HART®
FIELD COMMUNICATIONS PROTOCOL

A Technical Overview
(Second Edition)

Romilly Bowden

November 1999

FISHER-ROSEMOUNT
Managing The Process Better.™

Bowden, Romilly
HART Field Communications
Protocol: a technical
overvier (2nd ed.)

DATE DUE	BORROWER'S NAME	MD [redacted]
~~5-22-00~~	~~Greg Hedges / AMTown~~ 515	754 3848 TC
10/12/01	~~Pierre LeVasseur~~	~~Hawk~~ ✓

CONTENTS

CHAPTER 3. TRANSACTION PROCEDURE, CODING AND MESSAGE STRUCTURE.

CHAPTER 4. COMMANDS, DATA AND STATUS.

PREFACE

The HART® protocol has become a de facto standard protocol for communication with Smart field devices in industrial process applications. HART is an acronym for "Highway Addressable Remote Transducer". The HART specification defines the physical form of transmission, transaction procedures, message structure, data formats, and a set of commands. It also allows a designer the freedom to define device-specific commands where appropriate.

The HART protocol was originally developed by Rosemount Inc. However, to encourage the use of digitally communicating field devices, Rosemount has passed all rights in the protocol to the HART Communication Foundation, and the HART protocol is freely available for anyone to use. An ever-increasing number of suppliers offer products using this protocol.

This booklet introduces the HART protocol, and gives some technical details of the current implementation (Revision 5.3). We hope it will help you to understand the features and benefits of the HART protocol, how it works and what it can do. It may also help you understand the complete specification documentation, by providing an alternative less-formal treatment. But it cannot be regarded as a replacement for the complete HART specification – if you are designing a HART-compatible instrument or host system, you will need the complete specification. This booklet does not cover all the details of HART, nor will it necessarily be kept up-to-date. Under no circumstances should it be taken as definitive.

The booklet is designed to be read sequentially, with new topics being introduced as they are needed, sometimes at a simple level in one place, then with more detail later. To make it easier to use for reference, it has short sections, each on a specific topic. The index at the end of the booklet will direct you to one or more sections dealing with any listed subject.

Chapter 1 provides an overview of "smart" instrumentation and the HART protocol. Chapter 2 describes the physical signalling method and the transmission medium. Chapter 3 describes the transaction procedure, and the coding of characters and other data. Chapter 4 describes the commands used to operate a field device, and includes extensive reference tables. Chapter 5 introduces the Device Description Language, an important technique for interoperability.

A glossary of technical terms and abbreviations follows Chapter 5. The explanations given are aimed particularly at the relationship of the term to the HART protocol. You may like to refer to this for further explanation of any unfamiliar words or concepts, or as a reminder of HART usage.

A list of further HART-related documents, software tools and contact addresses is also included for reference.

The author would welcome any comments or suggestions on the content or presentation of this booklet.

HART is a registered trademark of the HART Communication Foundation.

[This page intentionally blank]

CHAPTER 1. "SMART" INSTRUMENTS AND THE HART PROTOCOL

1.1 Introduction

This chapter introduces the main concepts of digital communication with field instruments, as implemented by Fisher-Rosemount in its "Smart Family" of transmitters using the HART protocol.

1.2 "Smart"

The description "smart" for a field device has been used in the sense of "intelligent", to describe any device which includes a microprocessor. Typically, this would imply extra functionality, above what had previously been provided in similar non-microprocessor-based instruments. For example, a smart transmitter might provide better accuracy through the use of a numerical calculation to compensate for sensor non-linearity or temperature dependence. It might be able to operate with a variety of different sensor types. It might combine two or more measurements into a single new measurement (for example volume flow rate and temperature into mass flow). Or it might allow re-ranging or semi-automatic calibration. Often, it would provide internal diagnostic self-test functions to simplify maintenance procedures.

As well as giving better performance, this extra functionality can reduce the processing needed in the host (control system), and may also result in a range of instruments being reduced to a single model, with advantages in manufacturing and inventory management.

1.3 Configurators

To make use of these extra features, "smart" devices usually need a plug-in "configurator", a box with a display and a number of push-buttons for the user to set up and control the instrument. (Providing these as a local operator interface on the device itself is generally too expensive, and clumsy, for field-mounting units, but may be appropriate for more complex panel-mounting instruments.)

1.4 Digital communication

A logical next step is to allow the instrument and its "configurator" box to be separated by a greater distance, by using properly-specified serial communications between them. A further step combines this communication on to the two wires already used to connect the device back to the central control room. This brings us to Fisher-Rosemount's present use of the word "smart", to describe field devices in which the analogue signal, digital communication and (generally) power co-exist on the same pair of wires.

With such instruments, the advantages of digital communication are obtained, while retaining compatibility with the analogue signal inputs required by existing systems. Now, in addition to using digital communication to set up and control the field device, it becomes possible to read the measured variable over the communication link. Without modification, these instruments are ready for fully-digital system use.

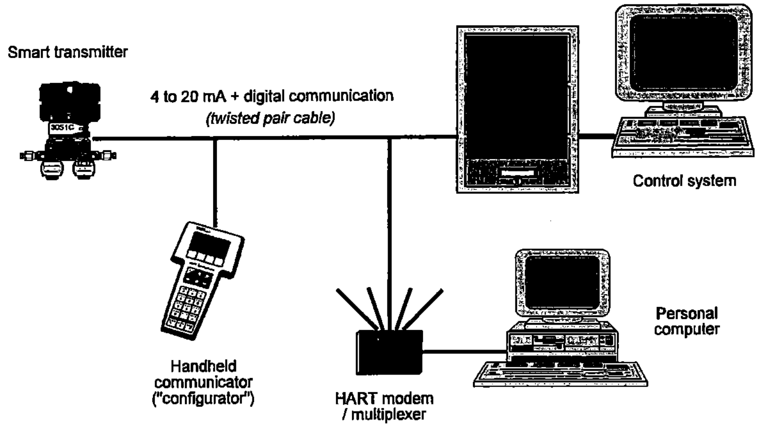


Figure 1-1. A Smart system

1.5 Reading the measured variable by digital communication

Using digital communication to read the measured variable, it becomes possible for a single instrument to provide more than one measurement. (Thus, for example, a Coriolis mass flow transmitter can provide mass flow rate, process temperature, density and totalized mass flow in a single message.) It becomes possible to check on the continued good health of the field device every time a measurement is made, giving a valuable increase in confidence and security. There is a further gain in accuracy, in that the intermediate steps of digital-to-analogue and analogue-to-digital conversion to and from the 4 to 20 mA signal are omitted.

Note, however, that the time taken to communicate the message adds an extra delay (dead time) to the measurement, which could adversely affect the control of fast loops. If this is a problem, it will be preferable to use the analogue value for control purposes. The higher communication speed of Fieldbus (see 1.15 below) will eventually remove this limitation.

1.6 Additional information

Digital communication also makes it worthwhile to keep additional information in the field device, to be read out when required. This leads to several useful possibilities. It can give process-related information such as tag number and a description of the measurement, and the instrument's calibrated range and units. Or it can give information about the device itself, acting as an electronic "label". Further, it can be used to keep records of maintenance-related activities such as the date of last calibration. Automated instrument management systems become possible, using accurate up-to-date information from the device itself.

1.7 Multidrop communication

If the measured variable is going to be read by digital communication, the analogue 4 to 20 mA signal is no longer required. It then becomes possible to connect multiple field devices in parallel to a single pair of wires, and to communicate with each one in turn to read its measurement (or other data). To do this, each device must have an "address", to which it will respond, and each request from the host must include this address as part of the message.

This "multidrop" connection can significantly reduce the cost of field wiring and host input interface electronics, and may be valuable in monitoring systems. Note, however, that the use of a cyclic scan means that each measurement is only examined at intervals, and the cycle time for a complete scan may be too long for high-speed control loops.

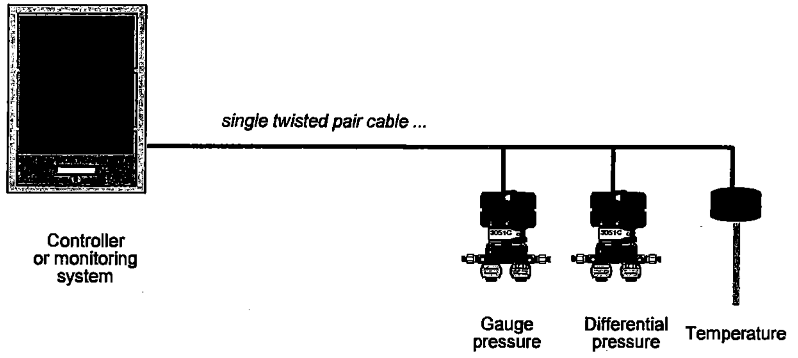


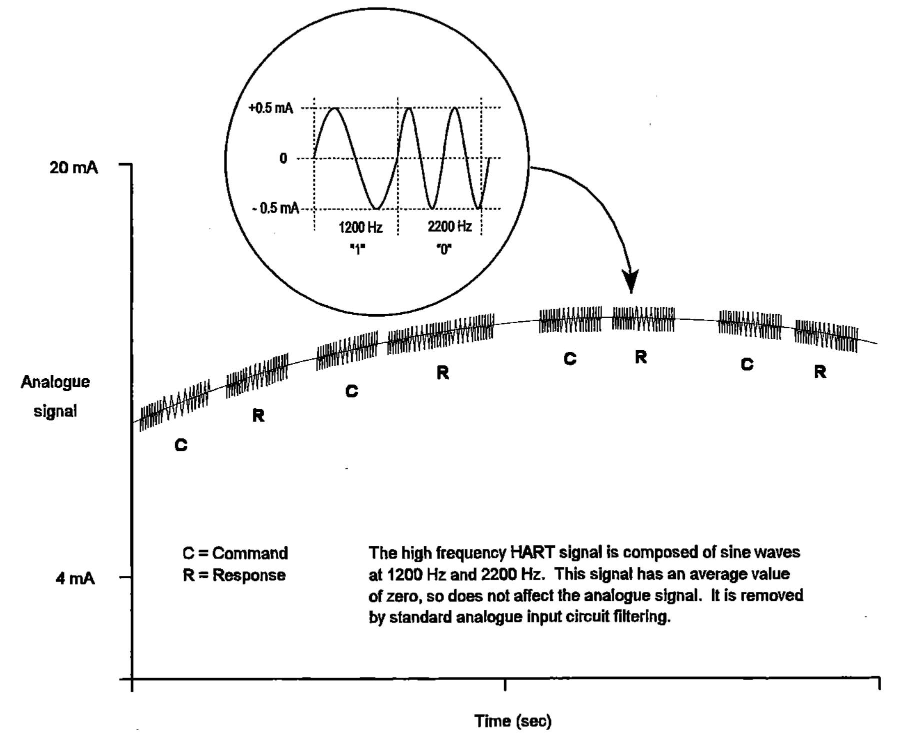
Figure 1-2. Multidrop communication

1.8 The HART protocol

To use these extra features easily with a range of different hosts and field devices, a communication standard is needed. This has to include specifications for the physical form of transmission, transaction procedures, message structure, data formats, and a set of commands to perform the required functions.

The HART protocol was developed by Rosemount Inc. for this purpose. HART is an acronym for "Highway Addressable Remote Transducer". To encourage the use of digitally communicating field devices, Rosemount Inc. has passed all rights in the protocol to the HART Communication Foundation (HCF) and the HART protocol is freely available for anyone to use. See 1.16 below.

The remainder of this booklet describes the HART protocol in detail. In brief, HART uses the Bell 202 standard frequency shift keying (f.s.k.) signal to communicate at 1200 baud, superimposed at a low level on the 4 to 20 mA analogue measurement signal. Having an average value of zero, an f.s.k. signal causes no interference with the analogue signal (see Figure 1-3).



The high frequency HART signal is composed of sine waves at 1200 Hz and 2200 Hz. This signal has an average value of zero, so does not affect the analogue signal. It is removed by standard analogue input circuit filtering.

Figure 1-3. The HART signal

HART is a master-slave protocol – a field device only replies when it is spoken to. There can be two masters (a control system and a hand-held HART Communicator, for example). Up to 15 slave devices can be connected to a single multidrop cable pair (up to four devices, in intrinsically-safe applications).

Each message (see Figure 1-4) includes the addresses of its source and its destination, to ensure that it is received by the correct device, and has a "checksum" to allow detection of any corruption of the message. The field device's status is included in every reply message, indicating its continued good

health. There may or may not be "data" included in a message, depending on the particular command. Two or three message transactions can be made each second.

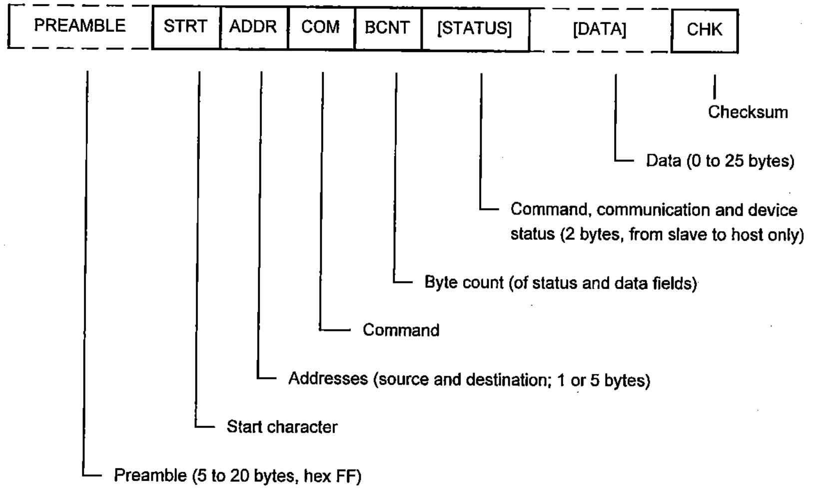


Figure 1-4. The HART message structure

1.9 Universal commands

The commands of the HART protocol are defined in three groups. The first group, "universal commands", provide functions which are implemented in all field devices. Table 1-1 lists these. See Table 4-4 for more details.

1.10 Common-practice commands

The second group, "common-practice commands", provide functions common to many field devices, but not all. If a device implements these functions, these commands should be used to perform them. Table 1-2 lists some of these. See Table 4-6 for a complete list and more details.

1.11 Device-specific commands

The third group, "device-specific commands" (previously called "transmitter-specific"), provide functions which are more or less unique to a particular field device. Table 1-3 lists a few examples.

Table 1-1. Universal commands.

Command number(s)	Function
0, 11	Read manufacturer and device type
1	Read primary variable (PV) and units
2	Read current output and percent of range
3	Read up to four pre-defined dynamic variables
6	Write polling address
12, 17	Read or write 32-character message
13, 18	Read or write 8-character tag, 16-character description, date
14	Read sensor serial number and limits
15	Read transmitter range, units and damping time constant
16, 19	Read or write final assembly number

Table 1-2. Some common-practice commands.

Command number(s)	Function
33	Read a selection of up to four dynamic variables
34	Write damping time constant
35	Write transmitter range
36, 37	Re-range (set span and zero)
40	Set fixed output current
41	Perform self-test
42	Perform master reset
43	Trim (set) PV zero
44	Write PV units
45, 46	Trim DAC zero and gain
47	Write transfer function (square root, linear, etc.)
48	Read additional device status
49	Write sensor serial number
50, 51	Read or write dynamic variable assignments

Table 1-3. Examples of device-specific commands.

Command(s)	Device	Function
128, 129	1151S	Read or write materials of construction
130, 131	3044C	Read or write sensor type
138, 139	8712	Read or write low flow cutoff value
146	9712	Start, stop or clear totalizer
146, 147	1054A	Read or write alarm relay set point
153, 154	9712	Read or write density calibration factor
166	3680	Write gamma source

1.12 Output devices

So far in this chapter, we have described "smart" and HART in terms of measuring instruments and inputs to control and monitoring systems; indeed this is what HART was originally designed for. But the protocol is now also used for output devices – valve positioners and current-to-pressure transducers. Significant benefits are obtained by making enhanced diagnostic information available from these devices, to the control system or to a maintenance management computer.

1.13 HART products

A wide range of products is now available using the HART protocol. Table 1-4 shows many of these, and notes the areas in which Fisher-Rosemount offers products. "The HART Book", published from time to time by GGH Marketing Communications, is a useful reference source.

1.14 Device Description Language

The HART "Device Description Language" (DDL) is a formal language (like a simple computer programming language), which lets a device designer describe completely and unambiguously what a field instrument looks like when you talk to it through the "window" of its digital communication link. The Device Description includes a definition of accessible variables, commands, and operating procedures. It also includes the menu structure which a host device can use for a human operator.

Device Descriptions make it easy to upgrade hosts to support new field devices, without re-writing software. Device Descriptions can be used by any suitably-designed host device (handheld communicator, control system or instrument management system) to automatically provide a correct and complete user interface for each field device. Device Descriptions allow interoperability and a degree of interchangeability between smart instruments from different manufacturers, even though the instruments' functions may be implemented in different ways. Users can choose the best instrument for each application, without being locked in to a single supplier for the complete system.

Although simple host systems can be designed without using Device Descriptions, they will be limited to using universal and common-practice commands, or will need custom upgrades for each new field device. See Chapter 5 for more detail on Device Descriptions and the Device Description Language.

Table 1-4. Some available HART products

Product category	Description	Fisher-Rosemount products
Field devices	Analytical (pH, conductivity, flue gas oxygen, ORP, RCL, DO)	✓
	Density (Coriolis, nucleonic)	✓
	Flow (DP, magnetic, vortex, ultrasonic, gas flow / multivariable)	✓
	Level (displacement, capacitive, hydrostatic, ultrasonic, microwave, impedance)	✓
	Mass flow (Coriolis, thermal, vortex, multivariable)	✓
	Pressure (absolute, differential, gauge)	✓
	Temperature (RTD, thermocouple, infra-red, mV)	✓
	I-to-P transducer	✓
	Valve position controller	✓
	Process controller (PID, advanced)	
Host devices	Distributed Control System (DCS)	✓
	Programmable Logic Controller (PLC)	
	Single Loop Controller	
	Handheld Communicator	✓
	Handheld PC	
	Flow Computer	
	Hydrostatic Tank Gauge	✓
Miscellaneous components	Modems (standalone, PC card: ISA, PCMCIA)	✓
	Modem IC chips	
	Multiplexers	✓
	Protocol converters	
	Intrinsic safety barriers and repeaters	
Software	Process monitoring	
	SCADA	
	Telemetry	
	Instrument configuration	✓
	Instrument calibration	✓
	Instrument diagnostics	✓
	Instrument / maintenance management	✓

1.15 Fieldbus

Fisher-Rosemount continues to work with others, both in the Fieldbus Foundation and in the national and international standards committees, to define a worldwide higher-speed field communication standard, generally referred to as "Fieldbus". Like HART, this is designed to be used as the lowest level in a hierarchical structure of functional devices and communication networks, such as that shown in Figure 1-5. However, the performance being demanded is significantly greater than that provided by today's instruments, both in communication speed and quantity of data. Power over the bus, multidrop and intrinsically-safe operation are required features, but compatibility with analogue systems is *not* retained. Several communication speed options are proposed.

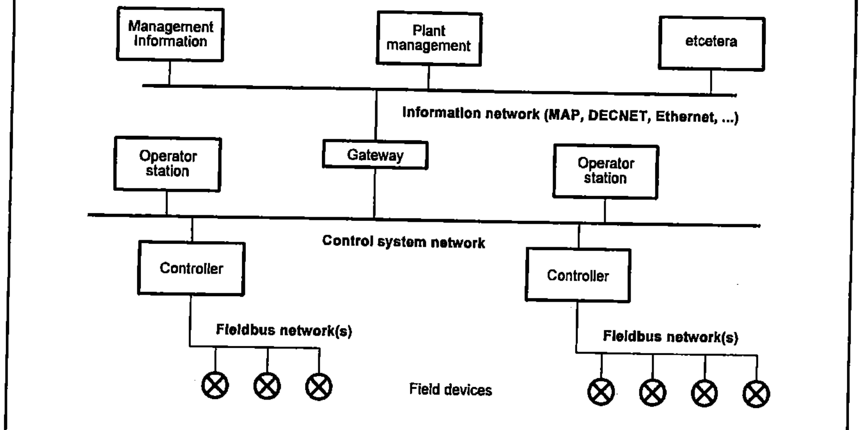


Figure 1-5. A Fieldbus system with hierarchical structure

Today's smart instruments are part of a continuing evolutionary trend, from pneumatic instruments (3 to 15 psi), through analogue electronic (4 to 20 mA) and simultaneous analogue/digital communication (HART) to full digital communication (Fieldbus). This is part of the general move towards a wider use of digital communication in instrument and control systems, leading to eventual integration with Management Information Systems.

The HART protocol already allows instrument manufacturers and users to get many of the benefits of, and to gain experience with, digital field communications, while keeping compatibility with existing analogue systems. This experience will help both manufacturers and users to judge the benefits and problems of using digital communication in plant operation. We expect the HART protocol to have a long life, alongside Fieldbus, for upgrades and extensions to existing systems, and for applications where compatibility with analogue signals continues to be important.

1.16 The HART Communication Foundation

The HART Communication Foundation (HCF) is a not-for-profit Foundation dedicated to promoting and supporting the use of the HART protocol. To encourage the widespread use of HART, Rosemount Inc. has transferred the registered trademark and all rights in the protocol to the HCF. The protocol remains open and free for all to use without royalties. The HCF is supported by its members (over 60 at a recent count), and can provide documentation, training and support to all users or interested parties.

1.17 Summary

This chapter has described the evolution of the present state of "smart" field devices and the HART protocol. Major features are summarised in Table 1-5 below. This combination of features is unique to the HART protocol; the resulting benefits provide powerful reasons for instrument suppliers and users to make use of this protocol.

Table 1-5. Major features of field devices using HART

Features	Benefits
"Smart" electronics	Improved accuracy. Wider functional range reduces inventory.
HART – an "open" protocol.	Users are not locked in to a single supplier.
Two-wire system.	Can use existing field wiring.
Simultaneous analogue and digital communication.	Compatible with existing analogue systems, but ready for fully-digital systems.
Multidrop option.	Allows economy in field wiring.
Multimaster protocol.	Can use hand held communicator without disturbing the control system.
Status with every message.	Improved data integrity.
Remote self-test and adjustment.	Invaluable for inaccessible instruments.
Extensive on-line instrument data.	Accurate records for maintenance and inventory control.
Access to on-line diagnostic data.	Improved performance. Reduced cost of maintenance procedures.
Universal and common-practice commands.	Operation with new devices.
Device-specific commands.	Allow innovation in field device design.
Read device identity (tag).	Easy tracing of field wiring.
"Set output" command.	Easy checking of loop integrity.
Bell 202 standard.	Proven reliability. Low cost modem ICs available to manufacturers.
Device Description Language.	Interoperability of devices from different suppliers.

CHAPTER 2. THE PHYSICAL SIGNAL

2.1 Introduction

This chapter describes the physical signalling method and transmission medium of the HART protocol. These correspond to layer 1 – the physical layer – of the OSI protocol reference model.

2.2 Frequency-shift keying

HART uses a frequency-shift keying technique to superimpose digital communication on to the 4 to 20 mA current loop connecting the central system to the field device. Two different frequencies (1200 Hz and 2200 Hz respectively) are used to represent binary 1 and 0.

These sine-wave tones are superimposed on the d.c. signal at a low level (see Figure 1-3). The average value of a sine-wave signal is zero, so no d.c. component is added to the existing 4 to 20 mA signal, no matter what the digital data may be. Consequently, most existing analogue instruments continue to work as usual – the low-pass filtering usually present effectively removes the communication signal.[†]

The data rate used is 1200 baud. That is to say, binary digits are transmitted at a rate of 1200 per second. This means that a 1 is represented by a single cycle of 1200 Hz, while a 0 is represented by approximately two cycles of 2200 Hz.

This choice of signalling frequencies and transmission rate accords with the American "Bell 202" standard, one of several used to send digital information over telephone networks. As a result of this, suitable integrated circuit modem chips are widely available at low cost. In the USA, it is permissible to transmit this signal over the public telephone network. Unfortunately, this standard is not approved for use over European public telephone networks. (In Europe, back-to-back modems could be used to convert Bell 202 to RS-232 and thence to CCITT standard V.22 or V.23, if operation over a public network is required).

2.3 Signal levels

The HART protocol specifies that master devices (a host control system or a hand-held communicator) transmit a voltage signal, whereas slave (field) devices transmit a current signal. (Recall that the normal operation of a 2-wire transmitter is to control the loop current; it is easy to extend this control to generate the small high-frequency component of the HART communication signal.)

[†] Fast sampling analogue-to-digital converters used in some control systems (especially PLCs) may be troubled by the presence of the HART signal. Using a voltage-sensitive input and an external filter should resolve this problem. (A single-pole 10 Hz low-pass filter reduces the communication signal to a ripple of about ±0.01% of the full-scale analogue signal).

The current signal is converted into a corresponding voltage by the loop load resistor, so all devices use voltage-sensitive receiver circuits. The specified peak-to-peak signal levels are shown in Table 2-1. Ideally, the wave shape is sinusoidal, but a trapezoidal waveform is acceptable within limits (see the full HART specification). A square wave is not acceptable.

Table 2-1. HART signal levels

Master transmitted signal	min 400 mV p-p max 600 mV p-p
Slave transmitted signal	min 0.8 mA p-p max 1.2 mA p-p
Minimum slave signal, converted by a load of 230 Ω	184 mV p-p
Maximum slave signal, converted by a load of 1100 Ω	1320 mV p-p
Receiver sensitivity (must receive correctly)	120 mV to 2.0 V p-p
Receiver threshold (must ignore)	80 mV p-p

For output circuits from a control system to a valve positioner, the same signal levels are used, but the field (slave) device also uses voltage signalling. In this case, the impedance of the field device forms the loop load resistor. See 2.13 below.

The receiver sensitivity specification allows for some attenuation of the signal due to cable or other component effects. The receiver threshold specification reduces the likelihood of interference from external signals, and prevents crosstalk from other HART signals running in adjacent cables, or sharing less-than-ideal grounding or power supply systems.

2.4 The connection loop

The conventional connection circuit for a two-wire loop-powered transmitter is shown in Figure 2-1. In practice, the three items (the power supply unit PSU, the transmitter Tx and the load resistor RL) may be connected in any order, and any point in the circuit may be grounded. The HART specification allows load resistors between 230 and 1100 Ω.

The HART communication signal must be introduced into, and detected from, the field loop. The power supply is almost a short circuit at the HART signalling frequencies, so a communicating device (a hand-held communicator or the communication circuitry of a host control system) cannot be connected directly across it. Instead, it should be connected either to the two wires to the field (at A and B), or across the load resistor (at B and C), in which case the circuit is completed through the power supply. Of course, connecting in the field, directly across the field device, is equally acceptable.

A HART communicator must not present any d.c. load to the line. To ensure this, it should include, or be connected through, a capacitor of about 5 μF or more. Even with capacitors present, care may be needed with grounding, to avoid an a.c. ground connection bypassing the high-frequency HART signal. Full galvanic isolation of the host connection eliminates this possibility.

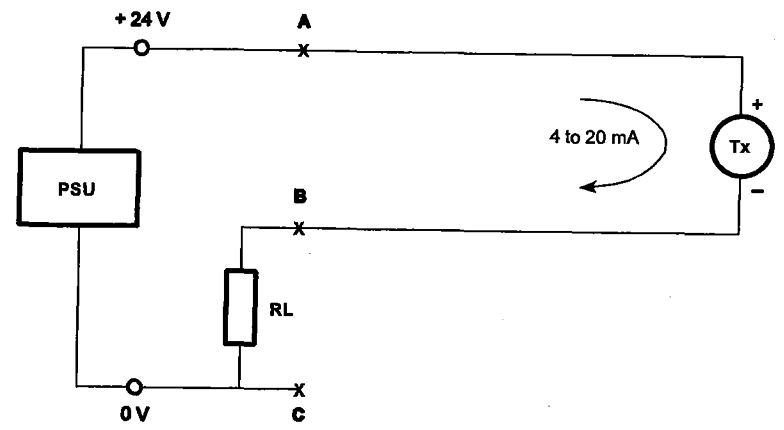


Figure 2-1. The two-wire current loop

2.5 Active-source devices

Some HART devices run on local power, and provide an active source for their 4 to 20 mA output and HART communication, instead of using the two-wire loop scheme shown above. Connection of these devices is shown in Figure 2-2 below; any communicating device is connected across the load resistor at B and C (or in the field, directly across the field device).

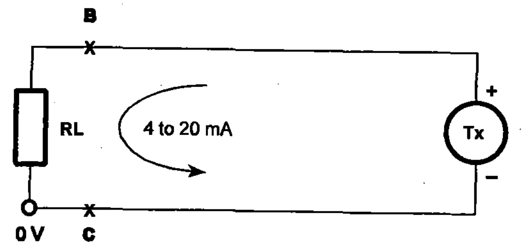


Figure 2-2. The current loop for an active-source field device

2.6 Multidrop operation

The HART protocol includes a destination address in each message. By giving each slave device a different address, a number of such devices can be connected in parallel across a single pair of field wires. Each one then accepts only messages addressed specifically to it (or broadcast messages). Since the analogue 4 to 20 mA signals would all add together to give a meaningless total, the act of setting a non-zero polling address is also used to park the analogue signal at 4 mA (enough to power the device), thus reducing the total power requirement. Up to 15 field devices are allowed in a multidrop system. Figure 2-3 shows three multidropped two-wire field devices.

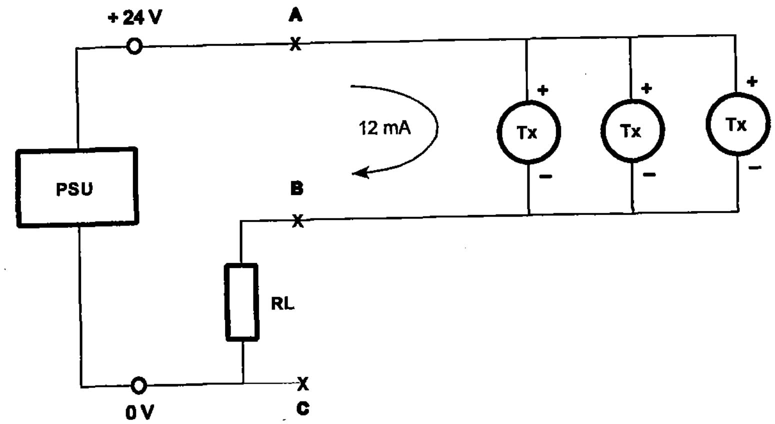


Figure 2-3. A two-wire multidrop loop

In point-to-point (single slave device) operation, the primary variable can be read either as an analogue value, or by digital communication. In the multidrop mode, digital communication must be used to read the primary variable, since the analogue signal is no longer available.

It is possible to mix two-wire current loop and active-source devices in a multidrop scheme, but because of their different methods of connection, a third wire is needed to the field, as shown in Figure 2-4 below. Current flow is shown by the arrows. The upper transmitters are two-wire loop-powered; the lower transmitters are separately-powered active-source devices. If "twisted triple" cable is not available, such a mixed system should be constructed using two separate twisted pairs, connected together at the load resistor. A communicating device can still be connected either across A and B, or across B and C, or across a field device, for communication to any field device.

2.7 Device characteristics

To allow HART systems to be designed reliably without detailed information on each device in the system, limits are specified for the impedances presented by any single device. See Table 2-2.

Notice that the primary master shunt impedance is specified on the assumption that it includes the loop load resistor. If this is not the case, the device's shunt impedance needs to be higher, so that the combination meets the specification.

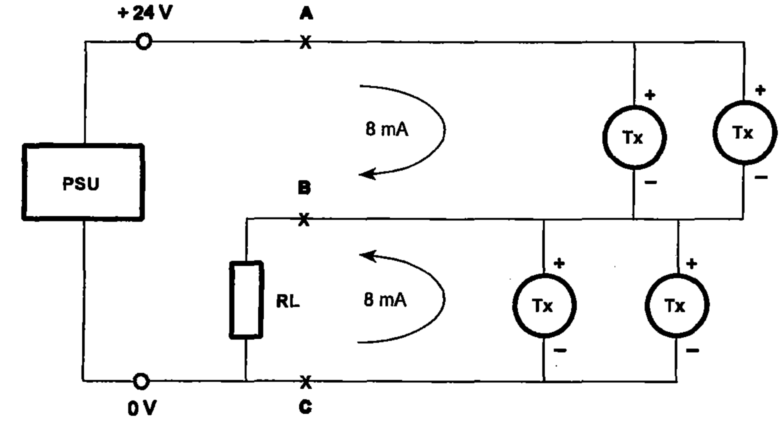


Figure 2-4. Mixed two-wire and active-source devices

Table 2-2. Impedance specifications

Primary master (including load resistor)[1]	Shunt impedance (receiving)	230 to 1100 Ω
	Maximum source impedance (sending)[2]	700 Ω
Secondary master[1]	Minimum shunt impedance (receiving)	5 kΩ
	Maximum source impedance (sending)	100 Ω
Slave device	Minimum shunt resistance	100 kΩ
	Maximum shunt capacitance[3]	5000 pF
Miscellaneous devices[4] (total)	Minimum shunt impedance	10 kΩ
	Maximum series impedance	100 Ω

Notes:

1. There are also separate limits on the reactive (inductive or capacitive) component of the master device impedance limits.
2. The primary master's source impedance when sending must also be no greater than its shunt impedance when receiving.
3. The 5000 pF limit on slave device shunt capacitance is a recommendation rather than an absolute limit. Devices having higher values must state their "CN" factor. Capacitance number CN is the actual device capacitance, divided by 5000 pF. (For example, a device with a capacitance of 22000 pF has a CN of 4.4, normally quoted as the next higher integer, 5.)
4. A "miscellaneous device" is any passive instrument in the loop, such as a local current indicator.

2.8 Signal attenuation and distortion – the 65 μs limit

In any network containing resistance and capacitance, signals are attenuated, and delayed (shifted in phase), as they pass through. The amount of attenuation and delay depends on the frequency of the signal, relative to the "cut-off" frequency of the network. To ensure reliable reception of the HART signal across the load resistor, the signal from the field device must not be attenuated by more than 3 db (a factor of 0.707). This allows a small safety margin for the lowest transmitted signal (0.8 mA), the lowest permitted load resistor (230 ohms), and the most insensitive receiver (120 mV) (see Table 2-1). In addition, the two signalling frequencies must not be delayed unequally by more than about 50 μs, or the composite waveform will be distorted and the data recovery circuits may fail to separate the two frequencies correctly.

To ensure that these conditions are met, the HART specification imposes a minimum cut-off frequency of 2500 Hz (at 3 db attenuation), slightly above the highest HART signalling frequency. A simple resistance-capacitance circuit will meet this requirement if it has an RC time constant value of 65 μs or less. (This means: multiply together the circuit resistance R and the circuit capacitance C. Include the units; remember ohms × farads = seconds, for example 250 Ω × 0.1 μF = 25 μs.)

This may all seem rather complicated – just remember that a HART system must be designed to have an RC time constant of 65 μs or less. In a simple case, the resistance R is the sum of the load resistor and the cable resistance, and the capacitance C is the sum of the cable capacitance and the capacitances of the connected devices. To allow high capacitance, keep the load resistor as low as possible (but not less than the 230 Ω limit) – 250 Ω is a commonly-used value. What this means in terms of the permissible number of devices and cable lengths is discussed in 2.9 below.

If there are other devices in series with the loop, such as a local current indicator, chart recorder, or IS barrier, the series resistance of these components (in so far as it is not shunted by a capacitor at HART signal frequencies) needs to be added in to the value for R.

2.9 Cabling

For all but the shortest cable runs, the field wiring of a HART system should use a screened twisted pair cable. For cable lengths above 1500 m, use individually-screened twisted pairs to avoid possible crosstalk between pairs. Below 1500 m, overall-screened multiple twisted pairs are acceptable, but in this case, it is important not to use the other pairs for any signals which might interfere with the HART communication. (They can be used for other HART lines, or for pure analogue lines, providing the HART limits on rate of change of analogue signal are met – see 2.12 below.)

If the cable is longer than a few metres, its resistance and capacitance may become significant in the RC time constant limitation (see 2.8 above). Of course, its resistance may also be important in the loop voltage drop calculation which any two-wire loop-powered system requires.

The relevant cable parameters depend on conductor diameter, insulation type and insulation thickness. This is the insulation which covers and separates the two copper conductors – outer protective coverings are not important in this respect. For HART signal calculations, the important parameters are the capacitance measured from one conductor to all others and screen *(not between the two conductors of a pair, as commonly quoted)*, and the resistance of both conductors in series.

If possible, when estimating the effect of cable length on the HART signal, use real values measured or specified for the particular cable used in the actual installation. Otherwise, a rough estimate of capacitance and resistance can be made from a knowledge of the insulating material and the conductor size, using Table 2-3. In general, the lowest capacitance cables have thinner conductors, and therefore higher resistance. Typical combinations for some common cable types are shown in Table 2-4.

Table 2-3. Cable parameters

Insulation	Capacitance
PVC	300 - 400 pF/m
polyethylene	150 - 200 pF/m
polyethylene foam	75 - 100 pF/m

Conductors			Resistance (both conductors in series)
Area	Diameter	AWG	
2.0 mm^2	1.6 mm	14	17 Ω/km
1.3 mm^2	1.3 mm	16	28 Ω/km
0.8 mm^2	1.0 mm	18	45 Ω/km
0.5 mm^2	0.8 mm	20	70 Ω/km
0.3 mm^2	0.6 mm	22	110 Ω/km
0.2 mm^2	0.5 mm	24	160 Ω/km

Table 2-4. Some typical cables.

Cable type	Insulation	Example	Capacitance	Resistance
Instrumentation-grade screened twisted pair	PVC	BS5308 part 2	400 pF/m	24 - 80 Ω/km
	polyethylene	BS5308 part 1	200 pF/m	24 - 80 Ω/km
	polyethylene, foam	Kerpen 7093	100 pF/m	36 Ω/km
Overall-screened multi-core	PVC	Belden 8441	270 pF/m	110 Ω/km
Computer-grade screened twisted pair	polyethylene	Belden 9873	180 pF/m	75 Ω/km
Low-capacitance (RS-485 / RS-422)	polyethylene or proprietary, foam	Belden 9729	73 pF/m	160 Ω/km

In the simple case of a single field device and a single host, with a 250 Ω load and no other significant resistance, the 65 μs rule would allow 0.26 μF total capacitance. Allowing 0.01 μF for device capacitance (5000 pF each for one field device and a possible secondary master), the cable capacitance could be up to 0.25 μF. However, allowing for the cable resistance reduces the permitted total capacitance and therefore the cable length. For a typical 1 mm^2 polyethylene-insulated instrumentation cable with 200 pF/m capacitance and 36 Ω/km resistance, the 65 μs rule allows 1100 metres of cable. Using the best of the cables in the table (100 pF/m and 36 Ω/km), 2000 metres is possible (still well short of the specified maximum HART cable length of 3000 metres). See Table 2-5 below.

Multidrop operation reduces the possible cable length, since the capacitance of the field devices uses more of the allowance. The effect of a high CN number is very significant in multidrop systems. Table 2-5 shows some examples of this.

Table 2-5. Maximum length for typical 1 mm^2 cables

Field devices	Cable insulation		
	PVC	Polyethylene	Polyethylene foam
1 (CN = 1)	600 m	1100 m	2000 m
10 multidrop (CN = 1)	500 m	900 m	1600 m
10 multidrop (CN = 4.4)	85 m	150 m	250 m

Notes: These lengths assume a 250 Ω load resistor and no miscellaneous devices.

Cable capacitances are taken as 400 pF/m, 200 pF/m and 100 pF/m respectively, for PVC, polyethylene and polyethylene foam insulation.

2.10 Grounding

To prevent interference by external signals, it is important to ground the system properly. In particular, the signal loop should be grounded, if at all, at one point only. The cable screening must be connected to ground, at one point only, and must not be connected to instrument or junction box cases unless these are isolated from ground. The single ground point will usually be at or near the primary master (for example, the control system).

2.11 Power supply

Power for a two-wire instrument loop is typically 24V d.c. As always, the voltage must be sufficient to provide the necessary lift-off voltage for the field device, taking into account voltage drops in the cable and load resistor, and a passive IS barrier if one is present. Smart devices may take up to 22 mA to indicate an alarm condition; use this value to calculate the worst loop voltage drop.

There are additional communication-related specifications for the power supply for a HART loop; these are shown in Table 2-6 below. The ripple and noise specifications are designed to prevent direct interference with the HART signals. The impedance limit ensures that HART signals see the power supply as a low impedance path, and prevents inadvertent coupling and crosstalk between multiple HART loops powered from a common supply. (The resistance of output fuses, if any, must be included, when measuring this value.)

Table 2-6. Power supply specifications

Maximum ripple (47 to 125 Hz)	0.2 V p-p
Maximum noise (500 Hz to 10 kHz)	1.2 mV rms
Maximum series impedance (500 Hz to 10 kHz)	10 Ω

2.12 Analogue signal bandwidth

To avoid interference with the superimposed HART communication signal, the rate-of-change of the analogue output of a HART-compatible transmitter must be limited above 25 Hz by a filter giving 40 db/decade attenuation. The HART receiver is specified to reject any signal which could be produced by a 16 mA square wave, passed through such a filter.

2.13 Output devices

For output devices, the HART specifications are adapted to take into account the different impedances of the master (control system) and slave (valve positioner or other transducer). In this case, the control system generates the 4 to 20 mA current signal, and is therefore a high impedance device (at least at d.c. and low frequencies). The valve positioner, on the other hand, has fairly low resistance, dropping perhaps 10 volts at 20 mA (a 500-ohm load). Ideally, the controller would maintain its high impedance up through the HART signal frequency band, and could impose a current modulation for the HART signal; the slave could use voltage modulation. In practice, many existing controllers do not meet this impedance characteristic, and some are upset by the appearance of HART signals on their output connections. They may also generate a rapidly-changing analogue output signal, which can interfere with HART communication (see 2.12 above).

The HART Communication Foundation is working on specifications to ensure good operation of HART for output devices. In the meantime, it is necessary to check carefully for compatibility, and it may be necessary to use a filter to isolate the controller output circuit from the HART signal. A separate technical note is available with more information on this subject.

2.14 Other devices

Other analogue devices such as local indicators or chart recorders can be included in the loop, as long as they meet the limits on series and shunt impedance for "miscellaneous devices" (see Table 2-2 above). In particular, if a chart recorder is connected to sense the voltage across an additional series resistor of more than a few ohms, it should be shunted by a capacitor to bypass the HART signal.

2.15 Intrinsic safety barriers

Systems using intrinsic safety (IS) barriers need special care. In addition to the usual check on loop voltage drop, the supply voltage to a passive shunt diode barrier must be reduced by 0.6 V to allow headroom for the HART signal. This avoids conduction by the zener diodes on signal peaks, which would introduce an error in the analogue signal. The series resistance of the barrier must be included in the RC time constant calculation for the 65 microsecond rule.

For the more complex active barriers, somewhat different considerations apply. A separate technical note is available with more information on this subject. Most suppliers now offer repeater/isolator barriers specifically designed to pass HART signals successfully.

Depending on their equivalent capacitance and other IS certification parameters, up to four field devices may be multidropped in an IS system, still leaving some of the hazardous side capacitance allowance for cabling.

2.16 Voltage-mode HART

An alternative physical layer has been defined for use in low-power field devices. This uses voltage modulation of the HART f.s.k. signal for communication in both directions, superimposed on a voltage-mode analogue signal of 1 to 5 volts. This involves changes to the permissible device impedance specifications, and is only workable for point-to-point (non-multidrop) applications. In addition, the possible signalling distance is much reduced: 150 metres should always be possible; 330 metres may be possible, depending on system details.

2.17 RS-485 HART

Some vendors (including Micro Motion) offer instruments using HART frame and message formats over an RS-485 physical layer, independently of the analogue output signal. This is a purely digital signal, not using the f.s.k. technique. With a balanced impedance-matched line, higher communication speeds are possible, up to 38400 bps, resulting in faster sampling rates for process measurements. At speeds other than 1200 bps, the transaction timing rules of HART have to be changed. Multidrop operation is supported.

At the time of writing, this mode has not been accepted by the HART Communication Foundation.

2.18 Summary

HART uses a frequency-shift keyed (f.s.k.) signal to communicate at 1200 baud, superimposed at a low level on the 4 to 20 mA analogue signal. Having an average value of zero, the f.s.k. signal causes no interference with the analogue signal.

If analogue signalling is not required, up to 15 field devices can be connected in parallel on the same pair of wires in a multidrop system.

The transmitted signal levels and receiver sensitivity are specified in such a way as to allow for signal attenuation, but reduce the likelihood of interference and crosstalk.

To avoid excessive attenuation or distortion of the HART signal, a limit is placed on the cut-off frequency of the line. This can be considered as a 65 μs limit on the RC time constant of the components of the system, including the cable capacitance. Low-capacitance cable types allow longer cable lengths, up to about 2000 m.

Grounding of the signal loop, and the cable screen, must be done properly, avoiding multiple ground connections.

A HART-compatible transmitter has a restricted analogue signal bandwidth, to avoid interference with the communication signal.

The use of IS barriers requires extra consideration. Most suppliers offer HART-compatible barriers.

Alternative voltage-modulation and RS-485 physical layers are used by a few vendors for instruments having special requirements.

[This page intentionally blank]

CHAPTER 3. TRANSACTION PROCEDURE, CODING AND MESSAGE STRUCTURE

3.1 Introduction

This chapter describes the transaction procedure, character coding and message structure of the HART protocol. These correspond to layer 2 – the data-link layer – of the OSI protocol reference model.

3.2 Master-slave operation

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

3.3 Multimaster operation

The HART protocol allows for two active masters in a system, one "primary" and one "secondary". Usually, the primary master would be the control system or other main host device, and the secondary master would be either a hand-held communicator or a maintenance computer. The two masters have different addresses, so each can positively identify replies to its own command messages.

3.4 Transaction procedure

HART is a half-duplex protocol; after completion of each message, the f.s.k. carrier signal must be switched off, to allow the other station to transmit. The carrier control timing rules state that the carrier should be turned on not more than 5 bit times before the start of the message (that is, the preamble) and turned off not more than 5 bit times after the end of the last byte of the message (the checksum).

The master is responsible for controlling message transactions. If there is no reply to a command within the expected time, the master should retry the message. After a few retries, the master should abort the transaction, since presumably the slave device or the communication link has failed.

After each transaction is completed, the master should pause for a short time before sending another command, to allow an opportunity for the other master to break in if it wishes. In this way, two masters (if they are present) take turns at communicating with the slave devices.

Typical message lengths and delays allow two transactions per second.

Table 3-1 below gives a simplified summary of these and other timing rules. Refer to the full HART documentation for complete specifications covering all circumstances.

3.5 Burst mode

To achieve a higher data rate, some field devices implement an optional "burst mode". When switched into this mode, a slave device repeatedly sends a data message, as though it had received a specific command to do so. Special commands (#107, #108, #109) are used to start and stop this mode of operation, and to choose which command should be assumed. (If burst mode is implemented, Commands #1, #2 and #3 must be supported; other commands are optional.) There is a short pause after each "burst" message, to allow a master device to send a command to stop the burst mode operation, or to initiate any other single transaction (after which burst messages will continue).

Generally, burst mode is only useful if there is just one field device attached to a pair of wires (since only one field device on a loop can be in burst mode at any one time). In burst mode, more than three messages can be transmitted per second.

Table 3-1 includes a simplified summary of the burst mode timing rules. Refer to the full HART documentation for complete specifications covering all circumstances.

Table 3-1. Summary of timing rules.

Device and message type	Time interval	
Unsynchronised primary master sends a command Unsynchronised secondary master sends a command Unsynchronised bursting slave bursts	≥ 305 ms ≥ 380 ms ≥ 305 ms	after continuous quiet on the bus
Synchronised master sends a command	20* - 75 ms ≥ 75 ms	after a response to the other master after a response to itself
Non-bursting slave responds to a command	0 - 256 ms	after the command
Synchronised bursting slave bursts	75 - 256 ms 0 - 20 ms	after its previous burst message after its response to the initial "enter burst mode" command, or after the response to any interposed command

Notes: Intervals are timed from the end of the checksum character (not from the end of the carrier).

When first connected to the bus, a device is "unsynchronised". It becomes "synchronised" when it has been monitoring bus activity and has recognised the type and end of a previous message.

If there is no response to a command, the bus again becomes "unsynchronised".

* A master need not wait 20 ms, following a *burst* message addressed to the other master (see 3.11 below).

3.6 Character coding

HART messages are coded as a series of 8-bit characters or "bytes". These are transmitted serially, using a conventional UART (Universal Asynchronous Receiver/Transmitter) function to serialize each byte, adding a start bit, an odd parity bit and a stop bit. These allow the receiving UART to identify the start of each character, and to detect bit errors due to electrical noise or other interference. The bit sequence for a complete character is shown in Figure 3-1. The least-significant data bit D0 is sent first.

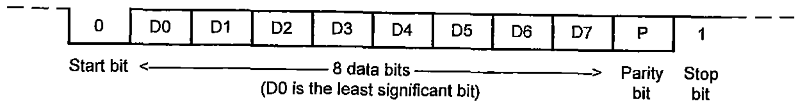


Figure 3-1. Character format

(Note that the serial port on an IBM-compatible PC cannot be set directly to this combination of 8 data bits plus parity, either by the DOS MODE command or by the IBM BASIC "OPEN COM" instruction. Most other programming languages do not have this problem. If necessary, the serial port can always be set up using low-level machine functions.)

Most asynchronous serial protocols allow inter-character periods at the idle signal level; however, inter-character gaps are not permitted in HART. This restriction is necessary, to meet the HART message timing specifications; indeed any gap longer than 1 byte-time may be detected as an error.

3.7 Message format

The HART message structure is repeated here in Figure 3-2 for convenient reference.

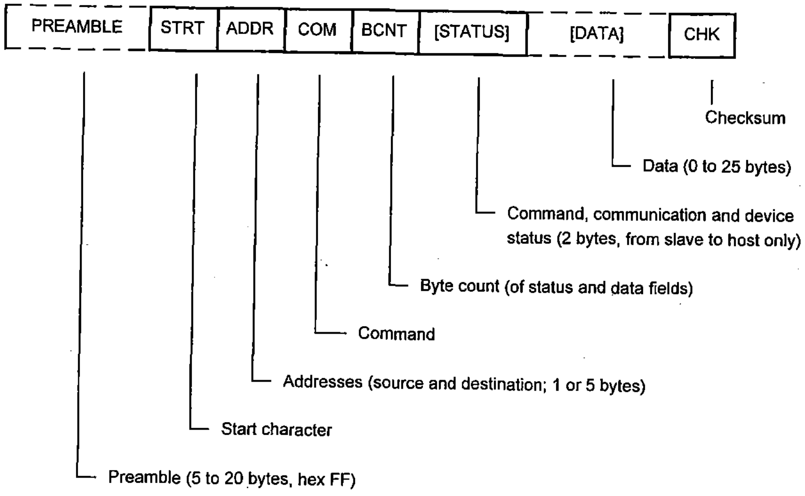


Figure 3-2. The HART message structure

The remainder of this chapter describes the Preamble, Start Character, Address, Byte Count and Checksum fields. See Chapter 4 for information on the Command, Status and Data fields. Complete example transactions are shown in Figures 3-5 and 3-6 below.

3.8 Long and short frame formats

Older HART instruments (up to and including HART Revision 4) always used a "short frame format". In this format, the address of the slave device is either 0, for non-multidropped devices using the 4-20 mA current signal for the measurement, or is in the range 1-15, for multidropped devices. This short address form is referred to as the "polling address".

HART Revision 5 introduced the "long frame format". In this, the address of a slave device is a worldwide-unique identifier, a 38-bit number derived from the manufacturer code, the device type code, and the device identification number. (Figure 3-6 shows the construction of the unique identifier.) This format gives extra security against the possible reception and acceptance of commands meant for another device, either due to external interference or due to excessive crosstalk in a badly-installed system. It also extends the addressing capability of the HART protocol to allow for larger networks (for example using a common radio link to a large number of remote field devices). Strictly, the unique identifier is not quite unique, since only the least-significant 6 bits of the 8-bit manufacturer code are included. In principle, there could be four devices with the same "unique identifier". (The HART Communication Foundation has issued recommendations on device type numbering which make this unlikely in practice.)

Most master devices should implement both long and short frame formats fully, so as to be able to deal correctly with existing field devices as well as new ones. Revision 5 (and later) field devices must always implement Command #0 ("Read unique identifier") in both frame formats. A master will normally use Command #0 in short frame format to identify a field device on first connection, when the unique identifier is not yet known. Since the reply to this command also includes the device's universal command revision level, the master can then determine which format to use for further commands to that field device. (See also 4.6 below.)

3.9 Preamble

The preamble consists of between five and twenty hexadecimal FF characters (all 1's). This allows the receiver to synchronize to the signal frequency and the incoming character stream, after initial detection of the HART signal, and also allows for any small delay in reversing the direction of transmission through the modem after an outgoing command.

A first attempt at communication, and any retries, should use 20 preamble characters, to have the best chance of success. The response to Command #0 tells a master how many preamble characters the slave would like to receive; a master can use Command #59 to tell the slave how many preambles to use in its replies.

3.10 Start character

The start character in a HART message has several possible values, indicating which frame format is being used, the source of the message, and whether this is a burst mode message. These are shown (*in hexadecimal*) in Table 3-2. When waiting for a message, receiving devices listen for any of these characters, as the first character after at least two FF characters, to indicate the start of the message.

Table 3-2. Start characters

Message type	Short frame	Long frame
Master to slave	02	82
Slave to master	06	86
Burst message from slave	01	81

These characters can be fully identified by the content of bits 0, 1, 2 and 7. It has been proposed that future enhancements to the HART protocol may use bits 5 and 6 of the Start character to indicate the presence of extra bytes between the Address and Command fields. However this has not yet been approved by the HART Communication Foundation.

3.11 Address

The address field contains both the master (host) and slave (field device) addresses for the message. These are contained in a single byte in the short frame format, or in five bytes in the long frame format.

In both formats, the most-significant bit is usually the single-bit address of the master device taking part in the transaction. Only two masters are allowed – for example a control system and a hand-held communicator. The most-significant bit of the address field distinguishes between these: primary masters (control systems or other permanently-connected hosts) use address 1, secondary masters use address 0. Burst messages are an exception – in these, the most-significant bit is set alternately to 0 and 1; this gives each master, in turn, an opportunity to interrupt the burst mode operation.

Also in both formats, the next-most-significant bit is set to 1 to indicate that this message comes from a field device in burst mode (which does not necessarily mean that this is itself a burst message).

In the short frame format, slave devices have polling addresses in the range 0 to 15. This number is included in binary form as the least-significant half of the single address byte. In the long frame format, the polling address is not used; instead, the remaining 38 bits of the five-byte address field hold the slave's "unique identifier" as an address. Figures 3-3 and 3-4 show the two address structures.

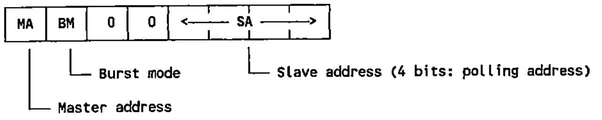


Figure 3-3. Short frame address structure

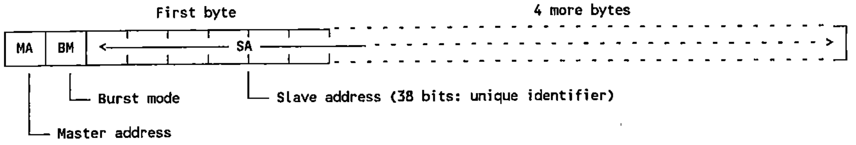


Figure 3-4. Long frame address structure

In the long frame format, 0 (38 zero bits) can be used as a broadcast address, for a message to be accepted by all slave devices. This is only possible if the data in the message determines which field device should reply; for example Command #11 ("Read unique identifier associated with tag") normally uses the broadcast address with a tag in the data field, so that all connected devices receive the message, but only the device with a matching tag replies.

3.12 Command

The command byte contains an integer (0 to hex FD or decimal 253) representing one of the HART commands. The received command code is echoed back by the slave device in its reply.

Chapter 4 gives details of many commands and their associated data.

3.13 Byte count

The byte count character contains an integer, the number of bytes which form the remainder of the message (that is, the status and data sections; the checksum byte is not included in this count). The receiving device uses this to identify the checksum byte and to know when the message is complete.

Because the data field is limited to 25 bytes maximum (see 3.15 below), the byte count is in the range 0 to 27.

3.14 Status

Status (also referred to as the "response code") is included only in reply messages from a slave. It consists of two bytes, reporting any outgoing communication errors, the status of the received command (such as that the device is busy, or does not recognise the command), and the operational state of the slave device.

The coding and meaning of status information is described in 4.14 below.

3.15 Data

Not all commands or responses contain data. For those that do, to conform to the overall transaction timing rules, the data field can never be more than 25 bytes. (It has been suggested that this limit should be relaxed for RS-485 HART, since higher communication speeds will generally be used.)

Data may be in the form of unsigned integers, floating point numbers or ASCII character strings. The number of bytes of data, and the data format used for each item, are specified for each command. Refer to Chapter 4 for more details.

3.16 Checksum

The checksum byte contains the exclusive-or ("longitudinal parity") of all the bytes which precede it in the message, starting with the "start" character. This provides a further check on transmission integrity, beyond that provided by the parity check on the 8 bits of each individual byte. The combination guarantees to detect any single burst of up to three corrupted bits in a message, and has an excellent chance of detecting longer or multiple bursts.

3.17 Example transactions

Figures 3-5 and 3-6 show examples of short frame and long frame transactions, with the meaning of each field explained. Within each message, byte values are shown in hexadecimal, with address fields further decomposed into binary to show their component parts.

3.18 Summary

HART is a master-slave protocol, with up to two active masters (a control system and a hand-held communicator, for example). Up to 15 slave devices can be connected to a single multidrop pair of wires.

Each message includes the addresses of its source and its destination. Two forms of slave addressing are used: a short form for older devices (and for initial device identification), and a long form, based on the unique identifier, for newer (HART Revision 5) devices.

The field device's status is included in every response message, indicating its continued good health. Parity checking and the checksum allow corruption of the message itself to be detected.

Two transactions can be made each second. In burst mode, three messages are sent each second.

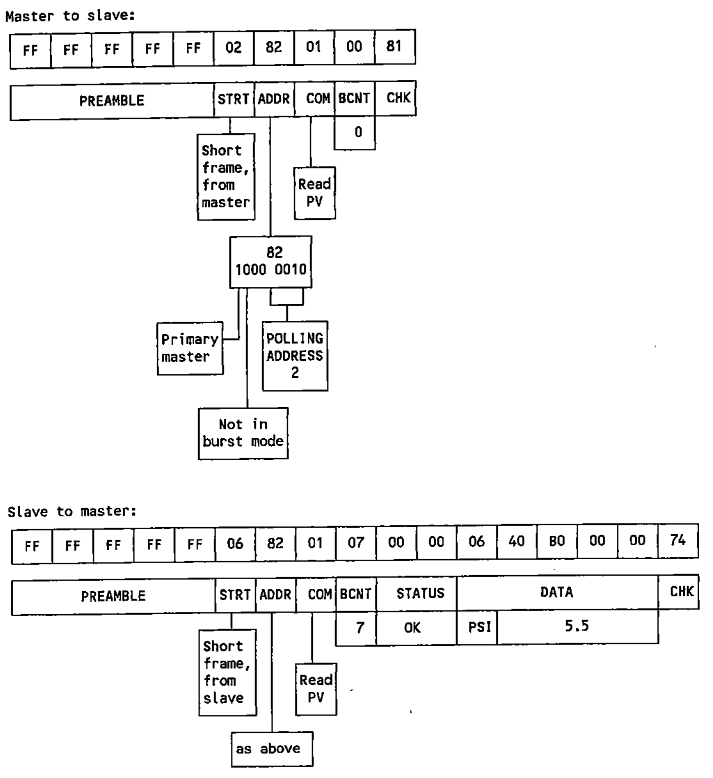


Figure 3-5. A short frame format transaction

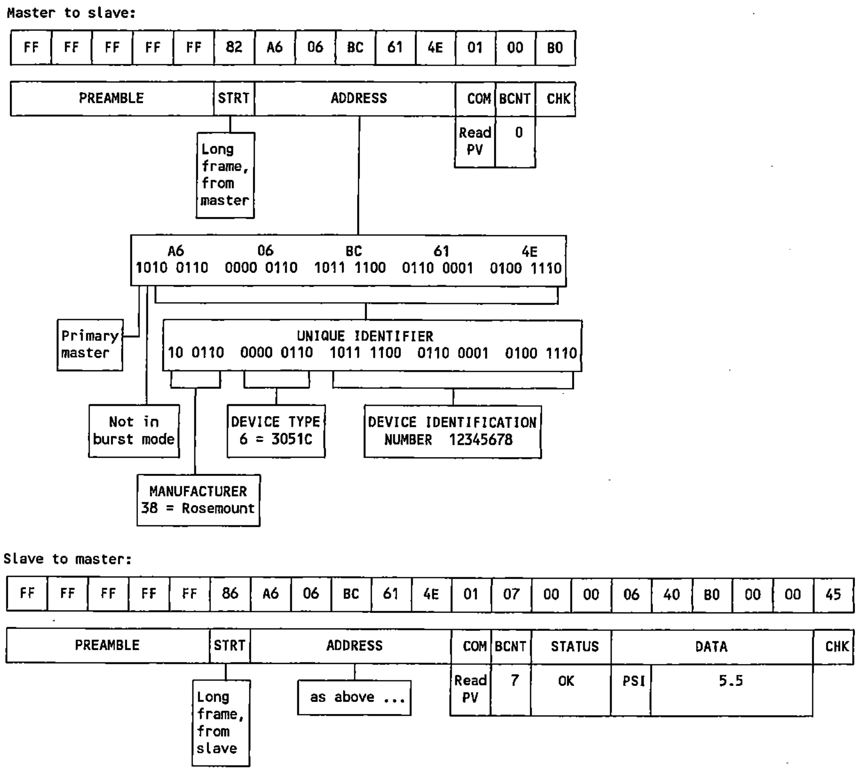


Figure 3-6. A long frame format transaction

[This page intentionally blank]

CHAPTER 4. COMMANDS, DATA AND STATUS

4.1 Introduction

This chapter describes the classification of HART commands, and gives details of the data structures used with many of them. The coding and meaning of HART status information is also described. This corresponds to layer 7 – the application layer – of the OSI protocol reference model.

Refer back to Figure 3-2 for the overall structure of a HART message. This chapter is concerned with the Command, Status and Data fields. (See Chapter 3 for information on the other fields.)

4.2 Commands

The command byte contains an integer (0 to hex FD or decimal 253), representing one of the HART commands. A few numbers (31, 127, 254 and 255) are reserved. "254" may become the basis of an expansion mechanism, if more command numbers are needed in future.

HART commands are defined in three groups: "universal", "common-practice" and "device-specific".

4.3 Universal commands

"Universal commands" are in the range 0 to 30. They provide functions which are implemented in all HART-conformant field devices. Table 4-1 summarises their functions. For more detail, see sections 4.6 to 4.9 and 4.13, especially Table 4-4, where the data structure for each command is shown. Some of these commands were different in earlier Revisions of HART; Table 4-5 shows those differences.

Table 4-1. Universal commands (summary)

Commands	Function
0, 11	Identify device (manufacturer, device type, revision levels)
1, 2, 3	Read measured variables
6	Set polling address (and multidrop mode)
12, 13, 17, 18	Read and write user-entered text information (tag, descriptor, date, message)
14, 15	Read device information (sensor serial number, sensor limits, alarm operation, range values, transfer function, damping time constant)
16, 19	Read and write final assembly number

4.4 Common-practice commands

"Common-practice commands" are in the range 32 to 126. They provide functions common to many field devices, but not all. If these functions are implemented in a device, these commands should be used to perform them. Table 4-2 summarises their functions; for more detail, see Table 4-5, where the data structure for each command is shown.

Table 4-2. Common-practice commands (summary)

Commands	Function
33, 61, 110	Read measured variables
34-37, 44, 47	Set operating parameters (range, damping time, PV units, transfer function)
38	Reset "configuration changed" flag
39	EEPROM control
40-42	Diagnostic functions (fixed current mode, self test, reset)
43, 45-46	Analogue input/output trim
48	Read additional device status
49	Write sensor serial number
50-56	Use of transmitter variables
57-58	Unit information (tag, descriptor, date)
59	Write number of preambles required
60, 62-70	Use of multiple analogue outputs
107-109	Burst mode control

Common-practice Commands #123 to #126 are "non-public". They are typically used by manufacturers to enter device-specific information during assembly, for example the device identification number, which will never be altered by users, or for direct memory read and write commands. Often, a password is needed to activate these commands.

4.5 Device-specific commands

"Device-specific commands" are in the range 128 to 253. They provide functions which are more or less unique to a particular field device. (Some documents refer to these as "transmitter-specific", but the term "device-specific" is to be preferred.) Table 1-3 showed some examples of device-specific commands.

In HART Revision 4 and earlier, device-specific commands always included the Device Type Code as the first byte of the data field, to ensure that a command never reached an incompatible device. This practice was dropped with HART Revision 5, since the use of Unique Identifiers now guarantees that the host has fully identified the field device before any other command can be sent.

Device-specific commands are not considered further in this booklet. Refer to the manufacturer's device-specific documentation for further information on the commands for any particular instrument.

4.6 Commands #0 and #11

Commands #0 and #11 are used to identify a field device. Since HART Revision 5, all commands use the long frame format, but Command #0 must also be accepted, and responded to, in the old short frame format. This lets a HART master identify an unknown field device, without previously knowing its Unique Identifier. The data in the reply to Command #0 includes the manufacturer identification code, the device type code, and the device ID number. From these, the master can build up the device's Unique Identifier, for subsequent use in long frame commands.

Referring to Tables 4-4 and 4-6, notice the difference in the data structures for Command #0 in earlier revisions. In Revision 4, the original "transmitter type code" is optionally split into two bytes: the manufacturer code and the device type code. This option is indicated by "254" ("expansion") in the first data byte, and the remaining bytes are moved up by two positions. In Revision 5, the expanded version is mandatory; in addition, the final assembly number is replaced by the device identification (ID) number. All HART masters must deal with all these cases, if they want to work with Revision 4 or earlier devices. (In a future HART Revision, it is proposed to add four further data bytes to the response to these commands, to identify the device's HART and functional specifications more exactly: common-practice command revision, common tables revision, data link revision and device family code.)

A master will commonly begin communication by using Command #0, with a polling address of 0, then perhaps scanning up from 1 to 15 if multidrop operation is expected. Alternatively, if the field devices are Revision 5 or later, and the tag(s) of the connected device(s) are already known (as they may well be, in a control system), the master can use Command #11, with a long-frame broadcast address of all 0's, and with the tag as data in the command. A field device will then respond only if its tag matches. The data in the reply to Command #11 is identical to that of Command #0, so the master can then construct the Unique Identifier in the usual way, for use in further commands.

4.7 Commands #1, #2 and #3

These commands are used to read measured variables in various forms. Commands #2 and #3 include the actual output current in mA. Like the real analogue output, this mA value represents the primary variable (PV) only when it is within the configured output range, not when the device is in multidrop mode, and not when the output is otherwise fixed, saturated or indicating a device fault by an out-of-range value. However, the PV and other dynamic variables returned in engineering units by these commands are *not* limited by the configured output range, but continue to follow the measurement out to the sensor limits. The percent of range value returned by Command #2 also follows the measurement out to the sensor limits, so can take values below 0% and above 100%.

Common-practice Command #61 is an equivalent to Command #3, for use with instruments having an analogue output other than current. Command #110 also returns the dynamic variables (without the analogue output signal level). Command #33 provides any selection of up to four transmitter variables. For multiple-output devices, Command #60 reads any selected analogue output signal level (and its percent of range), and Command #62 provides any selection of up to four analogue output levels. (See also section 4.12 for more on multivariable devices.)

4.8 Command #6

Command #6 sets the polling address of a field device. Set to 0, the device works normally in point-to-point mode, generating an analogue output signal. Set to any value from 1 to 15, the device is switched into multidrop mode, and its output current is fixed at 4 mA.

4.9 Commands #12 to #19

Commands #12 to #19 are used to read and write a selection of device information. Refer to Table 4-4 for details. In HART Revision 4 and earlier, these commands did not exist. Instead their functions were provided by Commands #4 and #5, with "block numbers" (0 to 4) used to select particular sets of data. Table 4-5 shows the data formats for these old commands.

4.10 Data

Not all commands or responses contain data. For those that do, up to 25 bytes can be included. Data may be represented as

- integers – 8, 16, 24 or 32 bits, unsigned
- floating point numbers – IEEE 754 single-precision floating point format
- ASCII character strings – usually packed 4 characters into each 3 bytes
- enumerated items from a standard list – coded as 8-bit integers.

Refer to the Glossary for details of these data formats.

If a command is unsuccessful (indicated by an "error" status), the response should not contain data. The response to a successful "write" or "command" command always includes the same set of variables as were contained in the command message; however, the values in the response are those actually used, taken from the field device's memory, so as to indicate any approximation involved.

The number of bytes of data, and the data format used for each item, are specified for each command (though a few are allowed to be truncated after the last data item used in the particular device). Refer to Tables 4-4 to 4-6 for details.

4.11 Enumerated items

Data items for which a choice is made from a list of alternatives are coded as a number corresponding to each alternative. Table 4-3 shows some of the standard enumerated lists defined in the HART specification. There are also many device-specific lists, for example for special materials or function options. Refer to the full HART specification and manufacturers' device documentation for more information.

Table 4-3. Enumerated variables

Variable	Values
Manufacturer identification	1 - 249, allocated by the HART Communication Foundation[1]
Device type	0 - 249, allocated by each manufacturer
Units	0 - 249: 6 = psi, 7 = bar, 32 = C, 33 = F, etc. etc.[1, 2]
Transfer function	0 = linear, 1 = square root, etc.[1]
Material	0 - 249: 2 = 316 stainless steel, 10 = PTFE, 18 = ceramic, etc. etc.[1]
Alarm selection	0 = low, 1 = high, 239 = hold last output value
Write protect	0 = not write-protected, 1 = write-protected
Burst mode control	0 = exit burst mode, 1 = enter burst mode
Physical signalling	0 = Bell 202 current, 1 = Bell 202 voltage, 2 = RS-485, 3 = RS-232

Notes: 1. Refer to the full HART specification for complete lists.

2. Individual codes are allocated for many combinations of fundamental units and their multiples, including metric, "imperial" and industry-specific units, such as kg/sec, imperial gallons/hour and barrels/day. Over 140 different unit codes have so far been allocated.

4.12 Multivariable transmitters

Some transmitters (and indeed, output devices) measure more than one physical quantity. There are HART commands which allow up to four measurements to be returned in a single message. In some multivariable devices, the set of measurements is predefined, but in others the user can select from the set of up to 250 "transmitter variables" defined for that instrument.

Common-practice Commands #50 to #56 are related to these transmitter variables, their sensors and ranges. In particular, in devices which support it, Command #51 allows the selection of transmitter variables for the primary, secondary, third (or tertiary) and fourth variables (PV, SV, TV and FV). These can then be read using Command #3. Alternatively, Command #33 specifies up to four transmitter variables to be included in the immediate reply message. See Table 4-6 for details of these commands.

Multivariable transmitters may also have the capability of generating more than one analogue output. By definition, analogue outputs numbered 1 to 4 represent the HART "dynamic variables" (PV, SV, TV and FV) respectively. (These are not necessarily all 4-to-20 mA current signals; in this respect, for example, even a frequency signal counts as analogue.) Common-practice Commands #60 and #62 to #70 are concerned with the configuration and control of these outputs. See Table 4-6 for details of these commands.

The numbering of transmitter variables may start at 0 or 1, according to the manufacturer's preference.

4.13 Command summary

Table 4-4 lists the functions and data structures for universal commands in HART Revision 5. Table 4-5 shows the differences in universal commands in earlier revisions of HART. (Host devices should implement these if they need to operate with older field devices.) Table 4-6 lists common-practice commands in HART Revision 5, and includes notes (n.n) of the revision in which certain features were introduced.

Notes: In these tables, data types are indicated as follows:

A	ASCII string (packed 4 characters per 3 bytes)
B	Bit-mapped flags
D	Date (3 bytes: day, month, year-1900)
F	Floating point (4 bytes IEEE 754)
H	Integers xxxxx yyy (xxxxx = hardware revision, yyy = physical signalling code)

Unmarked items are 8-, 16- or 24-bit integers (including enumerated code values).

Table 4-4. Universal commands in HART Revision 5

Command number and function	Data in command	(type)	Data in reply	(type)
0 Read unique identifier	none		Byte 0 "254" (expansion)	
			Byte 1 manufacturer identification code	
			Byte 2 manufacturer's device type code	
			Byte 3 number of preambles required	
			Byte 4 universal command revision	
			Byte 5 device-specific command revision	
			Byte 6 software revision	
			Byte 7 hardware revision	(H)
			Byte 8 device function flags*	(B)
			Byte 9-11 device ID number	
			Byte 12 ** common-practice command revision	
			Byte 13 ** common tables revision	
			Byte 14 ** data link revision	
			Byte 15 ** device family code	
			* Bit 0 = multisensor device; bit 1 = EEPROM control required; bit 2 = protocol bridge device. ** Proposed for a future HART revision – *not* in 5.3.	
1 Read primary variable	none		Byte 0 PV units code	
			Byte 1-4 primary variable	(F)
2 Read current and percent of range	none		Byte 0-3 current (mA)	(F)
			Byte 4-7 percent of range	(F)
3 Read current and four (predefined) dynamic variables	none		Byte 0-3 current (mA)	(F)
			Byte 4 PV units code	
			Byte 5-8 primary variable	(F)
			Byte 9 SV units code	
			Byte 10-13 secondary variable	(F)
			Byte 14 TV units code	
			Byte 15-18 third variable	(F)
			Byte 19 FV units code	
			Byte 20-23 fourth variable	(F)
			(truncated after last supported variable)	
6 Write polling address	Byte 0 polling address		as in command	
11 Read unique identifier associated with tag	Byte 0-5 tag	(A)	Byte 0-11 as Command #0	
12 Read message	none		Byte 0-23 message (32 characters)	(A)
13 Read tag, descriptor, date	none		Byte 0-5 tag (8 characters)	(A)
			Byte 6-17 descriptor (16 characters)	(A)
			Byte 18-20 date	(D)
14 Read PV sensor information	none		Byte 0-2 sensor serial number	
			Byte 3 units code for sensor limits & min. span	
			Byte 4-7 upper sensor limit	(F)
			Byte 8-11 lower sensor limit	(F)
			Byte 12-15 minimum span	(F)
15 Read output information	none		Byte 0 alarm select code	
			Byte 1 transfer function code	
			Byte 2 PV/range units code	
			Byte 3-6 upper range value	(F)
			Byte 7-10 lower range value	(F)
			Byte 11-14 damping value (seconds)	(F)
			Byte 15 write-protect code	
			Byte 16 private-label distributor code	
16 Read final assembly number	none		Byte 0-2 final assembly number	
17 Write message	Byte 0-23 message (32 chars)	(A)	as in command	
18 Write tag, descriptor, date	Byte 0-5 tag (8 characters)	(A)	as in command	
	Byte 6-17 descriptor (16 chars)	(A)		
	Byte 18-20 date	(D)		
19 Write final assembly number	Byte 0-2 final assembly number		as in command	

Table 4-5. Universal commands in HART Revisions 2, 3 and 4 (differences from Revision 5)

Command number and function	Data in command (type)	Data in reply (type)
0 Read unique identifier	none	Byte 0 transmitter type code * Byte 1 number of preambles Byte 2 universal command revision Byte 3 device-specific command revision Byte 4 software revision Byte 5 hardware revision (H) Byte 6 device function flags (B) Byte 7-9 final assembly number * Revision 4 introduced the expanded device type as an option (see Rev. 5, Table 4-4), with the remaining bytes moved up by two positions.
4 Read common static data (block 0): Read message	Byte 0 block number ("0")	Byte 0 block number ("0") Byte 1-24 message (A)
4 Read common static data (block 1): Read tag, descriptor, date	Byte 0 block number ("1")	Byte 0 block number ("1") Byte 1-6 tag (A) Byte 7-18 descriptor (A) Byte 19-21 date (D) Byte 22-24 "250"
4 Read common static data (block 2): Read sensor information	Byte 0 block number ("2")	Byte 0 block number ("2") Byte 1-3 sensor serial number Byte 4 units code for sensor limits & min. span Byte 5-8 upper sensor limit (F) Byte 9-12 lower sensor limit (F) Byte 13-16 minimum span (F) Byte 17-24 "250"
4 Read common static data (block 3): Read output information	Byte 0 block number ("3")	Byte 0 block number ("3") Byte 1 alarm select code Byte 2 transfer function code Byte 3 PV/range units code Byte 4-7 upper range value (F) Byte 8-11 lower range value (F) Byte 12-15 damping value (seconds) (F) Byte 16 write-protect code ("1" = protected) * Byte 17 private-label distributor code ** Byte 18-24 "250" * "250"or "251" in Revisions 2 and 3. ** "250" in Revisions 2 and 3.
5 Write common static data (block 0): Write message	Byte 0 block number ("0") Byte 1-24 message (A)	as in command
5 Write common static data (block 1): Write tag, descriptor, date	Byte 0 block number ("1") Byte 1-6 tag (A) Byte 7-18 descriptor (A) Byte 19-21 date (D) Byte 22-24 "250"	as in command
5 Write common static data (block 4): Write final assembly number	Byte 0 block number ("4") Byte 1-3 final assembly number Byte 4-24 "250"	as in command
11 - 19	These commands did not exist before Revision 5.0	

Table 4-6. Common-practice commands

Command number and function	Data in command (type)	Data in reply (type)
33 Read transmitter variables	Byte 0 transm. var. code for slot 0 Byte 1 transm. var. code for slot 1 Byte 2 transm. var. code for slot 2 Byte 3 transm. var. code for slot 3 (truncated after last requested code)	Byte 0 transm. variable code for slot 0 Byte 1 units code for slot 0 Byte 2-5 variable for slot 0 (F) Byte 6 transm. variable code for slot 1 Byte 7 units code for slot 1 Byte 8-11 variable for slot 1 (F) Byte 12 transm. variable code for slot 2 Byte 13 units code for slot 2 Byte 14-17 variable for slot 2 (F) Byte 18 transm. variable code for slot 3 Byte 19 units code for slot 3 Byte 20-23 variable for slot 3 (F) (truncated after last requested variable)
34 Write damping value	Byte 0-3 damping value (seconds) (F)	as in command
35 Write range values	Byte 0 range units code Byte 1-4 upper range value (F) Byte 5-8 lower range value (F)	as in command
36 Set upper range value (= push SPAN button)	none	none
37 Set lower range value (= push ZERO button)	none	none
38 Reset "configuration changed" flag	none	none
39 EEPROM control	Byte 0 EEPROM control code* * 0 = burn EEPROM, 1 = copy EEPROM to RAM.	as in command
40 Enter/exit fixed current mode	Byte 0-3 current (mA)* (F) * 0 = exit fixed current mode.	as in command
41 Perform device self-test	none	none
42 Perform master reset	none	none
43 Set (trim) PV zero	none	none
44 Write PV units	Byte 0 PV units code	as in command
45 Trim DAC zero	Byte 0-3 measured current (mA) (F)	as in command
46 Trim DAC gain	Byte 0-3 measured current (mA) (F)	as in command
47 Write transfer function	Byte 0 transfer function code	as in command
48 Read additional device status	none	Byte 0-5 device-specific status (B) Byte 6-7 operational modes (5.1) Byte 8-10 analogue outputs saturated* (5.1) (B) Byte 11-13 analogue outputs fixed* (5.1) (B) Byte 14-24 device-specific status (B) * 24 bits each: LSB ... MSB refers to AO #1 ...# 24. (Response is truncated after last byte implemented)

Command number and function	Data in command	(type)	Data in reply	(type)
49 Write PV sensor serial number	Byte 0-2 sensor serial number		as in command	
50 Read dynamic variable assignments (4.1)	none		Byte 0 PV transmitter variable code Byte 1 SV transmitter variable code Byte 2 TV transmitter variable code Byte 3 FV transmitter variable code	
51 Write dynamic variable assignments (4.1)	Byte 0 PV transmitter variable code Byte 1 SV transmitter variable code Byte 2 TV transmitter variable code Byte 3 FV transmitter variable code		as in command	
52 Set transmitter variable zero (4.1)	Byte 0 transmitter variable code		as in command	
53 Write transmitter variable units (4.1)	Byte 0 transmitter variable code Byte 1 transm. variable units code		as in command	
54 Read transmitter variable information (4.1)	Byte 0 transmitter variable code		Byte 0 transmitter variable code Byte 1-3 transm. var. sensor serial number Byte 4 transm. var. limits units code Byte 5-8 transm. variable upper limit Byte 9-12 transm. variable lower limit Byte 13-16 transm. var. damping value (sec.) Byte 17-20 transm. var. minimum span (5.0)	 (F) (F) (F) (F)
55 Write transmitter variable damping value (4.1)	Byte 0 transmitter variable code Byte 1-4 transmitter variable damping value (seconds)	 (F)	as in command	
56 Write transmitter variable sensor serial number (4.1)	Byte 0 transmitter variable code Byte 1-3 transmitter variable sensor serial number		as in command	
57 Read unit tag, descriptor, date (5.0)	none		Byte 0-5 unit tag (8 characters) Byte 6-17 unit descriptor (16 characters) Byte 18-20 unit date	(A) (A) (D)
58 Write unit tag, descriptor, date (5.0)	Byte 0-5 unit tag (8 characters) Byte 6-17 unit descriptor (16 chars) Byte 18-20 unit date	(A) (A) (D)	as in command	
59 Write number of response preambles (5.0)	Byte 0 number of response preambles		as in command	
60 Read analogue output and percent of range (5.1)	Byte 0 analogue out. number code		Byte 0 analogue output number code Byte 1 analogue output units code Byte 2-5 analogue output level Byte 6-9 analogue output percent of range	 (F) (F)

Command number and function	Data in command	(type)	Data in reply	(type)
61 Read dynamic variables and PV analogue output (5.1)	none		Byte 0 PV analogue output units code Byte 1-4 PV analogue output level Byte 5 PV units code Byte 6-9 Primary variable Byte 10 SV units code Byte 11-14 Secondary variable Byte 15 TV units code Byte 16-19 Third variable Byte 20 FV units code Byte 21-24 Fourth variable (truncated after last supported variable)	 (F) (F) (F) (F) (F)
62 Read analogue outputs (5.1)	Byte 0 analogue output number code for slot 0 Byte 1 analogue output number code for slot 1 Byte 2 analogue output number code for slot 2 Byte 3 analogue output number code for slot 3 (truncated after last requested code)		Byte 0 slot 0 analogue output number code Byte 1 slot 0 units code Byte 2-5 slot 0 level Byte 6 slot 1 analogue output number code Byte 7 slot 1 units code Byte 8-11 slot 1 level Byte 12 slot 2 analogue output number code Byte 13 slot 2 units code Byte 14-17 slot 2 level Byte 18 slot 3 analogue output number code Byte 19 slot 3 units code Byte 20-23 slot 3 level (truncated after last requested level)	 (F) (F) (F) (F)
63 Read analogue output information (5.1)	Byte 0 analogue output number code		Byte 0 analogue output number code Byte 1 analogue output alarm select code Byte 2 analogue out. transfer function code Byte 3 analogue output range units code Byte 4-7 analogue output upper range value Byte 8-11 analogue output lower range value Byte 12-15 analogue output additional damping value (sec)	 (F) (F) (F)
64 Write analogue output additional damping value (5.1)	Byte 0 analogue out. number code Byte 1-4 analogue output additional damping value (sec)	 (F)	as in command	
65 Write analogue output range values (5.1)	Byte 0 analogue out. number code Byte 1 an. out. range units code Byte 2-5 an. out. upper range value Byte 6-9 an. out. lower range value	 (F) (F)	as in command	
66 Enter/exit fixed analogue output mode (5.1)	Byte 0 analogue out. number code Byte 1 analogue output units code Byte 2-5 analogue output level* * "not a number" exits fixed output mode	 (F)	as in command	
67 Trim analogue output zero (5.1)	Byte 0 analogue out. number code Byte 1 analogue out. units code Byte 2-5 externally-measured analogue output level	 (F)	as in command	
68 Trim analogue output gain (5.1)	Byte 0 analogue out. number code Byte 1 analogue out. units code Byte 2-5 externally-measured analogue output level	 (F)	as in command	

Command number and function	Data in command	(type)	Data in reply	(type)
69 Write analogue output transfer function (5.1)	Byte 0 analogue out. number code Byte 1 an. out. transfer funct. code		as in command	
70 Read analogue output endpoint values (5.1)	Byte 0 analogue out. number code		Byte 0 analogue output number code Byte 1 analogue out. endpoint units code Byte 2-5 analogue out. upper endpoint value Byte 6-9 analogue out. lower endpoint value	 (F) (F)
107 Write burst mode transmitter variables (for Command #33) (5.1)	Byte 0 transm. var. code for slot 0 Byte 1 transm. var. code for slot 1 Byte 2 transm. var. code for slot 2 Byte 3 transm. var. code for slot 3		as in command	
108 Write burst mode command number (5.0)	Byte 0 burst mode command number		as in command	
109 Burst mode control (5.0)	Byte 0 burst mode control code (0 = exit, 1 = enter)		as in command	
110 Read all dynamic variables (5.0)	none		Byte 0 PV units code Byte 1-4 PV value Byte 5 SV units code Byte 6-9 SV value Byte 10 TV units code Byte 11-14 TV value Byte 15 FV units code Byte 16-19 FV value	 (F) (F) (F) (F)

4.14 Status

Two bytes of "status", also known as "response code", are included in every message from a field (slave) device. Between them, these two bytes convey three different types of information: communication errors, command responses, and field device status.

Table 4-7 shows how this is done. If an error is detected in the outgoing communication, the most-significant bit (bit 7) of the first byte is set to 1, and the details of the error are reported in the rest of that byte; the second byte is then all zeros. Otherwise, if communication was good, bit 7 of the first byte is 0, the remainder of the byte contains the "command response", indicating any problem with the received command, and the second byte contains the "field device status", indicating the operational state of the slave device.

Communication errors are mostly those which would be detected by a UART: parity, overrun and framing errors. In addition, the field device reports overflow of its receive buffer, and any discrepancy between the message content and the received checksum.

Command response codes (integers in the range 0 to 127) are categorised as either errors or warnings, and as having either a single meaning or multiple meanings. Table 4-8 shows the ranges specified for each type. (The rather strange allocations are the result of maintaining compatibility with earlier revisions of HART.) Table 4-7 shows command response codes which may be applicable to any transaction. Table 4-9 shows those which have multiple meanings specific to particular universal or common-practice commands; device-specific commands may re-use the same meanings for these codes, or define their own.

Field device status includes both fault conditions and abnormal operational modes or conditions, so does not necessarily imply a faulty device. Table 4-7 shows the decoding of this byte, and Table 4-10 describes the meaning of each of the reported conditions.

Table 4-7. Response codes

First byte:

Bit 7 = 1: Communication error		
Bit 6	(hex C0)	parity error
Bit 5	(hex A0)	overrun error
Bit 4	(hex 90)	framing error
Bit 3	(hex 88)	checksum error
Bit 2	(hex 84)	0 (reserved)
Bit 1	(hex 82)	rx buffer overflow
Bit 0	(hex 81)	(undefined)

or

Bit 7 = 0: Command response	
Bits 6 to 0 (decoded as an integer, not bit-mapped):	
0	no command-specific error
1	(undefined)
2	invalid selection
3	passed parameter too large
4	passed parameter too small
5	too few data bytes received
6	device-specific command error (rarely used)
7	in write-protect mode
8-15	multiple meanings (see Table 4-9)
16	access restricted
28	multiple meanings (see Table 4-9)
32	device is busy
64	command not implemented

Second byte:

(not used)	
Bit 7	
Bit 6	
Bit 5	all bits 0
Bit 4	(when a communication
Bit 3	error is reported in the first
Bit 2	byte)
Bit 1	
Bit 0	

or

Field device status		
Bit 7	(hex 80)	field device malfunction
Bit 6	(hex 40)	configuration changed
Bit 5	(hex 20)	cold start
Bit 4	(hex 10)	more status available
Bit 3	(hex 08)	analogue output current fixed
Bit 2	(hex 04)	analogue output saturated
Bit 1	(hex 02)	non-primary variable out of limits
Bit 0	(hex 01)	primary variable out of limits

Note: Hexadecimal equivalents are quoted assuming only a single bit is set. In reality, several bits may be set simultaneously, and the hex digits can be or'ed together.

Many devices offer more status information than can be coded into this single byte. For these, bit 4 of the device status byte may be set, indicating "more status available"; Command #48 can then be used to read the additional information. Originally, the use of data bytes in the reply to Command #48 was left open for device designers to specify, but since HART Revision 5.1, bytes 6 to 13 have specific meanings, indicating operating modes (as yet undefined) and the status of multiple analogue outputs (see Table 4-6). The remaining bytes can be allocated as the device designer wishes; they are typically bit-mapped to indicate individual conditions.

Table 4-8. Command response classification

	Error	Warning
Single-meaning	1 - 7 16 - 23 32 - 64	24 - 27 96 - 111
Multiple-meaning	9 - 13 15 28, 29 65 - 95	8 14 30, 31 112 - 127

Table 4-9. Multiple-meaning command response codes

Code	Commands	Alternative meanings
8*	1, 2, 3, 33, 60, 61, 62, 110 34, 55, 64 48	Update failure Set to nearest possible value Update in progress
9	35, 65 36, 37, 43, 52 45, 46, 67, 68	Lower range value too high Applied process too high Not in proper current mode (fixed at 4 mA or 20 mA)
10	6 35, 65 36, 37, 43, 52	Multidrop not supported (Revision 4 and earlier) Lower range value too low Applied process too low
11	35, 65 40, 45, 46, 66, 67, 68 53	Upper range value too high In multidrop mode Invalid transmitter variable code
12	35, 65 53, 66, 67, 68	Upper range value too low Invalid units code
13	35, 65 69	Both range values out of limits Invalid transfer function code
14*	35, 36, 65 37	Span too small Pushed upper range value over limit
15	65, 66, 67, 68, 69	Invalid analogue output number code
28	65	Invalid range units code

Note: * Codes 8 and 14 are classified as "warnings"; the remainder are "errors".

Table 4-10. Field device status

Status indication	Meaning
Field device malfunction	Measurements should not be trusted
Configuration changed	Set whenever a configuration change is made by any host, or through a local operator interface. A primary master should recognise this bit as a warning to re-read any cached configuration information, and should then clear the bit by sending Command #38
Cold start	Set, for one transaction only, when a field device is powered up
More status available	Use Command #48 to retrieve further information
Analogue output fixed	The device is in multidrop mode, or the output has been set at a fixed value for testing. This bit applies only to Analogue Output #1. In a multi-output device, Command #48 may return similar status information for the other outputs
Analogue output saturated	The measurement (for Analogue Output #1) is out of range. A linear overrange band, typically -0.63% (3.9 mA) to +105% (20.8 mA) is often allowed, before this status bit is set. In a multi-output device, Command #48 may return similar status information for the other outputs
Primary variable out of limits	The primary measurement is outside the sensor operating limits. So, not only is the analogue signal unreliable, but so is the digital value read by HART commands
Non-primary variable out of limits	As above, but for one or more other variables. There is no way to identify the specific variable (unless Command #48 gives further information)

4.15 Summary

HART commands are defined in three groups: "universal", "common-practice" and "device-specific".

Data is represented as integers, floating point numbers, ASCII text strings or enumerated item lists.

Commands to identify a field device, to read process variables, to set multidrop operation, and to handle multivariable devices, are defined. Data structures for all universal and common-practice commands are shown in the tables.

Messages from a field device include status ("response code") information relating to communication errors, the command transaction and the condition of the device itself.

CHAPTER 5. DEVICE DESCRIPTION LANGUAGE

5.1. Introduction

The HART "Device Description Language" (DDL) solves a number of problems relating to the introduction and use of new smart devices. The use of a common communication protocol is not enough to ensure useful communication, as different instruments are likely to have individual variations on the data available, its use and its meaning. In the past, this has meant that a host device (handheld communicator, control system or instrument management system) has needed a software upgrade to accommodate product-specific features of each new field device. This involved a significant development cost, and often there would be a time lag before it could be done. Even then, the host would usually talk to only a limited range of field devices, often restricted to the host vendor's own products. With the increasing success of the HART protocol, and the rapidly-growing number of suppliers using it, it has become quite impossible to keep host software up-to-date in this way. Users have been left with hosts unable to take full advantage of the wide range of instruments on the market.

DDL overcomes this difficulty. Device Descriptions make it easy to upgrade hosts to support new field devices, without re-writing any software. Device Descriptions can be used by any suitably-designed host device to automatically provide a correct and complete user interface for each field device. Device Descriptions allow full interoperability and a degree of interchangeability between smart instruments from different manufacturers. Users can choose the best instrument for each application, without being locked in to a single supplier for a complete system.

This chapter provides an introduction to the HART Device Description Language, and to the generation and use of Device Descriptions. However, not all features are described here. The complete language specification, development tools and training are available from the HART Communication Foundation.

5.2 What is Device Description Language?

DDL is a formal language, used to describe completely and unambiguously, what a field instrument looks like when you talk to it through the "window" of its digital communication link. It forms an additional "user layer" on top of the OSI protocol reference model. DDL includes descriptions of accessible variables, the instrument's command set, and operating procedures such as calibration. It also includes a description of a menu structure which a host device can use for a human operator. The Device Description (DD), written in a readable text format, consists of a list of items ("objects") with a description of the features ("attributes" or "properties") of each. Some example fragments from an (imaginary) flowmeter DD are shown in Figure 5-1.

```
VARIABLE low_flow_cutoff
        {
        LABEL [low_flow_cutoff];
        HELP "Low Flow Cutoff – The value below which the process variable will indicate zero,
                  to prevent noise or a small zero error being interpreted as a real flow rate.";
        TYPE FLOAT
                {
                DISPLAY_FORMAT "6.4f";            /* ##.#### */
                }
        CONSTANT_UNIT "%";
        HANDLING READ & WRITE;
        }

MENU configure_io
        {
        LABEL [configure_io];

        ITEMS
                {
                flow_units,                       /* variable */
                rerange,                          /* edit-display */
                low_flow_cutoff,                  /* variable */
                flow_tube_config,                 /* menu */
                pulse_output_config               /* menu */
                }
        }

COMMAND write_low_flow_cutoff
        {
        NUMBER 137;
        OPERATION WRITE;
        TRANSACTION
                {
                REQUEST
                        {
                        low_flow_cutoff
                        }
                REPLY
                        {
                        response_code,
                        device_status,
                        low_flow_cutoff
                        }
                }
        RESPONSE_CODES
                {
                0, SUCCESS,                       [no_command_specific_errors];
                3, DATA_ENTRY_ERROR,              [passed_parameter_too_large];
                4, DATA_ENTRY_ERROR,              [passed_parameter_too_small];
                5, MISC_ERROR,                    [too_few_data_bytes_received];
                7, MODE_ERROR,                    [in_write_protect_mode];
                }
        }
```

Figure 5-1. Fragments of a Device Description

Notes: Upper-case words are keywords of DDL. Text surrounded by /*... */ is a comment.

Items in square brackets [ ] are references to the standard dictionary.

Other names are internal cross-references, for example the MENU ITEM "low_flow_cutoff" refers to the VARIABLE "low_flow_cutoff".

The principal constructs or object types of DDL (summarised in Table 5-1 below) are as follows:

- Variables – Any item of data contained in the field device: measurements, operating parameters, device information. Among the attributes of a variable are a label for display, and the specific data type. Table 5-2 lists the main data types available.
- Commands – HART commands which the device will accept, defined by the content of their request and reply data fields, and the response codes implemented.
- Menus – Menus for presentation to an operator (specifically for a handheld communicator), defined as a list of other items (variables, displays, methods, further menus)
- Edit Displays – Displays for presentation to the operator (specifically for a handheld communicator), defined as a list of variables for display and/or editing, and procedures ("methods") to be executed before or after editing.
- Methods – Defined sequences of interactions with the field device and with the operator, executed by a host to achieve specified operations on the field device (e.g. calibration or re-ranging), using a subset of the ANSI C programming language. A library of built-in functions is available for use within methods, including sending commands to the field device, inspecting responses, displaying messages to the operator, and accepting keyboard input from the operator. These allow a method to deal correctly with error and failure conditions, as well as normal operation. A method can also be used to warn the operator of the implications before proceeding with an action which might adversely affect system behaviour.

Variables (and other objects) can also be grouped into "arrays", "collections" and "relations", to express functional similarities or relationships relevant to the way they are used.

Arithmetic, logical or conditional expressions are permitted in the definition of many attributes. In this way, the device can be treated differently depending on present circumstances (for example, configuration parameters or operating modes).

A standard dictionary is used to provide multiple language translations of common phrases (see Figure 5-2). This provides several benefits:

- shorter compiled DDs using dictionary reference numbers instead of text.
- instant translation into several languages.
- consistent terminology between manufacturers.

<table>
<tr><td>[251,2]</td><td>square_root</td><td>"Sq root",
"|033 Racine carree",
"|049 Radiziert"</td></tr>
<tr><td>[251,3]</td><td>linear_with_input</td><td>"Linear with input",
"|033 Lineaire avec entree",
"|049 Linear z. Eingang"</td></tr>
<tr><td>[254,2]</td><td>passed_parameter_too_large</td><td>"Value was too high",
"|033 Val trop haute",
"|049 Wert war zu hoch"</td></tr>
<tr><td>[301,34]</td><td>remove_from_auto_befor_send</td><td>"WARN – Remove loop from automatic control before sending. You may return loop to automatic control after sending.",
"|049 WARNUNG – Vor Senden die automatische Steuerung abschalten. Steuerg. ev. wieder einschalten."</td></tr>
</table>

Note: International telephone dial codes are used to identify languages other than US English.

Figure 5-2. Entries in the standard dictionary

5.3 Benefits of DDL

The major benefit of DDL for suppliers is that it decouples the development of host and field devices. Each designer can complete product development, with the assurance that the new product will interoperate correctly with current and older devices, as well as with future devices not yet invented. In addition, a simulation program can be used to "test" the user interface of the DD, allowing iterative evaluation and improvement, even before the device itself is built.

For the user, the major benefit is the ability to mix products from different suppliers, with confidence that each can be used to its full capability. Easy field upgrades allow host devices to accept new field devices. Innovation in new instruments is encouraged.

The use of a standard dictionary both provides instant translation into the supported languages, and encourages designers to follow consistent implementations of common tasks. "Interoperable" DDs (see section 5.4) encourage this further, and make true interchangeability possible.

A host system keeping records of configuration or instrument management data can use DDs to construct appropriate database record structures for each instrument type, either fully automatically, or interactively with the assistance of a human operator.

Table 5-1. Principal DDL object types

DDL object type	Attributes	Comments
VARIABLE	name	For cross-reference
	LABEL	Displayable text
	TYPE	Data type. See Table 5-2
	CLASS	Classification of use[1]
	HANDLING	READ, WRITE or READ & WRITE
	CONSTANT_UNIT	Unchangeable units text
	VALIDITY	Meaningful existence[2]
	HELP	Displayable text
	READ_ & WRITE_TIMEOUTs	Times a host may have to wait for read or write to complete
COMMAND	name	For cross-reference
	NUMBER	HART command number
	OPERATION	READ, WRITE or COMMAND[3]
	TRANSACTION	
	REQUEST	Data included in command
	REPLY	Data included in reply
	RESPONSE_CODES	List of sets: {value, type, description, help}
MENU	name	For cross-reference
	LABEL	Displayable text
	ITEMS	List of variables, menus, edit displays and/or methods
EDIT_DISPLAY	name	For cross-reference
	LABEL	Displayable text
	DISPLAY_ITEMS	List of variables for display only
	EDIT_ITEMS	List of variables for editing
	PRE_EDIT_ACTIONS	List of methods to be performed before editing
	POST_EDIT_ACTIONS	List of methods to be performed after editing
METHOD	name	For cross-reference
	LABEL	Displayable text
	CLASS	Classification of use[1]
	DEFINITION	Sequence of C statements
	VALIDITY	Meaningful existence[2]
	HELP	Displayable text

Notes:
1. CLASS classifies a variable or method according to how it is used by the field device. Examples are: DEVICE, INPUT, CORRECTION, DYNAMIC, DIAGNOSTIC, SERVICE).
2. VALIDITY defines circumstances in which a variable or method is valid, that is, has a meaningful existence. It is usually specified as a conditional expression, evaluating to TRUE or FALSE depending on present values of other variables.
3. A COMMAND command is one that affects device operation in some way other than by writing a new value to a variable (for example, "set zero").

Table 5-2. Principal DDL data types

Variable data type	Sub-attributes	Comments
Arithmetic:		
FLOAT		Four-byte floating point (IEEE 754)
DOUBLE		Eight-byte floating point (IEEE 754)
INTEGER	size (in bytes)	
UNSIGNED_INTEGER	size (in bytes)	
	DISPLAY_FORMAT	Defines display (C "printf" format string)
	EDIT_FORMAT	Defines data entry (C "scanf" format string)
	MIN_VALUE	Upper limit for entered value of variable
	MAX_VALUE	Lower limit for entered value of variable
	SCALING_FACTOR	Multiplier to convert value of variable for display
Enumerated:		
ENUMERATED	size (in bytes)	List of sets: {value, description, help}
BIT_ENUMERATED	size (in bytes)	List of sets: {value, description, help, functional-class[1], status_class[2], actions}
Strings:		
ASCII	length (in characters)	Full ISO Latin 1 character set available
PACKED_ASCII	length (in characters)	Restricted ASCII character set (see Glossary)
PASSWORD	length (in characters)	Usually displayed as *****
BITSTRING	length (in bits)	Interpretation is not specified.
Date and time:		
DATE		Three bytes: day, month, year-1900.
TIME		Format not yet defined
DATE_AND_TIME		Format not yet defined
DURATION		Format not yet defined

Notes:
1. "Functional-class" indicates the class of use for each bit (as for other variable types).
2. "Status-class" classifies each device status bit according to its cause, duration, correctability and scope.

5.4 Creating a Device Description

Figure 5-3 represents the process of generating and using a Device Description. Creation of the DD is the responsibility of the field device designer, who first writes the DD in DDL "source" text form. This can also form an important part of the device's specification, since it is human-readable, and describes explicitly how the device will appear to the outside world. The designer can omit the standard HART "universal" and "common-practice" data and commands from the DD, and instead simply refer to them as "imported". The HART Communication Foundation is also developing "interoperable" DDs for common instrument types (pressure, temperature, level, various types of flowmeter, valve positioner, etc.) These will encourage consistency of parameter use and operating procedures in devices of the same type from different suppliers.

The DD text is then passed through a "tokeniser" program, which compresses it, replacing standard keywords and text phrases by numbers (using the standard dictionary for reference). This creates a "tokenised" or "binary" form of the description; this is the distributable version used by host system builders and service organisations.

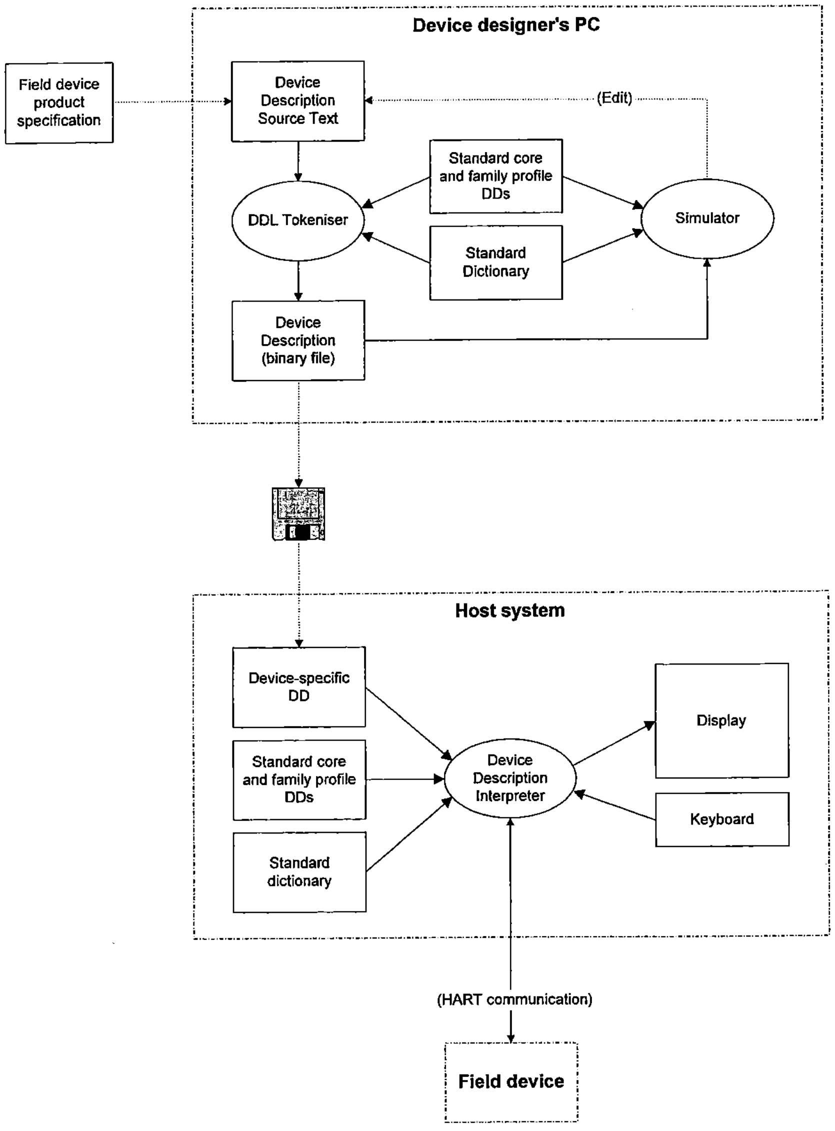


Figure 5-3. Generation and use of a Device Description

Using a simulator program, the designer can demonstrate the proposed user interface, and improve it if necessary by editing the DD source text. (The core DDs and standard dictionary, and the tokeniser and simulator programs, are available from the HART Communication Foundation, to run on an IBM-compatible PC.)

DDL writers should be aware that it is possible to describe, in DDL, things which are not permitted by other parts of the HART protocol specification. Such usage must be avoided.

5.5 Using a Device Description

The host device designer's task is to write a program which can interpret Device Descriptions (a "DDL engine"), generating the described screen displays, operator interactions and HART commands, in the particular hardware and software environment of the host system. In general, this program should be capable of providing all the services which the DDL language can specify; however, some applications may not need a complete implementation. This DD interpreter, when running in the final system, operates on the DD for an identified field device, referring as necessary to the standard core (universal and common-practice) DDs and the standard text dictionary.

Once this general-purpose DD interpreter (and a linker, if required – see below) is written and tested, the host designer's task is complete. The resulting system should work with any device presented to it. In the event that the host does not possess a copy of the DD for a particular field device, it can fall back on the standard universal and common-practice command DDs, or on "interoperable" DDs if the device implements these. Thus a host can provide at least a basic capability for any device, until such time as the specific DD can be loaded in.

In some host systems, the set of device-specific DDs and the standard DDs may be pre-combined into an internal form for more efficient operation (less memory use, for example). This might be done off-line by a "linker" program (probably running in a PC), which takes binary DDs for a number of devices, combines them, and translates them into the form the interpreter needs. Or it might be done on-line, whenever a new DD is introduced into the system.

5.6 Distribution of Device Descriptions

For a host to work fully with a particular set of field devices, it needs to have the appropriate set of binary DDs available. Depending on the physical form of the host device, DDs could be loaded in by copying from a floppy disk, by transfer through a serial port, or simply by plugging in a memory module containing the information.

In any case, the host system builder (or service engineer, for later updates) needs to have the relevant set of DD files for the field devices to be used. Typically, users will find that the host system supplier includes a basic set of DDs, and offers a service for installing more. However, it should be quite feasible for users to undertake this installation themselves if they wish, given appropriate equipment and training by the host system supplier.

To provide a central resource for manufacturers and users to obtain DDs from other vendors, the HART Communication Foundation maintains an archive of registered and tested DDs. These are available to subscribers, with regular updates as needed. It may be that some instrument vendors will also supply their own DDs directly to end users.

As memory cost and power consumption continue to fall in the future, it is possible that the DD might be included in the field device itself, so that it could be read out by a host when it is first connected. This would be a very attractive proposition, since it would completely remove the need for field upgrades to host systems. However, the HART protocol does not at present include suitable commands for this, and is also rather slow for the purpose – it might take many minutes to perform the upload of a full DD. (Uploading just the incremental DD differences in a new device type could be attractive.)

5.7 Fieldbus

Fieldbuses have exactly the same need for Device Descriptions, for all the same reasons. In fact, the idea for DDL originated in the International Fieldbus Group a few years ago. Any fieldbus with pretensions to being an international standard should include a DDL. It cannot be exactly the same as the HART DDL, since the details of the language must depend to some extent on the protocol it is describing, but it must provide the same benefits of real interoperability between instruments from different manufacturers. The Fieldbus Foundation DDL is based closely on the HART version, with appropriate modifications and extensions.

5.8 Summary

The HART Device Description Language provides the means to completely describe the behaviour of a field device, as seen through its communication link. The language offers a set of object types and attributes, from which a description of data, commands, menus and displays can be built up. "Methods" allow operator interactions to be specified. Existing DDs can be imported and re-used.

The DDL mechanism allows interoperability between equipment from different suppliers – an important requirement for the widespread acceptance of digital field communications.

DDL is being used by many members of the HART Communication Foundation as they introduce new HART-based products. The model 275 Universal HART Communicator, and other host systems implementing DDL, will be able to work with any of these, both now and in the future.

The HART Communication Foundation maintains an archive for instrument manufacturers to register their Device Descriptions, and manages the distribution of these to host system designers, service departments or others who need them.

GLOSSARY

This glossary contains explanations of some technical terms and abbreviations, particularly as they relate to the HART protocol.

Address.

In communications technology, the address of a device is a code number used to identify that device, so that messages can be delivered correctly. The destination device's address is included as part of the message frame, so that listening devices can know whether the message is intended for them or not. The source device's address may also be included. HART uses two forms of addressing for field devices: a "polling address" of 0 to 15, and a "unique identifier" of 38 bits. The HART master address is 1 for a primary master, 0 for a secondary master.

HART also uses a "broadcast address" (38 bits, all zeros) in messages intended for all field devices.

Aliasing.

Aliasing is an effect which can occur in sampling measurement systems, when the measurement signal changes faster than can be properly represented by the samples, specifically, if the measurement contains frequencies higher than about half the sampling frequency. (For measurement purposes, sampling twice per cycle can usually be considered an adequate representation; at least the average over the cycle will be correct.) Aliasing can result in the appearance of errors in the sampled values, at beat frequencies between the signal variation and the sampling rate. Low frequency errors (or even steady a d.c. offset) may be caused, which could badly affect the operation of a control loop.

If the sampling rate cannot be increased to avoid the effect, filtering the measurement signal before sampling, to remove higher-frequency components, is the proper cure. See also "Damping value".

ASCII.

ASCII (American Standard Code for Information Interchange) is a widely-used code defined by ANSI (the American National Standards Institute). It represents the alphabet (upper and lower case), numbers 0 to 9, and common punctuation characters, as 7-bit binary codes. In addition, a number of codes are allocated for "control" functions, such as "Start of Text", "End of Text", "Carriage Return", "Form Feed", "Tab", "Bell", "Backspace" and others.

When 7-bit ASCII is transmitted in 8-bit bytes, the most-significant bit is usually either set to 0, or used as a parity check bit.

The original ASCII code did not include European characters with accents. Extended ASCII codes are now in use, which use 8 bits, and include extra characters, but these are not well standardised. The ISO 8859/1 "Latin 1" character set is compatible with ASCII for the first 128 characters, but also includes a good selection of accented and other characters for European languages in the second 128 characters.

For use in tags, descriptors and messages, the HART protocol uses only a subset of the original 7-bit ASCII characters – those represented by codes with a most-significant hex digit of 2, 3, 4 or 5. This range includes the digits 0 to 9, upper-case A to Z, and common punctuation characters. Lower-case and accented letters are not included. This subset is represented in HART by 6 bits, by removing the most-significant bit of the 7 bit ASCII code. (The most-significant hex digits 2, 3, 4 and 5 become 2, 3, 0 and 1 respectively). This allows four characters to be packed into three bytes, reducing the length of text data items for improved transmission efficiency.

Asynchronous transmission.

To recover meaningful data from a serial bit stream arriving over a communication channel, the receiving device needs to identify the beginning and end of each bit and character. This timing information can be provided in various ways:

In "synchronous" communication, clock information is transmitted on a separate line, or is embedded in the bit stream in such a way that it can be extracted by the receiver.

In "asynchronous" ("without a clock") transmission, timing is defined by starting each character with a start bit (always 0) and following the character by a stop bit (always 1). Within a character, the bit timing is then defined by the baud rate (agreed by both parties, or, sometimes, detected automatically).

HART uses asynchronous serial transmission, sending the least-significant bit first in each character. In principle, there could be any amount of idle time (at the logical 1 level) between characters; however, to meet the overall transaction timing requirements, this is not permitted in HART.

Baud rate.

The baud rate of a communication channel is the number of data symbols transmitted each second. Some systems code more than one data bit into each symbol (often by combining phase and amplitude modulation), so as to provide more possible values for each symbol, and therefore a higher bit rate.

The HART protocol specifies a 1200 baud transmission rate, with only two distinct values for each symbol (frequencies of 1200 or 2200 Hz); thus, each symbol represents only one data bit, and the data rate is 1200 bits per second (bps), the same as the baud rate.

Bell 202.

Bell 202 is a U.S. standard, originated by AT&T (the Bell Telephone Company). It uses 1200 Hz and 2200 Hz as 1 and 0 respectively, at 1200 baud. Bell 202 is a full duplex communication standard, using a different pair of frequencies for its reverse channel.

HART uses Bell 202 signals, but is a half-duplex system, so the reverse channel frequencies are not used.

Some other Bell standards have European (CCITT) equivalents; Bell 202 does not.

Binary.

Numbers can be represented in any "base". Our normal counting uses decimal (base 10), in which any number is represented by the digits 0 to 9, written as multipliers for successive powers of 10 from right to left (units, tens, hundreds, etc.). In a binary representation, only digits 0 and 1 are used, and successive digits from right to left represent multipliers for successive powers of 2. Computers nearly always use binary representation for numbers and other data coded into numeric form, to match the two-state "on/off" switching mechanism of most digital electronic circuits.

Bit.

A bit, or binary digit, represents a single item of "yes/no" information.

Numerical and alphabetic information can be coded into a number of bits for computer or communication purposes, for example using the ASCII code for alphanumeric characters or the IEEE 754 code for floating point numbers.

Byte.

A byte is a set of bits (usually 8), treated as an entity. Eight bits is often a convenient sized piece of data for a computer to handle. This is because it is a power of two, and is large enough to contain a useful range of character codes.

Capacitance number (CN).

HART specifies a capacitance limit of 5000 pF between the two terminals of a field device. However, it is allowable to exceed this value, and quote a "capacitance number" or "CN", which is the multiple of 5000 pF actually present in the device. For simplicity, the multiplier is normally rounded up to the next whole number.

CCITT.

The CCITT (in English, the International Telegraph and Telephone Consultative Committee) is the international standards organisation responsible for modem and other communication standards, both for telephone and radio systems. The V-numbered protocols (V.21, V.22, etc.) are CCITT standards.

Character.

Either:

an alphabetic, numeric or other text symbol, which can be represented by a binary code (for example ASCII, see above), or

a transmitted sequence of bits which contain data (which may, or may not, consist of a character in the above sense). This "character" is often considered as including the start, parity and stop bits surrounding the real data.

Checksum.

An additional byte or bytes appended to a message, containing the arithmetic sum of all previous bytes (usually ignoring any carry beyond the number of bits allocated for the checksum). In practice, the term is often loosely applied to the longitudinal parity check used in HART, and sometimes to more-complex schemes such as cyclic redundancy checks.

Compiler.

A computer program whose function is to convert a programmer's human-readable "source code" program into an machine-executable code version. A tokeniser (q.v.) is sometimes loosely referred to as a compiler, though it does not in fact generate executable code.

Crosstalk.

Crosstalk is the unintentional physical coupling of signals from one circuit to another. It can be caused by capacitance between circuits, by inductive coupling, or by common impedances shared between the two circuits.

The HART signal levels and cable specifications are designed to reduce crosstalk to levels at which it will not cause significant interference. If, despite this, crosstalk is high enough to be received by a device on another HART loop, the use of unique identifiers prevents a message being accepted and acted upon by the wrong device.

Cyclic Redundancy Check (CRC).

A cyclic redundancy check (CRC) is a complex check character, generated by a succession of bit shifting and exclusive-oring operations on each character of a message. Several standard CRC algorithms are in use, giving various levels of protection against different types of corruption. A common one is "CRC-16".

HART does not use a CRC. See also "Checksum" and "Longitudinal Redundancy Check".

Damping value.

The "damping value" or "damping time constant" in a HART field device is a smoothing time constant applied to the primary variable before its value is made available, either by digital communication or as the analogue output signal. It can be used to reduce unwanted "noise" from a measurement, or to reduce aliasing (q.v.) in a host system using a slow scan cycle (for example via a multiplexer).

Device Description (DD).

The HART Device Description is a complete and unambiguous description of a field device, written in Device Description Language (q.v.). Loading a Device Description into a host device allows that host to communicate fully with the corresponding field device, without any custom programming. See Chapter 5 for a full discussion of this subject.

Device Description Language (DDL).

The HART Device Description Language (usually abbreviated to DDL) is a formal language used to describe field devices, their commands, accessible data, display requirements and operating procedures. See "Device Description" above, and Chapter 5. The Fieldbus Foundation is developing a similar DDL for fieldbus devices.

Duplex.

Duplex communication means that communication is possible in both directions (as opposed to simplex, which is communication in one direction only – radio broadcasting, for example).

In half-duplex systems, the two stations take turns to transmit. In full-duplex, both can transmit and receive simultaneously.

HART uses half-duplex communication.

Dynamic variable.

In HART, the four principal measured variables (primary, secondary, third and fourth) are referred to as the "dynamic variables". This is a special use of the word – in general it could also be said that any process-related variable, which changes its value without user intervention, is "dynamic".

Enumerated variable.

A variable which can take only certain values, and for which those values are represented by some other set of symbols (usually successive integer numbers), is said to be "enumerated". For example, in the HART protocol, the list of registered manufacturer names is represented by the manufacturer code (1 to 249). Even a numerical variable may be enumerated, if not all values are permitted.

Exclusive Or.

"Exclusive Or" (sometimes shortened to "ex-or") is the logical combination function of two logical (0 or 1) values, such that the result is true (1) if one or other of the values is true, but not both. That is, the two values must be different. Otherwise, the result is false (0).

This concept is extended to bytes of data, by taking the corresponding bits of each byte, exclusive-oring them, and putting the result in the corresponding position in a "result" byte.

The HART checksum uses an exclusive-or of the message bytes as a check against corruption. To generate the checksum for a message, each byte is exclusive-ored into the previous result.

See also "Longitudinal Redundancy Check".

Field.

Either:

> the area of a process plant outside the control room, where measurements are made, to and from which communication is provided, or
>
> a part of a message devoted to a particular function, for example the "address field" or the "command field". This could consist of one or several characters or bytes within the message. The size and other rules for the interpretation of each field are a part of the protocol specification.

Floating point.

Floating point represents a number in two parts: an exponent E and a mantissa M. The number represented is $M\times2^E$ (M times 2 to the power of E). This allows a uniform proportional precision over a wide numerical range.

The IEEE 754 standard single precision form, used in the HART protocol for all measured variables and range-related values, has a sign bit (1 = negative), an 8-bit exponent (biased by +127 to make it always positive) and a 23-bit mantissa. An additional leading 1 bit is assumed for the mantissa, with the stated part as a fractional addition to this. This format allows a numerical range of positive or negative values from 10^{-38} to 10^{+38}, with a resolution of 1 in 10^7, that is, 0.00001% of value. It fits into four bytes, as follows:

SEEEEEEE	EMMMMMMM	MMMMMMMM	MMMMMMMM

where S is the sign bit, E is the biased exponent and M is the fractional part of the mantissa.

Frame.

The message frame is the structure of the set of characters or bytes making up a single complete message. It is made up of a number of individual fields, containing the separate items within the message (address, data, etc.). The sequence of fields forming the frame is a part of the protocol specification.

Frequency-shift keying.

Frequency-shift keying (f.s.k.) is a method of modulating digital information for transmission over paths with poor propagation characteristics. Two different frequencies are used to represent 0 and 1, usually in the audio frequency range (300 to 3000 Hz). Such a signal can be transmitted successfully over telephone systems. An f.s.k. signal can also be modulated on to a radio carrier, or, as in HART, on to a d.c. current or voltage.

Half-duplex.

Transmission in both directions, but only one direction at a time. See "Duplex".

Handshaking.

Handshaking is part of many communication protocols. It is the method used to control the flow of information, so that the receiver is not overloaded. Without handshaking, the receiver must be ready and able to accept a message at any time. Handshaking may consist of a defined sequence of special characters or messages, or may use separate control signals (as in RS-232).

HART does not use handshaking, beyond what is defined by the sequence of messages making up the transaction procedure. (HART messages are always short enough that they can easily be received and stored in a temporary buffer, and, if need be, not actually dealt with until a short time later.)

HART®.

The HART protocol is a widely-used open protocol for communication with Smart transmitters. HART is an acronym for "Highway Addressable Remote Transducer".

HART is a registered trademark of the HART Communication Foundation.

HART Communication Foundation (HCF).

The HART Communication Foundation (HCF) was formed in 1993, to promote and support the HART protocol, taking over from the earlier HART User Group. The HCF is a "not for profit" foundation, supported by its membership. It offers training and tools to help manufacturers of HART-compatible products.

The HCF office is in Austin, Texas, USA. Telephone +1 (512) 794-0369; fax +1 (512) 794-3904.

Hexadecimal.

Hexadecimal (base 16) representation of numbers (hex for short) is commonly used to describe the value of a data byte. One hex digit takes values 0 to 15, written as 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Four bits are expressed in this way by a single hex digit. An 8-bit byte can be expressed as two hex digits, for example hex 43 represents binary 01000011 (decimal 67), and hex FF represents binary 11111111 (decimal 255).

Host.

A device which contains the communication "master" function. Typically a control system, or instrument management computer.

Integer.

An integer is a whole number (0, 1, 2, etc.), either positive or negative.

In the HART protocol, integers are transmitted as 8, 16 or 24 bits, and are always regarded as unsigned (that is, positive).

Interpreter.

A computer program whose function is to execute another program written in a different programming language. The interpreter provides executable code segments corresponding to each possible instruction in the original language, so that it can "run" that program as a series of calls to those code segments. Often the original program is part-compiled or "tokenised" to compress it, before it is passed to the interpreter. See also "Compiler" and "Tokeniser".

Linker.

A computer program whose function is to combine together several program sections or files into one, to create a complete working program or file. HART Device Descriptions are combined by a linker, to reduce the amount of memory space used in the HART Communicator.

Longitudinal Redundancy Check (LRC).

This form of message integrity check, otherwise known as "longitudinal parity", uses a check character containing an odd or even parity bit for the corresponding bits in each character of the message (usually including address and control characters as well as the real data).

HART uses this form of error checking. In conjunction with the parity bit for each individual character, this allows detection of multiple errors per character.

See also "Exclusive Or".

Master-slave.

A master-slave communication system is one in which message transactions are always initiated by a master device (for example, a central controller) and slave devices only respond to requests received. This organisation keeps the design of slave devices (such as field-mounting transmitters) simple, and puts the responsibility for recovering from errors entirely in the more-complex central master. HART is a master-slave protocol.

See also "Multimaster".

Modem.

A modem ("modulator / demodulator") is a device which converts binary digital signals to and from an f.s.k. form. Most commercial modems provide RS-232 signal levels on the binary digital side. A modem does not provide a data coding mechanism, only a conversion of the physical form of signal used.

The term may be applied to a box which includes supporting functions such as signal interface circuits, connectors, power supply, auto-dialling or auto-answer functions, etc., or to an integrated circuit chip which just performs the main conversion task.

Low-power CMOS modem chips are available for use in HART field devices.

Modulation.

Modulation is the process by which a carrier signal is varied in some way to represent an information signal. This can allow accurate transmission of the information, even over an imperfect signal path, or a path which could not convey the information signal in its original form.

"Demodulation" is the inverse process of recovering the original information at the receiver.

HART uses "frequency-shift keying" as a method of modulating binary coded data on to a d.c. current or voltage signal. Within limits, attenuation of the audio frequencies used has no effect on the accuracy of the reconstructed binary data stream at the receiver.

Multidrop.

In a multidrop communication system, more than two devices are connected together on a single transmission medium. In such a system, each device must have a unique address. A message including a destination address can then be recognised by the device it is intended for.

HART can be operated in a multidrop mode, with up to 15 slave devices on one pair of wires. The analogue signals cannot be used in this mode, since they would simply add together in the common wiring. To save power, the output current of each slave device is set at its minimum value.

Multimaster.

Multimaster communication systems can have more than one master device. For this to work, the masters need a specified way to allow each other an opportunity to transmit.

HART is a simple multimaster system. After receiving a reply from a slave device, a master waits for a short time before starting another transaction. This allows a second master to break in if it wishes.

Multiplexer.

A multiplexer is a device which can connect one circuit to any one of a number of other circuits; in other words, it acts as a switch, normally under software control.

HART masters often include a multiplexer to allow a single serial communication channel and HART modem to serve several HART loops, thus saving cost. However, there is a penalty in performance: since only one transaction can be underway at a time through one modem, the scan cycle is increased proportionally to the number of devices scanned. In addition, the master should follow the "unsynchronised" timing rules, and wait 305 ms (380 ms for a secondary master) before transmitting, in case another master is already active on the loop, making the total transaction time anything up to 700 or 800 ms.

Off-line / on-line.

A communication device is said to be "on-line" when connected in its working environment, in an operational condition. It is "off-line" when disconnected or switched off, or perhaps when connected in a temporary environment such as a workshop.

OSI model.

The "OSI" ("Open Systems Interconnection") reference model is a defined way of structuring the specification and implementation of a communication protocol into "layers", each of which has a specific function. It originated from the International Standards Organisation (ISO). In any implementation, each function should be performed, if at all, in the appropriate layer, and the interfaces between layers should be well-defined.

There is no implication that different "OSI model conformant" protocols will be able to inter-communicate directly. However, the implementation of gateways translating between different

protocols should be easier than for non-OSI protocols.

HART implements layers 1 (the physical layer), 2 (the data-link layer) and 7 (the application layer) of the OSI model. Layers 3 (the network layer), 4 (the transport layer), 5 (the session layer), and 6 (the presentation layer), are not relevant to this type of local network.

Parallel.

See "Serial".

Parity bit.

A parity bit is often appended to the data bits in a character. This may be "even" or "odd", by agreement of both parties. For odd parity, the extra bit is a 1 or 0, so as to make the total number of 1's in the data and parity together, odd. For even parity, it makes the total number of 1's even.

Using the parity bit, the receiver can detect any single bit error within the data and parity part of the character. The UART will detect this as a "parity error". (Note that it is not possible to correct the error, since there is no way to tell which individual bit was corrupted. Also note that a single parity bit cannot guarantee to detect errors affecting more than one bit in a character.)

HART appends an odd parity bit to each byte transmitted. Further security against data corruption is provided by the message checksum.

Point-to-point.

In a point-to-point communication system, only two communicating devices are connected together. The wiring goes from one "point" to the other "point".

HART is usually operated in point-to-point mode, in which case the analogue signal can be used.

See also "Multidrop".

Process variable.

This is a general term describing any measured quantity originating in the process (pressure, temperature, flow, etc.), or a value derived from these, such as the HART "dynamic variables".

Protocol.

A communication protocol is a set of rules to be used in generating or receiving a message. It may include specifications for transaction rules (master-slave relationship, acknowledgement, timeouts, error-recovery), message structure (start character, addressing, data formats, error checking), coding (text and numeric data formats) and physical signal characteristics (modulation techniques, signal type, signal level, transmission medium).

Redundancy.

Redundant information is information which is additional to the real information being transmitted, but is generated in some way from that information. The use of redundancy allows the detection, and in some cases the correction, of errors introduced in the transmission of data.

The HART protocol includes redundant information in the form of an odd parity bit for each byte, a checksum character for each message, and the echoing of address, command and data fields from the host, in the reply from a slave device.

RS-232.

RS-232 is a well-known standard for serial asynchronous communication, originally designed for the connection of computer terminals and modems – "Data Terminal Equipment" (DTE) and "Data Communications Equipment" (DCE) respectively – over distances less than 50 feet. Specified by the EIA (Electronic Industries Association), it defines connectors, signal meanings, and signal voltage levels. In most applications, many of the handshaking and other control lines of the standard are not used, leading to a large number of minor variants.

For all practical purposes, the CCITT V.24 standard is the same as RS-232.

Many computers provide an RS-232 port, which can be used to connect a Bell 202 modem to link into a HART network.

RS-485.

RS-485 is an EIA (Electronic Industries Association) communication standard, using two-wire balanced circuits with a differential signal of 5V, for good noise-immunity. Impedance-matched lines are often used. RS-485 allows higher speeds and much greater distances than RS-232. Depending on their design, up to 32 devices can be connected together on a single pair of wires, in multidrop and/or multimaster configurations.

Some vendors (including Fisher-Rosemount) offer instruments using the HART frame and message formats, with an RS-485 physical layer. When higher speeds than the standard 1200 bps of HART are used, the timing rules have to be modified. This variant has not yet been accepted by the HART Communication Foundation.

Serial.

Transmission of digital information from one device to another can be organised in two ways. In serial form, one bit is sent after another, on a single transmission path. In parallel form, several bits

(often a byte of 8 bits) are sent simultaneously on a number of paths equal to the number of bits involved. (In this context, a transmission path is a wire, or a pair of wires, or other medium, used to convey a single binary digit value).

In either case, there could be an additional path carrying a clock signal defining instants at which the data signal(s) should be considered as meaningful.

HART uses serial transmission. Since HART is asynchronous, there is no extra clock signal.

Simplex.

Transmission in one direction only. See also "Duplex".

Smart Family Interface (SFI).

Rosemount's "Smart Family Interface", a hand-held configuration device for HART instruments. Now more commonly known as a "hand-held configurator", "hand-held terminal", "hand-held communicator" or "HART Communicator".

Start and stop bits.

In asynchronous communication, start and stop bits are used to indicate the beginning and end of a character. The start bit is a 0, perhaps following an idle period of 1-level signal. The stop bit is a 1, which ensures that the next start bit is recognisable even if there is no idle period.

Some protocols (not HART) extend the stop bit to occupy 1½ or 2 bit times, and may allow further extended idle periods at the "1" level, between characters.

The receiving UART derives sampling times for the individual bits by timing from the leading edge of the start bit (using the specified or assumed baud rate). The UART will detect a "framing error" if the stop bit does not appear at the right time due to corruption of the signal or other fault.

Synchronous.

See "Asynchronous".

Timeout.

If an expected event does not occur within a specified time, this time period, and the non-event itself, are both referred to as a "timeout".

In the HART protocol, there are timeouts for the response by a slave to a message from the master station, and for the pause after each transaction, to allow the other master to transmit.

Tokeniser.

A computer program whose function is to replace keywords and other text in a programmer's human-eadable "source code" by numerical coded reference symbols ("tokens"). This significantly compresses the source code, but does not actually produce machine-executable code. The resulting okenised form is used as data for an "interpreter" program to work on. (Many forms of BASIC work n this way.)

The HART DDL tokeniser converts the original text form of a DD into a smaller tokenised or "binary" form for distribution and use by the DD interpreter in a host device.

See also "Compiler" and "Interpreter".

Transaction.

The series of messages used to convey a piece of useful information from one station to another. This might include acknowledgements and/or retries after detection of errors.

UART.

A UART (Universal Asynchronous Receiver Transmitter) provides the electronics needed to convert a byte of data (usually presented by the processor in parallel form) to and from serial form, and to add or remove the start, parity and stop bits. It usually takes the form of an integrated circuit chip, and can be configured to use 7- or 8-bit data, odd, even or no parity, and any standard baud rate. At the receiving end, the UART checks parity and the character frame format, and reports any errors to its controlling processor.

Unique identifier.

The unique identifier is a HART concept; it is a 38-bit integer formed from the manufacturer identification code, the device type code, and a device identifier (serial number). This number is virtually unique for every HART field device in the world, and is used as the device address in long frame messages (see section 3.8). This ensures that messages are never accepted and acted upon by the wrong device. See also "Crosstalk".

Variable.

In the mathematical sense (and in HART), a "variable" is any item of data which can take various values. This has nothing to do with data type: text strings are just as much variables as are numeric quantities. Nor does it relate to whether the value varies often, or only when "configured". See also "Dynamic variable", "Enumerated variable" and "Process variable".

FURTHER INFORMATION

This section lists a selection of references for further information on the HART protocol.

Documents available from the HART Communication Foundation:

HART® - SMART Communications Protocol, Protocol Specification –
Document Revision 5.3; HCF Document Number HCF_SPEC-10.

HART® - SMART Communications Protocol, FSK Physical Layer Specification –
Document Revision 7.2; HCF Document Number HCF_SPEC-53.
(Document Revision 8.0, HCF Document HCF_SPEC-54, is due out shortly.)

HART® - SMART Communications Protocol, Voltage Modulation Physical Layer Specification –
Document Revision 1.0; HCF Document Number HCF_SPEC-71.

HART® - SMART Communications Protocol, Data Link Layer Specification –
Document Revision 7.0; HCF Document Number HCF_SPEC-81.

HART® - SMART Communications Protocol, Command Summary Specification –
Document Revision 7.0; HCF Document Number HCF_SPEC-99.

HART® - SMART Communications Protocol, Universal Command Specification –
Document Revision 5.1; HCF Document Number HCF_SPEC-127.

HART® - SMART Communications Protocol, Common Practice Command Specification –
Document Revision 7.0; HCF Document Number HCF_SPEC-151.

HART® - SMART Communications Protocol, Common Tables –
Document Revision 7.0; HCF Document Number HCF_SPEC-182.

HART® - SMART Communications Protocol, Command Specific Response Code Definitions –
Document Revision 4.0; HCF Document Number HCF_SPEC-307.

HART Physical Layer Test Procedure –
Document Revision 1.0 Preliminary; HCF Document Number HCF_TEST-2.

Device Description Language Specification –
Document Revision 10.0; HCF Document Number HCF_SPEC-500.

NCR 20C15 Modem Application Note: A HART Master Demonstration Circuit –
Document Revision 2.0; HCF Document Number HCF_LIT-14.

NCR 20C15 Modem Application Note: A HART Slave Demonstration Circuit –
Document Revision 2.0; HCF Document Number HCF_LIT-15.

Documents available from other sources:

"Rosemount SMART Transmitters in Intrinsically Safe Systems" (Romilly Bowden, Feb 1991) –
A 24-page note discussing technical aspects of how IS barriers and isolators can (and in some cases cannot) be used in HART networks. Available from Fisher-Rosemount Ltd.

"HART® compatibility with analogue control systems" (Romilly Bowden, June 1995) –
A 10-page note discussing compatibility issues to consider, when using HART field devices with "non-HART-compatible" hosts. Available from Fisher-Rosemount Ltd.

"The HART® Book", published from time to time by Fieldbus.com Ltd. –
A useful reference to HART products and suppliers. Available from the publisher.

Software available from the HART Communication Foundation:

HTEST: HART Master Simulator –
A DOS PC program which executes interpreted "C" scripts to simulate a HART master. Builds and transmits HART messages, and receives, displays and logs responses from a field device. See HTEST Application Manual, HART Master Simulator, Document Revision 10.0; HCF Document Number HCF_LIT-17.

XMTR: HART Slave Simulator –
A DOS PC program which simulates a HART slave. Allows configurable responses, including error conditions. Also offers host (master) and monitor modes. See XMTR Application Manual, HART Slave Simulator, Document Revision 10.0; HCF Document Number HCF_LIT-16.

ANALYS: HART bus analyser –
A DOS PC program which monitors HART network messages. Records every message, with timing in milliseconds, and any errors detected. See HART Bus Analyser User Manual, Revision 1.0; temporary Fisher-Rosemount document (un-numbered).

H-Sim: HART network simulator –
A Windows 3.1 PC program which solves a set of equations representing a simple circuit model of a HART network. Given cable and device parameters, calculates received signal level versus cable length. Currently at Version 0.5. (Also available from the author.)

Contact information:

HART Communication Foundation
9390 Research Boulevard
Suite I-350
Austin, TX 78759
USA.

Tel: +1 512 794 0369
Fax: +1 512 794 3904
E-mail: keithk@hartcomm.org
Internet: http://www.hartcomm.org/

Fisher-Rosemount Limited
Heath Place
Bognor Regis
West Sussex PO22 9SH
United Kingdom.

Tel: +44 1243 863121
Fax: +44 1243 867554
Internet: http://www.frco.com/

Romilly Bowden
at Fisher-Rosemount Limited
(address above)

Tel: +44 1243 845486
Fax: +44 1243 845354
E-mail: romilly.bowden@frco.com

or at
6 Inglewood Drive
Bognor Regis
West Sussex PO21 4JX
United Kingdom.

Tel./Fax: +44 1243 265770
E-mail: romilly@romilly.co.uk
Internet: http://www.romilly.co.uk/

The HART Book
Fieldbus.pub Ltd.
1 West Street
Titchfield
Hants PO14 4DH
United Kingdom.

Tel: +44 1329 846166
Fax: +44 1329 512063
E-mail: phil@fieldbus.com
Internet: http://www.fieldbus.com/hart/

Some further Fisher-Rosemount offices are listed inside the back cover.

INDEX

References preceded by a hash sign (#) are to HART command numbers. Details of these can be found in tables 4-4, 4-5 and 4-6. See also under "command #n" for further references in the text.

Where more than one reference is listed, **bold** type indicates those giving more extensive information.

Resource Center
Fisher-Rosemount Systems
Austin Texas

Fisher-Rosemount Europe

Austria
Fisher-Rosemount AG
Industrie-Zentrum NO Sud
Strasse 2a Obj. M29
2351 Wr Neudorf
Austria
Tel: 43 2236 607
Fax: 43 2236 60744

Belgium
Fisher-Rosemount NV/SA
De Kleetlaan 4
B-1831 Diegem
Belgium
Tel: 32 2 716 77 11
Fax: 32 2 725 83 00

Denmark
Fisher-Rosemount A/S
Hejrevang 11
DK-3450 Alleroed
Denmark
Tel: 45 48 17 03 33
Fax: 45 48 17 02 44

England
Fisher-Rosemount Limited
Heath Place
Bognor Regis
West Sussex
PO22 2SH
Tel: 01243 863 121
Fax: 01243 867 554

France
Fisher-Rosemount SA
2 Place Gustave Eiffel
Silic 247
94568 Rungis Cedex
France

Germany
Fisher-Rosemount GmbH & Co
Argelsrieder Feld 7
82234 Wessling
Germany
Tel: 49 8153 270
Fax: 49 8153 27 172

Italy
Fisher-Rosemount Italia srl
Via Pavia, 21
1-20053 Muggio
Milano
Italy
Tel: 39 39 27021
Fax: 39 39 278 0750

Netherlands
Fisher-Rosemount BV
Patrijsweg 140
2289 EZ Rijswijk
The Netherlands
Tel: 31 70 340 96 66
Fax: 31 70 390 68 15

Portugal
Fisher-Rosemount A/S
Rosemount Portugal Instrumentos, LDA
Rua Alfredo Da Silva No 8
Bloco C, Piso O Norte
2720 Alfragide
Portugal
Tel: 351 1 471 3208
Fax: 351 1 471 3338

Spain
Fisher-Rosemount SA
Crta Fuencarral - Alcobendas Km, 12,200
Edificio Auge, 1 Plantas 5a-6a
28049 Madrid
Spain
Tel: 34 1 358 9141
Fax: 34 1 358 91 45

Sweden
Fisher-Rosemount AB
Kanikenasbanken 6, Box 1053
S-651 15 Karlstad
Sweden
Tel: 46 54 2 19 99 90
Fax: 46 54 2 11 28 04

Switzerland
Fisher-Rosemount AG
Blegistrasse 21
6341 Baar
Switzerland
Tel: 41 42 33 61 11
Fax: 41 42 31 87 40

Visit us on our Website: **http://www.fisher-rosemount.com**

FISHER-ROSEMOUNT

Managing The Process Better.™