Hart at a Glance

Frame Coding

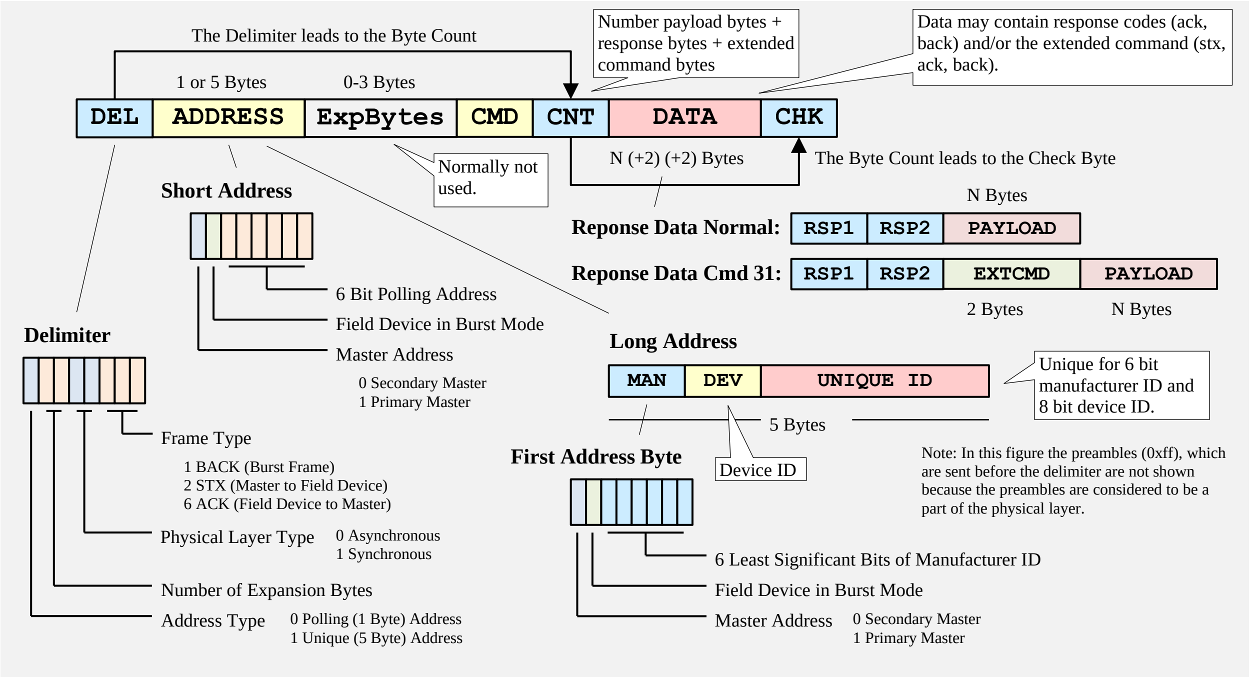
The Delimiter leads to the Byte Count
1 or 5 Bytes
0-3 Bytes
Number payload bytes + response bytes + extended command bytes
Data may contain response codes (ack, back) and/or the extended command (stx, ack, back).
DEL
ADDRESS
ExpBytes
CMD
CNT
DATA
CHK
Normally not used.
N (+2) (+2) Bytes
The Byte Count leads to the Check Byte
Short Address
6 Bit Polling Address
Field Device in Burst Mode
Master Address
0 Secondary Master
1 Primary Master
N Bytes
Reponse Data Normal:
RSP1
RSP2
PAYLOAD
Reponse Data Cmd 31:
RSP1
RSP2
EXTCMD
PAYLOAD
2 Bytes
N Bytes
Delimiter
Long Address
MAN
DEV
UNIQUE ID
Unique for 6 bit manufacturer ID and 8 bit device ID.
5 Bytes
Frame Type
1 BACK (Burst Frame)
2 STX (Master to Field Device)
6 ACK (Field Device to Master)
First Address Byte
Device ID
Note: In this figure the preambles (0xff), which are sent before the delimiter are not shown because the preambles are considered to be a part of the physical layer.
Physical Layer Type
0 Asynchronous
1 Synchronous
6 Least Significant Bits of Manufacturer ID
Number of Expansion Bytes
Field Device in Burst Mode
Address Type
0 Polling (1 Byte) Address
1 Unique (5 Byte) Address
Master Address
0 Secondary Master
1 Primary Master

Figure 1: The Basic Coding of a Hart Frame

The figure above is giving an overview of the coding of a Hart frame. Usually Hart services are composed of a request (stx) by the master followed the response (ack) of a slave. Bursts (back) are frames looking like a response (including response codes) but sent by the slave without any request. The slave is sending these frames in burst mode within defined time slots following the rules of the protocol specification. In fact Hart is a token passing protocol which allows also the slave to be a token holder and send burst frames.

The following chapter is showing a list of Hart commands which are used very often. The list is showing the major differences between Hart 5.3, Hart 6 and Hart 7.4.

New items in Hart 6 are marked with yellow color while new items of Hart 7.4 are marked by blue color.

However, the following is not replacing any specification and is not showing the details which are needed for an implementation. The details has to be taken from the Hart specifications which are provided by the FieldComm Group: Hart Specifications.
That the listed commands are most commonly used is not the opinion of the HCF but the opinion of the author of this document.

Commonly Used Commands

No	Title	Request Data			Response Data		
Universal							
00	**Read Unique Identifier**	None			0	int8	254
					1		Manufacturer ID
					2		Short device ID
					3		Number preambles request
					4		Hart revision
					5		Device revision
					6		Software revision
					7		Hw rev and signaling code
					8		Flags
					9	int24	DevUniqueID
					12	int8	Number preambles response
					13		Maximum number device variables
					14	int16	Configuration change counter
					16	int8	Extended device status
					17	int16	Extended manufacturer code
					19		Extended label distributor code
					21	int8	Device profile
01	**Read Primary Variable**	None			0	int8	PV Units
					1	float	Primary variable
02	**Read Current and Percent of Range**	None			0	float	Current
					1	float	Percent of range
03	**Read Current and Dyn. Variables**	None			0	float	Current
					4	int8	PV1 units code
					5	float	PV1 value
					9	int8	PV2 units code
					10	float	PV2 value
					14	int8	PV3 units code
					15	float	PV3 value
					19	int8	PV4 units code
					20	float	PV4 value
06	**Write Polling Address**	0	int8	Polling Address	0	int8	PV Units
		1	int8	Loop current mode	1	int8	Loop current mode
07	**Read Loop Configuration**	None			0	int8	Polling address
					1		Loop current mode
08	**Read Dyn. Vars Classification**	None			0	int8	PV1 classification
					1		PV2 classification
					2		PV3 classification
					3		PV4 classification

No	Title	Request Data			Response Data		
Universal							
09	**Read Device Variables with Status**	0	int8	Slot0: Device variable code	0	int8	Extended device status
		1		Slot1: Device variable code	1	Slot0: Device variable properties	
		2		Slot2: Device variable code	1	int8	Device variable code
		3		Slot3: Device variable code	2		Device variable classification
		4	int8	Slot4: Device variable code	3		Device variable units code
		5		Slot5: Device variable code	4	float	Device variable value
		6		Slot6: Device variable code	8	int8	Device variable status
		7		Slot7: Device variable code	9	struct	Slot1: Device variable properties
					17		Slot2: Device variable properties
					25		Slot3: Device variable properties
					33	struct	Slot4: Device variable properties
					41		Slot5: Device variable properties
					49		Slot6: Device variable properties
					57		Slot7: Device variable properties
					65	time	Time stamp slot0
11	**Read Unique ID by Short Tag**	0	pac6	Tag name (packed ascii) 6 bytes = 8 characters	Same as command 0 read unique identifier		
12	**Read Message**	None			0	pac24	Message (packed ascii) 24 bytes = 32 characters
13	**Read Tag, Descriptor, Date**	None			0	pac6	Short tag (packed ascii) 6 bytes = 8 characters
					6	pac12	Descriptor (packed ascii) 12 bytes = 16 characters
					18	int8	Day
					19		Month
					20		Year (offset to 1900)
14	**Read Primary Variable Transducer Information**	None			0	int24	Transducer serial number
					3	int8	Units code
					4	float	Upper transducer limit
					8		Lower transducer limit
					12		Minimum span
15	**Read Device Information**	None			0	int8	Alarm selection code
					1		Transfer function code
					2		Units code
					3	float	PV upper range value (for 20 mA)
					7		PV lower range value (for 4 mA)
					11		PV damping value
					15	int8	Write protect code
					16		Reserved, must be set to 250
					17		PV analog channel flags
16	**Read Ass. Num**	None			0	int24	Final assembly number
17	**Write Message**	Same as response command 12			Same as response command 12		
18	**Write Tag, Descriptor, Date**	Same as response command 13			Same as response command 13		
19	**Write Ass. Num**	Same as response command 16			Same as response command 16		
20	**Read Long Tag**	None			0	str32	Long tag: 32 ISO Latin-1 characters
21	**Read Unique ID by Long Tag**	0	str32	Long tag: 32 ISO Latin-1 characters	Same as command 0 read unique identifier		
22	**Write Long Tag**	Same as response command 20			Same as response command 20		

No	Title	Request Data			Response Data		
Universal / Common Practice							
38	**Reset Config Changed Flag**	None			None		
		0	int16	Configuration change counter	0	int16	Configuration change counter
48	**Read Additional Device Status**	None					
		0	int8[5]	Transmitter specific status	0	int8[5]	Transmitter specific status
					6	int8[2]	Operating mode
		6	int8	Extended device status	6	int8	Extended device status
		7		Device operating mode	7		Device operating mode
					8	int8[3]	Analog output status
		8	int8	Standard status 0	8	int8	Standard status 0
		9		Standard status 1	9		Standard status 1
		10		Analog channel saturated	10		Analog channel saturated
					11	int8[3]	Analog output fixed
		11	int8	Standard status 2	11	int8	Standard status 2
		12		Standard status 3	12		Standard status 3
		13		Analog channel fixed	13		Analog channel fixed
					14	int8[3]	Transmitter specific status
		14	int8[10]	Transmitter specific status	14	int8[10]	Transmitter specific status
Common Practice							
33	**Read Device Variables**	0	int8	Slot0: Device variable code	0	Slot0: Device variable properties	
		1		Slot1: Device variable code	0	int8	Device variable code
		2		Slot2: Device variable code	1		Device variable units code
		3		Slot3: Device variable code	2	float	Device variable value
					6	struct	Slot1: Device variable properties
					12		Slot2: Device variable properties
					18		Slot3: Device variable properties
34	**Write Prim. Var. Damping**	0	float	PV 1 damping value	0	float	PV 1 damping value
35	**Write Prim. Var. Range Values**	0	int8	Units code	0	int8	Units code
		1	float	Upper range value	1	float	Upper range value
		5		Lower range value	5		Lower range value
36	**Set Prim. Var. Upper Range**	None			None		
37	**Set Prim. Var. Lower Range**	None			None		
40	**Enter/Exit Fixed Current**	0	float	Current value	0	float	Actual current value
42	**Device Reset**	None			None		
43	**Set Primary Variable Zero**	None			None		
44	**Write Prim. Var. Units**	0	int8	PV 1 units code	0	int8	PV 1 units code
45	**Trim Prim. Var. Current Zero**	0	float	Measured current value	0	float	Actual current value
46	**Trim Prim. Var. Current Gain**	0	float	Measured current value	0	float	Actual current value
50	**Read Dynamic Variable Assignments**	None			0	int8	PV 1 variable code
					1		PV 2 variable code
					2		PV 3 variable code
					3		PV 4 variable code

No	Title	Request Data			Response Data		
Common Practice							
51	**Write Dynamic Variable Assignments**	0	int8	PV 1 variable code	0	int8	PV 1 variable code
		1		PV 2 variable code	1		PV 2 variable code
		2		PV 3 variable code	2		PV 3 variable code
		3		PV 4 variable code	3		PV 4 variable code
54	**Read Device Variable Information**	0	int8	Device variable code	0	int8	Device variable code
					1	int24	Sensor serial number
					4	int8	Units code
					5	float	Variable upper limit
					9		Variable lower limit
					13		Variable damping
					17		Variable minimum span
					21	int8	Variable classification
					22		Variable family
					23	time	Acquisition period
					27	bin8	Variable properties
71	**Lock Device**	0	int8	Lock code	0	int8	Lock code
76	**Read Lock State**	None			0	int8	Lock status
78	**Read Aggregated Commands**	0	int8	Number of commands requested	0	int8	Extended device status
		1	str[]	Array of command requests struct { int16 command int8 byteCount int8[] requestData }	1	int8	Number of commands requested
					2	str[]	Array of command responses struct { int16 command int8 byteCount int8 responseCode int8[] responseData }
79[1]	**Write Device Variable**	0	int8	Device Variable Code	0	int8	Device Variable Code
		1		DV command code	1		DV command code
		2		DV units code	2		DV units code
		3	float	DV value	3	float	DV value
		7	int8	DV status	7	int8	DV status
103	**Write Burst Period**	0	int8	Burst message	0	int8	Burst message
		1	time	Update period	1	time	Update period
		5		Maximum update period	5		Maximum update period
104	**Write Burst Trigger**	0	int8	Burst message	0	int8	Burst message
		1		Trigger mode selection code	1		Trigger mode selection code
		2		Device variable classification for trigger level	2		Device variable classification for trigger level
		3		Units code	3		Units code
		4	float	Trigger level	4	float	Trigger level

[1] Used to simulate the value of a device variable

No	Title	Request Data			Response Data		
Common Practice							
105	**Read Burst Mode Configuration**	None			0	int8	Burst mode control code
					1	int8	Burst command number
					2	int8	Burst command slot 0
					3	int8	Burst command slot 1
					4	int8	Burst command slot 2
					5	int8	Burst command slot 3
		0	int8	Burst message	0	int8	Burst mode control code
					1		0x1f (31) command expansion
					2		DV code slot0
					3		DV code slot1
					4		DV code slot2
					5		DV code slot3
					6		DV code slot4
					7		DV code slot5
					8		DV code slot6
					9		DV code slot7
					10		Burst message
					11		Maximum number of burst messages
					12	int16	Extended command number
					14	time	Update time
					18		Maximum update time
					22	int8	Burst trigger mode code
					23		DV classification for trigger value
					24		Units code
					25	float	trigger value
106	**Flush Delayed Responses**	None			None		
107	**Write Burst Device Variables**	0	int8	DV code slot 0	0	int8	DV code slot 0
		1		DV code slot 1	1		DV code slot 1
		2		DV code slot 2	2		DV code slot 2
		3		DV code slot 3	3		DV code slot 3
		4	int8	DV code slot 4	4	int8	DV code slot 4
		5		DV code slot 5	5		DV code slot 5
		6		DV code slot 6	6		DV code slot 6
		7		DV code slot 7	7		DV code slot 7
		8		Burst message	8		Burst message
108	**Write Burst Mode Command**	0	int8	Command number for the burst response	0	int8	Command number of the burst response
109	**Burst Mode Control**	0	int8	Burst mode control code	0	int8	Burst mode control code
113	**Catch Device Variable**	0	int8	Destination DV code	0	int8	Destination DV code
		1		Capture mode code	1		Capture mode code
		2		Source slave manufacturer ID	2	int8[5]	Source slave address
		3		Source slave device type			
		2	int16	Source slave expanded device type			
		4	int8[3]	Source slave device ID			
		7	int8	Source command number	7	int8	Source command number
		8		Source slot number	8		Source slot number
		9	float	Shed time for this mapping	9	float	Shed time for this mapping
		7	int8	0x1f (31) command expansion	7	int8	0x1f (31) command expansion
		8		Source slot number	8		Source slot number
		9	float	Shed time for this mapping	9	float	Shed time for this mapping
		13	int16	Ext source command number	13	int16	Ext source command number

No	Title	Request Data			Response Data		
Common Practice							
114	**Read Caught Device Variable**	0	int8	Destination DV code	0	int8	Destination DV code
					1		Capture mode code
					2	int8[5]	Source slave address
					7	int8	Source command number
					8		Source slot number
					9	float	Shed time for this mapping
					7	int8	0x1f (31) command expansion
					8		Source slot number
					9	float	Shed time for this mapping
					13	int16	Ext source command number
523	**Read Condensed Status Mapping Array**	0	int8	Starting index status map	0	int8	Actual starting index
		1		Number of entries to read	1		Number of entries returned
					2	int4[]	Status map codes array
524	**Write Condensed Status Mapping Array**	0	int8	Starting index status map	0	int8	Actual starting index
		1		Number of entries to write	1		Number of entries returned
		2	int4[]	Status map codes array	2	int4[]	Status map codes array
525	**Reset Condensed Status Map**	None			None		
526	**Write Status Simulation Mode**	0	int8	Status simulation mode	0	int8	Status simulation mode
527	**Simulate Status Bit**	0	int8	Status bit index	0	int8	Status bit index
		1		Status bit value	1		Status bit value

Response Codes

As response code 1 is command specific it is documented together with the command specifications. However response code 2 is of general nature and contains 8 bit flags with the following meaning.

Flag Number / Meaning	Description
Bit #7 Field Device Malfunction	The device has detected a hardware error or failure. Further information may be available through the Read Additional Transmitter Status Command, #48.
Bit #6 Configuration Changed	A write or set command has been executed.
Bit #5 Cold Start	Power has been removed and reapplied resulting in the reinstallations of the setup information. The first command to recognize this condition will automatically reset this flag. This flag may also be set following a Master Reset or a Self Test.
Bit #4 More Status Available	More status information is available than can be returned in the Field Device Status. Command #48, Read Additional Status Information, will provide this additional status information.
Bit #3 Primary Variable Analog Output Fixed	The analog and digital analog outputs for the Primary Variable are held at the requested value. They will not respond to the applied process.
Bit #2 Primary Variable Analog Output Saturated	The analog and digital analog outputs for the Primary Variable are beyond their limits and no longer represent the true applied process.
Bit #1 Non Primary Variable Out of Limits	The process applied to a sensor, other than that of the Primary Variable, is beyond the operating limits of the device. The Read Additional Transmitter Status Command, #48, may be required to identify the variable.
Bit #0 Primary Variable Out of Limits	The process applied to the sensor for the Primary Variable is beyond the operating limits of the device.

Data Types

Float IEEE 754

The following summarizes the IEEE 754 and recommends that standards are referred to for implementation.

The floating point values passed by the protocol are based on the IEEE 754 single precision floating point standard.

Data Byte	#0	#1	#2	#3
	SEEEEEEE	EMMMMMMM	MMMMMMMM	MMMMMMMM

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

The value of the floating point number described above is obtained by multiplying 2, raised to the power of the unbiased exponent, by the 24-bit mantissa. The 24-bit mantissa is composed of an assumed most significant bit of 1, a decimal point following the 1, and the 23 bits of the mantissa.

$$S1.M \cdot 2^{(E-127)}$$

The floating point parameters not used by a device will be filled with 7F A0 00 00: Not-a-Number.

Double IEEE 754

The following summarizes the IEEE 754 and recommends that standards are referred to for implementation.

The floating point values passed by the protocol are based on the IEEE 754 single precision floating point standard.

Data Byte	#0	#1	#2	#3
	SEEEEEEE	EEEEMMMM	MMMMMMMM	MMMMMMMM
Data Byte	#4	#5	#6	#7
	MMMMMMMM	MMMMMMMM	MMMMMMMM	MMMMMMMM

S - Sign of the mantissa; 1 = negative
E - Exponent; Biased by 1023 decimal in two's complement format
M - Mantissa; 52 least significant bits, fractional portion

The value of the floating point number described above is obtained by multiplying 2, raised to the power of the unbiased exponent, by the 53-bit mantissa. The 53-bit mantissa is composed of an assumed most significant bit of 1, a decimal point following the 1, and the 52 bits of the mantissa.

$$S1.M \cdot 2^{(E-1023)}$$

Packed ASCII

The packed ASCII Format uses 6 Bit to encode a character. Therefore 4 characters in the original string require 3 octets in the resulting data. It is recommended to provide strings always as a multiple ordinal of 4 characters

Construction of Packed-ASCII characters:

a) Truncate Bit #6 and #7 of each ASCII character.
b) Pack four, 6 bit-ASCII characters into three bytes.

Reconstruction of ASCII characters:

a) Unpack the four, 6-bit ASCII characters.
b) Place the complement of Bit #5 of each unpacked, 6-bit ASCII character into Bit #6.
c) Set Bit #7 of each of the unpacked ASCII characters to zero.
d) The Packed ASCII code (hexadecimal) allows the representation of the following characters.

CHAR	CODE	CHAR	CODE	CHAR	CODE	CHAR	CODE
@	00	**P**	10	**Space**	20	**0**	30
A	01	**Q**	11	**!**	21	**1**	31
B	02	**R**	12	**"**	22	**2**	32
C	03	**S**	13	**#**	23	**3**	33
D	04	**T**	14	**$**	24	**4**	34
E	05	**U**	15	**%**	25	**5**	35
F	06	**V**	16	**&**	26	**6**	36
G	07	**W**	17	**'**	27	**7**	37
H	08	**X**	18	**(**	28	**8**	38
I	09	**Y**	19	**)**	29	**9**	39
J	0A	**Z**	1A	*	2A	**:**	3A
K	0B	[	1B	**+**	2B	**;**	3B
L	0C	\	1C	**,**	2C	**<**	3C
M	0D	**]**	1D	**-**	2D	**=**	3D
N	0E	**^**	1E	**.**	2E	**>**	3E
O	0F	_	1F	**/**	2F	**?**	3F

e) Note: The implementation of the function is assuming that the packed ascii string should be an ordinal multiple of 3. If the length of the passed string is not an ordinal multiple of 4 the missing packed ascii characters are replaced by spaces.