately complete. The slave initiates the DR and begins processing the command for later retrieval by the master. Some time later the master issues the same exact command. The slave is not finished and issues a DR_RUNNING. Later the command completes and the master retrieves the result.

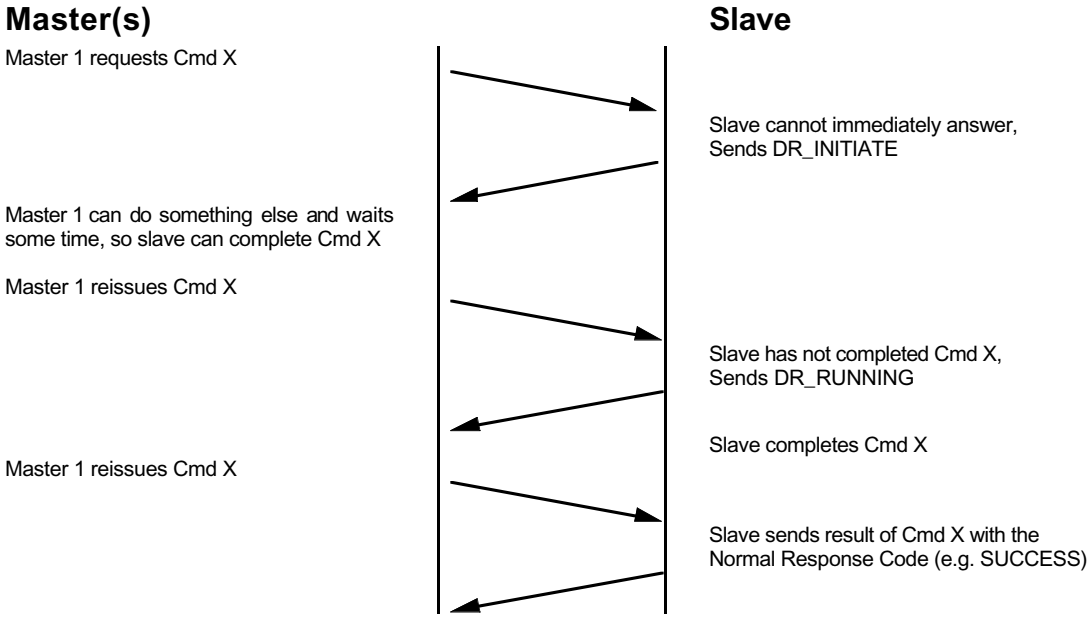

Figure 23. Normal DRM Operation

10.6.2 Use of DR_CONFLICT Response Code

If a DR_CONFLICT is received, the master delays its retry to allow slave to complete the pending DR. Masters should follow this procedure to minimize repetitive polling that, in turn, would delay the field device's completion of the DR.

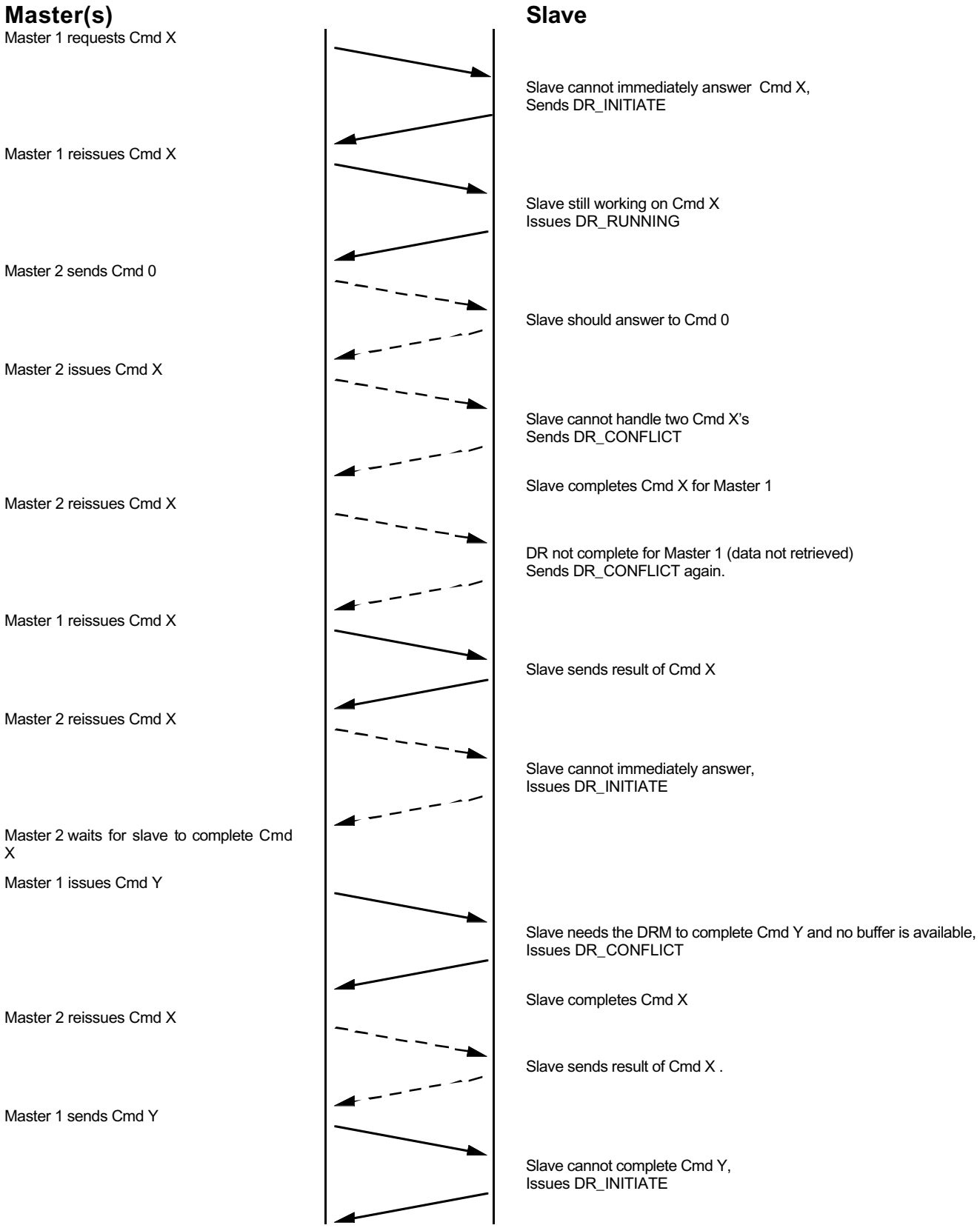

Figure 24. Command Responses During DR Processing

10.6.3 Multiple DR Buffers

More sophisticated slave devices will have two DR buffers, one for each master. Some devices (e.g., multiplexers and I/O systems) may have many more. Multiple DR buffers allow a DR from each master to be queued. However, the slave must be careful to protect against inconsistencies. If implemented properly, out-of-order-completion of pending delayed responses is possible.

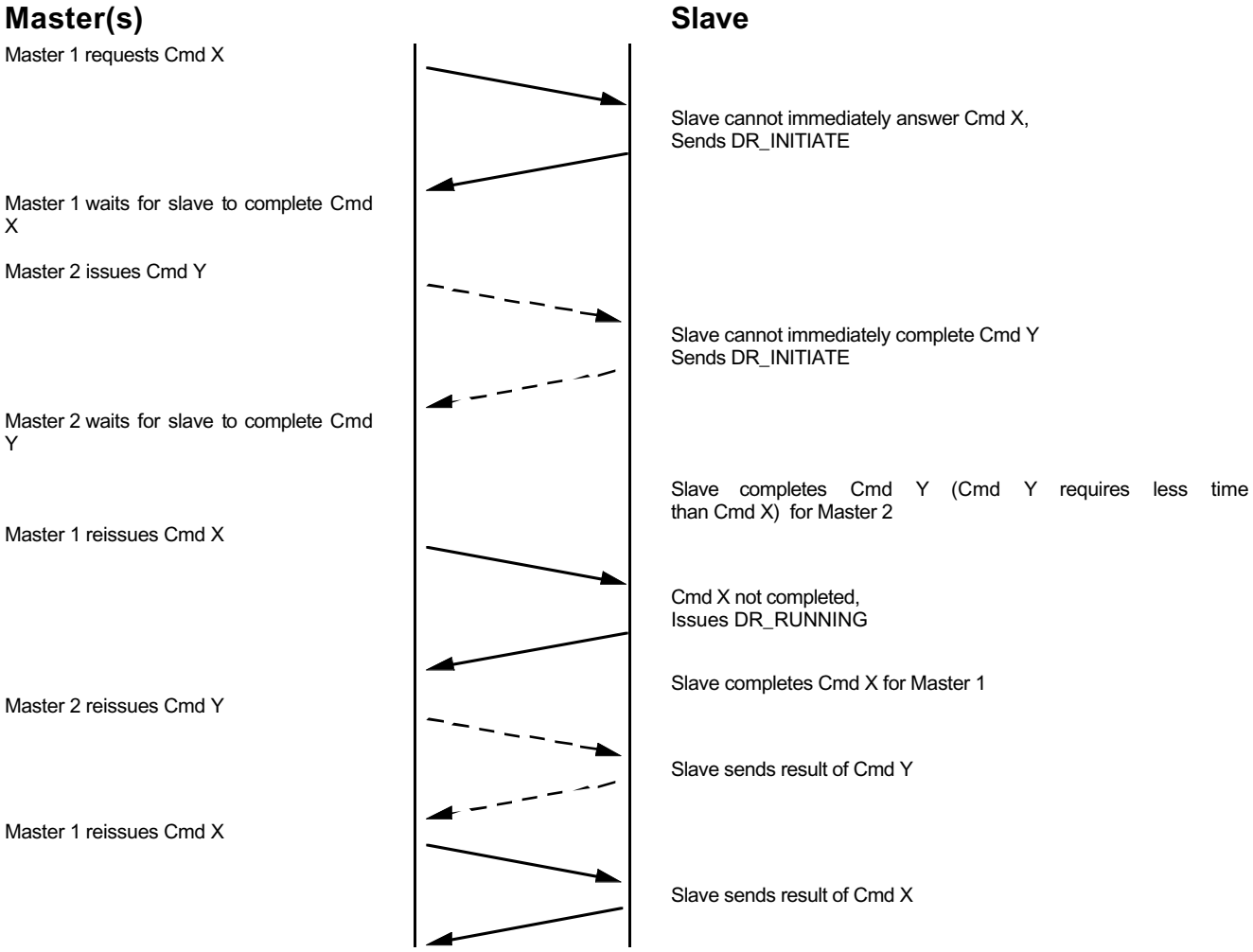

Figure 25. Slave with Multiple DR Buffers

10.6.4 Bridge Device Use of DRM

Bridge devices (e.g., multiplexers and I/O systems) function as slaves for the master that issues a request and as master for slaves that are addressed. Such devices are identified by setting the Protocol_Bridge_Device bit in the Flags byte of Identity Commands.

Bridge devices may introduce delays in relaying a command to and from the HART Field Device or Sub-Device. As a result, bridge devices are allowed to use the Delayed Response Mechanism on every command.

An additional Response Code (DR_DEAD) is used if the slave connected to the bridge device fails to reply at all. A master receiving DR_DEAD must assume that a serious error has occurred. Since an intelligent bridge should automatically retry the command if the slave does not answer, the master should immediately notify the user that the slave device is not responding.

11. HOST CONFORMANCE CLASSIFICATIONS

Host conformance classes identify host capabilities based on the level of host functionality. This functionality encompasses the level of data access and manipulation provided by the host and includes references to Common Practice and Device Family Commands. A host may choose to implement any function regardless of its class. However, conformance to a specific class may be claimed only when a host has implemented all of the functions in that class, particularly with regard to Common Practice and Device Family Command support for connected field devices.

Each conformance class includes the functions of all the lower classes. Class 1 hosts afford minimal host capabilities while Class 5 hosts provide the greatest functionality. Class 1 is the minimum classification that can be claimed by any host application. Host Applications must clearly and visibly indicate their capabilities using the classifications listed in Table 16.

Table 16. Host Conformance Classes

Class	Description
0	The host does not meet the minimum requirements of Conformance Class 1.
1	The host can utilize cyclical process data from any field devices.
2	The host can supply the user with basic identification and configuration data about any field device.
3	The host can perform basic configuration of any field device. Minimum level required to be classified as a "Generic Host".
4	The host provides basic commissioning and calibration support for any device.
5	The host is capable of accessing all field device data items and all device-specific commands for any field device.

Note: See Section 10.3 for Master network management requirements. Depending on the system architecture the network management requirements may be applicable to the Host Application.

11.1 Host Conformance Class 0

Class 0 identifies only those hosts that do not meet the minimum requirements of Conformance Class 1. In other words, any host that cannot access and utilize the cyclical process data supplied by all HART field devices fall into this conformance class.

11.2 Host Conformance Class 1

Class 1 host can perform the minimum set of recommended capabilities. A Class 1 host shall be able to access and utilize cyclical process data from any field device. Cyclical data is provided via HART Commands 1, 2, 3, and 9 (see Table 17). This data may be polled by the master or acquired from field devices in burst mode.

Note: The number of Device Variables is indicated in byte 13 of all Identity Commands and indicates the Device Variables that may be accessed using Command 9.

Compliance with Host Conformance Class 1 requires supporting commands in Table 17.

Table 17. Host Conformance Class 1 Commands

Command 1	Read Primary Variable
Command 2	Read Loop Current and Percent of Range
Command 3	Read All Dynamic Variables and Loop Current
Command 9	Read Device Variables with Status

11.3 Host Conformance Class 2

Class 2 hosts can supply the user with basic identification and configuration data about any field device. A Class 2 host can, for example:

- Indicate the Device Variable Classifications for all cyclical process data (i. e., for all Dynamic Variables and Device Variables);
- Display the Tag, Long Tag, Device Type, and Private Label Distributor;
- Show the Message, Descriptor and Date;
- Display the Loop Current configuration (i.e., Polling Address, Range Values and Loop Current Mode); and
- Decode the Command 48 Response Data Bytes sub-fields.

Compliance with Host Conformance Class 2 requires supporting all Class 1 requirements plus the commands in Table 18.

Table 18. Host Conformance Class 2 Commands

Command 7	Read Loop Configuration
Command 8	Read Dynamic Variable Families
Command 11	Read Unique Identifier Associated with Tag
Command 12	Read Message
Command 13	Read Tag, Descriptor, Date
Command 14	Read Primary Variable Transducer Information
Command 15	Read Device Information
Command 16	Read Final Assembly Number
Command 20	Read Long Tag
Command 48	Read Additional Transmitter Status
Command 50	Read Dynamic Variable Assignments
Command 54	Read Device Variable Information

Note: Command 48 support in a Class 2 may be limited to the display of the hexadecimal values in the Response Data Byte sub-field for device-specific status. The Extended Device Status, Operating Mode, Analog Channel Saturated, and Analog Channel Fixed data fields should be decoded.

11.4 Host Conformance Class 3 - Generic Host

This Conformance Class defines the minimum capabilities required for a host to be classified as a "Generic Host". A Generic Host can perform basic configuration of any field device even though the host does not have special drivers for the field device. Basic configuration includes:

- HART communications (e.g., the Tag, Long Tag, Response Preambles, Flush Delayed Response Buffers and Polling Address)
- Loop Current operation (e.g., Range Values, Damping Constants)
- Simple diagnostic procedures (Loop Current testing, Self Test and Device Rest)
- Units Code selection for cyclical process data

Compliance with Host Conformance Class 3 requires supporting all Class 2 requirements plus the commands in Table 19.

Table 19. Host Conformance Class 3 Commands

Command 6	Write Polling Address
Command 17	Write Message
Command 18	Write Tag, Descriptor, Date
Command 19	Write Final Assembly Number
Command 22	Write Long Tag
Command 34	Write Primary Variable Damping Value
Command 35	Write Primary Variable Range Values
Command 36	Set Primary Variable Upper Range Value
Command 37	Set Primary Variable Lower Range Value
Command 38	Reset Configuration Changed Flag
Command 40	Enter/Exit Fixed Primary Variable Current Mode
Command 41	Perform Transmitter Self Test
Command 42	Perform Device Reset
Command 44	Write Primary Variable Units
Command 51	Write Dynamic Variable Assignments
Command 53	Write Device Variable Units
Command 55	Write Device Variable Damping Value
Command 59	Write Number of Response Preambles
Command 72	Squawk
Command 79	Write Device Variable
Command 106	Flush Delayed Response Buffers
Command 107	Write Burst Transmitter Variables
Command 108	Write Burst Mode Command Number
Command 109	Burst Mode Control
Command 113	Catch Device Variable

11.5 Host Conformance Class 4

Class 4 hosts provide basic commissioning and calibration support for any field device. Calibration is performed using the Loop Current Commands and the Transducer Trim Commands (see Table 20). Basic field device commissioning shall be supported using the Device Family Commands (see the *Device Families Command Specification*)

Compliance with Host Conformance Class 4 requires supporting all Class 3 requirements, the commands in Table 20 and the Device Family Commands. Since Device Families are added periodically to the Protocol, Class 4 hosts shall list the Device Families or the Protocol major and minor revision level supported.

Table 20. Host Conformance Class 4 Commands

Command 45	Trim Loop Current Zero
Command 46	Trim Loop Current Gain
Command 80	Read Device Variable Trim Points
Command 81	Read Device Variable Trim Guidelines
Command 82	Write Device Variable Trim Point
Command 83	Reset Device Variable Trim

11.6 Host Conformance Class 5 - Requirements for a "Universal Host"

This Host Conformance Class indicates the maximum HART support by a host. Only hosts meeting Class 5 requirements may be classified as a Universal Host.

Compliance with Host Conformance Class 5 requires supporting all Class 4 requirements. In addition, the host must be capable of accessing all field device data items and all device-specific commands for any field device. Access to device-specific commands and data items may be via installable drivers (e.g., Device Descriptions). However, the device-specific drivers must be able to be developed and installed by the field device manufacturer. A Universal Host must be "open" and allow support to be incorporated for any field device from any manufacturer.

ANNEX A. REVISION HISTORY

A1 Changes from Rev 7.1 to Rev 8.0

1. These new sections were added as part of the format revisions for all HART Protocol Specification documents: Scope, Reference, Definitions, Symbols/Abbreviations.
2. Replaced "transmitter" with "device" to demonstrate applicability of commands to many device types.
3. Section 5 Data Types was expanded to include ISO Latin-1 Strings. Specification of all data types clarified. As a result, many tables and figures were added. In addition, requirements for lookup tables were incorporated from Section 6.3 of the previous version.
4. Section 6 Field Device Revision Rules updated for HART 6. In addition, these requirement were elevated to a separate section (i.e. was Section 6.1 in previous version)
5. The Command Number Partitions, Command Requirements, and Command Status Bytes are now consolidated in the Section 7 Application Layer Interface. These were formally found in Sections 2.1, 3 and 6.2 in the previous revision. Section 7.2 Data Field is new and explains, for example, the use of extended command numbers.
6. Section 8, Dynamic and Device Variables, was clarified and figures added (this was Section 5 in previous version).
7. Section 9, Field Device Identification, is clarified and expanded from Sections 6.1 and 6.6—6.9of the previous version.
8. Network Management requirements (Section 10) was added to include information formerly spread across many documents. Some parts that existed previously were updated to reflect changes in HART Rev. 6. Identification of field devices, Delayed Response Mechanism, Sub-Devices are some of the requirements added or clarified.
9. Host Conformance Class moved to Section 11 and requirements clarified.

A2 Changes from Rev 7.0 to Rev 7.1

The document was translated from a MultiMate document to Microsoft Word. As a result of this translation the document format was altered. No other modifications were made to the document.

A3 Changes from Rev 6.0 - Final to Rev 7.0 - Final

1. This revision adds commands for devices with Multiple Analog Outputs and Analog Outputs other than Current.
2. This revision adds More Status Available to the Field Device Status Byte of the Response Codes.
3. Summarized Release Notes from Rev 5 to Rev 6.0 - Final

Page	Line	Change	Text
TP	4	Replace	"6.0" by "7.0"
TP	5	Replace	"14 February 1990" by "11 October 1990"
TP	6	Replace	"14 February 1990" by "11 October 1990"
TP	7	Replace	"14 February 1990" by "11 October 1990"
TP	8	Replace	"D8900069;" by "D9000035;"
2	19	Insert	"P. V."
2	31	Insert	"Primary Variable"
2	44	Insert	"60 * Read Analog Output and Percent of Range..."
2	49	Replace	"110" by "110*"
3	8	Insert	"Primary Variable"
3	9	Insert	"Primary Variable"
3	9	Delete	"(Push SPAN Button)"
3	10	Insert	"Primary Variable"
3	10	Delete	"(Push ZERO Button)"
3	13	Insert	"Primary Variable"
3	16	Insert	"Primary Variable"
3	18	Insert	"64 * Write Analog Output Additional Damping..."
3	36	Insert	"Primary Variable Current"
3	37	Insert	"Primary Variable Current"
3	45	Insert	"67 Trim Analog Output Zero 68 Trim Analog..."
7	5	Replace	"Command Specific" by "Command-Specific"
7	10	Delete	"they"
7	12	Replace	"#32 and #64" by "#32, Busy, or #64, Command.."
7	24	Insert	"Command-Specific"
7	34	Replace	"Reponse" by "Response"
9	24	Replace	"Reserved, set to zero." by "More Status..."
9	29	Replace	"Bit #3 Output..." by "Bit #3 Primary Variable. -."
9	34	Insert	"Primary Variable"
9	35	Replace	"current" by "analog outputs for the Primary..."
9	36	Replace	"cannot respond to..." by "no Longer represent..."
12	13	Replace	"the analog output." by "Analog Output #1."
12	14	Replace	"the analog output..." by "Analog Output #1..."
12	16	Replace	"the analog output." by "Analog Output #1."
12	21	Insert	"When more than one Analog Output exists in a "

Page	Line	Change	Text
12	31	Replace	"Variables," by "Variables and their..."
12	33	Delete	"to select the variable for the analog output,"
14	27	Replace	"type code," by "Type Code,"
15	22	Insert	"EXAMPLE PRIMARY VARIABLE CURRENT"
15	24	Insert	"Primary Variable"
15	24	Replace	"4rnA" by "4 ma"
15	25	Insert	"Primary Variable Current"
15	26	Insert	"Primary Variable"
15	26	Replace	"2OmA" by "20 mA"
15	27	Insert	"Primary Variable Current"
15	28	Insert	"Primary Variable"
16	32	Replace	"Rosemount Devices" by "Rosemount's devices"
18	6	Delete	"the"
18	6	Insert	"#1"
18	12	Replace	"the Analog Output...." by "Analog Output #1...."
18	16	Delete	"the"
18	16	Insert	"#1"
18	17	Delete	"the"
18	17	Insert	"#1"
18	17	Replace	"the current" by "this Analog Output"
18	18	Replace	"4 milliamperes;" by "its minimum;"
18	18	Replace	"#4," by "#3, Primary Variable Analog"
18	19	Delete	"Current"
18	20	Delete	"the"
18	21	Insert	"#1"
18	22	Insert	"The operation of Analog Outputs other than #1..."
18	36	Delete	"the"
18	36	Insert	"#1"
18	38	Insert	"Primary Variable"
18	39	Insert	"Primary Variable Current"
18	39	Insert	"Primary Variable Current"
18	40	Replace	"#46" by "#46;"
18	42	Insert	"In addition, Command #66, Enter/Exit Fixed..."
18	47	Replace	"information ." by "information."
20	26	Delete	"the"
20	26	Insert	"#1"
20	27	Replace	"4 milliamperes." by "its minimum."
21	14	Delete	"the"
21	14	Insert	"#1"

A4 Major Modifications from Rev 5 to Rev 6.0 - Final

1. This revision incorporates Unique Identifier, Burst Mode Operation, Polling Address, and other changes pertaining to the Extended Frame Format.
2. A decimal point and integer has been added to the HART document revision numbering system.
3. Rearranged document and renamed several sections.
4. Changed most occurrences of "transmitter" to "field device".
5. Deleted all references to the Transmitter Type Code being returned in the Transmitter-Specific Commands.
6. Deleted Command #4, Read Common Static Data.
7. Deleted Command #5, Write Common Static Data.
8. Removed Expansion from each of the Transmitter Command Class Partitions.
9. Changed the titles of several commands.
10. Added Command #11, Read Unique Identifier Associated with Tag.
11. Added Command #12, Read Message.
12. Added Command #13, Read Tag, Descriptor, Date.
13. Added Command #14, Read Primary Sensor Information.
14. Added Command #15, Read Output Information.
15. Added Command #16, Read Final Assembly Number.
16. Added Command #17, Write Message.
17. Added Command #18, Write Tag, Descriptor, Date.
18. Added Command #19, Write Final Assembly Number.
19. Added Command #57, Read Unit Tag, Descriptor, Date.
20. Added Command #58, Write Unit Tag, Descriptor, Date.
21. Added Command #59, Write Number of Response Preambles.
22. Added Command #108, Write Burst Mode Command Number.
23. Added Command #109, Burst Mode Control.
24. Added Command #110, Read All Dynamic Variables.
25. Added Command #111, Transfer Service Control.
26. Added Command #112, Transfer Service.
27. Changed Communications Error Summary Bit #2 from Message Time-out Reserved.

28. Changed the Command-Specific Response Code bit assignments from #0 - #3 to #0 - #6 and expanded the number of codes to 127.
29. Moved Command Response Summary, Bit #4, Transmitter Fault to Command-Specific Response Code #16 and renamed it Access Restricted.
30. Changed Command Response Summary Bit #5 to Response Code #32.
31. Changed Command Response Summary Bit #6 to Response Code #64.
32. Changed Field Device Status Bit #4 from In Burst Mode to Reserved.
33. Added section on IEEE 754 Floating Point Format.
34. Added section on ASCII Data Format.
35. Added section on Compatibility.
36. Removed section on Model 268 Compatibility.
37. Added section on DAC Trim Sequence.
38. Added section on Unique Identifier.
39. Added section on Polling Address.
40. Added section on Burst Mode.
    (Refer to document Revision 6, D8900069, for detailed information.)

A5 Major Modifications from Rev 4 to Rev 5

1. This revision incorporates Write Protect Mode and adds Transmitter Variable Commands.
2. Added sections on variable descriptions, Write Protect Mode, Private Label Distributor, and Transmitter Type Code Expansion.
3. Added Command #50, Read Dynamic Variable Assignments.
4. Added Command #51, Write Dynamic Variable Assignments.
5. Added Command #52, Set Transmitter Variable Zero.
6. Added Command #53, Write Transmitter Variable Units.
7. Added Command #54, Read Transmitter Variable Damping Value.
8. Added Command #55, Write Transmitter Variable Damping Value.
9. Added Command #56, Write Transmitter Variable Sensor Serial Number.

A6 Major Modifications from Initial Rev 3 to Rev 4

Added Command #49, Write Sensor Serial Number.
(Refer to document Revision 3, D8700038, and Revision 4, D8900067, for detailed information.)