HART Master Protocol Manual

HART Link Level Protocol to allow Network 3000 Controllers with the ACCOL Custom Module to serve as masters to HART Devices

Bristol Babcock

D4068

Issue: March, 2000

Notice

The information in this document is subject to change without notice. Every effort has been made to supply complete and accurate information. However, Bristol Babcock assumes no responsibility for any errors that may appear in this document.

Bristol Babcock does not guarantee the accuracy, sufficiency or suitability of the software delivered herewith. The Customer shall inspect and test such software and other materials to his/her satisfaction before using them with important data.

There are no warranties, expressed or implied, including those of merchantability and fitness for a particular purpose, concerning the software and other materials delivered herewith.

Request for Additional Instructions

Additional copies of instruction manuals may be ordered from the address below per attention of the Sales Order Processing Department. List the instruction book numbers or give the complete model, serial or software version number. Furnish a return address that includes the name of the person who will receive the material. Billing for extra copies will be according to current pricing schedules.

A Few Words About Bristol Babcock

For over 100 years, Bristol® has been providing innovative solutions for the measurement and control industry. Our product lines range from simple analog chart recorders, to sophisticated digital remote process controllers and flow computers, all the way to turnkey SCADA systems. Over the years, we have become a leading supplier to the electronic gas measurement, water purification, and wastewater treatment industries.

On off-shore oil platforms, on natural gas pipelines, and maybe even at your local water company, there are Bristol Babcock instruments, controllers, and systems running year-in and year-out to provide accurate and timely data to our customers.

Getting Additional Information

In addition to the information contained in this manual, you may receive additional assistance in using this product from the following sources:

Contacting Bristol Babcock Directly

Bristol Babcock's world headquarters are located at 1100 Buckingham Street, Watertown, Connecticut 06795 U.S.A.

Our main phone numbers are:

(860) 945-2200
(860) 945-2213 (FAX)

Regular office hours are Monday through Friday, 8:00AM to 4:30PM Eastern Time, excluding holidays and scheduled factory shutdowns. During other hours, callers may leave messages using Bristol's voice mail system.

Telephone Support - Technical Questions

During regular business hours, Bristol Babcock's Application Support Group can provide telephone support for your technical questions.

For technical questions regarding **configuration of the HART Protocol in ACCOL, other ACCOL software products** (as well as Open BSI and UOI) **call (860) 945-2265.**

For technical questions regarding Bristol's **Enterprise Server®** and **Enterprise Workstation® products**, **call (860) 945-2286.**

For technical questions regarding Bristol **hardware products call (860) 945-2463.**

You can e-mail our Application Support Group at: **bsupport@bristolbabcock.com**

The Application Support Group also maintains a bulletin board for downloading software updates to customers. **To access the bulletin board, dial (860) 945-2251** (Modem settings: 14.4K baud maximum, No parity, 8 data bits, 1 Stop bit).

For assistance in interfacing Bristol Babcock hardware to radios, contact **Communication Technologies** in Orlando, FL at (407) 629-9463 or (407) 629-9464.

Telephone Support - Non-Technical Questions, Product Orders, etc.

Questions of a non-technical nature (product orders, literature requests, price and delivery information, etc.) should be directed to the nearest regional sales office (listed below) or to your local Bristol sales office or Bristol-authorized sales representative.

U.S. Regional Sales Offices
Northeast (Watertown) (860) 945-2262
Southeast (Birmingham) (205) 980-2010
Midwest (Chicago) (630) 571-6052
Western (Los Angeles) (909) 923-8488
Southwest (Houston) (713) 685-6200

Principal International Sales Offices:
Bristol Babcock Ltd (UK): (011-441) 562-820-001
Bristol of Canada: (416) 675-3820
Bristol Babcock SA Mouy (France): (011-334) 431-1515

Please call the main Bristol Babcock number (860-945-2200) if you are unsure which office covers your particular area.

Visit our Site on the World Wide Web

For general information about Bristol Babcock and its products, please visit our site on the World Wide Web at:

www.bristolbabcock.com

Training Courses

Bristol Babcock's Training Department offers a wide variety of courses in Bristol hardware and software at our Watertown, Connecticut headquarters, and at selected Bristol regional offices, throughout the year. Contact our Training Department at (860) 945-2385 for course information, enrollment, pricing, and schedules.

Who Should Read This Manual?

This manual is intended for a system engineer or programmer who is responsible for configuring a Network 3000 series controller (33xx) to function as a master to a HART device. This manual assumes familiarity with HART Terminology, ACCOL tools software, and ACCOL programming.

1. Introduction

The HART (Highway Addressable Remote Transmitter) Master interface allows a Bristol Babcock 33XX series controller to communicate with HART compatible field devices. The 33XX acts as a HART Master device. The communications protocol used is defined in reference 1 listed at the end of this manual.

This implementation is based on the specifications for HART Rev 3 Documents (see ref. 1). Certain facilities allowed in Rev 5 (such as Long Addresses and Burst Mode) are also implemented (see ref. 2).

This implementation uses Bristol Babcock's HART Device Interface (HDI) board which can be accessed using any RS232/RS485 communication port. For information on the HDI board, see the associated product information package (ref. 3).

2. Overview

2.1. HART Operational Overview

2.1.1. General Description

HART is a master/slave protocol, with the master originating each transaction, and the slaves only generating replies when asked for by the master. HART allows for 2 'masters' on a single link. There would normally be a control-system master (the 'primary' master), and, when required, a hand-held communicator used as the 'secondary' master, for maintenance purposes. The primary and secondary masters must be configured to use separate addresses.

A full description of the protocol is given in the references 1 and 2. Note the protocol is being continuously developed by the HART Communications Foundation, hence the newest facilities may not be available using this interface.

2.1.2. Device addressing

Before HART Revision 5, all devices used 'short-frame format', where the address of the field devices was always between 0 and 15 (the 'polling address'). Where a single field device was used, this address would normally be 0. On a multi-drop link, the address of each device would need to be set uniquely between 1 and 15 (thus a maximum of 15 devices on the link).

HART Revision 5 introduced 'long-frame format'. The address of each device is made up of a 38-bit number, which allows a unique address to be allocated to every field device manufactured. It also allows for more than 15 devices on a link.

All devices will always respond to a Command 0 with a 'short-frame address' of 0 (when on a point-to-point link). This allows the user to obtain the 'basic information' required to set up a link to the device (see section *'Getting the basic information'*)

2.1.3. Burst Mode

'Burst Mode' is also supported, for use when only a single field device is connected. In this mode (not supported by all field devices), the field device can be configured to repeatedly send data without being asked for by the master.

Once a node is placed in Burst Mode, it will return data about 3 times per second, with a short gap to allow each of the 2 masters to transmit a message (e.g. to stop burst mode). Burst Mode responses do not contain an address as such, but the address bit is alternated between 0 and 1 each message, so as to allow each master in turn to intervene. Burst Mode is only useful if (on average) 3 messages per second are required, rather than 2. Unless it is absolutely necessary, the user is recommended to avoid use of burst mode.

2.2. ACCOL HART Implementation Overview

2.2.1. General Description

The Custom Module implementation gives powerful, flexible access to the HART link level protocol. This can mean that there is lot of information for the user to understand, but defaults are used to make basic use easier.

There is no PORTSTATUS Module support (as baud rate and other line parameters are considered to be fixed).

2.2.2. Timing considerations.

Dual-master arbitration is achieved by the master having just received a reply from a slave holding off for 75ms to allow the other master to commence a transaction (which it will commence between 20 and 75ms after detecting the end of the response to the other master). For operation to run smoothly, each master should try to use the link without longer delays. For instance, if an ACCOL program is written to read a value from a slave every 1 second, then the dual-master arbitration logic will not function, and any clash of both masters transmitting will cause retries to occur. The situation will always recover (as Primary and Secondary masters use differing delays after an error, of 305ms and 380ms). But it is best if at least one of the masters tries to use the link at maximum throughput, with a delay of less than 75ms between the end of one transaction and the start of the next. The ACCOL program is therefore best written with this in mind, e.g. on a continuous rate, using a WAIT FOR on a logical 'DONE' signal, or with 2 Custom module calls queued at once.

If, in a dual master situation, a master misses the 20-75ms window, then, to avoid clashes with another master, this implementation holds off for a period before trying to transmit. The other master may transmit during this hold-off period, in which case the transmit will be commenced after the next 20-75ms window. If the other master does not transmit, then the hold-off time is given by the SLAVE RESPONSE TIMEOUT signal.

In multi-drop mode only one transmitter can be accessed at a time, therefore this mode is not suitable where frequent access is required. (e.g. if 6 field devices are attached to a multi-drop link, the fastest all primary values can be read would be 3 seconds).

2.3. HART Command Codes

The following tables list the command/function codes available (use in the 'COMMAND CODE' signal in the HART CUSTOM LIST). Some message formats were changed for HART Revision 5. This is a summary only, for full details refer to the official HART documentation, or the device manufacturer's documentation. HART commands are divided into 'Universal Commands', 'Common Practice Commands' and 'Device Specific Commands'. For details of 'Device Specific Commands' refer to the device manufacturer's documentation. Most devices only support a subset of the commands.

Tables 1 and 2 give the format codes and units codes which are used in tables 3 ,4 and 5

2.3.1.1. Table 1 - Format codes

(see section *'ACCOL Data Format Codes'* for full ACCOL format descriptions.)

Format code	Description
C1	Use CST1:0 - 4-byte IEEE floating point
A	Use CST9:n (HART packed ASCII, n=no of characters to decode (4 characters per 3 bytes)
VU	Use nVU format (in BYT or WRD mode, with 'n' depending on number of bytes/words to translate)
H	Byte Integer, packed as xxxxx yyy (xxxxx= hardware revision, yyy= physical signalling code) Use VU and extract using ACCOL code.
B	Bit-mapped information. Use BYT,BIT,VL to access bits in a byte to/from ACCOL logical signals..
D	Date, stored as 3 bytes (day,month,year-1900). Use format of BYT,3VU

2.3.1.2. Table 2 - Units Codes

These take the values 0 to 249, and may be allocated by the device manufacturer as appropriate. The codes allocated at Hart Rev 3 are listed below (others have subsequently been allocated). Refer to individual device documentation for particular allocations, or to the latest HART official documentation. NOTE: the ACCOL 'Units Text' is NOT automatically updated by the Custom Module - it is up to the ACCOL Programmer to allocate the correct Units Text at compile time.

CODE	UNITS
0	Undefined
1	inches H_2O @ 68 degrees Fahrenheit
2	inches Hg @ 0 degrees Celsius
3	feet H_2O @ 68 degrees Fahrenheit
4	millimeters H_2O @ 68 degrees Fahrenheit
5	millimeters Hg @ 0 degrees Celsius
6	pounds / $inch^2$
7	bars
8	millibars
9	grams / $centimeter^2$
10	kilograms / $centimeter^2$
11	pascals
12	kilopascals
13	torr @ 0 degrees Celsius
14	atmospheres
15	undefined
16	gallons / minute
17	liters / minute
18	imperial gallons / minute
19	$meters^3$ / hour
20	feet / second
21	meters / second
22 to 31.	undefined
32	degrees Celsius
33	degrees Fahrenheit
34	degrees Rankin
35	degrees Kelvin
36	millivolts
37	ohms
38	hertz
39	undefined
40	gallons
41	liters
42	imperial gallons
43	$meters^3$
44	feet
45	meters
46	barrels
47 to 249	undefined
250	Not Used
251	Reserved
252	Reserved
253	Special
254	Expansion
255	Reserved

2.3.1.3. Table 3 - Universal Commands in HART Revision 5

Command number and function		Data in command		format code	Data in reply		format code
0	Read unique identifier		none		Byte 0	'254' (expansion)	VU
					Byte 1	manufacturer identification code	VU
					Byte 2	manufacturer's device type code	VU
					Byte 3	number of preambles required	VU
					Byte 4	universal command revision	VU
					Byte 5	device-specific command revision	VU
					Byte 6	software revision	VU
					Byte 7	hardware revision	H
					Byte 8	device function flags *	B
					Byte 9-11	device ID number	VU
					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	VU
					Byte 1-4	primary variable	C1
2	Read current and percent of range		none		Byte 0-3	current (mA)	C1
					Byte 4-7	percent of range	C1
3	Read current and four (predefined) dynamic variables		none		Byte 0-3	current (mA)	C1
					Byte 4	PV units code	VU
					Byte 5-8	primary variable	C1
					Byte 9	SV units code	VU
					Byte 10-13	secondary variable	C1
					Byte 14	TV units code	VU
					Byte 15-18	third variable	C1
					Byte 19	FV units code	VU
					Byte 20-23	fourth variable	C1
					(truncated after last supported variable)		
6	Write polling address	Byte 0	polling address	VU		as in command	VU
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	VU
					Byte 3	units code for sensor limits and min.	VU
					Byte 4-7	span	C1
					Byte 8-11	upper sensor limit	C1
					Byte 12-15	lower sensor limit minimum span	C1
15	Read output information		none		Byte 0	alarm select code	VU
					Byte 1	transfer function code	VU
					Byte 2	PV/range units code	VU
					Byte 3-6	upper range value	C1
					Byte 7-10	lower range value	C1
					Byte 11-14	damping value (seconds)	C1
					Byte 15	write-protect code	VU
					Byte 16	private-label distributor code	VU
16	Read final assembly number		none		Byte 0-2	final assembly number	VU
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 6-16	descriptor (16 chars)	A			
		Byte 18-20	date	D			
19	Write final assembly number	Byte 0-2	final assembly number	VU		as in command	

2.3.1.4. Table 4 - Universal Commands in HART Revision 2,3 and 4 (as different from Rev 5)

Command number and function		Data in command		format code	Data in reply		format code
0	Read unique identifier		none		Byte 0 Byte 1 Byte 2 Byte 3 Byte 4 Byte 5 Byte 6 Byte 7-9	transmitter type code * number of preambles universal command revision device-specific command revision software revision hardware revision device function flags final assembly number	VU VU VU VU VU H B VU
					* Revision 4 introduced the expanded device type as an option with the remaining bytes moved up by two positions.		
4	Read common static data (block 0): Read message	Byte 0	block number ("0")	VU	Byte 0 Byte 1-24	block number ("0") message	VU A
4	Read common static data (block 1): Read tag, descriptor, date	Byte 0	block number ("1")	VU	Byte 0 Byte 1-6 Byte 7-18 Byte 19-21 Byte 22-24	block number ("1") tag descriptor date "250" (unused)	VU A A D VU
4	Read common static data (block 2): Read sensor information	Byte 0	block number ("2")	VU	Byte 0 Byte 1-3 Byte 4 Byte 5-8 Byte 9-12 Byte 13-16 Byte 17-24	block number ("2") sensor serial number units code for sensor limits & min. span upper sensor limit lower sensor limit minimum span "250" (unused)	VU VU VU C1 C1 C1 VU
4	Read common static data (block 3): Read output information	Byte 0	block number ("3")	VU	Byte 0 Byte 1 Byte 2 Byte 3 Byte 44-7 Byte 8-11 Byte 12-15 Byte 16 Byte 17 Byte 18-24	block number ("3") alarm select code transfer function code PV/range units code upper range value lower range value damping value (seconds) write-protect code ("1" = protected) * private-label distributor code ** "250" (unused)	VU VU VU VU C1 C1 C1 VU VU VU
					* "250" or "251" in Revisions 2 and 3. ** "250"2 in Revisions 2 and 3.		
5	Write common static data (block 0): Write message	Byte 0 Byte 1-24	block number ("0") message	VU A		as in command	
5	Write common static data (block 1): Write tag, descriptor, date	Byte 0 Byte 1-6 Byte 7-18 Byte 19-21 Byte 22-24	block number ("0") tag descriptor date "250" (unused)	VU A A D VU		as in command	
5	Write common static data (block 4): Write final assembly number	Byte 0 Byte 1-3 Byte 4-24	block number ("4") final assembly number "250" (unused)	VU VU VU		as in command	
11-19		These commands did not exist before Revision 5.0					

2.3.1.5. Table 5 - Common Practice Commands

Command number and function		Data in command		format code	Data in reply		format code
33	Read transmitter variables	Byte 0 Byte 1 Byte 2 Byte 3	transm.var.code for slot 0 transm.var.code for slot 1 transm.var.code for slot 2 transm.var.code for slot 3	VU VU VU VU	Byte 0 Byte 1 Byte 2-5 Byte 6 Byte 7 Byte 8-11 Byte 12 Byte 13 Byte 14-17 Byte 18 Byte 19 Byte 20-23	transm. variable code for slot 0 units code for slot 0 variable for slot 0 transm. variable code for slot 1 units code for slot 1 variable for slot 1 transm. variable code for slot 2 units code for slot 2 variable for slot 2 transm. variable code for slot 33 units code for slot 2 variable for slot 3	VU VU C1 VU C1 VU C1 VU C1 VU VU C1
		(truncated after last requested code)			(truncated after last requested variable)		
34	Write damping value	Byte 0-3	damping value (seconds)	C1		as in command	
35	Write range values	Byte 0 Byte 1-4 Byte 5-8	range units code upper range value lower range value	VU C1 C1		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 *	VU		as in command	
		* 0=burn EEPROM, 1=copy EEPROM to RAM					
40	Enter/exit fixed current mode	Byte 0-3	current (mA)*	C1		as in command	
		* 0 = exit fixed current mode.					
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	VU		as in command	
45	Trim DAC zero	Byte 0-3	measured current (mA)	C1		as in command	
46	Trim DAC gain	Byte 0-3	measured current (mA)	C1		as in command	
47	Write transfer function	Byte 0	transfer function code	VU		as in command	
48	Read additional device status		none		Byte 0-5 Byte 6-7 Byte 8-10 Byte 11-13 Byte 14-24	device-specific status operational modes (5.1) analog outputs saturated* (5.1) analog outputs fixed* (5.1) device-specific status	B B B B B
					* 24 bits each: LSB ... MSB refers to AO ~1 .. ~24. (Response is truncated after last byte implemented)		
49	Write PV sensor serial number	Byte 0-2	sensor serial number	VU		as in command	
50	Read dynamic variable assignments (4.1)		none		Byte 0 Byte 1 Byte 2 Byte 3	PV transmitter variable code SV transmitter variable code TV transmitter variable code FV transmitter variable code	VU VU VU VU
51	Write dynamic variable assignments (4.1)	Byte 0 Byte 1 Byte 2 Byte 3	PV transm. variable code SV transm. variable code TV transm. variable code FV transm. variable	VU VU VU VU		as in command	

			code				
52	Set transmitter variable zero (4.1)	Byte 0	transmitter variable code	VU		as in command	
53	Write transmitter variable units (4.1)	Byte 0 Byte 1	transmitter variable code transm. var. units code	VU VU		as in command	
54	Read transmitter variable information (4.1)	Byte 0	transmitter variable code	VU	Byte 0 Byte 1-3 Byte 4 Byte 5-8 Byte 9-122 Byte 13-16 Byte 17-20	transmitter variable code transm. var. sensor serial number transm. var. limits units code transm. variable upper limit transm. variable lower limit transm. var. damping value (sec.) transm. var. minimum span (5.0)	VU VU VU C1 C1 C1 C1
55	Write transmitter variable damping value (4.1)	Byte 0 Byte 1-4	transmitter variable code transmitter variable damping value (seconds)	VU C1		as in command	
56	Write transmitter variable sensor serial number (4.1)	Byte 0 Byte 1-3	transmitter variable code transmitter variable sensor serial number	VU VU		as in command	
57	Read unit tag, descriptor, date (5.0)		none		Byte 0-5 Byte 6-17 Byte 18-20	unit tag (8 characters) unit descriptor (16 characters) unit date	A A D
58	Write unit tag, descriptor, date (5.0)	Byte 0-5 Byte 6-17 Byte 18-20	unit tag (8 characters) unit descriptor (16 chars) unit date	A A D		as in command	
59	Write number of response preambles (5.0)	Byte 0	number of response preambles	VU		as in command	
60	Read analog output and percent of range (5.1)	Byte 0	analog out. number code	VU	Byte 0 Byte 1 Byte 22-5 Byte 6-9	analog output number code analog output units code analog output level analog ue output percent of range	VU VU C1 C1
61	Read dynamic variables and PV analog output (5.1)		none		Byte 0 Byte 1-4 Byte 5 Byte 6-9 Byte 10 Byte 11-14 Byte 15 Byte 16-19 Byte 20 Byte 21-24	PV analog output units code PV analog output level PB units code Primary variable SV units code Secondary variable TV units code Third variable FV units code Fourth variable	VU C1 VU C1 VU C1 VU C1 VU C1
					(truncated after last supported variable)		
62	Read analog outputs (5.1)	Byte 0 Byte 1 Byte 2 Byte 3	analog output number code for slot 0 analog output number code for slot 1 analog output number code for slot 2 analog output number code for slot 3	VU VU VU VU	Byte 0 Byte 1 Byte 2-5 Byte 6 Byte 7 Byte 8-11 Byte 12 Byte 13 Byte 14-17 Byte 18 Byte 19 Byte 20-23	slot 0 analog output number code slot 0 units code slot 0 level slot 1analog output number code slot 1units code slot 1level slot 2 analog output number code slot 2 units code slot 2 level slot 3 analog output number code slot 3 units code slot 3 level	VU VU C1 VU VU C1 VU VU C1 VU VU C1
		(truncated after last requested code)			(truncated after last requested level)		
63	Read analog output information (5.1)	Byte 0	analog output number code	VU	Byte 0 Byte 1 Byte 2 Byte 3 Byte 4-7 Byte 8-11 Byte 12-15	analog output number code analog output alarm select code analog output transfer function code analog output range units code analog output upper range value analog output lower range value analog output additional damping value (sec.)	VU VU VU VU C1 C1 C1

64	Write analog output additional damping value (5.1)	Byte 0 analog out. number code VU Byte 1-4 analog out. additional damping value (sec.) C1	as in command	
65	Write analog output range values (5.1)	Byte 0 analog out. number code VU Byte 1 an.out. range units code VU Byte 2-5 an.out. upper range value C1 Byte 6-9 an.out. lower range value C1	as in command	
66	Enter/exit fixed analog output mode (5.1)	Byte 0 analog out. number code VU Byte 1 analog output units code VU Byte 2-5 analog output level* C1 * "not a number"" exits fixed output mode	as in command	
67	Trim analog output zero (5.1)	Byte 0 analog out. number code VU Byte 1 analog output units code VU Byte 2-5 externally-measured analog output level C1	as in command	
68	Trim analog output gain (5.1)	Byte 0 analog out. number code VU Byte 1 analog output units code VU Byte 2-5 externally-measured analog output level C1	as in command	
69	Write analog output transfer function (5.1)	Byte 0 analog out. number code VU Byte 1 an.out. transfer func.code VU	as in command	
70	Read analog output endpoint values (5.1)	Byte 0 analog out. number code VU	Byte 0 analog output number code VU Byte 1 analog output endpoint units code VU Byte 2-5 analog output upper endpoint value C1 Byte 6-9 analog output lower endpoint value C1	
107	Write burst mode transmitter variables (for Command #33) (5.1)	Byte 0 transm.var.code for slot 0 VU Byte 1 transm.var.code for slot 1 VU Byte 2 transm.var.code for slot 2 VU Byte 3 transm.var.code for slot 3 VU	as in command	
108	Write burst mode command number (5.0)	Byte 0 burst mode command number VU	as in command	
109	Burst mode control (5.0)	Byte 0 burst mode control code (0=exit, 1=enter) VU	as in command	
110	Read all dynamic variables (5.0)	none	Byte 0 PV units code VU Byte 1-4 PV value C1 Byte 5 SV units code VU Byte 6-9 SV value C1 Byte 10 TV units code VU Byte 11-14 TV value C1 Byte 15 FV units code VU Byte 16-19 FV value C1	

3. Using the Custom Module

3.1. HART Configuration Overview

Before an ACCOL load can be fully implemented, the following steps may need to be performed.

1. The physical configuration of the HART network may need to be designed or assessed.
2. The Network configuration and associated limitations needs to be assessed, particularly with reference to the required timing.
3. Information concerning all the communicating devices needs to be obtained (e.g. addresses, capabilities).

See the section *'Setting up HART devices and links'* for more information

3.2. ACCOL Configuration overview

The following steps are required to configure the ACCOL load in order to make the 33XX controller act as a HART master device.

1. Define a Custom Port. This is described in the section *'Defining the Custom Port'*.
2. Define the Custom Modules. This is described in the section *'Defining the Custom Modules'*.
3. Define the Signal Lists, known as the HART Custom Lists. This is described in the section *'Defining the HART Custom Lists'*.
4. If 'long frame format' is used, then the address needs to be set-up. This is described in the section *'Defining the Long-Frame Address Lists'*.
5. Define the HART status lists. This is where device status is returned. This is described in the section *'Defining the HART Status Lists'*.
6. Set up the required Output/Input lists and formats, using formats described in the section '*ACCOL Data Format Codes'.*
7. Define the HART Config lists, if required, to override certain operational defaults. This is described in the section *'Defining the Configuration Lists'*.

3.3. Defining the Custom Port

The Custom Port is associated with a specific HART link, communicating with either a single device (as a point-to-point link) or several devices (as a multi-drop link). The following is a list of the parameter field values for a Custom port when configured for HART mode.

- MODE: Set this field to 21 to indicate HART mode
- BAUD: Set this field to 1200 (though the Custom Module will accept other values)
- CHARACTER LENGTH: Set this field to 8 bits 'BIT_8'
- STOP BITS: Set this field to 'SBIT_1' for 1 stop bit
- PARITY: Set this field to 'PARITY_O' for odd parity
- P1: Can be used to adjust RTS hold on time after last character has cleared the USART buffer (but not the shift register). 1 character time is required for the last character (checksum) to clear the USART, hence this should be at minimum approx. 10ms. As the 3330 has a timing resolution of 4ms, this will default to 12ms (if P1 = 0). Otherwise, the user may provide his own millisecond value in P1 if this helps to solve problems.
- P2: Not used

3.4. Defining the Custom Modules

The following is a list of the terminal values for the custom module when configured for HART mode

- MODE: A value of 21 indicates HART mode.
- LIST: The number of the signal list that contains the signals used by this module to control the interface. This signal list is referred to as the HART signal list.
- STATUS: The value of this terminal is a status code representing the module's status. The status code is used to indicate various communication states and error conditions. Communication and processing of reply messages are aborted when the status code is negative. See Section 3.10 for a list of possible status code values and their definitions.

3.5. Defining the HART Custom Lists

3.5.1.1. Table 6 - HART Custom List Contents

Terminal name	Data Type	Description
Signal 1 PORT	Analog	Analog value defining communications port to use for this interface. Valid values are from 1 to 10, and correspond to the following ports: 1 = Port A, 2= Port B, 3=Port C, 4 = Port D, 5 = BIP 1, 6 = BIP 2, 7 = Port G, 8 = Port H, 9 = Port I, 10 = Port J.
Signal 2 FUNCTION	Analog	0 = Transaction with Short Frame Format (1 byte address 0 to 15) 1 = Transaction with Long Frame Format (5 byte address) 2 = Set Burst Mode on, with Short Frame Format 3 = Set Burst Mode on, with Long Frame Format 4 = Set Burst Mode off, with Short Frame Format 5 = Set Burst Mode off, with Long Frame Format For all burst mode commands, the COMMAND CODE signal contains the command used in burst mode responses
Signal 3 SLAVE ADDRESS	Analog	For Short addresses (FUNCTION=0), this has an integer value between 0 and 15. For Long Addresses, this contains the number of a Signal List containing 5 analog signals, each holding an integer value between 0 and 255, corresponding to the 5 bytes of the long address field on transmission order A long address of all zeroes can be used as a broadcast address, though a response must be received from one of the slaves

Signal 4 COMMAND CODE	Analog	Contains an integer value normally between 0 and 255 corresponding to the HART Command code. *(NOTE: Any HART Command code other than 0 requires long frame addressing.)* Values above 256 are treated as extended 2 byte command code, with the first byte being hex 'FE' (254), and the second byte taken as the value given by this signal minus 256. (Note: only 0 to 255 commands allowed with burst mode)
Signal 5 OUTPUT BYTE COUNT/ INPUT BYTE COUNT	Analog	The total number of bytes in the Data Field to transmit. On completion, it will contain the number of data bytes received (not including the status bytes)
Signal 6 HART DEVICE STATUS LIST	Analog	Refers to a list where a decoded version of the two device status bytes is returned. This is only valid provided the Custom Module Status signal indicates that a reply has been successfully received. The Custom Module performs no action on this field, but merely passes it back for action via ACCOL if required.
Signal 7 OUTPUT LIST	Analog	ACCOL Signal to hold the number of a list containing the signals to output, interpreted using the 'OUTPUT FORMAT'
Signal 8 OUTPUT FORMAT	Analog	The format used with the OUTPUT LIST
Signal 9 INPUT LIST	Analog	ACCOL Signal to hold the number of a list containing the signals to contain the data input, interpreted using the 'INPUT FORMAT'
Signal 10 INPUT FORMAT	Analog	The format used with the INPUT LIST
Signal 11 CONFIG LIST	Analog	If this has a value of 0, then defaults are used. Otherwise, the values of various configuration parameters are taken from a configuration list, described below. Note: this list is accessed directly at transmission/reception time, therefore values in it should not be changed until the transaction has completed.
Signal 12 DONE	Logical /Analog	Module execution status. A logical signal is turned ON on completion, while an analog signal is incremented.

3.6. Defining the Long-Frame Address Lists

If long frame addressing is used, then signal ‘SLAVE ADDRESS’ in the Custom List must refer to a list containing the 5 ACCOL analog signals as follows:

3.6.1.1. Table 7 - Long Frame Address List Contents

Terminal name	Data Type	Description
Signal 1 ADDRESS BYTE 1	Analog	This should contain lowest 6 bits of the manufacturer’s identification code. (e.g. as obtained from Byte 1 in the response to ‘Command 0’ for a Rev 5 device - see table 3 in section *‘HART Command Codes’*)
Signal 2 ADDRESS BYTE 2	Analog	This should contain the manufacturer’s device type code. (e.g. as obtained from Byte 2 in the response to ‘Command 0’ for a Rev 5 device - see table 3 in section *‘HART Command Codes’*)
Signal 3 ADDRESS BYTE 3	Analog	This should contain the first (least significant) byte of the device ID number. (e.g. as obtained from Byte 9 in the response to ‘Command 0’ for a Rev 5 device - see table 3 in section *‘HART Command Codes’*)
Signal 4 ADDRESS BYTE 4	Analog	This should contain the second byte of the device ID number. (e.g. as obtained from Byte 10 in the response to ‘Command 0’ for a Rev 5 device - see table 3 in section *‘HART Command Codes’*)
Signal 5 ADDRESS BYTE 5	Analog	This should contain the third byte (most significant) of the device ID number. (e.g. as obtained from Byte 11 in the response to ‘Command 0’ for a Rev 5 device - see table 3 in section *‘HART Command Codes’*)

3.7. Defining the HART Status Lists

The ‘HART DEVICE STATUS’ list contains the status code returned from the field device, decoded into ACCOL signals, (the contents being valid providing the CUSTOM STATUS indicates that a response has been received). It is up to the ACCOL user to check this field according to the application . For instance, the user may want to check this value for general status errors, and then set the questionable bit via ACCOL. Alternatively, he may want to set ACCOL Logical Signals according to a more detailed status analysis.

On a successful communication transaction (i.e. Custom Status >= 0 or if the error is an input format error), this list will contain device status information returned by the Field Device. For full details refer to HART Documentation.

3.7.1.1. Table 8 - HART Status List Contents

Terminal name	Data Type	Description
Signal 1 COMMAND RESPONSE	Analog	The response code returned within bits 0 to 6 of the status field. A non-zero value indicates some form of error or warning. See tables 9 and 10 below.
Signal 2 COMM. ERROR	Logical	Returned 'True' to indicate a comm. error detected by the field device., in which case the COMMAND RESPONSE has a different meaning (see tables 9 and 10 below). If true, the following logical signals will be set 'false'.
Signal 3 PRIMARY VARIABLE OUT OF LIMITS	Logical	Field Device Error
Signal 4 NON-PRIMARY VARIABLE OUT OF LIMITS	Logical	Field Device Error
Signal 5 ANALOG OUTPUT SATURATED	Logical	Field Device Error
Signal 6 ANALOG OUTPUT CURRENT FIXED	Logical	E.g. when device is in multi-drop mode. The device current output will normally be fixed at 4mA.
Signal 7 MORE STATUS AVAILABLE	Logical	Further device specific status information can be read via command number 48.
Signal 8 COLD START	Logical	First status report after power-up
Signal 9 CONFIG CHANGE	Logical	Configuration has been changed
Signal 10 FIELD DEVICE MALFUNCTION	Logical	Field Device Error

3.7.1.2. Table 9 - Command Response Code explanations

Meaning of Command Response when COMM. ERROR = TRUE		
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)

Meaning of Command Response when COMM. ERROR = FALSE	
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
7	in write-protect mode
8-15	multiple meanings (see Table 10)
16	access restricted
28	multiple meanings (see Table 10)
32	device is busy
64	command not interpreted

3.7.1.3. Table 10 - Command Response codes with Multiple Meanings

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, 337, 43, 52 45, 46, 67, 68	Lower range value too high Applied process too high Not in proper current mode (fixed at 4mA or 20mA)
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 to 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*	335, 36, 65 37	Span too small Pushed upper range value over limit
15	65, 66, 67, 68, 69	Invalid analog output number code
28	65	Invalid range units code

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

3.8. Defining the Configuration Lists

This list contains data used to redefine parameters of the protocol. All entries must be present. The defaults listed apply if the CONFIG LIST parameter in the Custom List is set to 0. Even if, for example, it is only required to set the number of preambles to 20, while leaving the rest of the parameters as their defaults, it is still necessary to provide list entries for ALL the parameters.

3.8.1.1. Table 11 - Config List Contents

Terminal name	Data Type	Description
Signal 1 PRIMARY / SECONDARY MASTER SELECT	Logical	True if we are identified as a Primary master for this link False for a Secondary master. Default is TRUE.
Signal 2 PRIMARY / SECONDARY MASTER DELAY WHEN NOT SYNCHED	Analog	Integer number of ms for an un-synchronized master to wait for an Idle time to detect that it can transmit on the link. Normal Values are 305ms for a Primary master, and 380ms for a Secondary master. Default is 305ms.
Signal 3 DELAY BEFORE TX, WHEN OTHER MASTER JUST TRANSMITTED	Analog	Number of ms to wait, to ensure that the other master doesn't initiate a transmit. Value of between 4 and 1000ms. Default is 20ms.
Signal 4 DELAY BEFORE TX, WHEN THIS MASTER JUST TRANSMITTED	Analog	Number of ms to wait, to ensure that the other master can initiate a transmit. Value of between 4 and 1000ms. Default is 75ms
Signal 5 NUMBER OF PREAMBLES ON TX	Analog	Number of Preambles (between 5 and 20) to send at the start of each message. Value to use may depend on device, and whether link is noisy when being turned around. Default is 5.
Signal 6 SLAVE RESPONSE TIMEOUT	Analog	Value of between 4 and 1000ms, timed from end of transmission of last character of transmitted message. Default is 256ms.

Signal 7 TRANSMIT TIMEOUT	Analog	Normal value of 256ms, timed to end of transmission of last character of transmitted message (allowing for variable number of preambles). Default is 256ms
Signal 8 NUMBER OF RETRIES	Analog	Number of communication attempts. A value of 0 will cause all errors to be passed back to the caller. Also, errors in messages detected to the other master may cause status errors. Default is 3.

3.9. ACCOL Data Format Codes

A full list of the available format codes is given below, though many are not directly applicable to HART.

()	Parenthesis are used to group a section of the Format for repetition. Parenthesis may be nested up to five levels.
SFn	This descriptor invokes Format number n where n is any valid Format number. At the end of Format n, processing continues with the descriptor following SFn.
DA	The value of the current signal in the I/O list is used to define the number of an analog Data Array to be used. The signal's type must be analog. Array mode is set active which causes cells in the Data Array to be used by field descriptors for input and output. The first cell in the array is used first and all columns of a row are used before going to the next row. This descriptor causes an increment to the next signal in the I/O list.
DL	The value of the current signal in the I/O list is used to define the number of a logical Data Array to be used. The signal's type must be analog. Array mode is set active which causes cells in the Data Array to be used by field descriptors for input and output. The first cell in the array is used first and all columns of a row are used before going to the next row. This descriptor causes an increment to the next signal in the I/O list.
DE	Array mode is ended. Field descriptors resume using signals in the I/O list.
DC	Array mode is set active. A Data Array must have been previously defined via the DA or DL field descriptors. Field descriptors resume using cells in the data array.
BIT	Bit alignment mode is set active. The data in a message is processed in units of bits. Lower order bits of a byte or word are processed before higher order bits. If Word alignment mode was previously active, any remaining bits of the current word are used before using the next data byte. If Byte alignment mode was previously active, any remaining bits of the current byte are used before using the next data byte. It is intended that Bit alignment mode be used to access single bit logical values and subfields within a byte or word.
BYT	Byte alignment mode is set active. The data in a message is processed in units of bytes. Each field begins with the low order bit of the next byte. Values are treated as being right justified within the byte. If Word alignment mode was previously active and the high order byte of the current word was now used, the high byte is used first before using the next data byte.
WRD	Word alignment mode is set active. The data in a message is processed in units of words. Values are treated as being the combination of two bytes. Each field begins with the low order bit of the next word. Either the low order byte or the high order byte can occur first in the message. Values are right justified within the word.
LBF	Low Byte First mode is set active. Word alignment mode will treat the first of two bytes as being the low order byte of the word.
HBF	High Byte First mode is set active. Word alignment mode will treat the first of two bytes as being the high order byte of the word.
VL	This field descriptor is used for input or output of logical values. It operates on either bits, bytes, or words depending on the alignment mode. For input, the current bit, byte, or word value in the message is tested for zero. A

value of zero is treated as false and a non-zero value is treated as true. The current signal in the I/O list or the current cell in the data array is set to reflect the true or false value.

Analog signals or cells are set to 0.0 for false and 1.0 for true. String signals are invalid.

For output, the current signal or cell is tested for true or false. If true, a bit, byte, or word value of 1 is put in the message. If false, a bit, byte, or word value of 0 is put in the message. Analog signals or cells with values of 0.0 are treated as being false. String signals are invalid.

This descriptor causes an increment to the next signal in the I/O list or to the next cell in the data array depending on array mode being active. It also causes an increment to the next bit, byte, or word in the message depending on the alignment mode.

VSn This field descriptor is used for input or output of signed (2's complement) binary values with a field width of n bits., If Bit mode is active, the next n bits in the message are used. If Byte or Word mode is active, the field is right justified in the byte or word. If Byte mode is active and n is greater than 8, multiple bytes will be used. If Word mode is active and n is greater than 16, multiple words will be used.

The value of n may range from 2 to 32. The default value for n if not specified is; 2 for Bit mode, 8 for Byte mode, and 16 for Word mode.

For input, the current signal in the I/O list or the current cell in the data array is set to the value of this field. Logical signals or cells are set to false if the value is zero and set to true if the value is non zero. String signals are invalid.

For output, the value of the current signal or cell is in the message. Logical signal or cell values of false are equivalent to 0 and values of true are equivalent to 1. String signals are invalid. Values are rounded to the next integer value and values too large for the field are output as the largest possible field value.

This descriptor causes an increment to the next signal in the I/O list or to the next cell in the data array depending on array mode being active.

VUn This field descriptor is the same as VSn with the following exceptions. The binary value is unsigned and n may range from 1 to 32. Negative values are output as zero. The maximum value in a 32 bit field is limited to a 31 bit number for both input and output.

BCDn This field descriptor is used for input or output of Binary Coded Decimal (BCD) values with a field width of n digits. If Bit mode is active, the next n*4 bits in the message are used with the first digit treated as the highest order digit. If Byte or Word mode is active, the digits are right justified within the byte or word. If Byte mode is active and n is greater than 2, multiple bytes will be used. If Word mode is active and n is greater than 2, multiple bytes will be used. If Word mode is active and n is greater than 4, multiple words will be used.

The value of n may range from 1 to 39. The default value for n if not specified is; 1 for Bit mode, 2 for Byte mode, and 4 for Word mode.

For input, the current signal in the I/O list or the current cell in the data array is set to the value of the field. Logical signals or cells are set to false if the value is zero and set to true if the value is non-zero. String signals are invalid.

For output, the value of the current signal or cell is put in the message. Logical signal or cell values of false are equivalent to 0 and values of true are equivalent to 1. String signals are invalid.

This descriptor causes an increment to the next signal in the I/O list or to the next cell in the data array depending on array mode being active.

Tn This field descriptor is used for input or output of ASCII text strings with a length of n characters. Each character is 8 bits. If Bit mode is active, the next n*8 bits in the message are used. If Byte mode is active, the next n bytes are used. If Word mode is active, the next n/2 words are used.

The value of n may range from 1 to 64. The value of n will default to the length of the String signal's value if it is not specified. Only string signals from the I/O list are valid.

Values too large will be truncated and values too small will be padded with space characters.

For input, the current string signal in the I/O list is set to the string value of the field. Space characters are substituted for non printable characters in the string.

For output, the value of the current string signal is put in the message.

This descriptor causes an increment to the next signal in the I/O list.

X This field descriptor is used to skip a bit, byte, or word depending on the alignment mode. For output, a value of 0 is put in the message for the current bit, byte, or word.

This descriptor causes an increment to the next bit, byte, or word in the message depending on the alignment mode.

It is possible for field descriptors VS, VU, BCD, and T to use a partial byte or word. If Byte alignment mode is active and there are unused bits in the current byte, switching to Bit alignment mode via the BIT field descriptor will allow the unused bits to be accessed. If Word alignment mode is active and there are unused bits or bytes in the current word, switching to Bit or Byte alignment mode will allow the unused bits or bytes to be accessed. This is useful when different data types are combined into the same byte or word.

For example, a word may contain a 3 digit BCD value in the low order 12 bits and 4 logical status values in the high order 4 bits. The Format sequence WRD BCD3 BIT 4VL will relate the BCD value with a signal or array cell and each of the four status bits with its own signal or array cell.

CST1:n This field specifies conversion is to be made using 32 bit floating point format (IEEE-754 single precision) The low byte is output/input first.

CST2:n This field specifies conversion is to be made using 32 bit floating point format (whipple format) The low byte is output/input first.

CST3:n This field specifies conversion is to be made using 32 bit floating point format (IEEE-754 single precision) The high byte is output/input first.

CST4:n This field specifies conversion is to be made using 32 bit floating point format (whipple format) The high byte is output/input first.

CST5:n 64 bit analog format. Input/Output value is an 8 byte field, LSB first, in standard 'Intel' 64 bit double precision format. NOTE: use of this format is not precise, as ACCOL signals cannot hold the full resolution.

CST6:n 64 bit analog format. Input/Output value is an 8 byte field, MSB first, in standard 'Intel' 64 bit double precision format. NOTE: use of this format is not precise, as ACCOL signals cannot hold the full resolution.

CST7:n 32 Bit unsigned Long integer format. Input/Output value as a 4 byte field, LSB first, without sign bit. NOTE: use of this format is not precise, as ACCOL signals cannot hold the full precision. Use this where full precision is not required. The

33xx system cannot support more than 31 bits.

CST8:n 24 Bit unsigned Long integer format. Input/Output value as a 3 byte field, LSB first, without sign bit. NOTE: use of this format is not precise, as ACCOL signals cannot hold the full precision. Use this where full precision is not required.

CST9:n HART Packed ASCII Input/Output. With 4 characters coded per 3 bytes. 'n' gives number of characters to use

3.10. Error and Status Codes

3.10.1.1.Table 12 - Error/Status messages

0	communication completed successfully
1	communication requested, waiting to send
2	command message sent, waiting for reply
10	burst received
11	burst cancelled
-2	invalid HART Custom List specified
-3	invalid Port specified
-4	invalid Slave Address specified
-5	invalid Address List
-6	invalid Command Code
-7	invalid Byte count
-8	invalid Function code
-9	invalid HART status signal
-10	invalid default list signal
-11	invalid default list signal type
-12	invalid output list
-13	invalid input list
-14	invalid output format number
-15	invalid input format number
-16	invalid no. of tries in defaults list
-17	invalid rx time-out in defaults list
-18	invalid tx time-out in defaults list
-19	invalid hold-off time-out in defaults list
-20	invalid hold-off time-out in defaults list
-21	invalid no. of preambles in defaults list
-22	invalid idle time-out in defaults list
-23	invalid primary/secondary select in defaults list
-24	reply inconsistency
-25	reply inconsistency
-26	input character overrun detected
-27	input character parity error detected
-28	input character framing error detected
-29	input security check (BCC) error detected
-30	timed out waiting for response
-32	transmit timed out waiting for CTS
-33	buffer overflow
-35	message received - not addressed to us
-36	unexpected burst message received addressed to us
-37	unexpected burst message received - not addressed to us
-38	internal error
-39	response to other master time-out
-40	too many preambles monitored
-41	internal error

The following are format errors and warnings. Those with positive signs are warnings - data sent or returned is usually valid.:

101	An input signal is control inhibited (formats only)
102	Attempt to store signal into a constant (formats only)
103	An input string signal value was truncated (formats only)
104	Attempt to store into a Read Only data array
116	Format error - attempt to go beyond end of buffer on read
117	Format error - list/format specifies more than totalcount
118	Format error - overflow in write regs data array
-101	Format error - unsupported field descriptor
-102	Format error - attempt to use signal beyond end of list
-103	Format error - too many levels of parenthesis
-104	Format error - unmatched right parenthesis
-105	Format error - sub format number does not exist
-106	Format error - too many levels of sub formats
-107	Format error - invalid data array number selected
-108	Format error - data array has not been defined
-109	Format error - attempt to use cell beyond end of array
-110	Format error - signal or cell type must be analog
-111	Format error - signal type must be string
-112	Format error - sig or cell type must be analog or logical
-113	Format error - BCD input value is invalid
-114	Format error - bad analog value for BCD output
-115	Format error - unexpected input signal store failure
-116	Format error - attempt to go beyond end of buffer
-117	Format error - invalid floating point value
-119	Format error - Floating point field overflow on output
-121	Format error - field length too big, or decimal point position error
-122	Format error - rep count not 1

4. Setting up HART devices and links

4.1. General considerations

For multi-drop operation, it is important to know the capabilities of all the transmitters planned to be on a link, for example

- Can all the transmitters support 'long addressing'?
- Can the 4-20 mA output be disabled, or compensated for electrically?
- Can the overall timing requirements for polling devices be met?

4.2. Physical set-up.

The overall physical design of Hart networks and cabling runs can be quite complex, and is beyond the scope of this document (refer to references 1 and 2).

Any RS232 or RS485 port can be used to communicate. A single Hart device can be used, in ***point-to-point*** mode, or alternatively, up to 15 HART devices can be connected to the HART Device Interface (HDI) board using ***multi-drop*** mode.

When a single field device is connected, the 4-20ma input can be read by linking to an 'Analog Input' card, and using an ACCOL 'ANIN' module. This may be useful if the primary variable is required to be read faster than it can be read through the HART protocol, though there may be reduced accuracy. In multi-drop operation, the 4-20ma signal cannot be used.

The HDI card has built in a 250 ohm load resistor, necessary for communication.

In multidrop mode, there are important considerations to take into account for Intrinsic Safety. Please refer to HART Documentation, and documentation on the individual field devices.

4.3. Getting the Basic Information.

In order to communicate successfully with a device, it is necessary to obtain certain basic information about the device. The information may be available from the device manufacturer's documentation, otherwise it is obtained by sending a Command 0 to the device, on a point-to-point link, with a short-frame address of zero. This only needs to be performed once as part of the setting up of the link, and is best done by writing a simple ACCOL load just to do this. The information obtained can then be transcribed into the target ACCOL load. The command 0 should be generally be sent with 20 preambles, so a CONFIG LIST (signal 11 in Custom List) should be used with NO OF PREAMBLES (signal 8 in CONFIG LIST) set to 20, and all other entries set to their default values.

The response to command 0 will either contain 10 bytes or more depending on the HART revision of the device. (see tables 3 and 4 in section *'HART Command Codes'*)

If the 'tag' of the field device is known (for Rev 5 devices) and is unique on the link, then a Command 11 can be used instead of Command 0. In this case, the Command 11 must be used with a 'Long Address' with all fields set to 0, and the output data should contain the tag of the required device. This allows the basic information to be obtained without the need to set up a point-to-point link.

The basic information of most relevance is as follows:

Universal Command Revision
If this is Rev 5, then the command structures are as described in table 3, otherwise for revisions 2 ,3 or 4, use table 4 (in section *'HART Command Codes').*

Number of preambles required for this device.
The 'pre-ambles' are bytes (of hex 'FF') which must precede each message sent to a device. Normally, the default of 5 pre-ambles is sufficient, but some devices may require up to 20 for consistent operation.

'Long Frame Address' information.(Rev 5 devices)
See table 7 (in section *'Defining the Long-Frame Address Lists'*) for the source of this 5 byte field.

5. Problem solving

5.1. Possible limitations

The HART protocol is an Open Protocol, and is implemented by many manufacturers. However there may be differences in the interpretation of the specification between products, which may cause problems. An example of this is a Rosemount-Pressure transmitter which appends an extra hex ‘ff’ character onto each message it transmits, and also on receive, it requires the carrier to be left on for a while after the last character is transmitted (hence the USART ‘auto-enables’ mode cannot be used). Hence, for all applications, it is recommended that a bench trial is performed with each field device to be used, before committing to field installation.

On the 33XX, the USART ‘Auto-enables’ is used at the start of transmission, but not at end of transmits/receive. This means that the transmission of the first character is synchronized with CTS, but that RTS is not dropped immediately after the last character transmitted, but is instead controlled by a timer. This was done due to the behavior of a test field device, which required an extra delay before RTS was dropped. This delay has been made configurable via parameter ‘P1’ in the ACCOL Load.

5.2. Use of Serial Line Analyser

If errors are occurring on the link, or there is difficulty in establishing communications, then it may be desirable to monitor the communications activity by using a Serial Line Analyser. Bell 202 de-modulation will normally be required for this.

5.3. HART Communicator

If communication with the field devices cannot be obtained using the 33XX, then it may be best to check network operation using a hand-held terminal, such as a ‘HART Communicator’ (as available from Fisher-Rosemount).

6. Example

This ACCOL load reads the current and percentage from a field device using Command 2, at a continuous rate. The Questionable bit is set on an error. The Device is connected to a HART Device Interface (HDI) board on Port B.

```
*TARGET 33XX-386EX    VERS: 0002
*SECURITY-CODES    666666   555555   444444   333333   222222   111111
*MEMORY
*COMMUNICATIONS
   PORT_A  SLAVE            9600
   PORT_B  CUSTOM           1200 BIT_8 SBIT_1   PARITY_O PARAM: 21 0 0
   PORT_C  UNUSED
   PORT_D  UNUSED
   BIP_1   UNUSED
   BIP_2   UNUSED
*PROCESS-I/O
*TASK 1  RATE:   C  PRI:     1
*BASENAMES
*SIGNALS
         PORT.1.1            A R1 W3 ME CE              2.0000000
         FUNC.1.1            A R1 W3 ME CE              1.0000000
         ADDR.1.1            A R1 W3 ME CE              48.0000000
         COMMAND.1.1         A R1 W3 ME CE              2.0000000
         COUNT.1.1           A R1 W3 ME CE              0.0000000
         HSTATUS.1.1         A R1 W3 ME CE              47.0000000
         OUTLIST.1.1         A R1 W3 ME CE              230.0000000
         OUTFMT.1.1          A R1 W3 ME CE              230.0000000
         INLIST.1.1          A R1 W3 ME CE              202.0000000
         INFMT.1.1           A R1 W3 ME CE              202.0000000
         DEFS.1.1            A R1 W3 ME CE              0.0000000
         DONE.1.1            L R1 W4 ME CE      0 ON    OFF

         COMM.RESP.1         A R1 W3 ME CE              0.0000000
         COMM..ERR.1         L R1 W4 ME CE      0 ON    OFF
         PV.OOL.1            L R1 W4 ME CE      0 ON    OFF
         NPV.OOL.1           L R1 W4 ME CE      0 ON    OFF
         AO.SAT.1            L R1 W4 ME CE      0 ON    OFF
         AO.FIXED.1          L R1 W4 ME CE      0 ON    OFF
         MORE.STAT.1         L R1 W4 ME CE      0 ON    OFF
         COLD.START.1        L R1 W4 ME CE      0 ON    OFF
         CONFIG.CHANGE.1     L R1 W4 ME CE      0 ON    OFF
         DEV.MALF.1          L R1 W4 ME CE      0 ON    OFF

         CST.STATUS.1        A R1 W3 ME CE              999.0000000

         ADD1.1.1            A R1 W3 ME CE              38.0000000
         ADD2.1.1            A R1 W3 ME CE              3.0000000
         ADD3.1.1            A R1 W3 ME CE              77.0000000
         ADD4.1.1            A R1 W3 ME CE              40.0000000
         ADD5.1.1            A R1 W3 ME CE              25.0000000
         CURRENT.1.1         A R1 W3 ME CE              0.0000000
         PERCENT.1.1         A R1 W3 ME CE              0.0000000

         LASTSTAT.1          A R1 W3 ME CE              0.0000000
         ERRORS.1            A R1 W3 ME CE              0.0000000
         OK.1                A R1 W3 ME CE              0.0000000
```

```
        CUSTOM.LIST.1          A R1 W3 ME CE                    1.0000000


*TASK 0
*TASK 1
  10 * C
  20 * C ------  COMMAND 2  - READ CURRENT/% (LONG ADDRESS IN LIST 48)
  22 * C
  25 * CALCULATOR COUNT.1.1=0
  30 * CUSTOM
    MODE                    21.0000000
    LIST                    CUSTOM.LIST.1
    STATUS                  CST.STATUS.1

  40  * CALCULATOR LASTFUN.1 = 1
  50  * WAIT FOR (DONE.1.1)  0.1
  60  * IF (CST.STATUS.1>=0.0)
  70  *  CALCULATOR OK.1 = OK.1+1
  80  *  C Check Low Byte of HSTATUS for a non-zero value. If so, the field device
  90  *  C has reported an error, so set the Q bit on the input signals  Can
  95  *  C Add check for valid range of numbers etc.
  100 *  IF (COMM.RESP.1 !=0 | COMM..ERR.1)
  110 *   CALCULATOR
                    10 :Q:CURRENT.1.1 = #ON
                    20 :Q:PERCENT.1.1 = #ON
  120 *  ELSE
  130 *   CALCULATOR
                    10 :Q:CURRENT.1.1 = #OFF
                    20 :Q:PERCENT.1.1 = #OFF
  140 *  ENDIF
  150 * ELSE
  160 *   CALCULATOR
                    10 ERRORS.1 = ERRORS.1+1
                    20 LASTSTAT.1 = CST.STATUS.1
                    30 :Q:CURRENT.1.1 = #ON
                    40 :Q:PERCENT.1.1 = #ON
  170 * WAIT DELAY 0.1 S
  180 * ENDIF

*LIST   1
10 PORT.1.1
20 FUNC.1.1
30 ADDR.1.1
40 COMMAND.1.1
50 COUNT.1.1
60 HSTATUS.1.1
70 OUTLIST.1.1
80 OUTFMT.1.1
90 INLIST.1.1
100 INFMT.1.1
110 DEFS.1.1
120 DONE.1.1

*LIST 47
10 COMM.RESP.1
20 COMM..ERR.1
30 PV.OOL.1
40 NPV.OOL.1
```

50 AO.SAT.1
60 AO.FIXED.1
70 MORE.STAT.1
80 COLD.START.1
90 CONFIG.CHANGE.1
100 DEV.MALF.1

*LIST 48
10 ADD1.1.1
20 ADD2.1.1
30 ADD3.1.1
40 ADD4.1.1
50 ADD5.1.1

*LIST 51
10 ERRORS.1
30 OK.1
50 LASTERR.1
90 LASTSTAT.1

*LIST 202
10 CURRENT.1.1
20 PERCENT.1.1

*FORMAT 202
10 2CST1:0

*LIST 50
20 HT.STATUS.1
30 DONE.1.1
40 OK.1
50 ERRORS.1
110 PORT.1.1
120 FUNC.1.1
130 ADDR.1.1
140 COMMAND.1.1
150 COUNT.1.1
160 HSTATUS.1.1
170 OUTLIST.1.1
180 OUTFMT.1.1
190 INLIST.1.1
200 INFMT.1.1
210 DEFS.1.1

*LIST 230
*FORMAT 230

7. References / Further Information

1. ROSEMOUNT INC. HART Documentation Package.
   a) Physical/Data Link Specification, Rev 3, 1987.
   b) 3051 Transmitter Specific Command Specification Rev 3, 1987
   c) Command Summary information, Rev 3 1987
   d) Common Practice Command Specification, Rev 3 1987
   e) Universal Command Specification, Rev 3 1987
   d) Common Tables, Rev 1, 1987

2. HART Field Communications Protocol, A Technical Description (Second Edition), Romilly Bowden, November 1995 (from Fisher-Rosemount-UK).

3. Product Information Package for HART Device Interface (HDI) Board – See Bristol Babcock document PIP-HART33XX.

READER RESPONSE FORM

Please help us make our documentation more useful to you! If you have a complaint, a suggestion, or a correction regarding this manual, please tell us by mailing this page with your comments. It's the only way we know we're doing our job by giving you correct, complete, and useful documentation.

DOCUMENT NUMBER: D4068
TITLE: *Hart Master Protocol Manual*
ISSUE DATE: March, 2000

COMMENT/COMPLAINT:

____
____
____
____
____
____
____
____
____
____
____
____
____

Mail or FAX this page to:
Bristol Babcock Inc.
1100 Buckingham Street
Watertown, CT 06795
FAX#: (860) 945-2213
Attn: Technical Publications Group, Dept. 314

Bristol Babcock Inc.
an FKI company

1100 Buckingham Street
Watertown, CT 06795
Telephone: (860) 945-2200

Return to the Table of Contents

Return to the List of Manuals